

# **ULİSAR**

## **Ulusal İnsansız Sualtı Aracı Projesi**

**Proje No: 105E127**

Prof.Dr. M. Kemal LEBLEBİCİOĞLU

MAYIS 2010  
ANKARA

**ÖNSÖZ**

Bu proje bir TÜBİTAK-1001 projesi olup 1.06.2006 yılında başlamıştır. Sualtı gözleme amacı ile yapılan bir insansız sualtı aracı hakkındadır. Proje bütçesi – bölüm payı dâhil olmak üzere - yaklaşık 462,000 TL'dir. Proje 31.05.2009 tarihinde sona ermiştir.

Bitirme raporu Ekim 2009 tarihinde verilmiş ve hakem istekleri doğrultusunda eksik olan kısımlar tamamlanmış ve sualtı aracı üzerinde gerekli deneyler yapılarak rapor tamamlanmıştır.

# İÇİNDEKİLER

ŞEKİL LİSTESİ.....	4
TABLO LİSTESİ.....	5
ÖZET.....	6
1. GİRİŞ.....	7
ÇOK MAKSATLI ULUSAL İNSANSIZ SUALTI ARACI PROJESİ (ULİSAR).....	7
2. GENEL BİLGİLER.....	9
2.1 Proje önerisinde neler var?.....	9
2.2 Projede neler yaptık?.....	10
2.3 Projede yapılan işlerin sınıflandırılması.....	12
2.4 Projede yapılan işler nerede kimler tarafından yapıldı?.....	12
3. GEREÇ VE YÖNTEM.....	13
3.1 Mekanik Tasarım Ve Üretim İle İlgili Gelişmeler.....	13
3.1.1 Barış Elektrik firmasına yapılan ziyaretler.....	13
3.1.2 Akustik modem için fiberglas plakadan yuva yapımı ve bu plakanın sabitlenmesi işlemi.....	13
3.1.3 Motorların gövdeye sabitlenmesi.....	13
3.1.4 Elektronik kart kutusunun alt çerçevesinin gövdeye kalıcı olarak yerleştirilmesi.....	14
3.1.5 Kamera ve kamera ışığı oyuklarının açılması bu ekipmanın açılan yuvalara yerleştirilmesi.....	14
3.1.6 Konnektörlerin takılması ve bu parçaların sızdırmazlığının sağlanması.....	15
3.1.7 İki yana saplamaların yapılması.....	17
3.1.8 Mesafe algılayıcılarının yerleşimi.....	17
3.1.9 Alt gövdeye suyun dolup boşalması için deliklerin açılması.....	17
3.1.10 Çizimlerin üretilen gövdeye göre yenilenmesi.....	17
3.1.11 Elektronik kartlardan çıkan önemli konnektörlerin bilgisayarda modellenmesi.....	17
3.1.12 Elektronik kart kutusunun yeniden tasarlanarak büyütülmesi, yeni kutunun tasarımı ve plakalarının üretimi.....	18
3.1.13 Uzun saplamalar kullanılarak kutunun montajının basitleştirilmesi... ..	18
3.1.14 Somunların Barış Elektrik'e gönderilecek çerçevenin içine geçirilmesi.....	18
3.1.15 Pillerin sabitlenmesi.....	19
3.1.16 Suya Kolaylıkla İniş Çıkış İçin Vinç Mekanizması Tasarımı.....	19
3.1.17 Gövde ile İlgili Hidrodinamik ve Isıl Analizler.....	25
3.2 Yazılım Çalışmaları.....	25
3.2.1 Yazılım Raporu Özeti.....	25
3.2.2 Denetim Bilgisayarı.....	28
3.2.3 İşletim Sistemi Görüntüsünün Hazırlanması ve Yüklenmesi.....	29
3.2.4 Çalıştırılabilir Kodun Derlenmesi ve Yüklenmesi.....	30
3.2.5 Görüntü Bilgisayarı.....	31
3.2.6 Operatör Bilgisayarı.....	32
3.2.7 Karşılaşılan Problemler.....	33
3.3 Deneysel Çalışmalar.....	33
4. Sonuç.....	37
TÜBİTAK PROJE ÖZET BİLGİ FORMU.....	38

# ŞEKİL LİSTESİ

Şekil 1.0.1 Sualtı-suüstü sistemi .....	8
Şekil 3.0.1 Kamera ve projektör yerleşimi.....	14
Şekil 3.0.2 Sızdırmaz bölge kablolama .....	16
Şekil 3.0.3 Konnektörlerin yerleşimi .....	16
Şekil 3.0.4 Hidrolik vinç.....	20
Şekil 3.0.5 Vinç tabanı .....	21
Şekil 3.0.6 Üçgen destek .....	22
Şekil 3.0.7 Vinç ve denizaltının bir görünümü .....	22
Şekil 3.0.8 Vinç ve denizaltının başka bir görünümü .....	23
Şekil 3.0.9 Denizaltı taşıyıcısından vince aktarım.....	23
Şekil 3.0.10 Denizaltıyı vinç ile havuz kenarına yaklaştırma işlemi .....	24
Şekil 3.0.11 Denizaltıyı vinç ile havuza indirme işlemi.....	24
Şekil 3.0.12 İlk yazılım tasarımı .....	27
Şekil 3.0.13 Operatör kullanıcı arayüzü .....	28
Şekil 3.0.14 Denizaltı aracını kontrol etmekte kullanılan kumanda.....	32
Şekil 3.0.15 Sızdırmazlık testlerinden bir görüntü.....	34
Şekil 3.0.16 Ağırlık kemerlerinin ayarlanması .....	35
Şekil 3.0.17 Denge testi çalışmaları.....	35
Şekil 3.0.18 Manevra testlerinden bir görüntü.....	36
Şekil 3.0.19 Çalışmalar sırasında operatör konsoluna gelen bir görüntü.....	36
Şekil 3.0.20 Çalışmalar sırasında çekilen bir fotoğraf.....	37

# **TABLO LİSTESİ**

Tablo 1 Kumandadaki düğme ve kolların işlevi.....	33
---	----

## ÖZET

ULİSAR projesi bir TÜBİTAK-1001 projesi olup 1.06.2006 yılında başlamıştır. Sualtı gözleme amacı ile yapılan bir insansız sualtı aracı hakkındadır. Temel fikir olarak, sualtı aracı ile beraber hareket eden ve sualtı aracına fiber kablo ile bağlı bir suüstü aracı olması düşünülmüştür. Sualtı ve suüstü araçları aralarında fiber üzerinden haberleşirken, suüstü aracı ile ana gemi arasındaki haberleşme rf üzerinden planlanmıştır. Böylece gerçek zamanlı navigasyon mümkün olacaktır. Mayın tarama görevleri için de kullanılabilmesi için gövde kompozit malzemedan imal edilmiştir. Oldukça kalabalık bir araştırmacı grubu projeye katkıda bulunmuşlardır. Bunların arasında farklı bölümlerden öğretim üyeleri, yüksek lisans ve doktora öğrencileri, şirketler ve mühendis subaylar vardır. Proje bütçesi – bölüm payı dâhil olmak üzere - yaklaşık 462,000 TL'dir. Proje sonunda istenilen hedeflere tam olarak ulaşılmasa da görüntülü navigasyon yapabilen, üzerinde kameralar, projektörler, sonar, sonar modem, akustik algılayıcılar ve benzeri cihazların yüklü olduğu, ilgili elektronik kartların tasarlanıp hazırlandığı, bu cihazlarla ilgili yazılımların hazırlandığı, kullanıcıya kolaylık sağlayacak şekilde bir operatör konsolunun bulunduğu bir sistem ortaya çıkmıştır.

Sualtı aracının deneme çalışmaları ODTÜ açık ve kapalı havuzlarında gerçekleştirilmiştir. Deneyler sonucunda navigasyon ve görüntü bilgileri elde edilmiştir. Araca eklenen her yeni modülden sonra yeni deneyler yapılmaktadır.

**Anahtar kelimeler:** İnsansız sualtı aracı (uuv), navigasyon, kompozit malzeme, fiber optik kablo, sürücüler, alıcılar, sonar, akustik modem, kamera, projektör.

# 1. GİRİŞ

## ÇOK MAKSATLI ULUSAL İNSANSIZ SUALTI ARACI PROJESİ (ULİSAR)

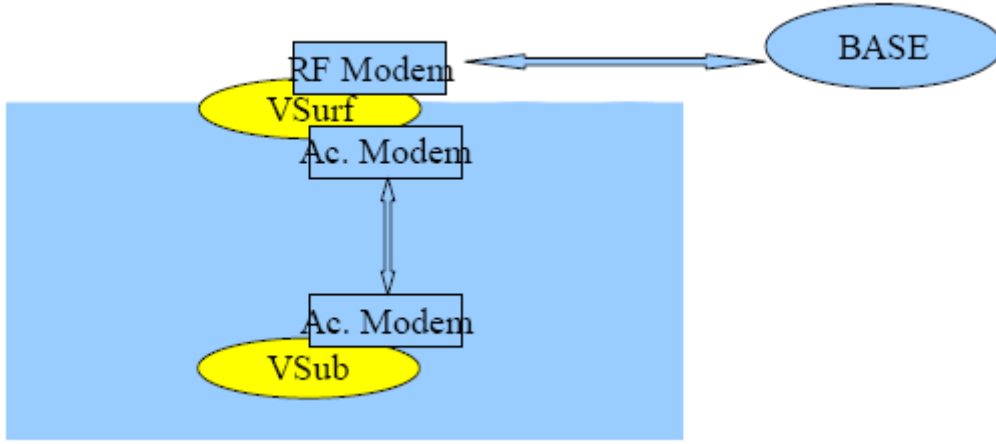
Ülkemizin doğalgaz, petrol boru hatları ile sualtı arkeoloji, petrol arama ve sismik çalışmaları gibi sivil ihtiyaçları ile mayın avlama, sörvey ve denizaltı kurtarma çalışmaları gibi askeri ihtiyaçları için ihtiyaç duyulan çok maksatlı sualtı aracının ülke imkanları ve yakalanan teknolojik seviye kullanılarak disiplinler arası bir sinerji yaratılmak ilk prototip tasarımının elde edilmesi projenin ana konusudur. Böylelikle sivil ve askeri uygulamalar için yurtdışından karşılanan ihtiyaçlar nedeniyle yurtdışına aktarılan milyonlarca dolar seviyesindeki kaynağın ülke içinde kalması ve kritik sualtı aracı geliştirme teknolojisinin ülkeye kazandırılması hedeflenmektedir. Üzerinde durulacak en önemli temel konulardan birisi de bu aracın bağlantısız olarak çalışabilmesinin amaçlanmış olmasıdır.

Bu konuda 1980 yıllarından itibaren dünyada yapılan araştırmalarda büyük bir artış olmuş ve dünyadaki bütün büyük üniversitelerde bu konudaki araştırmaların yapıldığı laboratuvarlar ve enstitüler kurulmuştur. Şu anda konu – en azından fiber optik kablo bağlantılı cihazlar açısından düşünüldüğünde – ticari uygulamalara dönüşmüştür. Yapılan çalışmalar, daha çok standart tasarımların, komponent bazındaki ilerlemeleri içine almasını sağlamak yönündedir (daha iyi kameralar, daha hafif ve güçlü aküler gibi).

Otonom cihazlar konusundaki gelişmeler oldukça sınırlıdır. Aracı kaybetme riskini göze almamak için, topoğrafyası kesin olarak belirlenmiş bölgelerde, basit sörvey araştırmaları yapabilecek şekilde kullanılan bazı sistemler vardır (Hatta bunlar arasında, kablolu çalışabilen sistemlerin kablosuz bir şekilde kullanılanları da sayılabilir). Bu sistemlere “engel tanımlayıcı” gibi bazı kritik durumları kendi kendine halletmesini sağlayabilecek “yapay zeka” diyebileceğimiz basit özelliklere sahip işletim sistemleri yüklenmiştir.

Bizim bu projede ilk önerdiğimiz kablosuz iletişimli sistemin tam benzeri hemen hiç araştırma amaçlı veya ticari sistem yoktur. Bu durumun nedenleri arasında akustik modemlerin oldukça sınırlı bir kapasiteye sahip olduklarını ve çalışma bölgelerinin 5 bin metreyi fazla geçmemesini söyleyebiliriz. Genel olarak dikey doğrultuda 6 bin metreye kadar bilgi iletişimi mümkün olsa da yatay doğrultuda – yansıma probleminden dolayı – çalışma bölgesi hızla azalmaktadır. Ayrıca dikey doğrultuda, sudaki sıcaklık, tuzluluk farklılaşması ve akıntılar yüzünden gerçek güvenilir haberleşme mesafesi çok fazla değildir. Bu problemin üstesinden gelebilmek için önerilen prototip sistemin en önemli özelliği, sualtı aracı ile beraber uyumlu olarak çalışan bir suüstü aracının olmasıdır. Bu sayede – şimdilik 100 metre derinliklere kadar – gerçek zamanlı görüntü bilgisi sualtı aracından suüstü aracına aktarılacaktır. Önce akustik modem olarak düşünülen bu iletişim şekli, daha sonra – satın alınabilecek akustik modemlerin yetersizliği, pahalılığı ve akustik haberleşmenin getireceği risklerden dolayı – fiber kablo üzerinden haberleşme olarak değiştirilmiştir. Bu şekilde bile oldukça orjinal bir tasarım

olduğundan söz edebiliriz. Suüstü aracı ile ana gemi arasındaki haberleşme ise RF bandında gerçek zamanda rahatlıkla yapılabilecektir (Şekil 1.1). Dolayısı ile, sınırlı yatay mesafede ve kablolu olarak çalışma problemlerinin ikisi birden çözülmüş olacaktır.



**Şekil 1.0.1 Sualtı-suüstü sistemi**

Raporun takip eden bölümlerinde projede genel olarak neler yapıldığı konusunda biraz daha detaylı bilgi verip sonuç bölümünde neler elde ettiğimiz ve nelere henüz ulaşamadığımız konusunda bilgi vereceğiz. Rapor haricinde daha detaylı bilgilere “ek” bölümlerinde ve bazı tez çalışmalarında ulaşabiliriz.

Aracın üzerindeki mekanik, elektronik ve yazılım çalışmaları tamamlandıkça çeşitli deneyler yapılmıştır. Bu deneyler öncelikler “sızdırmazlık” çalışmalarından başlamıştır. Sızdırmazlık garanti altına alındıktan sonra ise yüzerlik denge testleri yapılmıştır. Daha sonra birçok kez navigasyon ve görüntü alma ve görüntülü navigasyon deneyleri gerçekleştirilmiştir. Bu deneylere ait bilgiler ve görüntüler projenin ekinde verilen DVD içinde bulunmaktadır. Ayrıca proje CD’si içinde, proje ile ilgili resimler, sunuşlar ve projeden çıkan makaleler vardır.

DeneySEL çalışmalar, projenin donanım ve yazılım çalışmaları ilerledikçe sürekli olarak yapılacaktır. Deniz ve derin su şartlarındaki deneyler, Samsun 19 Mayıs Üniversitesi’nin tesisleri kullanılarak yapılması planlanmıştır. Bu arada Nisan 2010 ayı içerisinde, ODTÜ, Çankaya ve İTÜ’de yaptığımız çalışmalarla ilgili davetli sunuşlar yapılmıştır.



## 2. GENEL BİLGİLER

### 2.1 Proje önerisinde neler var?

ULİSAR projesinde, öncelikle ne amaçlandığını bu bölümde vermeye çalışacağız. Projede amaçlanan insansız ve kablo bağlantısı olmadan hareket edebilen bir sualtı aracı ilk-örneğini geliştirmektir. Geliştirilecek olan ilk-örneğin doğrudan gerçek hayatta kullanılacak şekilde ve boyutlarda olmasına gerek yoktu. Anladığımız kadarı ile kullanılmakta olan benzeri sualtı araçlarında boyut üzerindeki en önemli faktör enerji depolamakta kullanılan aküler, mühimmat, vs gibi yüklerin kapladığı alandır. Doğal olarak ta kullanılan elektrik motorlarının güçleri ve boyutları büyümektedir. Projenin olabilirliği açısından en kritik noktalar haberleşme ve her şeyden önce minimum istekleri sağlayan bir sualtı aracı olduğu için mümkün olduğunca boyutları küçük tutmakta yarar var diye düşünülmüştü. Ayrıca depolama, taşıma gibi sorunlarla karşılaşılmamış olacaktık. Bu düşünceler ışığında, araçla ilgili istek ve düşünebildiğimiz bazı özellikleri şöyle sıralamıştık:

- “Boyutları 1-1.5 metre uzunluğunda, 30-50 cm eninde olmalı, ağırlık 50 kiloyu fazla geçmemeli,
- Kablosuz iletişim için sonar tabanlı haberleşme sistemlerine sahip olmalı,
- Saniyede, en azından 100x100 pixel siyah-beyaz görüntüyü ana gemiye iletebilmeli (standart otopilot, navigasyon, sensor, vs., bilgilerinin haricinde). Aracın üzerinde “image compression” yapılabileceğini kabul edebiliriz.
- İletişimi rahatlatmak ve bazı başka özellikleri sağlayabilmek için, sualtı aracı ile aynı dik doğrultuda hareket etmesi düşünülen bir suüstü aracı olmalı,
- Suüstü aracının sonar+rf haberleşme sistemlerine sahip olması lazım. Sualtı aracı ile suüstü aracının en azından 200 m mesafe içinde haberleşebilmeleri gerekli,
- Bu araçların 30-60 dakika arasında çalışabilmeleri gerekli,
- Bu araçların hızları minimum 5 km/h olmalı,
- Yeterli paslanmazlık, dayanıklılık ve izolasyon özellikleri sağlanmalı,
- Mümkün olduğunca modüler bir tasarım sağlanmalı,
- (en azından) 50 metreye inebilmeli,
- Ana gemiyi temsilen bir komuta merkezi oluşturulmalı,
- Doğal olarak, çeşitli kameralar, sensörler taşınmalıdır. Kameralarla beraber çalışacak ışıldaklar olmalı. Projenin belki ilerideki aşamalarında, yük taşıma, robot kolu tasarımı gibi konulara girilmesi düşünülebilir.
- Otopilot ve navigasyon sistemlerinin tasarlanması gerekli. Ayrıca oldukça “akıllı” bir araç haline getirmemiz lazım. Mesela “pattern

## 2.2 Projede neler yaptık?

Projenin önerisinde belirtilen sistemi hemen hemen gerçekleştirememiz mümkün olmuştur. Ancak, daha önceden belirtildiği gibi, sualtı aracı ile suüstü aracı arasındaki iletişim fiber üzerinden yapılmaktadır. Ayrıca, maddi imkânsızlıklar sebebi ile şimdilik suüstü aracı yapılamamıştır.

Projede yapılan çalışmalara gelince, bunlar aşağıdaki gibi özetlenebilir (Bu çalışmaların hemen hepsi önceden raporlanmıştır):

1. **Elektronik kartların tasarımı:** Bazıları küçük olmakla beraber 12 adet kart tasarımı yapılmıştır. Son derece profesyonel bir iş çıkarılmıştır. Dolayısı ile zaman almıştır. Kartların üretimi bu konuda uzmanlaşmış özel bir şirkette yapılmıştır.
2. **Sürücülerin yazılması (gömülü sistem çalışmaları):** Elektrik motorları, kameralar, akustik alıcılar, diğer ölçü aletleri, sonar ve diğer aletlerin sürücü yazılımlarının yazılması işi yapılmıştır. Emek yoğun bir çalışma alanıdır. Projenin başından sonuna kadar devam etmiştir. Sistemde yapılan bazı değişiklikler yüzünden bazı sürücülerin birkaç kez yazılması gerekmiştir.
3. **Gövde tasarımı:** Aslında kompozit malzemeden olmasına gerek yoktur. Ancak projenin başlangıcında, Deniz Kuvvetlerinin insansız denizaltı ihalesinin ODTÜ'nün başını çektiği gruba verileceğinin söylenmesi veya hissettirilmesi dolayısı ile (aslında SSM'da yapılan toplantıda proje sunumu yapıldıktan sonra ihalesiz olarak projenin ODTÜ yönetimindeki gruba (ODTÜ, Aselsan, Havelsan, Barış Elektrik) verildiği söylenmiş ve anlaşmanın imzalanması beklenmeye başlanmıştı) gövdenin kompozit malzemeden yapılmasına karar verilmiştir. Bu seçim tasarımı çok etkilemiş (fiziksel ve maddi zorluklar) ve sonuçta şu anda ortaya çıkan gövde elde edilmiştir. Gövdenin şekilsel tasarımına paralel olarak, tasarımın hidrodinamik ve ısı incelenmesi tamamlanmıştır. Daha sonra, bir yandan sızdırmazlık çalışmaları sürdürülürken, öte yandan elektronik aletlerin gövdeye yerleştirilmesi konusunun üzerinde çalışılmış ve yerleştirme ile alet kutusu tasarımları elde edilmiştir.
4. **Araç içi veri iletişimi:** Aslında çok fazla adam-saat alabilecek bu çalışma konusu, RHEX isimli yürüyen robotta kullanılmak üzere geliştirilmiş bir üst düzey dil olan "URB" (universal robot bus) kullanılarak ve başka bir proje ile iş yükünü paylaşarak halledilebilir bir seviyeye indirilmiştir.
5. **Modelleme ve otopilot çalışmaları:** Aracın yaklaşık bir matematiksel modeli elde edilmiş ve bu model üzerinden otopilot yazılımları yapılmıştır. Bu çalışmalar doğrudan proje için gerekli olmasa bile sonrasında güdümlü hareketin sağlanması için gerekecektir. Ayrıca "yol planlama" ve "görev planlama" türü çalışmalarda yapılmıştır.

6. **Navigasyon çalışmaları:** Bilhassa sualtı aracının durum ve bilhassa davranış bilgisinin elde edilmesi, aracın kullanımı açısından çok önemlidir. Araç üzerindeki INS birimi ile birlikte çalışacak manyetik alıcılar ve diğer destekleyiciler için gerekli süzgeçleme çalışmaları yapılmıştır. Bu konudaki işlerin tamamen bittiğini söyleyemeyiz.
7. **Test çalışmaları:** Gömülü sistem çalışmaları, gövde tasarımı ve sızdırmazlık, navigasyon çalışmaları konularında çok sayıda test yapılmıştır. Bu şekilde ortaya çıkan ürünlerin güvenilirliği birbiri ile beraber çalışabilirliği denenmiştir. Ayrıca son 8 aydır, gövde çalışmalarının en önemli parçası olarak, ODTÜ’nde gerçekleştirdiğimiz havuz testleri vardır. Testlere devam edilecektir.

Aracın üzerindeki mekanik, elektronik ve yazılım çalışmaları tamamlandıkça çeşitli deneyler yapılmıştır. Bu deneyler öncelikler “sızdırmazlık” çalışmalarından başlamıştır. Sızdırmazlık garanti altına alındıktan sonra ise yüzerlik denge testleri yapılmıştır. Daha sonra birçok kez navigasyon ve görüntü alma ve görüntülü navigasyon deneyleri gerçekleştirilmiştir. Bu deneylere ait bilgiler ve görüntüler projenin ekinde verilen DVD içinde bulunmaktadır.

Deneysel çalışmalar, projenin donanım ve yazılım çalışmaları ilerledikçe sürekli olarak yapılacaktır. Deniz ve derin su şartlarındaki deneyler, Samsun 19 Mayıs Üniversitesinin tesisleri kullanılarak yapılması planlanmıştır.

8. **Proje yönetimi:** Temel olarak haftalık toplantılar düzenlenmiştir. Bazen daha küçük gruplar halinde belli çalışma gruplarına yönelik toplantılar organize edilmiştir. Ayrıca proje evraklarının ve faturaların düzenlenmesi, yazışmaların yapılması ve herşeyden önemlisi raporların yazılması gibi işler yapılmıştır.
9. **Diğer çalışmalar:** Kamera kalibrasyon işleri, başka bir projede çalışan bir arkadaşımız tarafından halledilmiştir. Aracın havuza indirilip çıkarılması sırasında büyük sorunlar çıkmış ve bu sorunlar hareketli bir vinç tasarımı- inşaatı sayesinde halledilmiştir. Sualtından alınacak görüntüler üzerinden yapılacak yol takibi, görüntü iyileştirme ve benzeri çalışmalar, internet ortamından elde edilen verilerle yapılabilmektedir. Şu anda sualtı kamerası ve sonar görüntülerinin birleştirilmesine yönelik çalışmalar yapılmaktadır.

Çok geniş bir proje olduğu için yapılan bütün farklı çalışmaları kısa zamanda bir araya getirmek kolay olmamıştır. Ayrıca bu çalışmalardan farklı olarak, ortaya çıkarılan ürünün TOBB ve Çankaya Üniversitelerindeki bazı toplantılarda, ODTÜ’nde yer alan “Deniz Sistemleri’2009” toplantısında ve Uluslararası IDEF’2009 silah fuarında sunumu yapılmıştır. Nisan 2010 ayı içerisinde, ODTÜ, Çankaya ve İTÜ’de yaptığımız çalışmalarla ilgili davetli sunumlar yapılmıştır.

### **2.3 Projede yapılan işlerin sınıflandırılması**

ULİSAR projesinde, çok farklı konularda çeşitli çalışmalar yapılması gerektiğinden, yapılan/yapılacak işleri belli gruplara ayırmak gereksinimi doğmuştur. Bu gruplar aşağıda verilmiştir:

1. Gövde grubu,
2. Yazılım grubu,
3. Donanım grubu,
4. Modelleme ve navigasyon grubu,
5. Diğer çalışmalar grubu.

### **2.4 Projede yapılan işler nerede kimler tarafından yapıldı?**

Projemize, başta ODTÜ Elektrik ve Makine Mühendisliklerinden öğretim üyeleri, asistanlar ve yüksek lisans ve doktora öğrencileri olmak üzere çok sayıda kişi, az ya da çok katkıda bulunmuştur. Ayrıca, ODTÜ-Teknokent firmalarından Yenel Limited ve kompozit gövde imalatının yapıldığı Barış Elektrik projede çok önemli çalışmalar yapmışlardır.

Proje çalışmaları ODTÜ, Elektrik Mühendisliği “robotics” laboratuvarında yer almıştır. Havuz testleri çoğunluğu ise ODTÜ, büyük yüzme havuzunda yapılmıştır. Proje toplantıları bazen robotics laboratuvarında, bazen de Elektrik Mühendisliğinin seminer odalarında yapılmıştır. Gövde imalatı Barış Elektrik Firmasının tesislerinde yer almıştır. Havuz vinci ise Ostim’de bir şirkete yaptırılmıştır.

Genel olarak belli iş gruplarından belli kişiler sorumlu tutulmuştur. Bunun detayları aşağıda verilmiştir:

1. **Elektronik kartların tasarımı:** Hüseyin Yiğitler, Emre Ege, Alper Kılıç, Yenel Limited.
2. **Sürücülerin yazılması (gömülü sistem çalışmaları):** Hüseyin Yiğitler, Emre Ege, Alper Kılıç, Örsan Aytekin, Afşar Saranlı.
3. **Gövde tasarımı:** Barış Elektrik, Necmettin Cevheri, Kemal İder, Levent Parnas, Haluk Aksel, Hüseyin Yiğitler, Mine Cüneyitoğlu, Kemal Leblebicioğlu.
4. **Araç içi veri iletişimi:** Hüseyin Yiğitler, Emre Ege, Afşar Saranlı.
5. **Modelleme ve otopilot çalışmaları:** Kadir İsiyel, Kemal Leblebicioğlu, Kemal Özgören, Koray Küçük.
6. **Navigasyon çalışmaları:** Hüseyin Yiğitler, Kemal Leblebicioğlu.
7. **Test çalışmaları:** Kemal Leblebicioğlu, Emre Ege, Barış Elektrik, Örsan Aytekin, Mine Cüneyitoğlu, Alper Kılıç, Seda Karadeniz.
8. **Proje yönetimi:** Kemal Leblebicioğlu, Emre Ege, Hüseyin Yiğitler.
9. **Diğer çalışmalar:** Afşar Saranlı, Emre Ege, Gözde Bozdağı Akar, İlkey Ulusoy, Halit Oğuztüzün, Yavuz Kırılı, Murat Deniz Aykın.

## 3. GEREÇ VE YÖNTEM

Proje çalışmalarından belli gruplara dahil olanlarının daha detaylı açıklamalarını bu bölümde bulabiliriz. Özellikle projenin son 6 ayı içinde yapılan çalışmalar burada vurgulanmıştır.

### **3.1 Mekanik Tasarım Ve Üretim İle İlgili Gelişmeler**

Bu bölümde mekanik tasarım ve üretim ile ilgili son 8 ayda yapılan çalışmalar anlatılmıştır. Bu kısımlarla ilgili fotoğraf ve şekiller “gövde resimleri” isimli dosyanın içindeki çeşitli alt dosyalarda CD’de verilmiştir. Bu alt dosyalarda, gövde ve elektronik kartlar kutusunun yapımı ile ilgili hemen bütün aşamaları izlemek mümkündür.

#### **3.1.1 Barış Elektrik firmasına yapılan ziyaretler**

Mart, Nisan ve Mayıs aylarında çoğunlukla haftalık bir ya da iki defa olmak üzere Barış Elektrik firmasına gidilerek araç üretiminin takibi yapıldı. Çeşitli parçaların montajı araç üzerinde konuşulup tartışılarak netleştirildi. Mekanik tasarım ile yapılan üretim arasındaki farklılıklar konuşuldu ve aracın çizimleri bu doğrultuda yeniden güncellendi. Su sızdırmaz kabinin üretilmiş halinin ölçüleri alındı ve bu ölçüler elektronik kart kutusunun en verimli olabilecek şekilde yeniden tasarımında kullanıldı.

#### **3.1.2 Akustik modem için fiberglas plakadan yuva yapımı ve bu plakanın sabitlenmesi işlemi**

Akustik modem ekipmanının ölçüleri alınarak bu parçanın rahatlıkla oturabileceği ve taşınabileceği bir plaka hazırlandı. Önce katı modele bu plaka eklendi ve parçanın gövdede nasıl durduğuna bakıldı. Daha sonra plaka üretildi ve denizaltının su alan ön kısmına kompozit malzeme yapıştırıcı epoksi ile monte edildi.

#### **3.1.3 Motorların gövdeye sabitlenmesi**

Yan kanatlar altındaki yuvalara ve su alan bölmelerdeki kısımlara dik olarak açılan yuvalara motorların civataları için uygun delikler açıldı ve motorlar kompozit gövde üzerinde tasarlanan yerlerine monte edildi.

### 3.1.4 Elektronik kart kutusunun alt çerçevesinin gövdeye kalıcı olarak yerleştirilmesi

Elektronik kart kutusunun yeniden tasarlanıp büyütülmesi kısmında detaylı olarak anlatılan kart kutusunun alt çerçevesi denizaltının su sızdırmaz kısmına iki yandan ölçüyü tam ortalayacak şekilde epoksi ile sabitlendi.

### 3.1.5 Kamera ve kamera ışığı oyuklarının açılması bu ekipmanın açılan yuvalara yerleştirilmesi

Aracın ön kısmına kamera ve kamera aydınlatma ışıkları için uygun boşluklar açıldı. Bu ekipmanlar açılan boşluklara soğuk silikonla sabitlendi. Bu sabitleme yöntemi ön yüzey geometrisinin düz yüzeylerden oluşmamasından dolayı tercih edildi. Bu şekilde aracın en ön kısmındaki boşluklar soğuk silikonla dolduruldu. Mekanik destek sağlaması için ayrıca altlardan ve yanlardan bu parçalar köpükle desteklendi. Böylece kamera ve aydınlatma ekipmanının sarsılmazlığı sağlanmaya çalışıldı (Şekil 3.1).



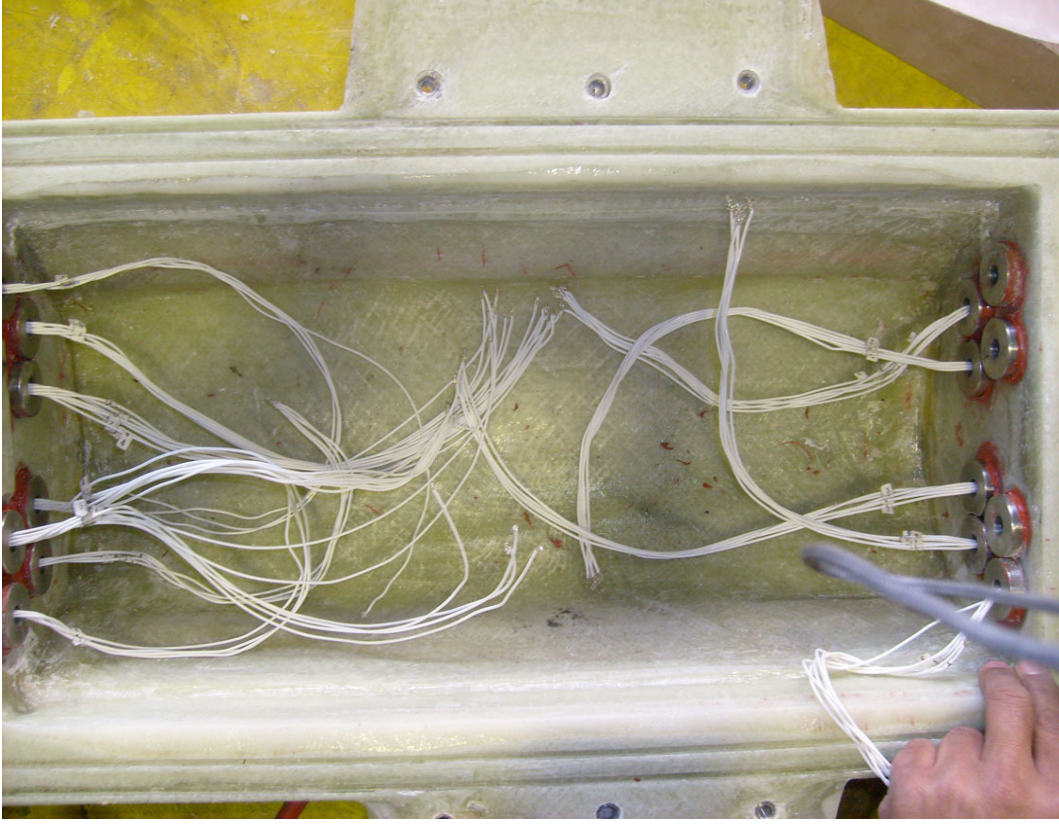
Şekil 3.0.1 Kamera ve projektör yerleşimi

### 3.1.6 Konnektörlerin takılması ve bu parçaların sızdırmazlığının sağlanması

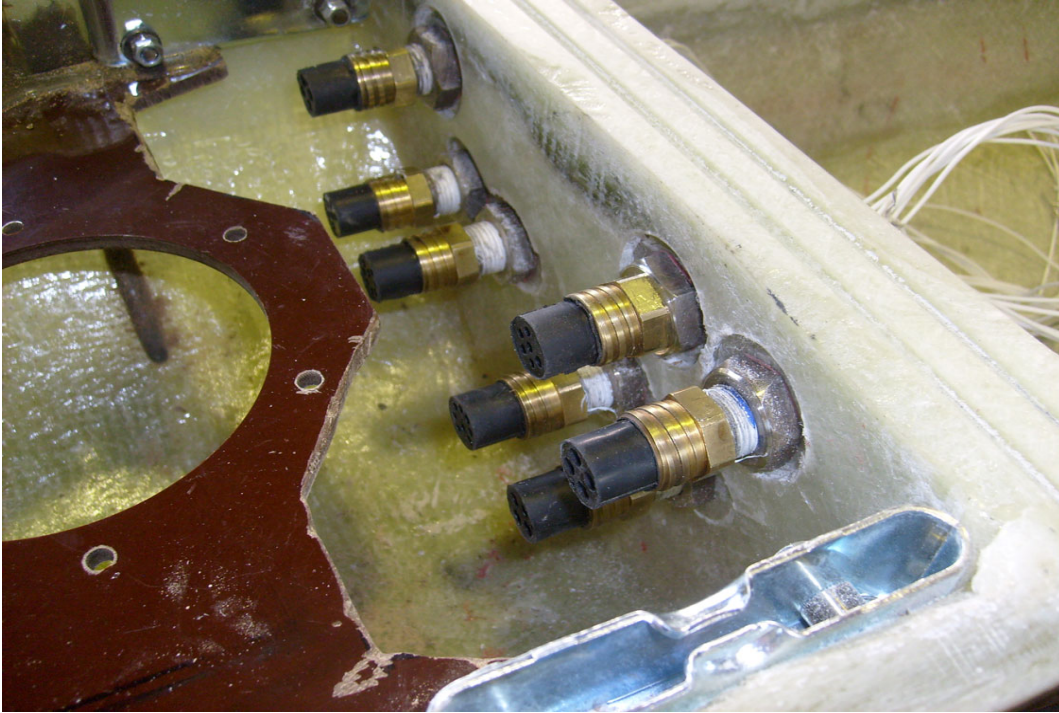
Bu işlem beklenilenden daha fazla süre aldı. Su sızdıran ve sızdırmayan bölümler arası duvarların et kalınlığı, katı modelde 15 mm olarak çizilmişti. Bu tasarıya göre su sızdırmayan özel su altı konnektörleri iki yana contaları denk gelecek şekilde uygun olarak yerleşiyor ve sızdırmazlık sağlıyordu. Ancak Barış Elektrik firmasında üretim yapılırken bu duvarlar 40 mm et kalınlığında üretildi. Bu değişikliğin sebebi, üretim yapılırken firmada kalınlığı 10 mm olan köpük yerine 15 mm olan köpükten bulunuyor olması olarak açıklandı. Ayrıca 15 mm köpük ile su sızdırmayan bölge genelinde daha iyi sızdırmazlık sağlanacağı düşünüldüğü belirtildi.

Ancak bu değişiklik sonucu 40 mm olarak üretilen yan duvarlara konnektörler monte edilemedi ve bu montaj için tornalama işlemi ile çelikten contaların oturabileceği yüzeylerin bulunduğu ara parçalar üretildi. Bu ara parçalar konnektörleri içine aldığı için her bir konnektörün toplamda kapladığı alan artmış oldu. Bu bulunan çözüme göre her bir konnektörün yuvasının gövdenin iç kısmına geçtiği yerlerde oldukça yüksek düzgünlüğe sahip yüzeyler gerekmektedir. Çalışma alanının darlığı nedeniyle (yaklaşık olarak 125 mm yarıçaplı yarımdaire içerisine 8 adet 40 mm çaplı daireler oturtma işlemi) bu işlem işçilik açısından oldukça zorlayıcı oldu.

Montaj esansında ara parçaların dış kısmına teflon bant sarılarak sızdırmazlığa katkıda bulunuldu. Teflon bant oldukça ince ve pürüzsüz yapıda olduğu için tornalama ya da delik açma işlemi sırasında oluşmuş olabilecek pürüzleri doldurabileceği için tercih edildi. Yapılan uygulama araştırmaları sonucunda bu bantın sızdırmazlık uygulamalarında sıklıkla kullanıldığı görüldü. Daha sonra üretilen bağlantılara sızdırmazlık testi yapıldı. Fakat ilk denemeler sonucu başarılı olunamadı. Daha fazla teflon bant sarılarak ve konnektör ara parçası kenarları özel sızdırmazlık macunu ile kaplanarak yapılan denemeler sonucunda başarıya ulaşıldı. Ancak yukarıda da belirtildiği gibi bu işlem planlanan zamandan çok daha uzun süre aldı (Bakınız: Şekil 3.2, Şekil 3.3).



**Şekil 3.0.2** Sızdırmaz bölge kablolama



**Şekil 3.0.3** Konnektörlerin yerleşimi



### **3.1.7 İki yana saplamaların yapılması**

Hem aracın taşınması esnasında mekanik destek sağlaması açısından hem de alt ve üst gövdeler arasında sızdırmazlığı sağlamak için kullanılacak M8 saplamalar gövdenin su sızdıran iki kısmının radyal yönde ortasına, aksenel yönde ise sızdırmaz kısma yakın ve duvarlara baskı yapacak şekilde yerleştirildi. Üst gövde monte edilip araç kapatılırken bu saplamalar da sıkılarak alt ve üst gövde arası daha sıkı bir bağlantı elde edilmektedir. Bu saplamaların ucuna takılan halkalı somunların halka kısmından ise araç asılabilmekte, bu da suya nakliyat esnasında oldukça kolaylık sağlamaktadır. Bu bağlantının yükünü alt gövde taşıdığı için aracın ağırlığı ve olası yüklemelere karşı yeterince dayanıklı bir mekanik tutanak oluşturulmuştur.

### **3.1.8 Mesafe algılayıcılarının yerleşimi**

Basınç algılayıcılar için gövdenin gerekli yerlerine delikler açılmış, bu sensörler yine soğuk silikonla gövdeye sabitlenmiştir.

### **3.1.9 Alt gövdeye suyun dolup boşalması için deliklerin açılması**

Aracın su alan kısımlarına suya batmada ve sudan çıkarken iç hacimdeki suyun boşalmasında kolaylık sağlayacak delikler açılmıştır. Bu deliklerin çevrelerine görselliğin bozulmaması ve ayrıca suyun akışının kolaylaşması için plastik parçalar geçirilmiştir.

### **3.1.10 Çizimlerin üretilen gövdeye göre yenilenmesi**

Yukarıda su sızdıran kısım ile sızdırmaz kısım arasındaki konnektörlerin bağlantısının anlatıldığı kısımda detaylı olarak anlatılan duvar et kalınlığı değişimi ölçülerek katı modele aktarıldı. Sızdırmaz kısım biraz küçülmüş olduğundan iç hacim bu yeni ölçülere göre yeniden düzenlendi.

### **3.1.11 Elektronik kartlardan çıkan önemli konnektörlerin bilgisayarda modellenmesi**

Konnektör bağlantıları göz önünde bulundurulmadan yapılan bir kutu tasarımının olası sakıncaları düşünülerek ekipler arası iletişim artırıldı. Elektronik kısımları tasarlayan ekiple uzun süreli iki toplantı yapılarak elektronik kartların mekanik bağlantı detayları konuşuldu. Kabloların geçeceği yerler planlanmaya çalışıldı.

### **3.1.12 Elektronik kart kutusunun yeniden tasarlanarak büyütülmesi, yeni kutunun tasarımı ve plakalarının üretimi**

Yapılan toplantılar sonucu katı model olarak bir ön tasarım çıkarıldı ve elektronik ekiple bunun üzerinde konuşuldu, revizyonlar yapıldı. Sağladığı tasarım esnekliği ve mekanik açıdan güvenilirliği yüksek olması açısından elektronik kart kutusunun dış hatlarının yine 20x20 sigma profilden yapılmasına karar verildi. Sızdırmaz hacmin en verimli şekilde kullanılmasına özen gösterildi. Bunun için olabilecek her yere elektronik devre konulabilecek yüzeylerin monte edilebileceği esnek bir yapı sağlanmaya çalışıldı. Barış Elektrik'te üretilen araçtan alınan silindirik hacim ölçülerini az bir pay bırakarak dolduran en büyük dikdörtgenler prizması hacmi bulundu ve kutu bu ebatlarda (dıştan dışa 160 x 210 x 380 mm) tasarlandı.

Konnektörlerin geçiş noktaları ve de parçaların olabildiğince hafifletilmesi göz önüne alınarak içinde boşluklar olan plakalar tasarlandı. Bu plakaların rahatlıkla sabitlenebilmesi için ara parçalar ve saplamalar kullanıldı. Yapılan tasarımdaki plakalar içlerindeki boşluklar ve gerekli boşlulardan düzgünce geçme gereksinimi nedeniyle Telmek A.Ş.'de cnc kontrollü lazer tezgahlarla üretildi. Montaj plakalarının üzerindeki çeşitli delikler ve sigma profilin yapısı sayesinde oldukça esnek bir tasarım hayata geçirilmiş oldu.

### **3.1.13 Uzun saplamalar kullanılarak kutunun montajının basitleştirilmesi**

Ancak yapılan kutu tasarımının araca takılıp çıkarılması işlemin bir süre boyunca sıklıkla tekrarlanacağı düşünülerek bu işlemin daha pratik hale getirilmesi düşünüldü. Ayrıca Barış Elektrik firması da çizimlerde görülen sızdırmaz kısmın iki yanındaki alüminyum profil gömme işlemini gerçekleştiremeyeceğini belirtti. Bunun sebebi daha önceden fazla yükseklik gerektirmeden (5 mm gibi) yapılabileceğini düşündükleri plaka sabitleyici somun sabitleme işleminin aslında oldukça miktar yükseklik gerektirdiğini (yaklaşık 50 mm) fark etmeleri olarak açıklandı. Buna ek olarak da bu işlemi sadece ortadaki düz bir yüzeye uygulayabileceklerini belirttiler. Bunun üzerine bir grup halinde firmaya gidildi ve kutu sabitleme sorununa çözüm arandı.

Çözüm olarak sigma profil köşe çatmasının değiştirilerek kutunun dört yanından boyu boyunca profil içerisindeki silindirik kanallık içinden geçebilen M4 saplama ile aracın tabanına sabitlenebilecek bir yapı oluşturuldu.

### **3.1.14 Somunların Barış Elektrik'e gönderilecek çerçevenin içine geçirilmesi**

Yukarıda anlatılan profil içinden geçen M4 saplamaların somun olarak karşılıkları gövde içerisine kalıcı olarak monte edilecek olan sigma profil çerçevesinin içine geçirildi. Bu işlem için değişik yöntemler arandı ve en sağlam olanını

bulunmaya çalışıldı, çünkü somunlar gövdeye kalıcı olarak gömülecek çerçevenin alt kısmında kalacaklardı ve zaman içerisinde yerlerinden oynamamaları gerekmekteydi.

Bu işlem için en uygun yöntem olarak somunları vinyl plakalara gömme işlemi seçildi. Maket bıçağı yardımıyla profil kanalından geçebilen ince uzun ve 2.5 kalınlığında vinyl plakaya altıgen biçimli somunun gömüleceği altıgen bir yuva açıldı. Somunun bu yuvalar içerisinde dönmediğinden emin olunacak şekilde kesim yapıldı. Yuvalara oturtulan somunların dönmemesi el ile kontrol edildi. Daha sonra, ekstra güvenlik önlemi olması amacıyla somunlar bu yuvalara arada kuvvetli bir yapıştırıcı olacak şekilde kalıcı olarak geçirildi. Daha sonra bu plakalar sigma profilin en alt kanalına saplamaların gireceği yerlere yerleştirildi. Bu kanal somunun yukarı aşağı hareketini engelleyici bir işleve sahip oldu. Daha sonra çerçeve içerisindeki somunlu plakalarla birlikte Barış Elektrik'e götürüldü ve burada hem somunlu plakalar çerçeveye kalıcı olarak hem de çerçevenin kendisi de alt gövdeye yine kalıcı olarak yapıştırıldı.

### **3.1.15 Pillerin sabitlenmesi**

Piller motorlar ve işlemciler olarak gruplandıktan sonra çerçevenin iç kısmında piller için ayrılan yere oturtuldular. Grup halinde uygun şekilde dizildikten sonra birbirlerine cırtcirt ile tutturulan piller ayrıca alt çerçevenin kenarları tarafından da sıkıştırıldı. Daha sonra pillerin üzeri uygun yerlerinde pil kutuplarının bağlı olduğu kabloların çıkışına uygun olacak şekilde kesilmiş yalıtkan lexan malzemesinden plaka ile örtüldü. Çerçevenin dışında iki yanda kalan üçer tane pil ise bu plakaya frezede kesilen daha küçük plakalar ile tutturuldu. Büyük lexan plaka sigma profil kutuya sabitlendi. Küçük plakalar da büyük plakaya sabitlendi ve diğer yanda da gövdenin iki yanına dayandı. Büyük lexan plakanın üzerine pillerin bağlantısı yapılan kartlar monte edildi. Böylece pillerin yukarı aşağı hareketi tamamen engellenmiş oldu.

### **3.1.16 Suya Kolaylıkla İniş Çıkış İçin Vinç Mekanizması Tasarımı**

Denizaltının ağırlığı 70 kg dolaylarındadır ve denizaltı yüzeyi türbülansı engelleyecek şekilde tasarlanmış yuvarlak hatlardan oluşmaktadır. Bu nedenlerle denizaltının havuza indirilmesini ve de özellikle havuzdan çıkarılmasını birden fazla kişi bile güçlüklerle gerçekleştirebilmektedir. Havuza aktarım ve havuzdan çıkarma işlemlerini rahatlatmak ve ekipmana zarar vermemek için bir vinç mekanizması düşünülmüştür. Bu mekanizmanın, deneylerin yapıldığı ODTÜ havuzuna ve gereken diğer yerlere standart bir aracın bagajında taşınarak götürülmesi gerekebileceğinden mekanizma sökülüp takılabilir elemanlardan oluşacak şekilde tasarlanmıştır.

Konsept tasarımı olarak, altında sarhoş tekerler olan bir taşıyıcı araba üzerinde ters L benzeri bir uzantısı olan ve uzantıya denizaltı bağlandığında bu ağırlığı dengeleyen ağırlıkların konabileceği bir sistem düşünülmüştür. Piyasadaki vinç firmaları gezilerek bu işi görebilecek hazır sistemlerin olup olmadığı

araştırılmıştır. Elektrikli bucurgat sistemleri görülmüştür; ancak bu tip bir sistemin monte edileceği L şeklinde yapının da tasarlanıp üretilmesi gerekmektedir. Bu alternatif karşılık aşağıda katalog bilgisi verilen “Pick - Up” vincinin de gereken ölçülerde serbestlik sağladığı görülmüştür. Hem maliyet hem de zaman açısından bu vinç sistemini hazır almak bucurgatlı bir sistem tasarlamaktan daha uygun bulunmuştur.

Şekilde (Şekil 3.4) görülen vinç manuel olarak hidrolik tahriklidir ve tek serbestlik derecesine sahiptir. Ucundaki yükü taşıdığı rota denizaltını ODTÜ havuzu ve benzeri ortamlara daldırmak için uygun ölçülerdedir. Bu vincin kaldırma kapasitesi istenenden oldukça yüksektir; ancak piyasada bulunabilen en küçük model budur. Bucurgatlı sistemlerde ise yük değişmektedir; ancak fiyat farkı çok olmamaktadır. Bunun yanı sıra istenen mukavemete sahip bir L yapı oluşturup kaynatmanın ve arabaya monte etmenin maliyeti çok daha yüksek olacaktır.

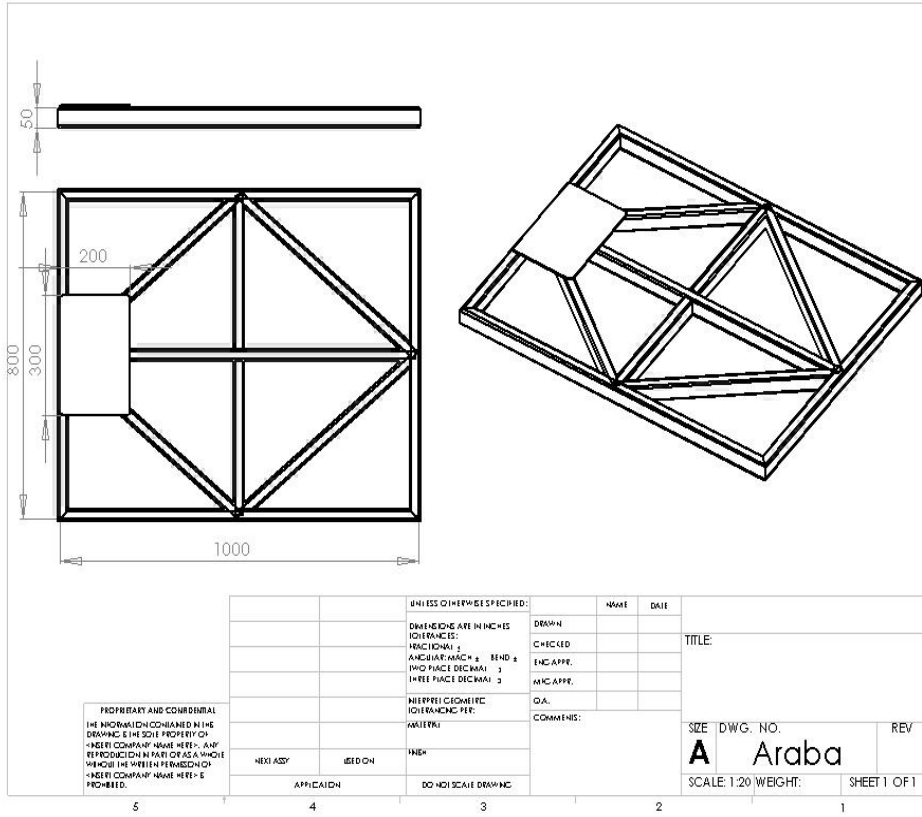
PICK-UP VİNCİ / PICK-UP CRANE / HEBEKRAN		NL-A027	170 \$
			Art.No. 59000014
Kaldırma kapasitesi / Capacity / Tragkraft	900 kg.		
Minimum yükseklik / Min. height / Min. Höhe	580 mm.		
Maksimum yükseklik / Max. height / Max. Höhe	2400 mm.		
Çalışma mesafesi / Boom length / Armlänge	870 - 1290 mm.		
Ölçüler / Dimension / Abmessungen	340 x 330 x 500 mm.		
Net ağırlık / Weight / Gewicht	52 kg.		
Brüt ağırlık / Gros weight / Bruttogewicht	55 kg.		

info@neillift.com.tr

153

Şekil 3.0.4 Hidrolik vinç

Bu vincin monte edileceği sistem katı model olarak tasarlanmış ve Duygu Makine firmasıyla görüşülerek burada üretilmiştir. Arabanın destek profilleri birbirine kaynakla tutturulmuştur. Yapıda vincin üzerine civatalanabildiği bir plaka bulunmaktadır. Destekli profil yapının (Şekil 3.5) altına ise taşıyıcı sistemler için yeterli sağlamlıkta sarhoş tekerler monte edilmiştir. Aşağıda Duygu Makine firmasına gönderilen tasarı dokümanı yer almaktadır. Bu tasarımda kullanılan 50x50 mm çelik profiller sağlamlığın garantilenmesi açısından 60x60 mm olan profillerle değiştirilmiştir. Aşağıdaki çizimde vincin montaj delikleri görülmemektedir ancak bu delikler vinç temin edilip Duygu Makine'ye nakledildikten sonra ölçü alınarak açılmıştır.



Şekil 3.0.5 Vinç tabanı

Denizaltının vince montesi alt kabuktan gelen iki adet saplamanın somunlarının uçlarındaki halkalardan olmaktadır. Birer emniyet halkası bu halkalardan geçirilir ve diğer uçlarına da birer halat takılır. Ancak bu iki noktadan gelen halatların karışmadan ve denizaltını fazla hareket ettirmeden yukarı vincin ucundaki tek zincirin ucuna takılabilmesi için iki tarafında ve üst tarafında M12 mapa bulunan üçgen şeklinde bir parça (Şekil 3.6) düşünülmüştür. Duygu Makine firmasında bu işlevi görecektek aparat da üretilmiştir. Firmaya gönderilen tasarım aşağıda verilmiştir ancak firma probleme biraz daha farklı bir çözüm getirmiştir. Altta ve ortadaki profil kalınlaştırılmış ve üçgen formu veren destek parçaları ortadan kaldırılmıştır. Bu üretim kolaylığını artırmıştır. Getirilen çözüm de işlevi yerine getirmede başarılı olmuştur.





**Şekil 3.0.8** Vinç ve denizaltının başka bir görünümü



**Şekil 3.0.9** Denizaltı taşıyıcısından vince aktarım



**Şekil 3.0.10** Denizaltıyı vinç ile havuz kenarına yaklaştırma işlemi



**Şekil 3.0.11** Denizaltıyı vinç ile havuza indirme işlemi



### **3.1.17 Gövde ile İlgili Hidrodinamik ve Isıl Analizler**

Bir insansız sualtı aracının kavramsal tasarımı esnasında gerçekleştirilen hidrodinamik ve ısıl analizler bu bölümdeki çalışmanın ana konusunu oluşturmaktadır. Ayrıca proje elemanlarından Necmettin Cevheri bu konuda, Prof. Dr. Haluk Aksel danışmanlığında bir yüksek lisans tezi bitirmiştir. Çok detaylı olarak yapılan çalışmalar, CD içindeki ekler bölümündeki “tezler” dosyasının altında bulunabilir.

Gövdenin şekline çeşitli sualtı aracı şekillerinin değerlendirilmesi sonucu karar verilmektedir. Boyutlar ise cihazların gövde içine yerleşiminden sonra ortaya çıkmaktadır. Elektronik ekipmanın bulunduğu bölmenin ısınmasına dair analizler ticari bir hesaplamalı akışkanlar dinamiği yazılımı olan FLUENT yardımı ile yapılmaktadır.

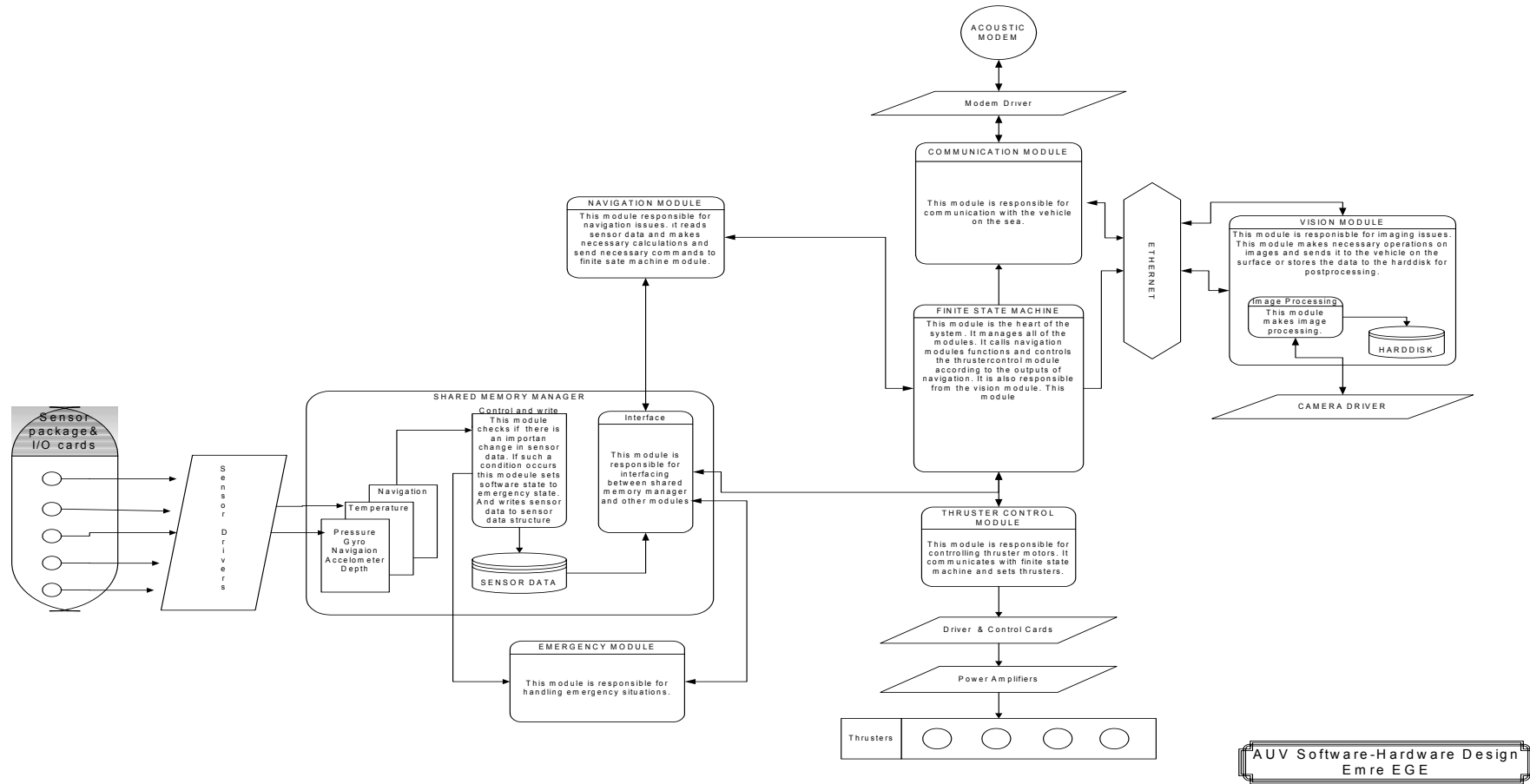
Önerilen tasarımın performans analizleri de FLUENT ile yapılmaktadır. Analizlerden önce bazı doğrulama çalışmaları gerçekleştirilmiştir. Bu çalışmalarda “Series 58” adı verilen bir grup gövdeden 4159, 4158 ve 4154 modelleri deneysel referans olarak alınmaktadır. Bu modellerin toplam direnç katsayıları yapılan analiz sonuçları ile kıyaslanmaktadır. Bu analizler esnasında ağ yoğunluğu, türbülans modeli, duvar yakınındaki akışı modelleme yöntemleri ve giriş türbülans yoğunluğu değiştirilerek çözümlere etkileri gözlemlenmektedir. Uygun bir türbülans modeli seçilerek aracın ileri ve yukarı yönde hareketleri incelenmekte ve verilen tasarım kriterlerini sağlaması şartı kontrol edilmektedir. Gövde çıkıntılarının dahil edildiği ve edilmediği modeller yaratılarak bu çıkıntıların toplam sürüklenme kuvvetine etkileri gözlemlenmektedir. Analizler sonucu gövdenin istenen hız kriterlerini sağladığı ve sızdırmaz orta bölümde aşırı ısınma oluşmadığı sonucuna varılmıştır.

## **3.2 Yazılım Çalışmaları**

Bu raporda ULİSAR projesi 6. döneminde yapılan çalışmalar anlatılmaktadır. Bu raporda önceki dönemlerde geliştirilen yazılımların hangi bilgisayarlara, nasıl derlenip yükleneceği hakkında bilgi verilmiştir. Yazılım grubu kısmındaki önceki dönemlerde yazılması gerekli yazılımlar yapıldığı için bu dönem içerisinde daha çok geliştirilen donanımlarla yazılım bütünleşme sağlaması çalışmaları yapılmıştır. Bütünleşme çalışmalarında bazı ufak tefek problemler dışında herhangi bir sorun ile karşılaşılmamıştır. Ancak laboratuvar denemelerinden sonra yapılacak olan havuz denemeleri öncesinde elimizdeki motorların sürüş devrelerinde çıkan arızalardan dolayı havuz denemeleri tam olarak yapılamamıştır. İlerleyen bölümlerde gerekli yazılımların derlemesinin ve yüklemesinin nasıl yapılacağı, bilgisayarlar ile elektronik kartlar arasındaki bağlantıların nasıl yapılacağı anlatılmıştır.

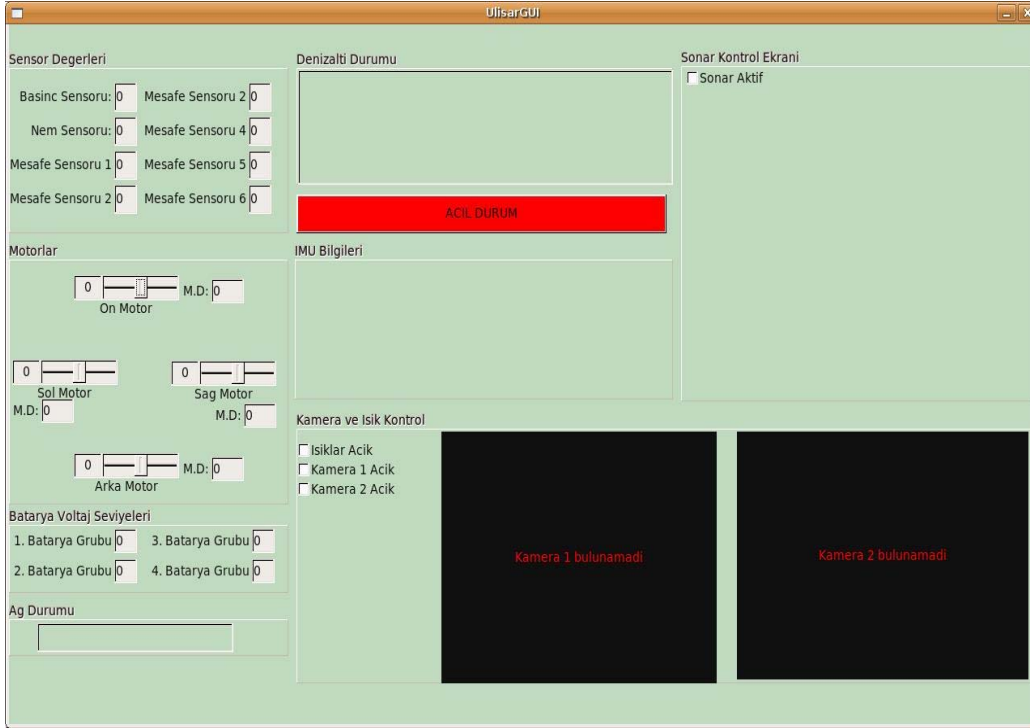
### **3.2.1 Yazılım Raporu Özeti**

Bu bölümde daha önce yapılan yazılım çalışmalarının bir kısmının kısa bir özeti verilmiştir. ULİSAR projesindeki yazılım çalışmaları genel olarak 3 kısımdan oluşmaktadır. Bunlar görüntü bilgisayarı yazılımı, denetim bilgisayarı yazılımı ve operatör bilgisayarı yazılımıdır. İlk olarak düşünülen yazılım tasarımı Şekil 3.7 de verilmiştir.



Şekil 3.0.12 İlk yazılım tasarımı

Ancak bu tasarım diğerk bir TÜBİTAK projesi olan SensorHex’de de kullanılan RhexLib kütüphanesinin kullanılmasına karar verildikten sonra değıştirilmiştir. Her iki projede de ortak olması amacı ile RhexLib kullanılmıştır. RhexLib kütüphanesi oldukça modüler bir özelliğe sahiptir. Yazılan modüller istenildiğı zaman kolayca çalıştırılıp durdurulabilir. Projede kullanılan denetim bilgisayarında QNX işletim sistemi, görüntü bilgisayarında DamnSmallLinux, operatör bilgisayarında ise UBUNTU 9.04 işletim sistemi bulunmaktadır. Denizaltıyı uzaktan kumanda etmek için kullanılan operatör yazılımındaki kullanıcı arayüzü Şekil 3.0.138 de verilmiştir.



Şekil 3.0.13 Operatör kullanıcı arayüzü

Bahsi geçen yazılımların nasıl derlenip yükleneceğı hakkındaki detaylı bilgi Ulisar Rapor6-HW-SW.doc adlı dosyada verilmiştir.

### 3.2.2 Denetim Bilgisayarı

Denetim bilgisayarı denizaltının kontrolü ve operatör bilgisayarının haberleşmesinden sorumludur. Denetim bilgisayarında aynı anda birçok işlem yapılacaktır ve bu işlemlerin gerçek zamanda yapılabiliyor olması gerekmektedir. Bunun yanında kullanılan yazılımın modüler olması gerekmektedir. Kullanılacak olan yazılıma istenilen özellikler yazılım bittikten sonra dahi kolay bir şekilde eklenip çıkartılabilmelidir.

Denetim bilgisayarı yazılımında yukarıda bahsedilen özelliklere sahip olan RHexLib kütüphanesi kullanılmaktadır. RHexLib yine bir TÜBİTAK 1001 projesi olan SensorHex projesinde geliştirilmiştir. URB temel olarak 3 öğeden oluşmaktadır. Bunlar bilgisayar, köprü (bridge) ve düğümdür (node). Bilgisayar

denetim yazılımının çalışacağı ve çevre birimlere komut gönderen/alan kısımdır. Bridge bilgisayardan alınan komutları düğümlere yollar, düğümden aldığı bilgileri de bilgisayara yollar. Düğümler çevre birimlerin bağlı olduğu mikroişlemcilerdir. Köprüden aldığı komutları işler ve eğer bir cevap isteniyorsa cevabı hazırlayarak köprüye yollar. ULİSAR projesinde köprü devresindeki mikroişlemci Silicon Lab. C8051F342 olarak seçilmiştir.

Köprü devresi ile bilgisayar arasında USB veriyolu kullanılmıştır. USB kullanılmasının en önemli nedenlerinden birisi kontrol bilgisayarında kullanılan bilgisayarda 6 adet USB çıkışı bulunmasıdır. Köprü devresi üzerinde USB-Serial dönüştürücü kullanılmıştır. Kontrol bilgisayarında yüklü olan QNX işletim sisteminde USB-Serial cihazının sürücüsünün adı “devc-serusb” dir. Bilgisayar açılınca işletim sistemi yüklenirken bu sürücünün de yüklenmesi sağlanmıştır. Sürücü yüklendikten sonra /dev/ klasörünün altında kaç adet USB-Serial dönüştürücü donanımı varsa o kadar sayıda device(cihaz) dosyası oluşturulur. Bu dosyaların adı “serusbNO” dur .(Örn. serusb1, serusb2). Sürücünün yüklenmesinden sonra bu cihaz dosyaları normal seri haberleşme kullanılır gibi kullanılabilir.

Ayrıca dönem içerisinde önceden ücretli ancak daha sonra ücretsiz olan QNX 6.4 işletim sistemine geçiş yapılmıştır. Önceki ve sonraki versiyonlarda yapılan değişikliklerden dolayı yeni sürüme geçişte bazı problemlerle karşılaşılınca da problemler çözülmüştür.

Denetim bilgisayarı ile operatör bilgisayarı arasındaki bağlantı fiberoptik kablo ile gerçekleştirilmektedir. Fiberoptik kablo kullanılırken hem denetim bilgisayarının bulunduğu denizaltı aracında hem de operatör bilgisayarının bulunduğu yere Fiberoptik – Ethernet çevirici kullanılmıştır. Kullanılan çevirici Moxa firmasının IMC 101 model çeviricisidir. Bu çevirici 12-48 V arasında çalışabilmektedir. Dolayısıyla operatör bilgisayarın tarafında çeviriciyi kullanabilmek için 12 V luk güç kaynağına ihtiyaç duyulmaktadır.

Denetim bilgisayarı işletim sistemi ve çalıştırılabilir yazılımı CF karta yüklenerek çalıştırılmaktadır. Aşağıdaki bölümlerde yüklemelerin ve derlemelerin nasıl yapıldığı anlatılmaktadır.

### **3.2.3 İşletim Sistemi Görüntüsünün Hazırlanması ve Yüklenmesi**

Bu bölümde denetim bilgisayarında kullanılan işletim sisteminin CF disk için görüntüsünün oluşturulması ve yüklenmesi anlatılmıştır. QNX de işletim sistemi oluşturmak için genellikle bir script dosyası hazırlanarak bu sayesinde işletim sisteminin görüntüsü oluşturulur. Bu dosyada oluşturulacak olan görüntü dosyasında hangi modüllerin ekleneceği, hangi kütüphanelerin kullanılacağı gibi bilgiler yer almaktadır. Bu dosya çalıştırdıktan sonra uzantısı .ifs olan görüntü dosyası oluşturulur. Oluşturulan bu görüntü dosyası daha sonra CF disk e yüklenebilir. Oluşturulan görüntü dosyasının yaklaşık olarak boyutu 850KB dir. Görüldüğü gibi çok küçük bir dosya oluşturulmaktadır. Ve bu şekilde işletim sistemi çok kısa bir zaman içerisinde(yaklaşık 8 sn) açılmaktadır. Ethernet

modülünün ve diğer çalışması gereken programlar için CF diskin altında bulunan /etc/rc.d/rc.local dosyasına aşağıdaki satırlar eklenmelidir.

```
# Start the io-net driver with just the tcpip protocol
echo Starting network drivers...
io-net -ptcpip &
/bin/waitfor /dev/socket 5

# Configure the network card driver

mount -Tio-net /lib/dll/devn-pcnet.so

/bin/waitfor /dev/io-net/ip0 5

# Start the USB server

echo Starting the USB server with EHCI, OHCI and UHCI support...

io-usb -dehci -dohci -duhci &

/bin/waitfor /dev/io-usb/io-usb 5

# Force contents of /etc/net.cfg to take effect

echo Loading network configuration...

netmanager

# Run the sshd server

echo Starting sshd...

sshd
```

### 3.2.4 Çalıştırılabilir Kodun Derlenmesi ve Yüklenmesi

ULİSAR projesi kapsamında geliştirilen denetim bilgisayar yazılımı aşağıda adresi verilen depoda tutulmaktadır. Bu depodan dosyaların indirilebilmesi için depo yöneticisi tarafından verilen hesap adı ve şifre kullanılmalıdır. Deponun adresi:

```
svn://pc saranli.cs.bilkent.edu.tr/projects
```

Dosyalar depodan indirildikten sonra aşağıdaki ayarlar yapılmalıdır. ULİSAR yazılımını derlerken bazı çevresel değişkenler kullanılmıştır. Bunları her seferinde tanımlamak için /PATH/RoboDevel altında setvars.sh adlı bir dosya ile

herhangi bir konsol açtığımızda bunların otomatik olarak tanımlanması sağlanmıştır. Bunu yapabilmek için /home/user/ dizini altındaki .bashrc dosyasına ekteki komutlar eklenmelidir.

```
## optional : will default to $USER if you do not define here
export ROBODEVEL_USER=username
#
## required : where local copy will reside
export ROBODEVEL_DIR=/PATH/RoboDevel
export ULISAR_TOPDIR=/PATH/Ulisar/Underwater/ControlComputer
export URB_DIR=/PATH/urb/trunk/cpu_util
#
## required : target platform for compilation
## valid choices are linuxx86,
qnx6x86
export ROBODEVEL_TARGET=linuxx86
#
## you can individually choose robot hardware types
## if you do not specify, that robot is not compiled
export ULISAR_HARDWARE=_ULISAR_
## last line required
to source this file
. $ROBODEVEL_DIR/setvars.sh
```

ROBODEVEL\_DIR çevre değişkeni ile tanımlı klasöre gidilip “make” komutu yazıldığında çalıştırılabilir dosya oluşturulmalıdır. Çalıştırılabilir dosya /PATH/Ulisar/Underwater/ControlComputer/Demo/Supervisor dizini altında “main” adındaki dosyadır. Bu dosya scp komutu ile denetim bilgisayarına yüklenmelidir. Buradaki PATH depodan dosyaları indirdiğimiz klasör adıdır.

Denetim bilgisayarında 1. USB ye köprü kartı üzerindeki 1. köprü bağlanmalıdır. 2. USB girişine ise köprü kartındaki 2. köprü bağlanmalıdır. Denetim bilgisayarındaki Ethernet çıkışlarından bir tanesi görüntü bilgisayarına bir tanesi ise fiberoptik-ethernet dönüştürücüye bağlanmalıdır. Denetim bilgisayarının 1. seri port çıkışına akustik modem bağlanmalıdır. Akustik modem şu anda kullanılmamakla birlikte ileride kullanılacaktır. Deneysel çalışmalarda iletişimin sorunsuz sağlanabilmesi için akustik modemin yanı sıra fiberoptik kablo bağlantısı kullanılmaktadır.

### 3.2.5 Görüntü Bilgisayarı

Görüntü bilgisayarında işletim sistemi olarak DSL-N(DamnSmallLinux-Not) bulunmaktadır. Bu işletim sisteminin nasıl yükleneceği 5. raporda detaylı olarak verilmiştir. Dolayısıyla burada tekrar yazılmamıştır. Denetim bilgisayarı çalıştırılabilir kod üretme kısmında bahsedilen işleri yapınca görüntü bilgisayarının da çalıştırılabilir kodu üretilmiş oluyor. Bu kod görüntü bilgisayarına yine “scp” komutunu yükleyerek gerçekleştirilmektedir.

Görüntü bilgisayar yığımındaki MPEG4 encoder kartının 1. girişine sol kamera 2. girişine ise sağ kamera bağlanmalıdır. Kodlanmış kamera görüntüleri denetim bilgisayarı üzerinden operatör arayüzüne iletilmektedir.

### 3.2.6 Operatör Bilgisayarı

Operatör bilgisayarında işletim sistemi olarak Ubuntu 9.04 bulunmaktadır. Bu işletim sistemi kurulumu Ubuntu 9.04 imaj dosyası internetten indirilerek ardından bir CD ye yazılarak yapılabilir. Operatör bilgisayarının çalışan kodu da denetim bilgisayarının çalışan kodunu üretirken üretilir. Operatör arayüzünü çalıştırmak için `=/PATH/Ulisar/Underwater/ControlComputer/Operator` dizini altına giderek burada `./start.sh 3000 192.168.2.22` kodu girilerek çalıştırılabilir. Bu komutun ardından ekrana gelen arayüzde “Demo” modu seçildiği zaman denizaltı aracını yönettiğimiz arayüz karşımıza çıkar. Bu arayüzde artık denizaltında tanımladığımız bütün fonksiyonları kullanabiliriz. Denizaltı aracını hareket ettirmek için Joystick kullanılmaktadır. Kumanda kolu resmi ve düğmelerin işlevleri aşağıda verilmiştir.

Denizaltı aracı bir kumanda sayesinde kontrol edilmektedir. Bu kumanda Şekil 3.0.14’de verilmiştir.



Şekil 3.0.14 Denizaltı aracını kontrol etmekte kullanılan kumanda

Şekil 3.0.14’ de verilen kumandadaki tuşların işlevler Tablo 1’ de verilmiştir.

Numara	İşlev
1	Kamera 1 açma-kapama
2	Kamera 2 açma-kapama
3	Işıkları açma-kapama
4	Sonar açma-kapama
5	Denizaltı aracının yönlendirilemesi. İleri-geri,sağ,sol.
6	Denizaltı aracının yunuslama(pitch) kontrolü



7	Kullanılmıyor
8	Denizaltı aracının dik olarak yukarı çıkması
9	Denizaltı aracının dik olarak aşağı inmesi
10,11	10 ve 11 numaralı düğmelere aynı anda basıldığında Acil Durum konumuna geçirmek

Tablo 1 Kumandadaki düğme ve kolların işlevi

### 3.2.7 Karşılaşılan Problemler

Bu dönem çalışmaları içinde girişte de belirtildiği gibi ağırlıklı olarak entegrasyon çalışmaları yapılmıştır. Bu çalışmalar sırasında ilk problem denetim bilgisayarında kullanılan QNX işletim sisteminin bir üst versiyona geçişi sırasında yaşandı. Yeni sistemdeki network yönetim sistemi tamamıyla değiştiğinden, eski konfigürasyon dosyalarında ve işletim sistemi imaj dosyasında bazı değişiklikler yapıldı. İkinci sorun ise denetim bilgisayarın ve köprü arasında kullanılan USB-Serial sürücüsünden kaynaklandı. Kullandığımız çeviriciye uygun sürücüyü bulmak ve çalıştırılabilir duruma getirmek birkaç günümüzü aldı. Bu sorunlar dışında çalışmamızın yavaşlamasını sağlayacak herhangi büyük bir problemle karşılaşmadı.

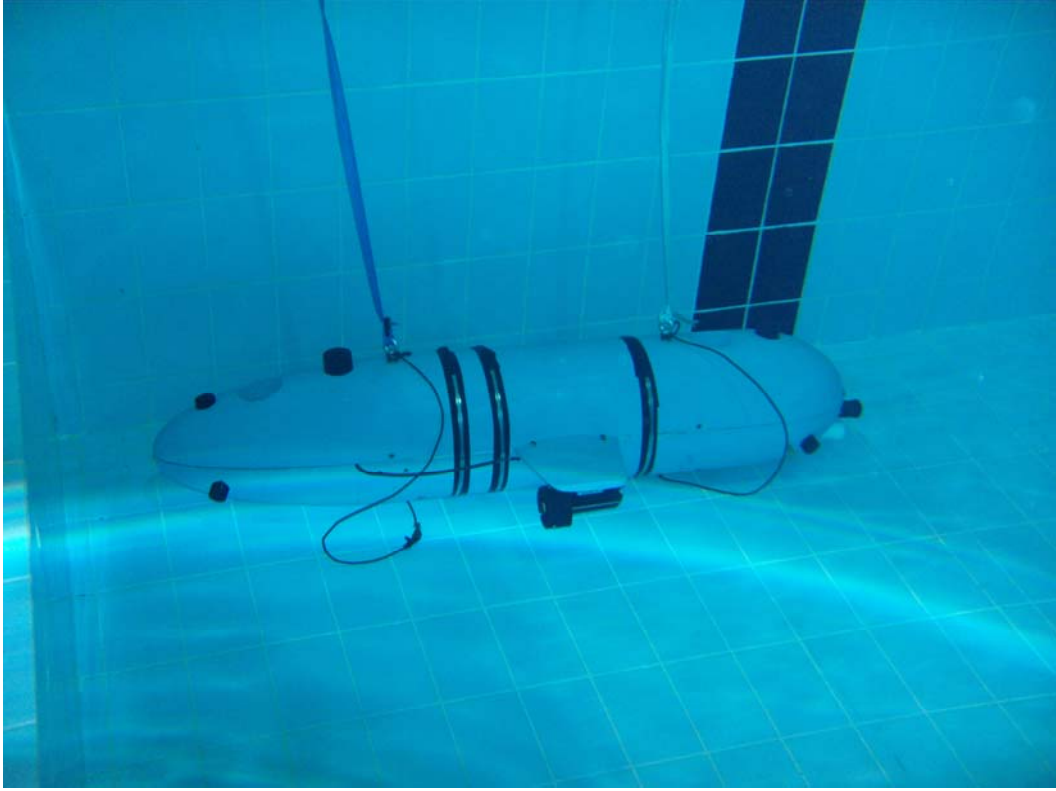
Sistemle ilgili olan en büyük sıkıntı motorlarımızın bir tanesi haricindeki diğerlerinin sürücü kartların havuz denemeleri öncesi bozulmuş olmasıdır. Bu aşamada motor sürücü kartlarının garanti kapsamı altında değişimi yapılacak olup daha sonra bu sürücüler yerine daha başka projelerde de kullandığımız kendi tasarımı olan DC Brushed motor sürücüleri kullanılması düşünülmektedir.

### 3.3 Deneysel Çalışmalar

Proje ile ilgili deneysel çalışmalarımız, projenin ilk aşamalarından beri sürekli olarak devam etmektedir. Öncelikle her bir eleman satın alındığında, işlerlik ve kullanım deneyleri yapılmıştır (Örnek olarak sonarlar, akustik modem, itici motorlar üzerinde yapılan deneylerden söz edilebilir).

Denizaltı gövdesi üretildikten sonra, sürekli bir şekilde, çeşitli aşamalarda ve farklı şekillerde sızdırmazlık deneyleri yapılmıştır (Bakınız Şekil 3.0.15). Bu deneyler sırasında, sızdırmazlığı sağlayabilmek için gövde üzerinde bazı değişiklikler yapılması gerektiği ortaya çıkmıştır. Bir sonraki aşamaya geçmeden bu hatalar giderilmiştir.

Bu bölümde asıl üzerinde duracağımız ODTÜ havuzlarında yapılan havuza indirme-havuzdan çıkarma, yüzerlik dengesi (pitch ve roll eksenlerindeki pasif dengenin sağlanması, yüzerliğin (buoyancy) yaklaşık olarak her deneyden önce nötr bir hale getirilmesi), navigasyon ve görüntü aktarma ve kayıt çalışmalarıdır.



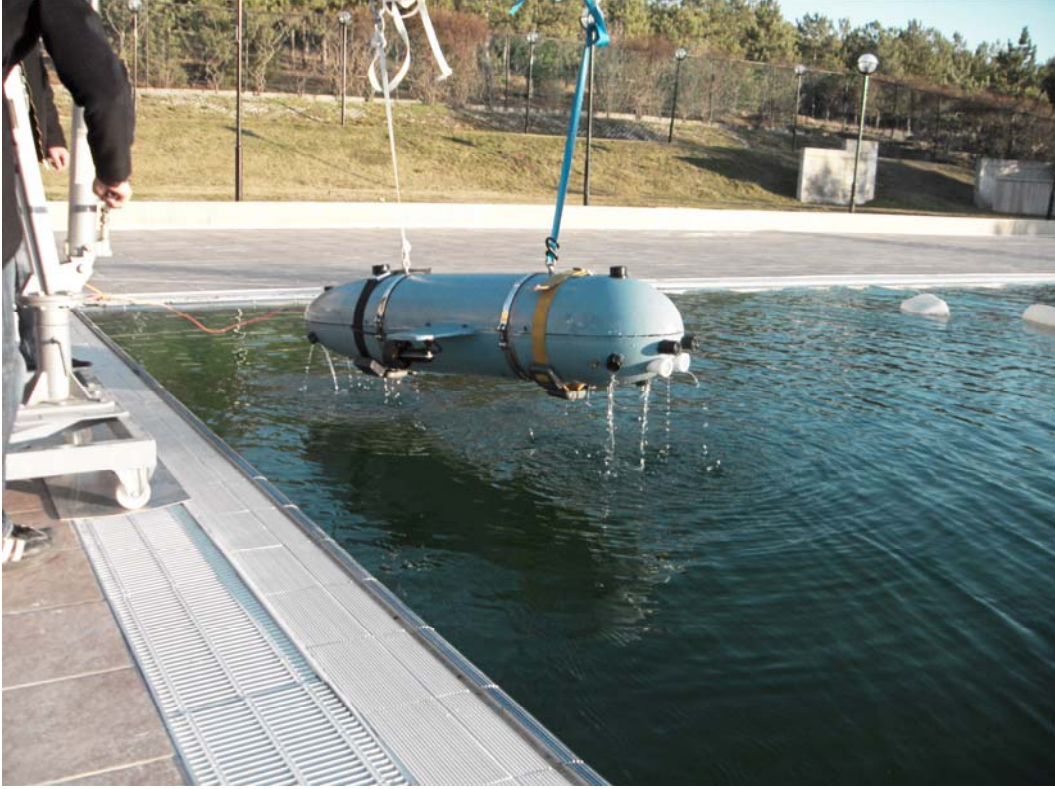
**Şekil 3.0.15** Sızdırmazlık testlerinden bir görüntü

Su altı aracının, vinç kullanılarak havuza indirme-kaldırma deneyleri hakkındaki resimler, “3.1.16 Suya Kolaylıkla İniş Çıkış İçin Vinç Mekanizması Tasarımı” ile ilgili bölümde verilmiştir.

Su altı aracı, normal olarak pozitif yüzerliğe sahip olacak şekilde tasarlanmıştır. Ancak aracın içine konacak olan sistemlerin, pillerin ağırlığına ve su özelliklerine göre her havuz denemesi öncesi bir yüzerlik ayarı yapılması gerekmektedir. Bunu sağlamak için aracın ön ve arka kısımlarına değişik ağırlıklarda ağırlık kemerleri bağlanmaktadır (Bakınız: Şekil 3.0.16, Şekil 3.0.17). Daha sonraki aşamalarda bu ağırlık kemerlerinin yerini araç içine konacak ağırlıklar alacaktır.

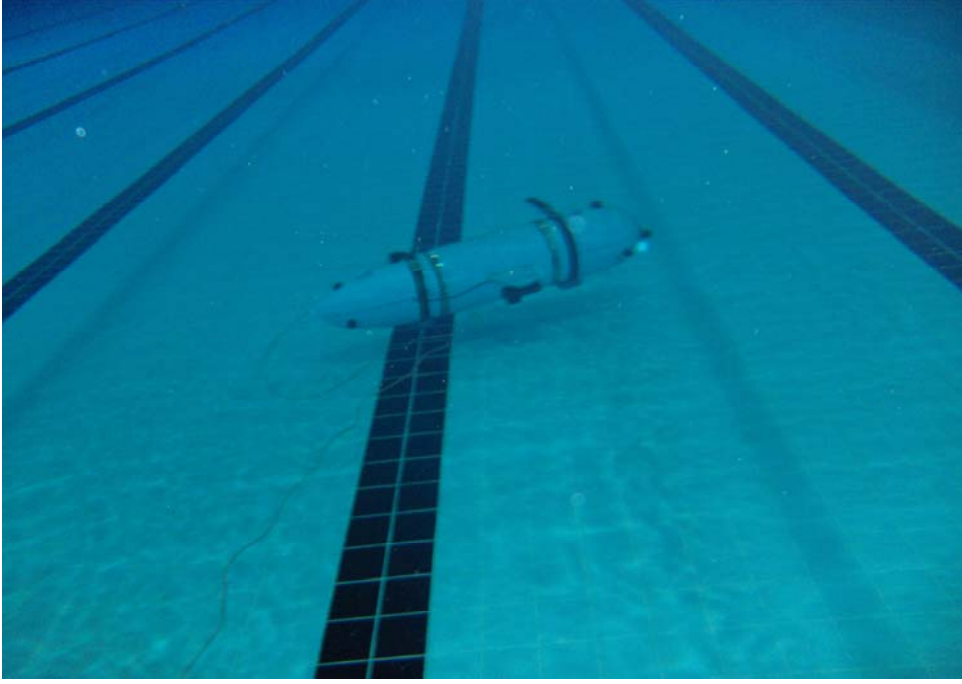


**Şekil 3.0.16** Ağırlık kemerlerinin ayarlanması

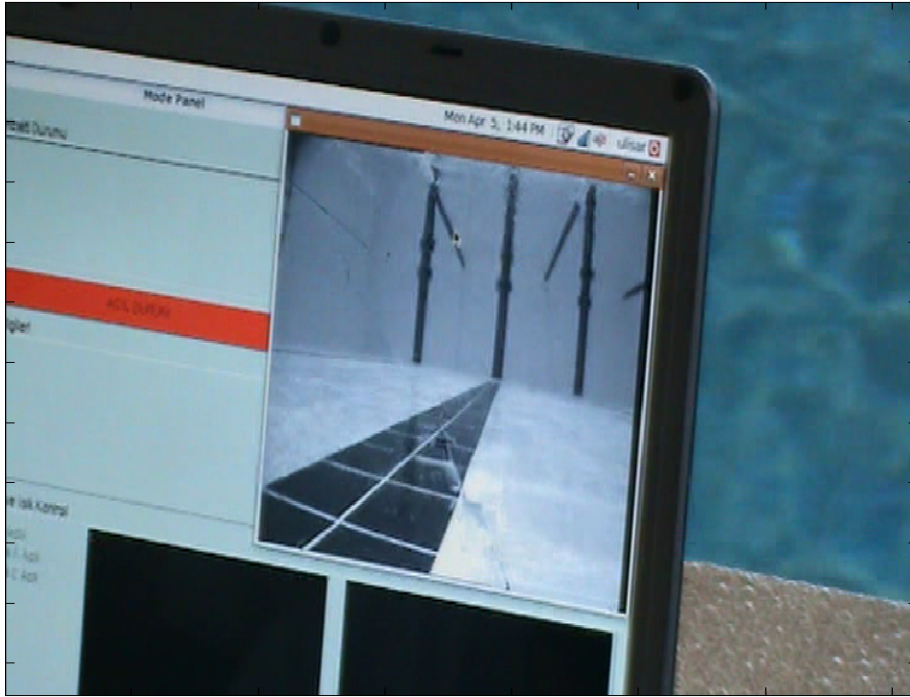


**Şekil 3.0.17** Denge testi çalışmaları

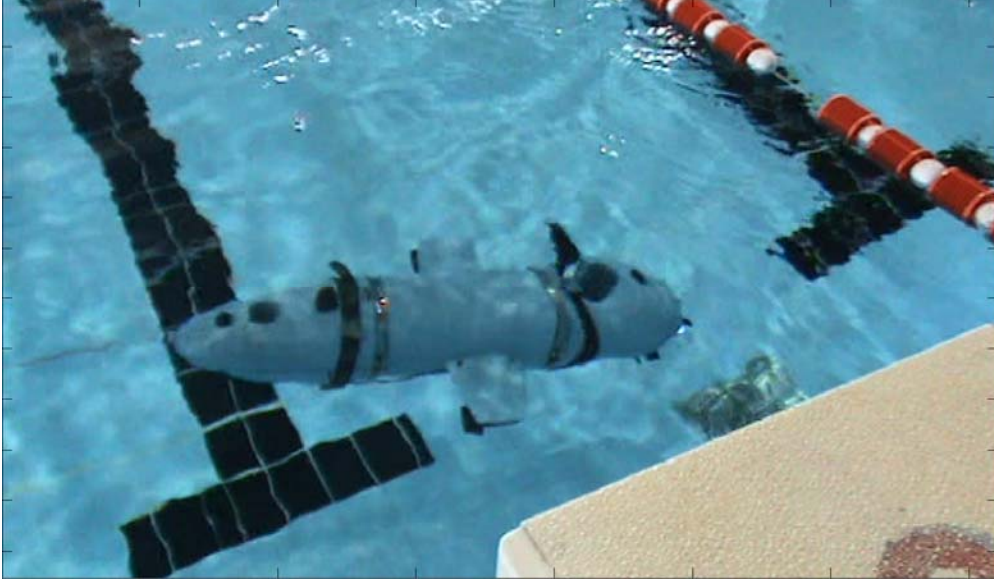
Denge testi çalışmalarından sonra görüntülü manevra testlerine geçilmiştir. Yaklaşık Mart 2010 ayının ortalarından itibaren bu son fazdaki testler devam etmektedir. Testler sonucunda edinilen bilgiler doğrultusunda kullanım kolaylığına yönelik iyileştirmeler yapılmıştır.



**Şekil 3.0.18** Manevra testlerinden bir görüntü



**Şekil 3.0.19** Çalışmalar sırasında operatör konsoluna gelen bir görüntü



**Şekil 3.0.20** Çalışmalar sırasında çekilen bir fotoğraf

## 4. Sonuç

Proje kapsamında insansız bir denizaltı aracı üretilmiş ve bunun tüm elektronik ve yazılım kısmı proje grubu tarafından tasarlanıp, üretilip geliştirilmiştir. Denizaltı aracının şu anda birçok işlevi çalışıyor olup bundan sonra denizaltı aracının navigasyon sistemine dip taramalı sonar vasıtası ile destek verilmesi ve otonom davranış göstermesi yönünde (güdümlü hareket için) çalışmalara devam edilecektir.

## TÜBİTAK PROJE ÖZET BİLGİ FORMU

<b>Proje No:</b> 105E127
<b>Proje Başlığı:</b> ÇOK MAKSATLI ULUSAL İNSANSIZ SUALTI ARACI PROJESİ (ULİSAR)
<b>Proje Yürütücüsü:</b> Prof. Dr. M. Kemal Leblebicioğlu
<b>Araştırmacılar:</b> Prof. Dr. Kemal İDER Prof. Dr. Kemal ÖZGÖREN Yüksek Müh. Haldun MİDOĞLU Prof. Dr. Uğur HALICI Y. Doç. Dr. İlkay ULUSOY Prof. Dr. Levent PARNAS Doç. Dr. Halit OĞUZTÜZÜN Prof. Dr. Haluk AKSEL Prof. Dr. Volkan ATALAY Prof. Dr. Gözde BOZDAĞI Prof. Dr. Yasemin Yardımcı ÇETİN Prof. Dr. S. Sencer KOÇ Y. Doç. Dr. Afşar SARANLI Yüksek Müh. Emre Ege Yüksek Müh. Hüseyin Yiğitler Yüksek Müh. Alper Kılıç Yüksek Müh. Necmettin Cevheri Seda Karadeniz Mine Cüneyitoğlu Yüksek Müh. Örsan Aytekin  <b>Diğer Araştırmacılar:</b> Bu arkadaşlar resmi olarak proje çalışanları arasında gözükmeyip tezleri dolayısı ile projeye katkıda bulunmuşlardır. Yüksek Müh. Deniz Üsteğmen Kadir Isıyel Yüksek Müh. Yavuz Kırılı

Yüksek Müh. Murat Deniz Aykın
<b>Projenin Yürütüldüğü Kuruluş ve Adresi:</b> ODTÜ, Elektrik ve elektronik Mühendisliği Bölümü, 06531, Ankara.
<b>Destekleyen Kuruluş(ların) Adı ve Adresi:</b> <ol style="list-style-type: none"> <li>1. Barış Elektrik Endüstrisi A.Ş.</li> <li>2. Yenel Ltd.</li> </ol>
<b>Projenin Başlangıç ve Bitiş Tarihleri:</b> 1.06.2006 – 31.05.2009
<b>Öz (en çok 70 kelime)</b> <p>Bu proje sualtı gözleme amacı ile yapılan bir insansız sualtı aracı hakkındadır. Temel fikir olarak, sualtı aracı ile beraber hareket eden ve sualtı aracına fiber kablo ile bağlı bir suüstü aracı olması düşünülmüştür. Sualtı ve suüstü araçları aralarında fiber üzerinden haberleşirken, suüstü aracı ile ana gemi arasındaki haberleşme rf üzerinden planlanmıştır. Böylece gerçek zamanlı navigasyon mümkün olacaktır. Mayın tarama görevleri için de kullanılabilmesi için gövde kompozit malzemedem imal edilmiştir.</p>
<b>Anahtar Kelimeler:</b> İnsansız sualtı aracı (uuv), navigasyon, kompozit malzeme, fiber optik kablo, sürücüler, alıcılar, sonar, akustik modem, kamera, projektör.
<b><u>Fikri Ürün Bildirim Formu</u> Sunuldu mu?</b> Gereklı Deęil
Fikri Ürün Bildirim Formu'nun tesliminden sonra 3 ay içerisinde patent başvurusu yapılmalıdır.
<b>Projeden Yapılan Yayınlar:</b> <ol style="list-style-type: none"> <li>1. H. Yiğitler, K. Leblebiciođlu, "Online Calibration of Strapdown Magnetometers", 9th IFAC Symposium on Robot Control, Nagaragawa Convention Center, Gifu, Japan, September 9-12, 2009.</li> <li>2. N. Cevheri, F. Şenel, K. İder, H. Aksel, L. Parnas, "Bir İnsansız Sualtı Aracı: ULİSAR Gövde Tasarımı", SAVTEK 2008, Savunma Teknolojileri Kongresi, Cilt I, s. 549-556, 26-27 Haziran 2008, ODTÜ, 2008.</li> <li>3. N. Cevheri, H. Aksel, "Hydrodynamic and Thermal Analysis of an Autonomous Underwater Vehicle (in Turkish)", 2008 FLUENT Users Conference, Bilkent, Ankara, October 7-9, 2008.</li> <li>4. K. Leblebiciođlu, "ULİSAR: İnsansız Bir Sualtı Aracı", 2008 TOBB Ekonomi ve Teknoloji Üniversitesi, Çok Erkinli ve Haberleşmeli Dinamik Kontrol Sistemleri Çalıştayı, 1 Temmuz, 2008.</li> </ol>
<b>Projeden Doğrudan Yapılan Tezler:</b> <ol style="list-style-type: none"> <li>1. K. Isıyel, "Autopilot Design and Guidance Control of ULİSAR UUV (Unmanned Underwater Vehicle)", Yüksek Lisans Tezi, ODTÜ, 2007.</li> <li>2. N. Cevheri, "Computer Aided Engineering of an Unmanned Underwater Vehicle", Yüksek</li> </ol>

Lisans Tezi, ODTÜ, 2010.



# ABSTRACT

## EFFICIENT CALIBRATION OF A MULTI-CAMERA MEASUREMENT SYSTEM USING A TARGET WITH KNOWN DYNAMICS

Aykın, Murat Deniz

M.S., Department of Electrical and Electronics Engineering

Supervisor : Assist. Prof. Dr. Afşar Saranlı

August 2008, 86 pages

Multi camera measurement systems are widely used to extract information about the 3D configuration or “state” of one or more real world objects. Camera calibration is the process of pre-determining all the remaining optical and geometric parameters of the measurement system which are either static or slowly varying. For a single camera, this consist of the internal parameters of the camera device optics and construction while for a multiple camera system, it also includes the geometric positioning of the individual cameras, namely “external” parameters. The calibration is a necessary step before any actual state measurements can be made from the system. In this thesis, such a multi-camera state measurement system and in particular the problem of procedurally effective and high performance calibration of such a system is considered.

This thesis presents a novel calibration algorithm which uses the known dynamics of a ballistically thrown target object and employs the Extended Kalman Filter (EKF) to calibrate the multi-camera system. The state-space representation of the target state is augmented with the unknown calibration parameters which are assumed to be static or slowly varying with respect

to the state. This results in a “super-state” vector. The EKF algorithm is used to recursively estimate this super-state hence resulting in the estimates of the static camera parameters. It is demonstrated by both simulation studies as well as actual experiments that when the ballistic path of the target is processed by the improved versions of the EKF algorithm, the camera calibration parameter estimates asymptotically converge to their actual values. Since the image frames of the target trajectory can be acquired first and then processed off-line, subsequent improvements of the EKF algorithm include *repeated* and *bidirectional* versions where the same calibration images are repeatedly used. Repeated EKF (R-EKF) provides convergence with a limited number of image frames when the initial target state is accurately provided while its bidirectional version (RB-EKF) improves calibration accuracy by also estimating the initial target state.

The primary contribution of the approach is that it provides a fast calibration procedure where there is no need for any standard or custom made calibration target plates covering the majority of camera field-of-view. Also, human assistance is minimized since all frame data is processed automatically and assistance is limited to making the target throws. The speed of convergence and accuracy of the results promise a field-applicable calibration procedure.

Keywords: Calibration and Identification, Visual Tracking, Dynamics.

## ÖZ

### ÇOK KAMERALI BİR ÖLÇÜM SİSTEMİNİN BİLİNER DİNAMİĞİ OLAN BİR HEDEF İLE KALİBRE EDİLMESİ

Aykın, Murat Deniz

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi : Yrd. Doç. Dr. Afşar Saranlı

Ağustos 2008, 86 sayfa

Çok kameralı sistemler bir veya birden çok objenin 3 boyutlu konfigürasyonu ya da “durum” kestirimi ile ilgili bilgi elde etmek amacıyla sıklıkla kullanılmaktadır. Kamera kalibrasyonu statik ya da yavaş değişen optik ve geometrik parametrelerin önceden belirlenmesidir. Kalibrasyon tek bir kamera için, kameranın yapımı ve optiği ile ilgili iç parametrelerin belirlenmesine karşılık gelirken, çok kameralı bir sistemde kameraların birbirlerine göre pozisyonlarının kestirimini, yani “dış” parametre kestirimini de içermektedir. Ölçüm sistemi ile durum kestirimi yapabilmek için kalibrasyon gerekli ve önemli bir adımdır. Bu tezde çok kameralı bir ölçüm sistemi ile bu sistemin yöntemsel olarak etkin ve yüksek performanslı kalibrasyonu problemi incelenmiştir.

Bu tez kapsamında balistik olarak fırlatılan bir objenin bilinen dinamiğinin genişletilmiş Kalman Filtresi (EKF) yardımıyla çok kameralı bir sistemin kalibrasyonu için kullanılması önerilmiştir. Hedef objenin dinamiğinin durum-uzayı gösteriminin bilinmeyen statik ya da yavaş değişen kamera parametreleri ile genişletilmesiyle “süper durum vektörü” elde edilir. EKF durum kestirme algoritması süper durum vektörünün tahmin edilmesi için kullanılmakta

ve böylece bilinmeyen kamera parametreleri de hesaplanmaktadır. Yapılan simülasyon çalışmaları ve gerçek deneyler hedefin ballistik yörüngesinin önerilen iki yeni EKF algoritması ile işlenmesi sonucunda kamera kalibrasyon parametrelerinin gerçek değerlerine sonuçta kararlı bir şekilde yakınsadığını göstermiştir. Kalibrasyon için gerekli olan görüntüler elde edildikten sonra çevirim dışı bir şekilde işlenmeleri mümkün olduğundan, aynı görüntülerin defalarca kullanıldığı *özyinelemeli* ve *iki yönlü* EKF versiyonları önerilmiştir. Özyinelemeli EKF (R-EKF) hedef objenin ilk durum vektörü doğru olarak verildiğinde parametrelerin sınırlı bir sayıda görüntü karesi kullanılarak yakınsamasını sağlarken iki yönlü özyinelemeli EKF (RB-EKF) ise parametre kestiriminin yanı sıra ilk durum vektörünün de kestirimini gerçekleştirdiğinden daha doğru sonuç elde edilmesini sağlamaktadır.

Önerilen algoritmanın en büyük katkısı önceden hazırlanmış desenli bir plakanın kameraların çoğunluğunun görüş alanını kapsayacak şekilde hizalanması gerekmeden hızlı ve kolay uygulanabilir bir kalibrasyon prosedürü sağlamasıdır. Ayrıca kalibrasyon karelerinin otomatik işlenmesi, insan desteğinin sadece hedef fırlatma ile sınırlanmasını sağlamıştır. Doğru ve hızlı bir şekilde sonuç üretilmesi algoritmanın sahada uygulanabilir bir yöntem olduğunu göstermiştir.

Anahtar Kelimeler: Kalibrasyon ve Tanımlama, Görsel İzleme, Dinamik Hareket.

EFFICIENT CALIBRATION OF A MULTI-CAMERA MEASUREMENT SYSTEM  
USING A TARGET WITH KNOWN DYNAMICS

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MURAT DENİZ AYKIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF MASTER OF SCIENCE  
IN  
ELECTRICAL AND ELECTRONICS ENGINEERING

AUGUST 2008

Approval of the thesis:

**EFFICIENT CALIBRATION OF A MULTI-CAMERA MEASUREMENT SYSTEM  
USING A TARGET WITH KNOWN DYNAMICS**

submitted by **MURAT DENİZ AYKIN** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen  
Dean, Graduate School of **Natural and Applied Sciences** \_\_\_\_\_

Prof. Dr. İsmet Erkmen  
Head of Department, **Electrical and Electronics Engineering** \_\_\_\_\_

Assist. Prof. Dr. Afşar Saranlı  
Supervisor, **Electrical and Electronics Engineering Dept., METU** \_\_\_\_\_

**Examining Committee Members:**

Prof. Dr. Mübeccel Demirekler  
Electrical and Electronics Engineering Dept., METU \_\_\_\_\_

Assist. Prof. Dr. Afşar Saranlı  
Electrical and Electronics Engineering Dept., METU \_\_\_\_\_

Prof. Dr. Gözde Bozdağı Akar  
Electrical and Electronics Engineering Dept., METU \_\_\_\_\_

Assoc. Prof. Dr. Aydın Alatan  
Electrical and Electronics Engineering Dept., METU \_\_\_\_\_

Assist. Prof. Dr. Buğra Koku  
Mechanical Engineering Dept., METU \_\_\_\_\_

**Date:**

**27.08.2008**

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last Name: MURAT DENİZ AYKIN

Signature :

# ABSTRACT

## EFFICIENT CALIBRATION OF A MULTI-CAMERA MEASUREMENT SYSTEM USING A TARGET WITH KNOWN DYNAMICS

Aykın, Murat Deniz

M.S., Department of Electrical and Electronics Engineering

Supervisor : Assist. Prof. Dr. Afşar Saranlı

August 2008, 86 pages

Multi camera measurement systems are widely used to extract information about the 3D configuration or “state” of one or more real world objects. Camera calibration is the process of pre-determining all the remaining optical and geometric parameters of the measurement system which are either static or slowly varying. For a single camera, this consist of the internal parameters of the camera device optics and construction while for a multiple camera system, it also includes the geometric positioning of the individual cameras, namely “external” parameters. The calibration is a necessary step before any actual state measurements can be made from the system. In this thesis, such a multi-camera state measurement system and in particular the problem of procedurally effective and high performance calibration of such a system is considered.

This thesis presents a novel calibration algorithm which uses the known dynamics of a ballistically thrown target object and employs the Extended Kalman Filter (EKF) to calibrate the multi-camera system. The state-space representation of the target state is augmented with the unknown calibration parameters which are assumed to be static or slowly varying with respect



to the state. This results in a “super-state” vector. The EKF algorithm is used to recursively estimate this super-state hence resulting in the estimates of the static camera parameters. It is demonstrated by both simulation studies as well as actual experiments that when the ballistic path of the target is processed by the improved versions of the EKF algorithm, the camera calibration parameter estimates asymptotically converge to their actual values. Since the image frames of the target trajectory can be acquired first and then processed off-line, subsequent improvements of the EKF algorithm include *repeated* and *bidirectional* versions where the same calibration images are repeatedly used. Repeated EKF (R-EKF) provides convergence with a limited number of image frames when the initial target state is accurately provided while its bidirectional version (RB-EKF) improves calibration accuracy by also estimating the initial target state.

The primary contribution of the approach is that it provides a fast calibration procedure where there is no need for any standard or custom made calibration target plates covering the majority of camera field-of-view. Also, human assistance is minimized since all frame data is processed automatically and assistance is limited to making the target throws. The speed of convergence and accuracy of the results promise a field-applicable calibration procedure.

Keywords: Calibration and Identification, Visual Tracking, Dynamics.

# ÖZ

## ÇOK KAMERALI BİR ÖLÇÜM SİSTEMİNİN BİLİLEN DİNAMİĞİ OLAN BİR HEDEF İLE KALİBRE EDİLMESİ

Aykın, Murat Deniz

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi : Yrd. Doç. Dr. Afşar Saranlı

Ağustos 2008, 86 sayfa

Çok kameralı sistemler bir veya birden çok objenin 3 boyutlu konfigürasyonu ya da “durum” kestirimi ile ilgili bilgi elde etmek amacıyla sıklıkla kullanılmaktadır. Kamera kalibrasyonu statik ya da yavaş değişen optik ve geometrik parametrelerin önceden belirlenmesidir. Kalibrasyon tek bir kamera için, kameranın yapımı ve optiği ile ilgili iç parametrelerin belirlenmesine karşılık gelirken, çok kameralı bir sistemde kameraların birbirlerine göre pozisyonlarının kestirimini, yani “dış” parametre kestirimini de içermektedir. Ölçüm sistemi ile durum kestirimi yapabilmek için kalibrasyon gerekli ve önemli bir adımdır. Bu tezde çok kameralı bir ölçüm sistemi ile bu sistemin yöntemsel olarak etkin ve yüksek performanslı kalibrasyonu problemi incelenmiştir.

Bu tez kapsamında balistik olarak fırlatılan bir objenin bilinen dinamiğinin genişletilmiş Kalman Filtresi (EKF) yardımıyla çok kameralı bir sistemin kalibrasyonu için kullanılması önerilmiştir. Hedef objenin dinamiğinin durum-uzayı gösteriminin bilinmeyen statik ya da yavaş değişen kamera parametreleri ile genişletilmesiyle “süper durum vektörü” elde edilir. EKF durum kestirme algoritması süper durum vektörünün tahmin edilmesi için kullanılmakta

ve böylece bilinmeyen kamera parametreleri de hesaplanmaktadır. Yapılan simülasyon çalışmaları ve gerçek deneyler hedefin ballistik yörüngesinin önerilen iki yeni EKF algoritması ile işlenmesi sonucunda kamera kalibrasyon parametrelerinin gerçek değerlerine sonuçta kararlı bir şekilde yakınsadığını göstermiştir. Kalibrasyon için gerekli olan görüntüler elde edildikten sonra çevirim dışı bir şekilde işlenmeleri mümkün olduğundan, aynı görüntülerin defalarca kullanıldığı *özyinelemeli ve iki yönlü* EKF versiyonları önerilmiştir. Özyinelemeli EKF (R-EKF) hedef objenin ilk durum vektörü doğru olarak verildiğinde parametrelerin sınırlı bir sayıda görüntü karesi kullanılarak yakınsamasını sağlarken iki yönlü özyinelemeli EKF (RB-EKF) ise parametre kestiriminin yanı sıra ilk durum vektörünün de kestirimini gerçekleştirdiğinden daha doğru sonuç elde edilmesini sağlamaktadır.

Önerilen algoritmanın en büyük katkısı önceden hazırlanmış desenli bir plakanın kameraların çoğunluğunun görüş alanını kapsayacak şekilde hizalanması gerekmeden hızlı ve kolay uygulanabilir bir kalibrasyon prosedürü sağlamasıdır. Ayrıca kalibrasyon karelerinin otomatik işlenmesi, insan desteğinin sadece hedef fırlatma ile sınırlanmasını sağlamıştır. Doğru ve hızlı bir şekilde sonuç üretilmesi algoritmanın sahada uygulanabilir bir yöntem olduğunu göstermiştir.

Anahtar Kelimeler: Kalibrasyon ve Tanımlama, Görsel İzleme, Dinamik Hareket.

*To my loving mother  
and to the memory of my beloved father*

## ACKNOWLEDGMENTS

I would like to express my deep and sincere gratitude to my supervisor, Professor Afşar Saranlı for his personal guidance, encouragement, constructive comments and for his outstanding support throughout this work. Special thanks to Professor Uluç Saranlı for the inspirational ideas and productive discussions during the course of this research.

I owe my most sincere gratitude to Professor Gözde Bozdağı Akar for her incredible understanding, hospitality and continuous support.

I wish to express my warm and sincere thanks to Mehmet Oğuz Bici, Anıl Aksay, Emrah Bala, Cem Vedat Işık, Berkan Solmaz and all colleagues from Multi Media Research Group, for their suggestions and technical supports and also for the very pleasant working environment.

I also wish to thank Cevahir Çıgla, Elif Vural, Engin Türetken, Eren Halıcı and Osman Serdar Gedik for their valuable friendship, entertainment and availability to help me whenever it was necessary.

During this work I have collaborated with many colleagues for whom I have great regard, and I wish to extend my warmest thanks to all those who have helped me with my work in the RHex Research Group.

Lastly I owe my loving thanks to my parents, Emel İnal and Kerim Serdar Aykın. They bore me, raised me, supported me, taught me, and loved me. To them I dedicate this thesis.

# TABLE OF CONTENTS

ABSTRACT . . . . .	iv
ÖZ . . . . .	vi
DEDICATION . . . . .	viii
ACKNOWLEDGMENTS . . . . .	ix
TABLE OF CONTENTS . . . . .	x
LIST OF TABLES . . . . .	xiii
LIST OF FIGURES . . . . .	xiv
LIST OF ABBREVIATIONS . . . . .	xviii
CHAPTERS	
1 Introduction . . . . .	1
1.1 General . . . . .	1
1.2 Scope and Contribution of the Thesis . . . . .	3
1.3 Outline of the Dissertation . . . . .	4
2 Background on Camera Calibration . . . . .	5
2.1 Internal Camera Calibration . . . . .	6
2.2 External Camera Calibration . . . . .	9
2.3 Object Detection . . . . .	11
2.4 Links between Calibration and State Estimation . . . . .	12
3 The Extended Kalman Filter . . . . .	14
3.1 General . . . . .	14
3.2 Overview of the EKF Algorithm . . . . .	15
3.3 Simultaneous State and Parameter Estimation . . . . .	17
3.3.1 The SSPE Algorithm . . . . .	17

	3.3.2	An Example on SSPE . . . . .	18
	3.3.3	An Example on Simultaneous Localization and Mapping . . . . .	20
4		Simultaneous Tracking and Camera Calibration . . . . .	22
	4.1	Ballistic Motion and Camera Measurement Models . . . . .	22
	4.1.1	Ballistic Motion Model . . . . .	23
	4.1.2	The Pinhole Camera Model . . . . .	24
	4.2	Raw measurements: Detection of the Target . . . . .	25
	4.3	Definition of Super-State . . . . .	27
	4.3.1	Augmented System Representation . . . . .	28
	4.3.2	Augmented System Equations . . . . .	30
5		Estimating the Super-State . . . . .	32
	5.1	Standard EKF Algorithm . . . . .	33
	5.2	Repeated EKF Algorithm . . . . .	34
	5.3	Repeated Bidirectional EKF Algorithm . . . . .	36
6		Simulation Experiments . . . . .	38
	6.1	Experiment 1: R-EKF / Free Fall Motion . . . . .	38
	6.2	Experiment 2: RB-EKF / Free Fall Motion . . . . .	42
	6.3	Experiment 3: Comparison of R-EKF and RB-EKF . . . . .	43
	6.4	Experiment 4: RB-EKF / Ballistic Motion . . . . .	49
	6.5	Experiment 5: RB-EKF / Multiple Trajectories . . . . .	52
	6.6	Discussion on Simulation Experiments . . . . .	55
7		Physical Experiments . . . . .	57
	7.1	The Multi Camera Setup Components . . . . .	57
	7.1.1	CTR-1472 MPEG-4 Compressor . . . . .	58
	7.1.2	Measurement Cameras . . . . .	58
	7.1.3	Connection Cables, Tripods and Clips . . . . .	59
	7.2	Internal Calibration . . . . .	61
	7.3	Experiment 1: R-EKF / Free Fall Motion . . . . .	63
	7.4	Experiment 2: RB-EKF / Free Fall Motion . . . . .	66
	7.5	Experiment 3: RB-EKF / Ballistic Motion . . . . .	68

7.6	Experiment 4: Perpendicular Camera Placement . . . . .	71
7.7	Experiment 5: Arbitrary Camera Placement . . . . .	74
7.8	Experiment 6: Using Multiple Trajectories for Arbitrary Camera Placement . . . . .	76
7.9	Discussion on Physical Experiments . . . . .	77
8	Conclusions and Future Work . . . . .	81
	REFERENCES . . . . .	83



## LIST OF TABLES

### TABLES

Table 3.1	The extended Kalman filter algorithm in discrete time . . . . .	16
Table 6.1	The final re-projection, angle and center errors for R-EKF and RB-EKF . . .	46
Table 6.2	The final re-projection, angle and center errors for single and double throw cases . . . . .	53
Table 7.1	The final parameter estimates for different initial positions of the calibration object and the corresponding re-projection pixel errors after 1000 throw iterations with R-EKF. . . . .	64
Table 7.2	The final parameter estimates for different initial positions of the calibration object and the corresponding re-projection pixel errors after 1000 throw iterations with RB-EKF. . . . .	68

## LIST OF FIGURES

### FIGURES

Figure 4.1	The Euclidean transformation between the world and camera coordinate frames. $C$ is the camera center, $X$ is the 3D world point, $x$ is the 2D projection onto the camera image plane . . . . .	25
Figure 4.2	Detection of the target calibration object. (a) Mean image of the calibration frames. (b) Standard deviation image of the calibration frames. (c) Threshold image of a calibration frame. (d) Binary image of a calibration frame. The center of the target is also marked. . . . .	27
Figure 4.3	Two-camera system assumed for illustration purposes to be mounted on a fixture with the unknown rotation and translation between them consisting the external calibration parameters. . . . .	28
Figure 4.4	First camera shown on the <i>basis coordinate frame</i> . The relation between the basis frame and <i>Cam1 coordinate frame</i> is represented with the rotation angles $\alpha^{c1}$ and $\beta^{c1}$ around $z$ and $x$ -axes respectively. . . . .	29
Figure 5.1	The flow chart for the repeated EKF algorithm. . . . .	35
Figure 5.2	The flow chart of the repeated bidirectional EKF algorithm. . . . .	37
Figure 6.1	The synthetic images of a free falling target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated EKF algorithm. . . . .	39
Figure 6.2	Convergence of the Repeated EKF Algorithm w.r.t <i>throw</i> iterations. All calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters. . . . .	40

Figure 6.3	The synthetic images of a free falling target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm. . . . .	42
Figure 6.4	Convergence of the Repeated Bidirectional EKF Algorithm w.r.t <i>throw</i> iterations. All calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters. . . . .	44
Figure 6.5	The synthetic images of a free falling target captured by a stereo camera system. The motion and observation noise variances are taken as 0.002 m and 0.1 pixels. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by (a) Repeated EKF; (b) Repeated Bidirectional EKF. . . . .	46
Figure 6.6	Convergence of the Repeated EKF Algorithm w.r.t <i>throw</i> iterations. Since the initial target state is not truly provided the calibration parameters are not accurately obtained (a) Rotation angles; (b) Translation parameters. . . . .	47
Figure 6.7	Convergence of the Repeated Bidirectional EKF Algorithm w.r.t <i>throw</i> iterations. Even though the initial target state is not truly provided the calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters. . . . .	48
Figure 6.8	The synthetic images of the ballistic trajectory of a thrown target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm. . . . .	49
Figure 6.9	Convergence of the Repeated EKF Algorithm w.r.t <i>throw</i> iterations. All calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters. . . . .	51
Figure 6.10	The cameras are placed perpendicular to each other and at the diagonal corners of a square with 1.5 meter edge length. . . . .	52
Figure 6.11	Convergence of the Repeated EKF Algorithm w.r.t <i>throw</i> iterations when two different target trajectories are used for calibration. All calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters. . . . .	54

Figure 7.1	Eurotech CTR-1472 frame grabber and MPEG-4 Compressor . . . . .	58
Figure 7.2	Outland Tech. UWC-300 underwater cameras (black cylinder, top) . . . . .	59
Figure 7.3	The connection cables and connectors. (a) PC104 stack which includes CTR-1472 MPEG-4 compressor and the manufactured cable set connecting the cameras to the stack. (b) CTR-1472 connector. . . . .	59
Figure 7.4	Tripod-clamp-camera trio and the calibration pattern. (Shown for a single camera) . . . . .	60
Figure 7.5	4 pieces of BWC-300 underwater cameras and synchronous video captur- ing with these cameras. . . . .	60
Figure 7.6	The calibration view obtained from 4 cameras synchronously. This kind of chessboard pattern shown in the figure is used for internal calibration purpose. . .	62
Figure 7.7	The extracted 2-D positions of a free falling target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated EKF algorithm. . . . .	64
Figure 7.8	Convergence of the Repeated EKF Algorithm w.r.t <i>throw</i> iterations, (a) Rotation angles; (b) Translation parameters. . . . .	65
Figure 7.9	The extracted 2-D positions of a free falling target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm. . . . .	66
Figure 7.10	Convergence of the Repeated Bidirectional EKF Algorithm w.r.t <i>throw</i> it- erations, (a) Rotation angles; (b) Translation parameters. . . . .	67
Figure 7.11	The extracted 2-D positions of a ballistically moving target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm. . . . .	69
Figure 7.12	Convergence of the Repeated Bidirectional EKF Algorithm w.r.t <i>throw</i> it- erations, (a) Rotation angles; (b) Translation parameters. . . . .	70
Figure 7.13	The cameras are placed at the diagonal corners of a square with 1 meter edge length. . . . .	71

Figure 7.14 The extracted 2-D positions of a ballistically moving target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm. . . . .	72
Figure 7.15 Convergence of the Repeated Bidirectional EKF Algorithm w.r.t <i>throw</i> iterations, (a) Rotation angles; (b) Translation parameters. . . . .	73
Figure 7.16 The extracted and re-projected 2-D positions of a ballistically moving target captured by two arbitrarily placed cameras. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm. . .	74
Figure 7.17 Convergence of the Repeated Bidirectional EKF Algorithm w.r.t <i>throw</i> iterations, (a) Rotation angles; (b) Translation parameters. . . . .	75
Figure 7.18 The extracted and re-projected 2-D positions of three ballistically moving targets captured by two arbitrarily placed cameras. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm for the (a) first; (b) second and (c) third trajectories. . . . .	78
Figure 7.19 Convergence of the Repeated Bidirectional EKF Algorithm w.r.t <i>throw</i> iterations, (a) Rotation angles; (b) Translation parameters. . . . .	79

## LIST OF ABBREVIATIONS

<b>Cam</b>	Camera
<b>EKF</b>	Extended Kalman Filter
<b>fps</b>	Frame per second
<b>FOV</b>	Field of View
<b>KF</b>	Kalman Filter
<b>MDP</b>	Markov Decision Process
<b>ML</b>	Maximum Likelihood
<b>MMSE</b>	Minimum Mean Square Error
<b>SLAM</b>	Simultaneous Localization and Mapping
<b>R-EKF</b>	Repeated EKF
<b>RB-EKF</b>	Repeated Bidirectional EKF
<b>SSPE</b>	Simultaneous State and Parameter Estimation
<b>STCC</b>	Simultaneous Tracking and Camera Calibration

# CHAPTER 1

## Introduction

### 1.1 General

The use of a multi-camera setup to extract information about both static and dynamic 3D configuration (state) of real world objects is an important problem of 3D vision [1, 2, 3]. In the particular application case of robot vision, the cameras may be part of a robotic platform (such as a mobile robot, unmanned aerial, land or underwater vehicle) and the state in question may belong to the platform in motion. Alternatively, the camera setup may be mounted externally as a measurement setup to externally measure the robot body state with high precision [4, 5]. The increase in precision is due to the possibility of off-line processing of data as well as abundance of computational hardware as compared to the case of being on a moving platform. Commercial multi-camera "motion capture" systems exist for this latter problem of precise robot body state measurement [6]. All of these systems necessarily require steps for calibration, namely the derivation of the internal parameters of each camera involved (internal calibration) as well as the parameters of the geometric arrangement of these cameras (external calibration) [7]. The collection of these calibration parameters characterize everything in the measurement setup and they can be used to extract the actual measurement variables (e.g. the 3D position of an external object).

Internal and external parameters show different characteristics with respect to rate of change or sensitivity to external disturbances. In general, camera itself is a rigidly constructed object and hence the internal parameters of the camera is more robust as compared to the external parameters. Given that the internal calibration parameters are obtained at low and medium accuracy levels temporal variations in these parameters for consumer-grade cameras are gen-

erally not significant [8]. Therefore, a procedurally efficient and fast method for the internal calibration is not considered critical since it is usually performed per camera and can remain valid for a long time. On the other hand, the external calibration, which characterizes the geometric arrangement of the individual cameras, is much less robust and has to be repeated every time the arrangement of the cameras is changed. For some applications where the cameras are mounted permanently, e.g. in a laboratory setup, these are much less prone to disturbances and a slow and manual calibration may be tolerated. For some other applications such as in robotics and for experimental measurements in the field, the situation is significantly different. A multi-camera setup in this case is often constructed on tripods and specifically to conduct one particular set of experiments. The setup is open to disturbances due to the field conditions such as loose ground, wind, accidental bumping to the tripods and similar. In these circumstances, the re-calibration of the setup may be necessary on a much more frequent basis and the re-calibration effort may constitute a large percentage of the experimental effort. This is clearly undesirable.

In our considered application problem, a multi-camera setup is deemed necessary to measure the body-state of an experimental hexapod platform: RHex [9]. Such a system is critical for good experimental practice with aim of generating ground truth data for the evaluation of the on-board state estimation, control and navigation algorithms. Since computationally more complex off-line state estimation algorithms can be used on an external system, those estimates can generally be considered as ground truth as compared to what can be achieved on-board.

RHex is a robotic platform claiming outdoor mobility on complex terrain. Advancing the state of research on this platform demands field experimentation in various environments. This is opposed to many mobile robotics platforms constrained to laboratory environment where a fixed measurement setup can be affordable. The application demands a fast and procedurally efficient calibration process (procedure and algorithm) since the cameras are to be carried to the site and suitably located for each experiment. There is high probability that a natural disturbance may move the setup and render the external calibration invalid. In each such occurrence, the external calibration has to be repeated, making ease of calibration procedure critical. If the calibration have to be performed frequently one needs an efficient calibration method that is both computationally fast but which also requires minimal human effort, e.g., by requiring minimal materials and special equipment such as specially built calibration pan-



els. The design of such a calibration method with competent calibration performance is the focus of the present thesis.

## 1.2 Scope and Contribution of the Thesis

In this thesis, a novel calibration algorithm for obtaining the external parameters of a camera setup is presented. The method uses the known dynamics of a target object for calibration. The aim is to exploit this extra knowledge about an easy to use target object to minimize the calibration effort and provide a fast and procedurally efficient technique. It is claimed that the proposed method is particularly beneficial for any application where frequent re-calibration is required such as outdoor field measurement applications using multiple-cameras.

In this thesis, we assume that a set of cameras, whose internal parameters are known apriori, are placed and oriented such that the required measurement area is within the overlapping section of the field-of-views (FOV) of all cameras. It is further assumed that their resulting external calibration parameters are to be determined. The motivation of the present work suggests that an easy to carry rigid target and simple to generate 3D motion with well known dynamics can be used to generate the calibration data set. A feasible example is a small rigid ball with well defined color and suitable mass which is ballistically thrown within the camera joint FOV. In 3D space, the motion of this ball is fully described by Newton's equations of motion. Assuming that wind and other secondary disturbances are negligible, the dynamics of this motion can therefore be modeled with good accuracy.

It is the claim and contribution of the present thesis that given enough calibration data (in the form of camera frames from all involved cameras) the repeated and bidirectional extensions of the EKF algorithm asymptotically converge to the estimates of the camera calibration parameters. We claim that standard EKF can provide usable estimates if there are large number of image frames while the proposed extensions can generate these estimates with much less number of image frames. The repeated use of the calibration data (R-EKF) provides convergence while its repeated and bi-directional use (RB-EKF) further improves the accuracy of the estimates.

The Kalman framework takes into account the process and observation noises such as wind effect and pixel errors. We also demonstrate that the RB-EKF version can tolerate significant

uncertainty in the initial state of the target (i.e. the initial position and velocity of the target). It should also be pointed out that despite our focus being on the use of the ball's motion to find the external calibration parameters of the camera setup, it is observed that an interesting complementary problem is also solved, namely that of estimating the state of a target with known dynamics with an uncalibrated camera setup.

At this point it should be stated that the proposed method is not limited to the present application scenario and can be extended to other areas: E.g., with the growing interest in 3D-TV, 3D multi-camera capture of sports events becomes important, requiring the external calibration of the camera setup. Due to the existence of a well defined ball, which follows a ballistic path, in most competition games (tennis, football, basketball, etc.), the method presented here is directly applicable.

### **1.3 Outline of the Dissertation**

The motivation and the context for this work together with the main contributions are presented in Chapter 1. Chapter 2 summarizes the state of the art relating to algorithms for internal and external calibration and also relating to basic video tracking methods. Chapter 3 focuses on the Kalman framework and the Extended Kalman Filter (EKF) and the usage in simultaneous state and parameter estimation (SSPE) problems. The definition of the considered calibration problem within an EKF framework and the augmentation of the target state with the unknown camera parameters to form the "super-state" is presented in Chapter 4. Chapter 5 presents the two approaches proposed for estimating the complete state with a limited number of measurement data, namely the R-EKF and RB-EKF algorithms. Chapter 6 and Chapter 7 presents the results of the experiments which are performed to evaluate the performance of the implemented calibration method, with both simulated and actual physical camera setups and image frame data. Finally, Chapter 8 gives the conclusions of this thesis together with possible avenues for future studies.

## CHAPTER 2

### Background on Camera Calibration

In all measurement systems, the calibration of the measurement instrument is a necessary step in order to find the transformation between the raw readings from the instrument to the actual physical measurement variables of interest. When the state of the world or the ego-state (state of the body on which the sensor(s) are mounted) is being measured through the use of cameras as sensors, the system needs calibration so that the raw camera images can be transformed into this measurement of the state. Essentially, calibration parameters can be seen as the collection of all parameters that characterize the transformation between the raw data from the instrument and the desired variables to be measured. Calibration is a compulsory initial step for any measurement system including a multi-camera 3D vision system.

For the particular case of camera calibration, we are concerned with the process of estimating the internal (intrinsic) and external (extrinsic) parameters of a camera system. The internal parameters characterize the optical properties of the lens system in the camera as well as the interaction of this lens system with the other components of the camera such as the sensor. In short, it is a collection of all parameters that determine how the image is being formed on the sensor of a single camera. External parameters on the other hand relate this camera unit (with its internal parameters) to a global reference frame, hence represent the 3D geometrical placement and orientation. If multiple cameras are involved, the external calibration parameter set represent this geometric specification for all cameras hence also describing their relative positions and orientations.

This chapter reviews the previous work on camera calibration and its historical evolution. Most of the previous work focus on internal calibration techniques for a single camera. However especially for the last two decades, the number and variety of the cameras have been

increasing rapidly while their prices have been decreasing, making multiple camera imaging systems more accessible to the research community. The necessity of external calibration methods for such multiple camera setups have encouraged recent studies on this subject.

A review of the state of the art techniques is introduced first for the internal calibration of a camera and then for the external calibration of a multiple camera network in sections 2.1 and 2.2 respectively. This is followed by an overview of object detection methods in section 2.3 which are necessary to generate *features* or *observations* for the calibration algorithms. Finally the relationship between calibration and dynamic state estimation (tracking) using Kalman Filtering is introduced in 2.4. Also in this last section, a survey of the earlier calibration methods utilizing the Extended Kalman Filter is presented.

## 2.1 Internal Camera Calibration

The image formation process can be seen as a mapping from the 3D world to the 2D image plane. The usual modeling of this mapping is by *central projective mapping*. In this mapping rays reflected from a 3D point in space is passed through a fixed point which is the *center of projection*. The rays intersect the *image plane*, which is placed at a specific distance from the center of projection, and their intersection forms the image point. Throughout this process involving a single camera, the depth information is lost due to the loss of one dimension.

Most cameras are modeled by the central projective mapping assumption [10]. The center of the lens of a camera is assumed as the center of projection. A ray of light from a 3D point in the world passes through the lens and produces an image point on a film or an opto-electronic capturing device (i.e., imaging sensor).

Homogenous coordinates allows affine transformations to be represented by a matrix and make calculations possible in projective space [11]. The mapping of a point in  $\mathbb{R}^3$  to  $\mathbb{R}^2$  on the image plane by a camera can be represented by a projection matrix  $\mathbf{P}$  in the homogenous coordinates. This  $\mathbf{P}$  matrix can be decomposed into two matrices  $\mathbf{K}$  representing the internal parameters of a camera and  $[\mathbf{R} \mid \mathbf{t}]$  representing the rotation and translation between the camera and the world coordinate frames. The transformation is given by Eqns. (2.1) and (2.2) and

further elaborated in section 4.1.2.

$$\mathbf{x} = \mathbf{P}\mathbf{X} \quad (2.1)$$

$$\mathbf{P} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}] \quad (2.2)$$

Internal calibration of a camera is therefore the task of obtaining the  $\mathbf{K}$  matrix which represents the inner characteristics of the camera. These calibration parameters are specifically

- *focal length*: the distance between the center of projection and the image plane,
- *pixel aspect ratio*: the width of a pixel divided by its height,
- *principal point offset*: the misplacement of the image origin,
- *skew*: the non-orthogonality of the camera axes.

Classic photogrammetric methods use non-linear optimization for solving all of the calibration parameters simultaneously [12, 13, 14]. The parameter values are directly computed by a non-iterative closed form algorithm. The algorithms are fast since no iterations are involved, but the lens distortion can not be incorporated into these algorithms and noisy measurements may cause quick deterioration of performance.

More recently, two-step methods which can successfully handle lens distortions have been proposed in the literature [15, 16, 17, 18, 19]. All of these techniques first estimate the calibration parameters using a closed form solution similar to earlier methods. Then, the parameters are improved iteratively through nonlinear optimization, taking the lens distortion into account.

These two-step methods make use of a special planar calibration pattern printed on a plate and placed in the camera FOV. By the use of a planar chess-board pattern, a large number of data points can be supplied and the algorithms yield reliable results by minimizing the average *re-projection error* on the collection of the points. Average re-projection error can be defined as the mean square error between the scene points visible in the actual camera and their projection in the reference camera whose characteristics are described by the employed camera model and the acquired calibration parameters [20].

As a further improvement to the aforementioned algorithms, a four-step calibration procedure is suggested by Heikkilä and Silvén [21]. The authors claim that these additional steps further

improve the compensation for distortion and provide the necessary correction. Also a linear method for solving the parameters of the inverse model is presented in this work.

The suggested algorithm given in [22] aims to generalize the camera model, and in particular, try to provide a solution for cameras which do not have a single effective viewpoint (termed as *non-central cameras* by the authors). To achieve this goal, a very general mapping between rays and pixels is used instead of a *central projection mapping*.

A recent survey summarizes the current approaches on camera calibration and present a comparison of different algorithms [8]. The efficiency of the bundle adjustment techniques are presented with experimental results. The changing profiles of the radial distortion for different color channels is also discussed.

The calibration methods which are proposed by Tsai [16] and Zhang [19] are widely accepted and popularly used for computer vision. Although these two-step methods are primarily internal calibration methods, it is suggested that they can be extended for calibrating the external parameters as well. However, these extensions for calibrating a camera network are subject to practical difficulties. This is due to the fact that successful external calibration requires that multiple different orientations of a 2D planar calibration pattern to be viewable from the joint-FOVs of all cameras in the network. Indeed, such a pattern can be seen from two (or more) different cameras only if both cameras are placed on the same side of the plane. Furthermore the pattern should be appropriately scaled in accordance with the magnification of the cameras and also in accordance with the placement and orientation of their FOVs.

The camera calibration toolbox which is part of the MATLAB package (from MathWorks Inc.) provides another algorithm used extensively for internal and partly for external calibration [23]. However the same visibility problem of the calibration plate still exists.

These popular camera calibration techniques yield reliable results by minimizing the overall re-projection error defined over the collection of data points that are provided by the planar chess-board pattern. However it is the very same planar fixed sized calibration object that causes significant visibility, scalability and procedural complexity problems when the methods are used for external calibration. It should be noted that these problems form the primary motivations for our approach where we rather provide a simple but moving calibration target avoiding these difficulties.

The basic idea behind our approach is the use of a simple to detect target which is moving with known dynamics. For this we propose a ballistically moving rigid object such as a ball with given color. The path followed by such a projectile can easily be viewed by multiple cameras looking from all possible angles, even when the cameras are facing each other. Furthermore, the calibration field can be utilized maximally by throwing the ball according to the distance between the cameras. So the present method not only solves the visibility problem but it also handles the scalability problem as well. The calibration target considered (i.e. a ball) can easily be obtained, and the calibration procedure is practical and fast. Finally, the knowledge of the global gravity vector provides additional information about the external calibration which is not present in other techniques (unless the calibration plate is not carefully aligned with the gravity vector).

## 2.2 External Camera Calibration

When a 3D scene is projected onto a plane, depth information is lost during the process. Hence it is not possible to extract the complete geometry of a scene from a single view. The 3D structure of a scene through image analysis can either be obtained by a multi camera system [24] or by taking multiple frames of the same scene with a single camera moving in space. For both cases a fundamental problem, that is called external calibration, is to obtain the camera positions and orientations for each image frame. Without this a priori knowledge of the acquisition system it is not possible to obtain the full 3D scene structure. It should be noted that we only consider standard image sensors and we do not consider specialized measuring devices that provide depth information directly, such as laser range scanners.

Multi camera external calibration is the task of obtaining the rotation and translation of each camera with respect to each other or relative to a global world coordinate frame. Let  $[\mathbf{R} \mid \mathbf{t}]$  matrix represent the rotation and translation for a camera. The  $\mathbf{R}$  matrix of size  $3 \times 3$  is called as the *rotation matrix* and represents the rotations around the coordinate axes. The  $\mathbf{t}$  matrix of size  $3 \times 1$  is called as the *translation vector* and represents the position of the camera along the three coordinate axes. Since the transpose of a rotation matrix is equal to its inverse [7],  $\mathbf{R}$  actually consists of only 3 unknowns. If the camera coordinate system is considered, the negative of the translation vector  $\mathbf{t}$  also represents the origin of the world reference frame in camera coordinates. Therefore a total of 6 parameters are required for the external calibration

of each camera, specifically 3 being for representing the orientation (rotation) of the camera and 3 being for the position (translation) of each camera.

The inverse problem of image formation is computing the camera location and orientation from the projection matrix. It involves solution of simultaneous nonlinear equations in several variables and is considered difficult. To accomplish this inversion a simple analytical technique that works well in practice is suggested in [25].

In [26] the issues concerning the calibration of multiple camera systems are discussed. It is argued that the ease of production of the calibration object hence that of the calibration process is in a trade off with the accuracy of calibration, because the most accurate calibration process would likely require a complicated calibration object and process. In the present thesis, we demonstrate that with the use of known target dynamics, this is not necessarily the case. We aim at a very simple calibration target and a simple process while also maintaining the accuracy of calibration.

The work in [27] also makes use of multiple images of a static chessboard calibration pattern. Cross projection errors between the cameras are minimized for extending single camera calibration method to multiple cameras.

Complete multi-camera calibration procedures, which uses a laser pointer for image generation, are presented by Han et al [28], and Svoboda et al [29]. The assumption of only reasonable overlap between camera subgroups makes the procedures of the algorithms to be easily applicable. The drawback of these methods appear to be the necessity of at least three cameras for calibration, hence making the calibration of stereo cameras impossible. The procedure also requires total darkness, rendering the outdoor applicability of these methods limited. The calibration methods presented in the present thesis on the other hand can seamlessly handle stereo cameras and can also work in daylight and outdoors provided that the target is detectable over the background scene.

Other approaches are presented in [30] and [31]. Instead of tracking a single calibration point in image sequences, a reference bar with light sources on it is moved giving additional depth information which is the constant length of the bar. Note that this additional information requires the construction of a rather complicated calibration bar with embedded laser pointers on it. In our approach constant gravitational acceleration gives equivalent additional informa-



tion without the associated complication in the calibration target or procedure.

## 2.3 Object Detection

Object detection is the process of detecting and determining the position of an object within a given image. Clearly, for the methods discussed in the present thesis, one needs to “detect” the calibration object, reliably determining its center position within the image frame, possibly at sub-pixel resolution. The problem can be very simple (such as in the case of a white ball with a black background farther away) or may get gradually harder (such as trying to detect the ball over a busy urban or natural background). The problem can be helped by the use of target color selection or target illumination (e.g. with IR LEDs mounted on the target) combined with appropriate filter on the cameras.

The algorithms on this subject are mostly focused on the application areas like face and car detection in the sequential video frames. Some of these basic methods are overviewed in this section.

Object detection algorithms that are specialized to certain objects of interest are commonly based on learning the characteristics of these objects [32, 33, 34, 35, 36, 37, 38, 39]. These algorithms need too many training examples and depend on huge training databases [40].

An extremely fast object detection technique is presented by Viola et al [37, 38]. This study not only uses learning to train the classifiers, but also improves the detection time of the features by representing the image in a new format. Furthermore, by connecting the classifiers in a cascade manner the background image is discarded and hence more time is spent on object-like regions. This study is improved with the introduction of a novel set of rotated Haar-like features by Lienhart et al [39].

Two probabilistic approaches are presented in the histogram based method for 3D object detection by Schneiderman et al [41] and visual learning method based on density estimation in a high dimensional space by Moghaddam et al [32].

Throughout our study we assume that the cameras are fixed, and only one very small calibration target is moving in the common camera FOV. Therefore the background image can approximately be acquired by obtaining the mean image of the video sequence, significantly

reducing the complexity of the target detection problem. For this reason we used Svoboda’s LED detection method [29] with some modification for the automatic detection of the calibration target, instead of using a complicated learning based algorithm. In this manner we not only can avoid the computational cost of a complex method, but avoid the need for a training database.

## 2.4 Links between Calibration and State Estimation

The approach in the present thesis is based on the idea that the estimation of static parameters can be integrated into a dynamic state estimation problem. The benefit of this approach is its ability to make use of the knowledge of target dynamics in order to better observe the calibration target positions, hence attaining high precision with a relatively coarse target object.

This idea is clearly not entirely new. For example, our problem has significant similarities with the Simultaneous Localization and Mapping (SLAM) approach, where noisy optimal state estimation (Kalman filtering) formulation is used to simultaneously estimate a moving robot’s position (dynamic state) and the map of the environment (static state) [42, 43].

This approach has the property that a generalized *system* and its associated *augmented state* is defined such that the unknown locations of the features of the environment (i.e. the map) are made part of the augmented state of the system and are subjected to the optimal estimation process. Here, the known dynamics of the robot platform, namely the *model of the system* is exploited for a better estimate than is possible simply by static observations of the robot position relative to known features of the environment. Moreover, the fact that the robot itself is moving makes static observations of landmarks with respect to a global reference frame impossible

In the present study, we build on this idea in a different context and with a different objective. Indeed, there are other instances of the idea in the existing literature, even in the present context. However, we believe that the specific motivation and solution approach considered in the present study to be of significantly more general than any other existing work in the literature, hence of considerable value.

For example, a number of studies attempt to give the camera orientations [44, 45] or rectify

the ground plane by tracking moving objects [46], while solving a planar relationship using the video sequences from  $n$  fixed cameras at unknown positions. The planarity constraint is ensured by placing the cameras high enough from the ground plane. We do not have such a constraint in the present work and provide a more general calibration algorithm as a result. An extension to these works is [47], which takes the topology of the camera overlap as an unknown and estimate a planar tracking correspondence model in addition to the calibration parameters of the large camera network.

Kalman Filter is used in [48] to recursively estimate the motion, pointwise structure, and focal length from feature correspondences, and in [49] for determining the unknown transformation between a camera and an IMU. No known dynamic model of a target is exploited in either of these studies hence the Kalman Filter may just be acting as a low pass filter.

Again in [50], a calibration algorithm based on Kalman Filter is presented. The method claim to reduce human intervention during calibration but can not fully eliminate it. One  $2D-3D$  correspondence of the moving object and at least 6 other points on a rigid object are needed to initialize EKF algorithm and guarantee its convergence. The necessity of a planar ground is also another constraint and drawback of this approach. We should note that for our case in the present thesis, neither any  $2D-3D$  correspondences nor any extra information is needed for the initialization of the algorithms.

## CHAPTER 3

### The Extended Kalman Filter

This chapter aims to give the basic knowledge about Kalman filtering framework and in particular the use of Extended Kalman filter (EKF) in simultaneous state and parameter estimation (SSPE) problems. The objective is to provide the adequate background to facilitate our discussion in chapter 4, which is built upon the assumption of a general knowledge on these issues.

A review of the EKF algorithm is given in section 3.2. The SSPE method is discussed in 3.3. Finally an example that illustrates the SLAM approach is given in 3.3.2.

#### 3.1 General

Kalman filter (KF) is the *optimal* state estimator for discrete-time linear dynamic systems driven by *white noise* [51]. KF is also proved to be the best linear state estimator in the linear *non-Gaussian* case.

Starting from an initial state which is assumed to be a random variable with a certain mean and covariance, Kalman filter estimates the next states by representing the dynamics of the system as a Markov Decision Process (MDP) which assumes that the state of the system depends only on the previous state and action [52]. Hence, KF is a recursive estimator, which predicts the state only from the previous time step and current measurement instead of the history of observations as in batch estimation methods. If the initial state, the process and the measurement noises are Gaussian and *mutually independent*, KF is the minimum mean square error (MMSE) estimator [53].

Although the linear system with *zero-mean white noise* assumption seems to restrict the applicable usage of the filter, through a number of extensions, this technique can be used in numerous problems. The extensions make it possible to map the problem at hand into one that meets the above constraints. For example most non-trivial systems are non-linear and they violate the linearity assumption of the basic KF. Such non-linearities in the process or observation models can be handled by the Extended Kalman filter (EKF) which is an approximation to the optimal estimate.

Occasionally some of the continuous-valued parameters of the system may be unknown. In such cases the state can be augmented by adding the unknown parameters and the EKF can be used for estimating both the state and the system parameters. This simultaneous state and parameter estimation (SSPE) method is popularly used in robot mapping problems and specifically named as the simultaneous localization and mapping (SLAM).

### 3.2 Overview of the EKF Algorithm

The Kalman filter is the best studied state estimation technique and it was invented by Swerling (1958) and Kalman (1960). KF assumes that the observations are linear functions of the state and the next state is a linear function of the previous state. Note that a Gaussian random variable which is passed through a linear function yields another Gaussian random variable. This knowledge plays an important role in the derivation of KF.

Unfortunately neither the systems nor the measurements can always be modeled by only linear transformations. In fact the state transitions and measurements are rarely linear since real life systems lack the simplicity that would enable us to use the KF directly. The non-linearity of the process or measurement models (or both) can be associated by the extended Kalman filter (EKF).

In EKF the state transition and observation models are assumed to be nonlinear, differentiable functions of the state. Let us express the state transition and observation equations:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_{k+1}) + \mathbf{w}_{k+1} \quad (3.1)$$

$$\mathbf{z}_{k+1} = h(\mathbf{x}_{k+1}) + \mathbf{v}_{k+1} \quad (3.2)$$

Here  $\mathbf{x}_{k+1}$  and  $\mathbf{x}_k$  are state vectors, and  $\mathbf{u}_{k+1}$  is the control vector at time  $k + 1$ . The process

and observation uncertainties at time  $k + 1$  are modeled by the random variables  $\mathbf{w}_{k+1}$  and  $\mathbf{v}_{k+1}$  respectively. The predicted state is computed from the present estimate by the state transition function  $f$  and similarly the measurement is computed from the predicted state by the observation function  $h$ . Since  $f$  and  $h$  are not directly applicable for the covariance calculation their Jacobian is calculated at each time step by using the latest predicted states. This Jacobian computation step is the key idea underlying the EKF approximation which is called *linearization*.

The Jacobian matrices can be used in KF equations resulting in the extended Kalman filter equations given in Table 3.1.  $\mathbf{Q}_{k+1}$  and  $\mathbf{R}_{k+1}$  are the covariance matrices of the posterior state

Table 3.1: The extended Kalman filter algorithm in discrete time

<p><b>Algorithm EKF</b>( <math>\mu_k, \Sigma_k, \mathbf{u}_{k+1}, \mathbf{z}_{k+1}</math> )</p> <p><i>Prediction:</i></p> $\bar{\mu}_{k+1} = f(\mu_k, \mathbf{u}_{k+1})$ $\bar{\Sigma}_{k+1} = \mathbf{F}_{k+1} \Sigma_k \mathbf{F}_{k+1}^T + \mathbf{Q}_{k+1}$ <p><i>Update:</i></p> $\mathbf{K}_{k+1} = \bar{\Sigma}_{k+1} \mathbf{H}_{k+1}^T (\mathbf{H}_{k+1} \bar{\Sigma}_{k+1} \mathbf{H}_{k+1}^T + \mathbf{R}_{k+1})^{-1}$ $\mu_{k+1} = \bar{\mu}_{k+1} + \mathbf{K}_{k+1} (\mathbf{z}_{k+1} - h(\bar{\mu}_{k+1}))$ $\Sigma_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1}) \bar{\Sigma}_{k+1}$ <p><i>Return</i>(<math>\mu_{k+1}, \Sigma_{k+1}</math>)</p>
--

and measurement noises. The state transition and measurement Jacobians are defined as

$$\mathbf{F}_{k+1} = \frac{\partial f}{\partial \mathbf{x}} \Big|_{\mu_k, \mathbf{u}_{k+1}}, \quad \mathbf{H}_{k+1} = \frac{\partial h}{\partial \mathbf{x}} \Big|_{\bar{\mu}_{k+1}} \quad (3.3)$$

Similar to the Kalman filter, the EKF represents the state at time  $k$  by the mean  $\mu_k$  and the covariance  $\Sigma_k$ . The filter updates these parameters when  $\mu_k$  and  $\Sigma_k$  are given as the input of the EKF together with the control  $\mathbf{u}_{k+1}$  and measurement  $\mathbf{z}_{k+1}$  for the next time step. The output is the estimate at time  $k + 1$ , represented by  $\mu_{k+1}$  and  $\Sigma_{k+1}$ .

In the prediction step the predicted belief  $\bar{\mu}_{k+1}$  and  $\bar{\Sigma}_{k+1}$  is calculated representing the belief for the next time step by only incorporating the control  $\mathbf{u}_{k+1}$  but not the measurement  $\mathbf{z}_{k+1}$ . In the update step the *Kalman gain*  $\mathbf{K}_{k+1}$  and *innovation*, which is the difference between the measurement  $\mathbf{z}_{k+1}$  and the expected measurement  $h(\bar{\mu}_{k+1})$ , are computed. Innovation is used to update the predicted belief into the desired one up to a degree specified by the Kalman gain. The derivation of the EKF algorithm is available in [51], [53], [54] and [55].

The major drawback of the extended Kalman filter is its non-optimality. The EKF may even quickly diverge if the modeling of the system is not correctly handled or if the initial state estimate is not close to the actual state. The unobservability of the process may also cause divergence of the EKF [55]. Yet the EKF can give reasonable results and in fact practically it is the standard estimation method for many applications such as navigation systems and GPS.

### 3.3 Simultaneous State and Parameter Estimation

The extended Kalman filter can be used for state estimation in nonlinear dynamic systems. In general the parameters of the state transition function  $f$  and measurement function  $h$  should be known for EKF employment. When there are unknown system parameters which are *continuous valued* or *slowly varying* it is still possible to obtain the state and the parameters simultaneously by means of an extended Kalman filter.

#### 3.3.1 The SSPE Algorithm

The simultaneous state and parameter estimation (SSPE) method suggests augmenting the *base state* by the unknown system parameters. Let us denote the unknown parameters by a vector  $\theta$ , the base state vector by  $\mathbf{x}$ , and the augmented state vector by  $\mathbf{y}$ . The state equation in (3.1) can be rewritten as

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \theta_k, \mathbf{u}_{k+1}) + \mathbf{w}_{k+1} \quad (3.4)$$

and the dynamic equation of the parameter vector is assumed to be time invariant

$$\theta_{k+1} = \theta_k \quad (3.5)$$

Then the *super-state*  $\mathbf{y}$  consist of  $\mathbf{x}$  and  $\theta$

$$\mathbf{y}_k \equiv \begin{bmatrix} \mathbf{x}_k \\ \theta \end{bmatrix} \quad (3.6)$$

Combining Eqns. (3.4), (3.5) and (3.6) we get the following expression for the super-state

$$\mathbf{y}_{k+1} = \hat{f}(\mathbf{y}_k, \mathbf{u}_{k+1}) + \hat{\mathbf{w}}_{k+1} \quad (3.7)$$

For completeness the minor update in equation (3.2) is also given as

$$\mathbf{z}_{k+1} = h(\mathbf{y}_{k+1}) + \mathbf{v}_{k+1} \quad (3.8)$$

The EKF can be used with these resulting state and observation equations (3.7) and (3.8) for the estimation of the entire super-state. Note that the measurement vector is not needed to be augmented or redefined as the state vector even if the measurement function  $h$  may also be a function of  $\theta$ .

The parameter dynamics are assumed to be constant by Eqn. (3.5). So their covariance will asymptotically converge to zero. In addition to this, if the process noise corresponding to these parameters is also assumed to be zero, then the filter gain will tend to zero which will result in wrong estimates. Thus the situation where the parameter variances are inclined to zero is undesirable. In order to overcome this problem an *artificial process noise*  $\mathbf{w}_{k+1}^\theta$  may be added to the unknown parameter vector. This correspond to replacing Eqn. (3.5) by

$$\theta_{k+1} = \theta_k + \mathbf{w}_{k+1}^\theta \quad (3.9)$$

The artificial noise  $\mathbf{w}_{k+1}^\theta$  prevents the variances of the parameter estimates from converging to zero, so that the calculated variances can be balanced with the estimation errors. This *pseudo-noise* assumption also enables the filter to estimate slowly varying parameters. The standard deviation of the process noise is generally chosen as a few percent of the guessed value of the parameter. Further *tuning of the filter* can be done as explained in [53].

### 3.3.2 An Example on SSPE

Let us now give an example about the use of EKF in a simultaneous state and parameter estimation problem. For simplicity assume that the base state is a one dimensional vector and the state equation be given as

$$x_{k+1} = a_{k+1} x_k + b_{k+1} u_{k+1} + w_{k+1}^{base} \quad (3.10)$$

where  $a_{k+1}$  and  $b_{k+1}$  are the unknown system parameters which may slowly vary in time, and  $w_{k+1}^{base}$  is the base state process noise at time  $k + 1$ . Let us also define the observation as a one dimensional vector and the related equation as

$$z_{k+1} = c_{k+1} x_{k+1} + v_{k+1} \quad (3.11)$$

where  $c_{k+1}$  is again an unknown measurement parameter.



As defined in section 3.3.1, the augmentation of the base state with the unknown parameters gives us a four-dimensional state vector  $\mathbf{y}_k$  and the corresponding process noise  $\mathbf{w}_k$ .

$$\mathbf{y}_k \equiv \begin{bmatrix} x_k \\ a_k \\ b_k \\ c_k \end{bmatrix} \quad \text{and} \quad \mathbf{w}_k \equiv \begin{bmatrix} w_k^{base} \\ w_k^a \\ w_k^b \\ w_k^c \end{bmatrix} \quad (3.12)$$

Note that  $w_k^a$ ,  $w_k^b$  and  $w_k^c$  are the pseudo process noises. The augmented state equation is then

$$\begin{aligned} \mathbf{y}_{k+1} &= \begin{bmatrix} y_k^1 \\ y_k^2 \\ y_k^3 \\ y_k^4 \end{bmatrix} = \begin{bmatrix} y_k^1 & y_k^2 + y_k^3 u_k \\ y_k^2 \\ y_k^3 \\ y_k^4 \end{bmatrix} + \begin{bmatrix} w_k^{base} \\ w_k^a \\ w_k^b \\ w_k^c \end{bmatrix} \\ &= f(\mathbf{y}_k, u_k) + \mathbf{w}_k \end{aligned} \quad (3.13)$$

which exactly has the same form as given in equation (3.7). The observation equation

$$\begin{aligned} z_{k+1} &= y_{k+1}^4 y_{k+1}^1 + v_{k+1} \\ &= h(\mathbf{y}_{k+1}) + v_{k+1} \end{aligned} \quad (3.14)$$

also has the form given in equation (3.8).

Let us define the process noise and measurement noise covariances.

$$\mathbf{Q} = \text{diag}(q_1, q_2, q_3, q_4) \quad \text{and} \quad \mathbf{R} = r_1 \quad (3.15)$$

For EKF implementation the Jacobian of the functions  $f$  and  $h$  should also be obtained. In our example these matrices are easily calculated as

$$\mathbf{F}_k = \begin{bmatrix} y_k^2 & y_k^1 & u_k & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{H}_k = \begin{bmatrix} y_{k+1}^4 & 0 & 0 & y_{k+1}^1 \end{bmatrix} \quad (3.16)$$

The super-state  $\mathbf{y}$  can now be estimated by using the EKF algorithm given in Table 3.1. With this simple example we have shown the steps of augmenting the state vector by the unknown system parameters and computing the necessary matrices to be able to use EKF for the parameter estimation. However note that the real life problems may be much more complicated than

our simple example. There may be too many unknown parameters and after the augmentation process the state vector dimension may become quite large. This will be shown by a robot mapping example.

### 3.3.3 An Example on Simultaneous Localization and Mapping

Simultaneous localization and mapping is one of the most fundamental problems in robotics. An autonomous vehicle or robot, which is moving in an unknown environment, should not only be able to estimate its state but also the unknown map parameters of its surroundings. This concurrent mapping and localization problem can in fact be treated as an SSPE problem.

Assume that a robot is moving on a planar surface with the ability to rotate to a desired direction. The base state of such a robot is composed of three elements  $p_k^x$ ,  $p_k^y$  and  $p_k^\theta$  which denote the robot's coordinates at time  $k$ . The 2D location of the robot is denoted by  $p_k^x$ ,  $p_k^y$  and the orientation is denoted by  $p_k^\theta$ . When the robot state is augmented with the unknown map parameters the combined vector is given by

$$\begin{aligned} \mathbf{y}_k &= \begin{pmatrix} x_k \\ \theta \end{pmatrix} \\ &= (p_k^x \ p_k^y \ p_k^\theta \ m_{1,x} \ m_{1,y} \ s_1 \ m_{2,x} \ m_{2,y} \ s_2 \ \dots \ m_{N,x} \ m_{N,y} \ s_N)^T \end{aligned} \quad (3.17)$$

where  $m_{i,x}$ ,  $m_{i,y}$  are the coordinates of the  $i$ -th landmark, for  $i = 1, \dots, N$  and  $s_i$  is its signature. The size of the base state is increased by  $3N$  where  $N$  denotes the number of landmarks in the map hence the dimension of the complete state is  $3N + 3$ .

In this example the number of landmarks are assumed to be predefined and known by the robot. A more general approach also assumes that the correspondences are not known and an incremental *maximum likelihood* (ML) estimator is used to determine these correspondences. In fact there are various SLAM algorithms that are produced to reply different requirements. Still, all these methods are built upon the basic SSPE algorithm presented in section 3.3.1.

There may be more than 1000 point landmarks which will result in a huge super-state vector even in the known correspondence case. The robot pose is defined by only three variables while the complete state dimension may reach up to thousands or even more. This clearly demonstrates that we may need to process a much larger super-state vector than the base state

that we want to estimate. Usually at each different orientation of the robot just a few landmarks are observable, so only a small part of the complete state is updated at each time step  $k$ . Instead of using the complete state with the basic EKF algorithm, the SLAM algorithms check whether the landmark has ever been seen before or not and updates the state by using the observed features only. In SLAM observing a landmark does not only improve the position estimate of that landmark but it also improves the robot pose estimate as well as the position estimates of the other landmarks. Since SLAM is not the main subject of this thesis it deemed not necessary to provide the complete SLAM algorithm. However the interested readers can find a complete study on SLAM in [54].

## CHAPTER 4

### Simultaneous Tracking and Camera Calibration

In this chapter the concept of simultaneous state and parameter estimation is applied to the problem of simultaneous tracking and camera calibration (STCC). The approach presented in this thesis suggests that an easy to carry rigid target and simple to generate 3D motion with well known dynamics can be used to generate the calibration data set. Newton's equations of motion fully describe the ballistic trajectory followed by a thrown target in 3D space, and they form the state transition equations for the base state. On the other hand external camera parameters can be taken as the static or slowly varying unknowns of the system and hence they can be made part of the state by augmenting the base state as described in section 3.3.1.

Dynamic motion model for a ballistically moving target and the model of an optical measurement device are given in section 4.1. The detection of the target from the camera views is explained in 4.2. Finally the problem is transformed into an SSPE form by the augmentation of the base state in section 4.3.

#### 4.1 Ballistic Motion and Camera Measurement Models

The state and observation equations of the system are needed for the use of an extended Kalman filter. The state equations of a ballistically thrown object and the observation equations of a camera under the assumption of central projection mapping is derived in sections 4.1.1 and 4.1.2 respectively.

### 4.1.1 Ballistic Motion Model

Objects under the influence of gravitational attraction follow a ballistic trajectory when thrown with an initial state (initial position and velocity vectors) if other disturbance effects like the air friction can be ignored. Under normal weather conditions (no wind, rain, or snow) and for short motion trajectories, the air-friction can be reasonably ignored for a dense object. Assume that  $y$ -axis coincide with the negative direction of gravity, hence placing the  $x$  and  $z$ -axes on the ground plane. Then the equations of motion for the trajectory of a ballistically moving object can be derived by using Newton's 2nd law. If the velocity of a thrown object is decomposed into its  $x$ ,  $y$  and  $z$  components the velocity update equation can be written as

$$\mathbf{V}_{k+1} = \begin{bmatrix} V_{k+1}^x \\ V_{k+1}^y \\ V_{k+1}^z \end{bmatrix} = \begin{bmatrix} V_k^x \\ V_k^y - g\Delta T \\ V_k^z \end{bmatrix} = \mathbf{V}_k + \begin{bmatrix} 0 \\ -g\Delta T \\ 0 \end{bmatrix} \quad (4.1)$$

The components which are parallel to the ground  $V_k^x$  and  $V_k^z$  do not change with respect to time since there are no force acting in these directions during the ballistic flight. Hence we can drop  $k$  from these components and use  $V^x$  and  $V^z$  hereafter. Gravitational force acts towards earth and therefore the velocity component in  $y$  direction decreases by an amount determined by the gravitational acceleration  $g$  and the time difference  $\Delta T$  between each time step. Integration of the velocity update equation w.r.t time yields the position update equation as

$$\begin{aligned} \mathbf{X}_{k+1} &= \begin{bmatrix} X_{k+1} \\ Y_{k+1} \\ Z_{k+1} \end{bmatrix} = \begin{bmatrix} X_k + V^x \Delta T \\ Y_k + V_k^y \Delta T - \frac{g\Delta T^2}{2} \\ Z_k + V^z \Delta T \end{bmatrix} \\ &= \mathbf{X}_k + \mathbf{V}_k \Delta T + \begin{bmatrix} 0 \\ -\frac{g\Delta T^2}{2} \\ 0 \end{bmatrix} \end{aligned} \quad (4.2)$$

The complete state equation in discrete-time can then be given by

$$\mathbf{s}_{k+1}^{dyn} = \mathbf{A} \mathbf{s}_k^{dyn} + \mathbf{u} \quad (4.3)$$

where  $\mathbf{s}_k^{dyn}$  stands for the dynamic system state at time  $k$ ,  $\mathbf{A}$  represents the state transition matrix and  $\mathbf{u}$  corresponds to the input as described in Eqns. (4.4), (4.5) and (4.6).

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta T & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

$$\mathbf{s}_k^{dyn} = \begin{bmatrix} X & Y & Z & V^x & V^y & V^z \end{bmatrix}_k^T \quad (4.5)$$

$$\mathbf{u} = \begin{bmatrix} 0 & -\frac{g\Delta T^2}{2} & 0 & 0 & -g\Delta T & 0 \end{bmatrix}^T \quad (4.6)$$

$[X \ Y \ Z]^T$ ,  $[V^X \ V^Y \ V^Z]^T$  are the position and velocity vectors of the ballistic target object and  $g$  represents the gravitational acceleration along the *negative*  $y$ -axis direction.  $\Delta T$  is the sampling period.

#### 4.1.2 The Pinhole Camera Model

In this section we give the pinhole camera model since it consist a fundamental part of the overall problem. A pinhole camera can be represented with a central projection mapping from the 3D world to a 2D image [7]. Using the homogenous coordinate system this mapping can be represented by a  $3 \times 4$  matrix  $\mathbf{P}$ , which can be decomposed into  $3 \times 3$   $\mathbf{K}$  and  $3 \times 4$   $[\mathbf{R} \ | \ \mathbf{t}]$  matrices. Here,  $\mathbf{K}$  represents the inner parameters of the camera such as focal length, aspect ratio, principal point offset, and skew, while  $\mathbf{R}$  and  $\mathbf{t}$  represent the external parameters such as camera orientation and displacement with respect to a world coordinate frame as illustrated in Figure 4.1. Here, we have vectors  $\mathbf{X}$  and  $\mathbf{x}$  as the homogenous points in 3D world and 2D image coordinates respectively. The defining equations of this projection are given in Eqns. (4.7), (4.8), and (4.9). Note that  $\mathbf{P}$  has 11 degrees of freedom: 5 for  $\mathbf{K}$  ( $\alpha_x$ ,  $\alpha_y$ ,  $x_0$ ,  $y_0$ ,  $s$ ), 3

for  $\mathbf{R}$  and 3 for  $\mathbf{t}$ .

$$\mathbf{x} = \mathbf{P}\mathbf{X} \quad (4.7)$$

$$\mathbf{P} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}] \quad (4.8)$$

$$\mathbf{K} = \begin{pmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.9)$$

Projection matrix,  $\mathbf{P}$ , can be divided into submatrices  $\mathbf{K}$  and  $[\mathbf{R} \mid \mathbf{t}]$  as shown in Eqn. (4.8). Note that the world and the camera coordinate frames are related due to a rotation and a translation.  $\mathbf{X}$ , is first multiplied with  $[\mathbf{R} \mid \mathbf{t}]$ , which corresponds to representing the 3D point in the camera coordinate frame. Then the resultant point passes through internal parameter matrix,  $\mathbf{K}$ , providing the 2D projection,  $\mathbf{x}$ . In Eqn. (4.9)  $\alpha_x$  and  $\alpha_y$  represent the focal length of the camera in terms of pixel dimensions towards  $x$  and  $y$  directions respectively;  $[x_0, y_0]^T$  is the principal point in terms of pixel dimensions and  $s$  is the skew parameter. Detailed information about camera models and multiple view geometry can be found in [7].

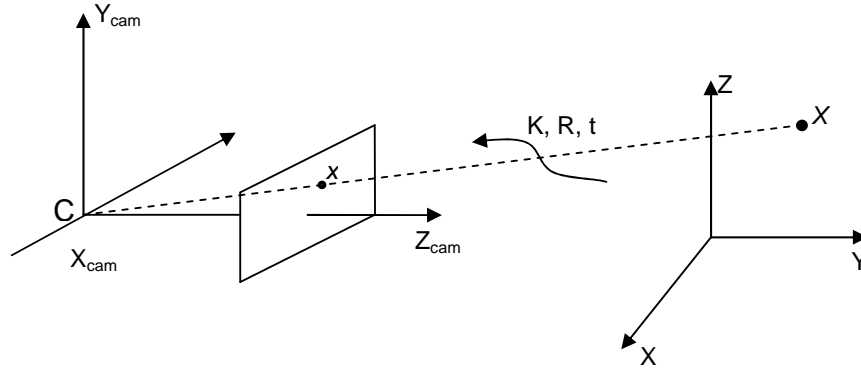


Figure 4.1: The Euclidean transformation between the world and camera coordinate frames.  $C$  is the camera center,  $X$  is the 3D world point,  $x$  is the 2D projection onto the camera image plane

## 4.2 Raw measurements: Detection of the Target

The EKF based algorithms discussed in this work clearly need the pixel coordinates of the calibration target in the image frames, possibly at sub-pixel resolution. In a simulation the target can be selected as a point mass with a predefined 3D trajectory, hence its projection, which absolutely corresponds to a single point on the image frame, can directly be computed

via pinhole camera model. However in reality, there is no exact knowledge about the actual target trajectory. Additionally, the calibration target may be chosen as a small ball which will definitely cover many pixels in the image frames. Therefore to provide an easy and robust calibration method, detection of the target and extraction of its center pixel location should be performed automatically for each calibration frame.

Using fixed cameras with a single moving target in the calibration frames reduces the complexity of the object detection problem. Hence the detection and localization steps at each calibration frame can be performed in an easy and reliable manner by the following algorithm:

- For each camera the mean of the calibration frames are obtained. Each mean image approximately provides the static background scene for the corresponding camera.
- The background image is subtracted from each calibration frame to obtain the target position at that frame.
- This difference image is converted to binary scale. Dilation and erosion operations are applied consecutively. Then,
  - if only a single blob exists, the pixels within the blob are assumed to correspond to the target projection. The center of the target (in pixel coordinates) is calculated by averaging the pixel coordinates.
  - if there are more than one blob, the operator is asked to select the target by clicking on the image that is being processed, so that the blob corresponding to the target can be separated from the others and averaging the pixels on that blob yields the target center.
  - else if there are no blobs left, the localization can not be achieved.

This algorithm can be used to distinguish any moving target on a constant background with subpixel accuracy. The examples of a mean, standard deviation, difference and binary images are shown in 4.2.

Note that if colored cameras are used, then the target color can particularly be chosen different from the background for easier detection. For example if the dominant surrounding color is green then the target can be chosen as red colored ball, and the given algorithm can exactly



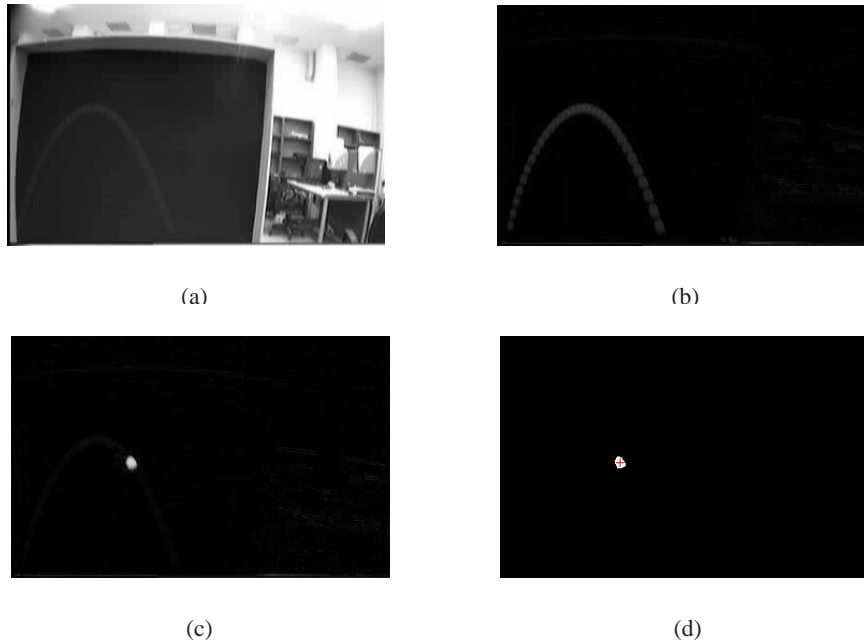


Figure 4.2: Detection of the target calibration object. (a) Mean image of the calibration frames. (b) Standard deviation image of the calibration frames. (c) Threshold image of a calibration frame. (d) Binary image of a calibration frame. The center of the target is also marked.

be applied in the red band. Such an algorithm is used by Svoboda et al in [29] for automatic detection of LED projections.

When the localization of the target is not achieved no measurement is available for the corresponding frame. Hence the measurement update equations in the EKF algorithm can not be executed at that time step. But the state can still be updated and the filter can progress to the next time step. So missing of the target position in a few frames is not a big deal since omission of the measurement update once in a while does not practically prevent the filter to converge.

### 4.3 Definition of Super-State

The motion model, the camera model and the detection of the target in the calibration frames are explained in the previous parts. From now on the transformation of the problem into the SSPE framework is demonstrated. In section 4.3.1 the external camera parameters are defined and the base state is augmented with these unknown parameters, and in section 4.3.2 the state

and measurement equations are extended for the overall state.

### 4.3.1 Augmented System Representation

In the following discussion, the system is described and the augmented state representation is defined for the two camera case. The results can quite easily be generalized to a multi camera problem but this generalization is considered to be beyond the scope of this section.

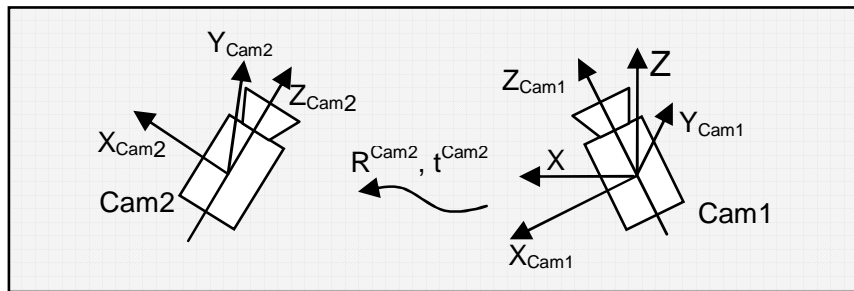


Figure 4.3: Two-camera system assumed for illustration purposes to be mounted on a fixture with the unknown rotation and translation between them consisting the external calibration parameters.

The system is composed of two arbitrarily placed cameras whose internal calibration parameters are predetermined. For illustration of the basic ideas, suppose the two cameras are rigidly mounted on a fixture as shown in 4.3. Assume the cameras are labeled as Cam1 and Cam2. Our aim is to find the unknown but constant rotation and translation of the cameras with respect to the basis frame.

Let  $x$ ,  $y$  and  $z$ -axes form a right handed reference frame that we call the *basis coordinate frame*. Without losing generality centers of the basis frame and the *Cam1 coordinate frame* are assumed to be coincident as shown in 4.4. Actually the necessity of a separate basis frame instead of using Cam1-frame as the reference frame arises from the fact that the gravitational pull is towards the center of the *earth*. Therefore  $y$ -axis should always coincide with the negative gravity vector direction in order to use Eqn. (4.3).

Note that as long as the center of the calibration target object thrown into the camera FOV is the only processed feature in the image, the rotation of the fixture around  $y$ -axis is undetectable. To visualize this, assume that the mounting fixture is rotated around  $y$ -axis. The position of the trajectory in the image plane will change. However, since the global loca-

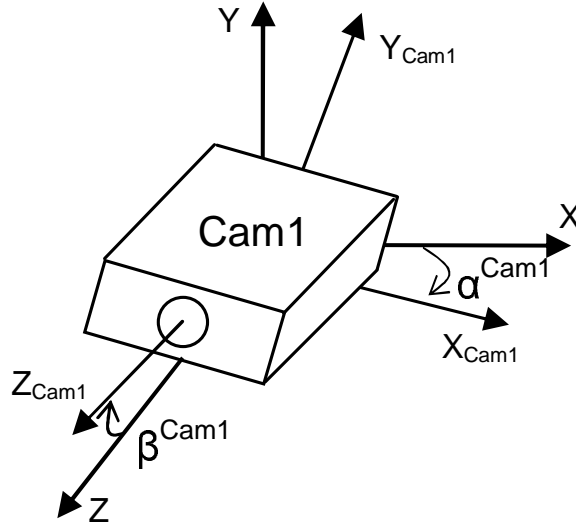


Figure 4.4: First camera shown on the *basis coordinate frame*. The relation between the basis frame and *Cam1 coordinate frame* is represented with the rotation angles  $\alpha^{c1}$  and  $\beta^{c1}$  around  $z$  and  $x$ -axes respectively.

tion of this trajectory is also unknown (except for the direction of the gravity vector which is observable from the trajectory) we cannot deduce this global rotation angle. Therefore, we conclude that the measurements taken from the target are invariant of the rotation around  $y$ -axis.

On the other hand, a rotation of Cam1 around  $z$  and  $x$ -axes will cause the object to accelerate at a different direction in the image plane measurements, indicating the global direction of gravity. Therefore these rotations of the fixture,  $\alpha^{c1}$  and  $\beta^{c1}$ , are measurable and are calibration unknowns of the process which should be determined. Rotation and translation of Cam2, on the other hand, is completely independent from the basis frame, and therefore all these 6 parameters (3 rotations and 3 translations) are calibration parameters that are also to be determined.

The EKF algorithm can be used in order to track the state of the calibration object and simultaneously estimate the static unknown parameters of the camera system. Dynamic object state,  $\mathbf{s}^{dyn}$ , can be augmented with the static camera parameters to form a super-state, which will be used as the state vector in the EKF algorithm formulation.

The calibration object state,  $\mathbf{s}^{dyn}$ , is already defined in Eqn. (4.5). Now, let  $\mathbf{s}^{sta}$  represent the static calibration parameters. Then the augmented super-state is the concatenation of  $\mathbf{s}^{dyn}$  with  $\mathbf{s}^{sta}$  as given in Eqn. (4.10). This augmentation procedure enables us to use such a powerful

noisy state estimation tool, EKF, for static parameter estimation.

$$\mathbf{s} = \begin{pmatrix} \mathbf{s}^{dyn} \\ \mathbf{s}^{sta} \end{pmatrix}, \quad (4.10)$$

$$\mathbf{s}^{sta} = (\alpha^{c1} \beta^{c1} \alpha^{c2} \beta^{c2} \Gamma^{c2} C_x^{c2} C_y^{c2} C_z^{c2})^T \quad (4.11)$$

where  $\alpha^{c1}$  and  $\beta^{c1}$  are the rotation angles of Cam1 around  $z$  and  $x$ -axes respectively;  $\alpha^{c2}$ ,  $\beta^{c2}$  and  $\Gamma^{c2}$  are the rotation angles of Cam2 around  $z$ - $x$ - $z$ -axes respectively, and finally  $C_x^{c2}$ ,  $C_y^{c2}$  and  $C_z^{c2}$  represent the center location of Cam2 with respect to the basis frame.

### 4.3.2 Augmented System Equations

We will now define the system state and measurement equations in a state space representation which are necessary for the EKF implementation. Since the camera parameters are invariant in time, their update matrix can be represented with an  $8 \times 8$  identity matrix and the overall state update equation can be written as

$$\mathbf{s}_{k+1} = \hat{\mathbf{A}}\mathbf{s}_k + \hat{\mathbf{u}} \quad (4.12)$$

where

$$\hat{\mathbf{A}} = \begin{pmatrix} \mathbf{A} & \mathbf{0}_{6 \times 8} \\ \mathbf{0}_{8 \times 6} & \mathbf{I}_{8 \times 8} \end{pmatrix}, \quad \hat{\mathbf{u}} = \begin{pmatrix} \mathbf{u} \\ \mathbf{0}_{8 \times 1} \end{pmatrix} \quad (4.13)$$

The measurement equations (4.14) and (4.15) are directly written by using the pinhole camera model which is explained in section 4.1.2. The tracked point (i.e. the center of the calibration object) is projected into the camera planes; hence our measurements are the pixel locations of the projection. 3D target position  $(X \ Y \ Z)^T$  is projected to give us corresponding 2D measurements  $(u^{c1} \ v^{c1})^T$  and  $(u^{c2} \ v^{c2})^T$ .

$$\lambda \begin{pmatrix} u^{c1} \\ v^{c1} \\ 1 \end{pmatrix}_k = \mathbf{K}_1 \left[ \mathbf{R}^{c1}(\alpha, \beta) \quad \mathbf{0}_{3 \times 1} \right] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}_k \quad (4.14)$$

$$\lambda \begin{pmatrix} u^{c2} \\ v^{c2} \\ 1 \end{pmatrix}_k = \mathbf{K}_2 \left[ \mathbf{R}^{c2}(\alpha, \beta, \Gamma) \quad \hat{\mathbf{t}}^{c2} \right] \begin{pmatrix} Z \\ Y \\ Z \\ 1 \end{pmatrix}_k \quad (4.15)$$

$$\hat{\mathbf{t}}^{c2} = -\mathbf{R}^{c2}(\alpha, \beta, \Gamma) \begin{pmatrix} C_x^{c2} \\ C_y^{c2} \\ C_z^{c2} \end{pmatrix} \quad (4.16)$$

$\mathbf{K}_1$  and  $\mathbf{K}_2$  are the internal calibration matrices for Cam1 and Cam2.  $\mathbf{R}^{c1}$  and  $\mathbf{R}^{c2}$  stands for the rotation matrices of Cam1 and Cam2 with respect to the *basis frame*. Finally  $\hat{\mathbf{t}}^{c2}$  is the translation of Cam2 in the world coordinate frame and it can be formulated in terms of the state variables as in Eqn. (4.16).

## CHAPTER 5

### Estimating the Super-State

In order to use the Kalman filter, the observations should be linear functions of the state and the next state should be a linear function of the previous state. In our problem although the state transition equation is a linear one, the measurement equations are highly nonlinear which removes the possibility of direct KF employment at its simplest form. Instead, an extended Kalman filter, which can handle the non-linearities of the observation equation, should be used.

Let the measurements  $(u^{c1} \ v^{c1})^T$  and  $(u^{c2} \ v^{c2})^T$  be concatenated to form the complete measurement vector,  $\mathbf{z}$ , as

$$\mathbf{z} = (u^{c1} \ v^{c1} \ u^{c2} \ v^{c2})^T \quad (5.1)$$

Then the disturbances affecting the process and observations should be modeled by adding random white Gaussian noises  $\mathbf{v}$  and  $\mathbf{w}$  to obtain the usual EKF relations

$$\mathbf{s}_{k+1} = \hat{\mathbf{A}}\mathbf{s}_k + \hat{\mathbf{u}} + \mathbf{w}_{k+1} \quad (5.2)$$

$$\mathbf{z}_{k+1} = h(\mathbf{s}_{k+1}, \mathbf{u}_{k+1}) + \mathbf{v}_{k+1}. \quad (5.3)$$

where  $h$  is obtained from (4.14), and (4.15). The process noise  $\mathbf{w}$  models the disturbances affecting the target motion such as air friction while the observation noise  $\mathbf{v}$  models the disturbances affecting the cameras such as possible vibrations during filming or other pixel errors. The process and observation noise covariances are given as

$$\mathbf{Q} = \text{diag}(q_1, q_2, \dots, q_{13}, q_{14}) \quad \text{and} \quad \mathbf{R} = \text{diag}(r_1, r_2, r_3, r_4). \quad (5.4)$$

For EKF implementation the Jacobian of  $h$  is computed by MATLAB 7.1. Note that together with the state transition matrix  $\hat{\mathbf{A}}$ , this Jacobian matrix  $\mathbf{H}$  determines how well internal states

of our system can be inferred by knowledge of its outputs, in other words the *observability* of the system. For that purpose numerical matrix entries can be regularly checked during the EKF iterations. The Jacobian calculation is a simple procedure, however since our observation equations are highly nonlinear the resultant  $\mathbf{H}$  matrix is extremely complex to give in the content of this thesis.

At this point all the necessary equations and matrices are obtained to reduce the problem to an EKF framework.

## 5.1 Standard EKF Algorithm

In most applications, EKF is used for tracking a platform in real time. Sensor measurements are continuously processed by the EKF together with the system dynamics and a suboptimal state estimation is performed. Even if the initial state of the system is not known exactly, the filter converges to the true state in time.

In our problem we have checked the convergence of the EKF by implementing the filter for the calibration of a two camera system in a simulation environment. Even though the super-state is set to a wrong initial value, EKF can yield accurate estimates of the static calibration parameters as well as the final target state. However, for the filter to converge to the true state, about 1000 filter iterations are performed and hence that many data points of the flying target is used during the iterations. Although in a simulation environment data points can be supplied as much as desired by increasing the fps rates of the cameras, for real camera setups that can not be done. For a standard 30 fps camera setup, it takes more than half a minute to capture 1000 frames. That is too long time for a ballistically moving target to stay in the joint-FOV of the cameras. For reasonable camera poses, when the calibration object is thrown into the joint-FOV it may stay there for about one second and only 30 measurements can be obtained by a 30 fps camera setup. How can we satisfy the EKF to converge with so few measurements?

In order to solve this convergence problem of EKF with limited number of calibration data, 2 different algorithms are proposed: Repeated EKF and Repeated Bidirectional EKF. As understood from their names the essence of these methods depends on using the same measurements repeatedly.

## 5.2 Repeated EKF Algorithm

Let us first divide the state,  $\mathbf{s}$  into its sub-parts:

$$\mathbf{s} = \begin{pmatrix} \mathbf{X} & \mathbf{V} & \mathbf{C} \end{pmatrix}^T \quad (5.5)$$

where  $\mathbf{X}$  corresponds to the position,  $\mathbf{V}$  corresponds to the velocity of the calibration object, and  $\mathbf{C}$  stands for the calibration parameters. Let us continue with the assumption that a target is thrown into the joint FOV of a two camera system, and the ballistic trajectory of the target is simultaneously captured by each camera yielding 30 frames to be used for calibration. Starting from a wrong initial state, the calibration parameters would not be obtained accurately after only 30 filter iterations.

The very first idea to solve this problem is to repeatedly use the measurements at hand. Let us define the *filter* and *throw* iterations:

- *Filter Iteration*: The extraction of the next state from the present state; in other words a single iteration of the EKF
- *Throw Iteration*: All filter iterations for one set of measurement data (i.e., 30 iterations of the filter).

After the completion of one throw iteration the state will not probably converge, but it will be estimated to some extent. We suggest keeping the calibration parameters at their final values and starting the extended Kalman filter again and again (Figure 5.1). So, even though the same observations are used, the state can be estimated better and better. Since the calibration is performed off-line, there is enough time for the repetition of the iterations.

The idea here is to make many throw iterations and hope for the calibration parameters to converge to their real values in a reasonable time. The calibration parameters are kept and provided to the next throw iteration while the target state is reset to its initial value, so the path followed by the ball is estimated better and better since the calibration parameters approach towards their real values and vice versa. This “*keep the static parameters, reset the dynamic state*” idea forms the essence of the **repeated EKF** (R-EKF) algorithm.

It should be reminded that simple KF yields the optimal solution with a single use of the measurements. So repeated use of the same measurements can not improve the results even



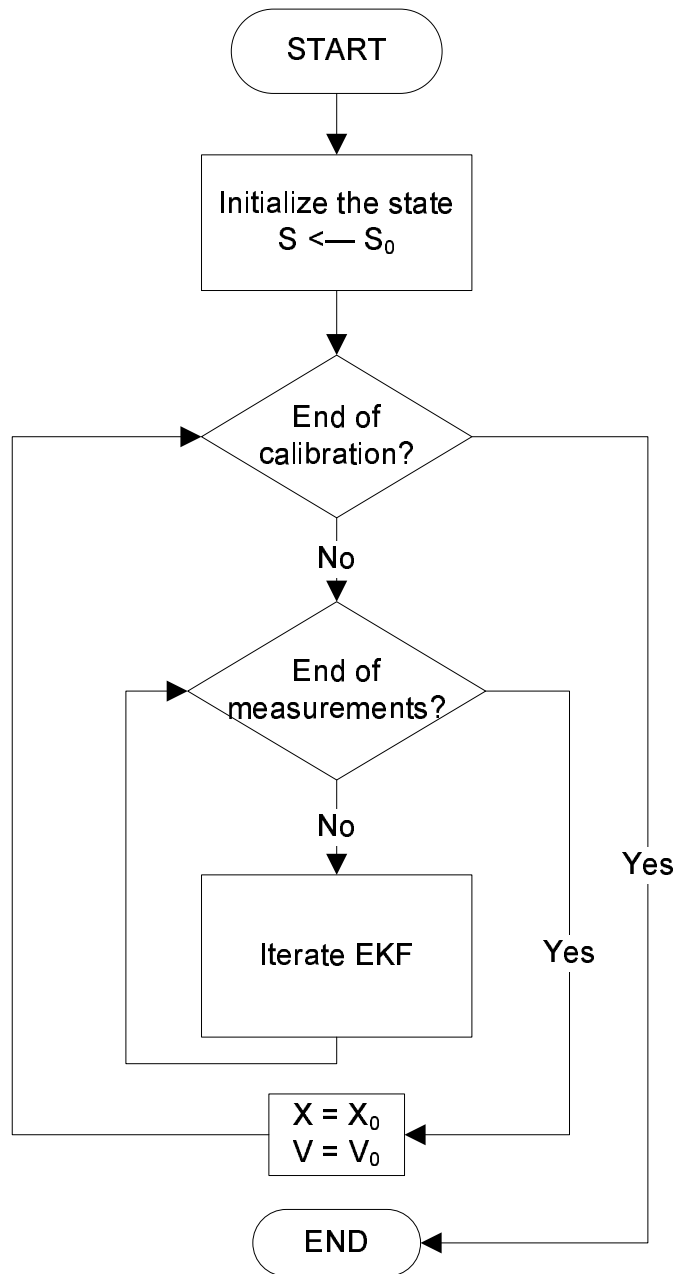


Figure 5.1: The flow chart for the repeated EKF algorithm.

after smoothing the state. On the other hand the EKF linearizes the non-linear functions at the predicted state, hence linearization performance can be improved by smoothing the state. Therefore even by using the same measurements, better estimates can be obtained by an EKF due to this improvement in the linearization.

The drawback of the R-EKF algorithm is that the results are highly dependent on the initial uncertainty of the dynamic state. Actually not only the camera parameters but also the target position and velocity are not available at the beginning, because unfortunately the target is thrown at an arbitrary position with an unknown velocity. The initial state of the calibration target is not estimated during the iterations, but always the originally given  $[\mathbf{X}_0 \mathbf{V}_0]^T$  vector is used. So the estimate for this initial state vector should be provided via external measurements as in [50]. Human factor and measurement errors which may badly affect the calibration results is inevitable with this approach.

### 5.3 Repeated Bidirectional EKF Algorithm

The second idea is to implement EKF in both forward and backward directions. The calibration object state and the camera parameters are again estimated repeatedly, but this time in two directions. In other words the target motion is first tracked in the forward direction and then in the reverse direction. This algorithm remedies the main problem of R-EKF since the initial target state is predicted when the reverse motion estimation is performed.

The flow sequence of the **repeated bidirectional EKF** (RB-EKF) algorithm is shown by the flowchart in Figure 5.2. The augmented state is set to  $\mathbf{s}_0$  initially and *forward throw* estimation is performed till all the frames are swept one time. Then the target trajectory is estimated from backwards by using the measurements in reverse order. Note that the velocity vector should be negated at the end of a *forward* and *backward throw* iteration while the position vector and camera parameters remain same.

RB-EKF yields both the dynamic and static states with negligible error in a short time. As stated beforehand the accuracy of the estimations depends on the initial state uncertainty, hence being able to estimate the initial state reduces this dependency. Removing the necessity of measuring the initial target state and hence avoiding the human related errors, this method not only provides an easily applicable calibration, but it also reduces the final error.

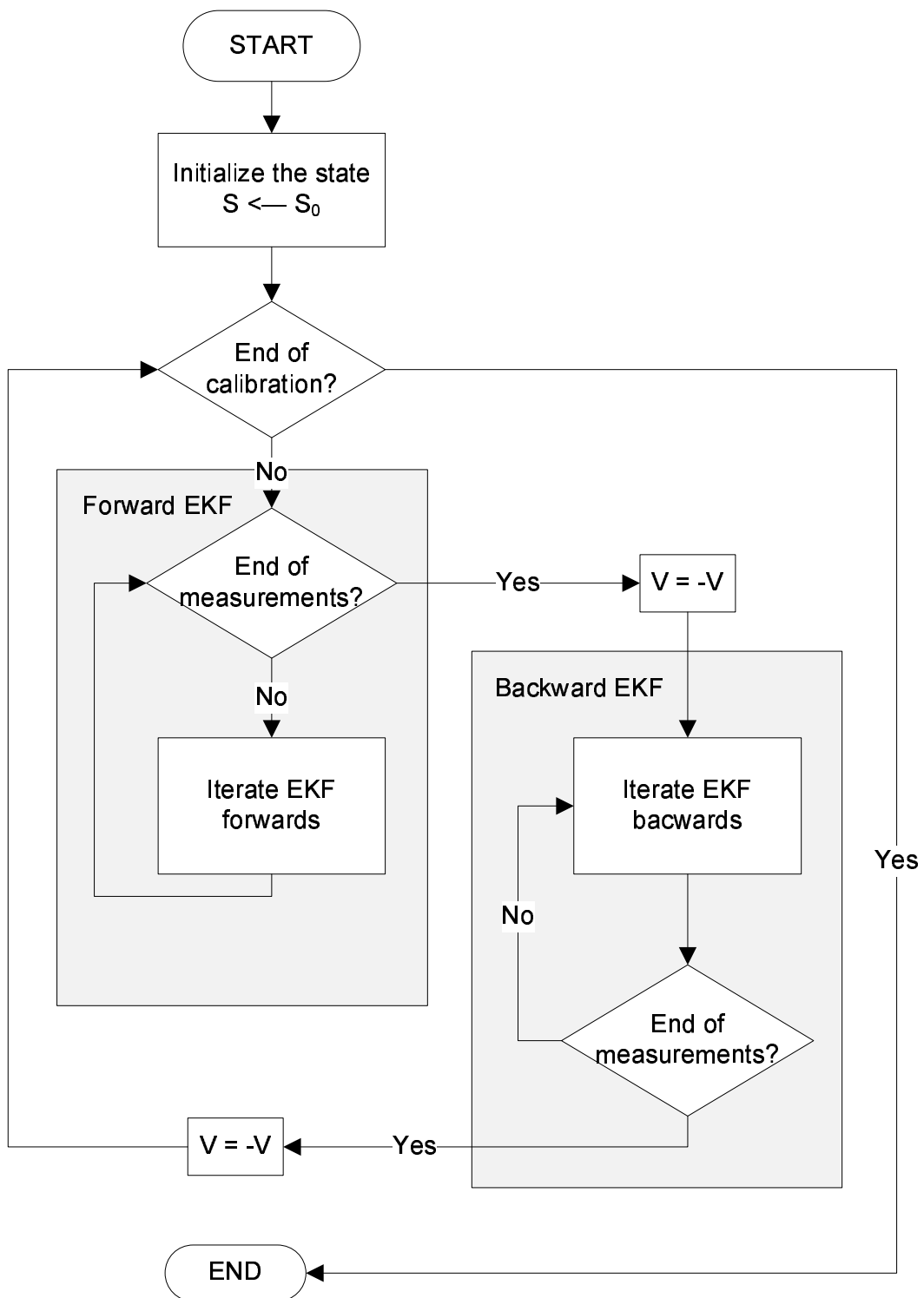


Figure 5.2: The flow chart of the repeated bidirectional EKF algorithm.

## CHAPTER 6

### Simulation Experiments

In this chapter the experiments which are performed on a computer simulation environment are presented. The proposed calibration method is tested on a simulation platform prepared in MATLAB7.1. For data generation, instead of the real camera images of a dynamically moving target, the dynamic motion model and pinhole camera model are used to generate simulated data points.

The experiments are chosen to identify the scenarios that can be encountered in real life problems. First a target in free fall is used for calibration with R-EKF and RB-EKF algorithms in sections 6.1 and 6.2 respectively. Section 6.3 focuses on the comparison of R-EKF with RB-EKF when the initial target state is not known. White Gaussian random process and measurement noises are also added on the measurement data for a more realistic evaluation. In section 6.4 the performance of RB-EKF is investigated for a ballistic trajectory and then multiple trajectories are used for calibration in section 6.5. Finally a general discussion of the simulation experiments is given in section 6.6.

#### 6.1 Experiment 1: R-EKF / Free Fall Motion

In our first simulation experiment we try to initially show that R-EKF method can yield camera poses under some facilitative but restrictive assumptions. As discussed in Chapter 5, the performance of the R-EKF algorithm is highly dependent on the initial uncertainty of the dynamic state. In order to use R-EKF efficiently, the initial state of the calibration target should be measured and provided to initialize the EKF, which makes R-EKF an impractical algorithm to use on the field. Since this experiment is a preliminary testing of the concept, we assume

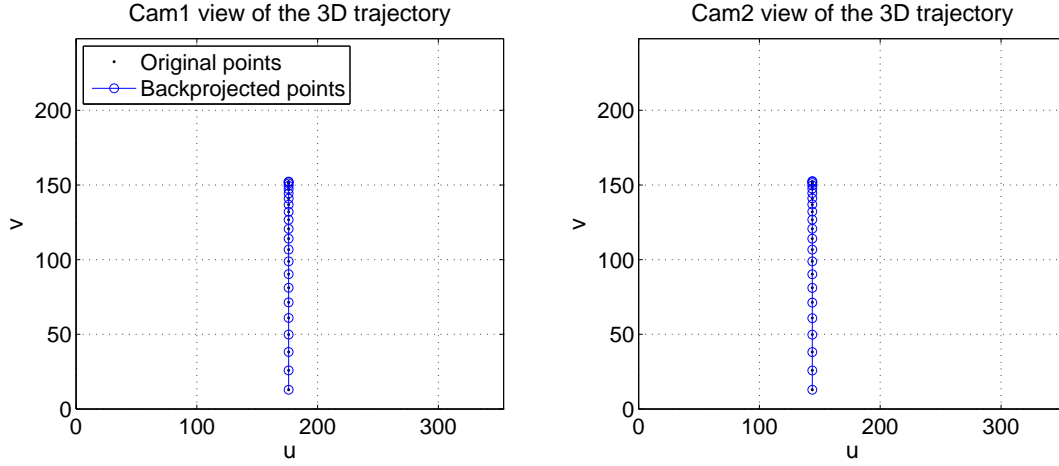


Figure 6.1: The synthetic images of a free falling target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated EKF algorithm.

that the initial target state is known even though such an assumption is not realistic.

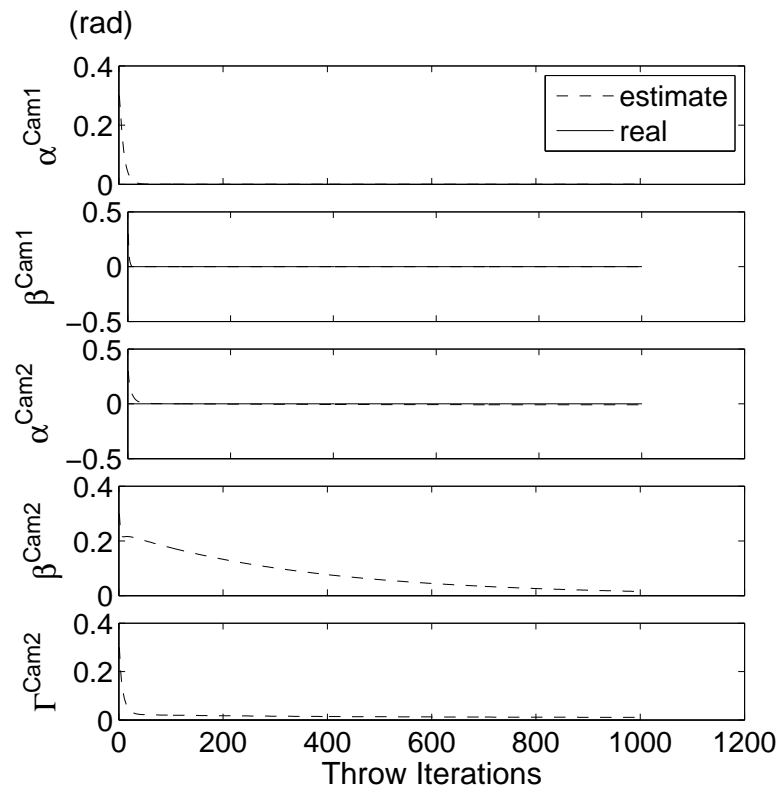
For a real case the initial position of the calibration target can approximately be measured by means of a ruler although that will not be practical. However to obtain its initial velocity is even harder. Yet a *free falling* target may reduce the unknowns since the initial velocity of a free falling object is a well known  $\vec{0}$  m/s vector. So, a target can be dropped into the joint FOV of a multi camera system and the captured image sequence can be used for the external calibration. In this experiment we try to inspect the convergence performance of the R-EKF algorithm by using the synthetic data of a free falling target.

In our simulation setup two cameras are placed in parallel with 0.2 m distance between them and zero rotation with respect to the basis frame. Such a configuration is similar to a human's vision system, so for easy understanding one can assume that a stereo camera is mounted on a planar platform which is placed parallel to the ground. When a point mass falls freely in front of the camera setup, the image sequences obtained by these two cameras are shown in Figure 6.1.

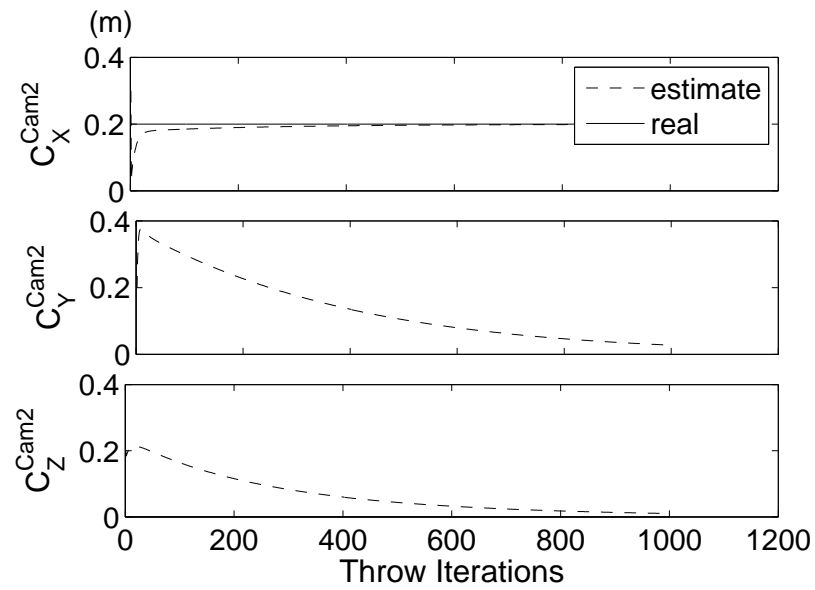
The target is dropped from 0.25 m above, 1.8 m in front of Cam1. So its initial condition vector is equal to

$$\mathbf{X}_0 = [0 \ 0.25 \ 1.8]^T, \quad \mathbf{V}_0 = [0 \ 0 \ 0]^T.$$

The camera parameters are all equal to zero except the  $x$  position of Cam2 center, due to the



(a)



(b)

Figure 6.2: Convergence of the Repeated EKF Algorithm w.r.t *throw* iterations. All calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters.

0.2 m gap between the cameras in  $x$  direction. Hence the original parameter vector is equal to

$$\mathbf{C} = [0 \ 0 \ 0 \ 0 \ 0 \ 0.2 \ 0 \ 0]^T.$$

The actual initial target state,  $[\mathbf{X}_0 \ \mathbf{V}_0]$ , is exactly used for initializing the R-EKF, so

$$\hat{\mathbf{X}}_0 = [0 \ 0.25 \ 1.8]^T, \quad \hat{\mathbf{V}}_0 = [0 \ 0 \ 0]^T$$

and the initial estimate vector for the camera parameters is assumed to be

$$\hat{\mathbf{C}}_0 = [0.3 \ 0.3 \ 0.3 \ 0.3 \ 0.3 \ 0.3 \ 0.2 \ 0.2]^T.$$

As illustrated in Figure 5.1 during the throw iterations the initial state of the target  $[\mathbf{X}_0^T \ \mathbf{V}_0^T]^T$  is repeatedly used while the calibration parameter estimates  $\hat{\mathbf{C}}$  converge towards their actual values  $\mathbf{C}$ . The convergence curves of the rotation and translation parameters are shown in Figure 6.2

The evaluation of 1000 throw iterations by R-EKF algorithm has taken for about 70.9 seconds with a hyper-threaded Pentium4, 3.0GHz CPU. The camera parameters are estimated as

$$\hat{\mathbf{C}} = [0.0000 \ 0.0000 \ -0.0092 \ 0.0151 \ 0.0109 \ 0.1992 \ 0.0271 \ 0.0101]^T$$

which is almost same as the original parameter vector  $\mathbf{C}$ . The re-projection pixel error is computed as  $6.5712 \times 10^{-4}$  pixels for Cam1 and 0.1056 pixels for Cam2. Note that this is the error in the image plane. In a simulation since the real parameters are known we can obtain the norm of the difference between the real camera parameters and the estimated ones. The average of the rotation and translation errors for both cameras are obtained as 0.0208 radians and 0.0290 meters respectively.

In this experiment we have shown that if the initial state of the calibration object is measurable, then R-EKF algorithm can yield accurate results. However, since the initial state of the target is not estimated during iterations, the results are highly dependent on the initial state uncertainty. This is the major drawback of the R-EKF method. Also note that throughout this experiment we assume there are no disturbances which would cause noisy measurements. But unfortunately in real life unwanted disturbances do exist. That's why without noise analysis the actual performance of the R-EKF algorithm can not be fully understood. In fact due these disturbances we even may not be able to obtain the initial state of the calibration target accurately which may tragically diminish the success of the R-EKF method.

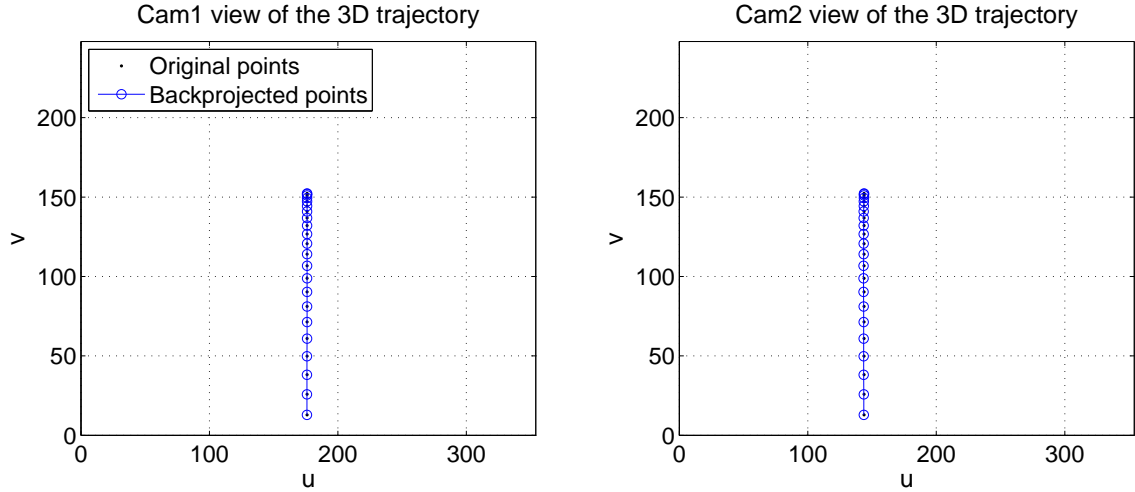


Figure 6.3: The synthetic images of a free falling target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm.

## 6.2 Experiment 2: RB-EKF / Free Fall Motion

In this section the first experiment is repeated for the RB-EKF algorithm. We try to show that RB-EKF converges under facilitating assumptions. The same orientation of the camera setup is used again to capture the same free falling motion of the target. Hence  $\mathbf{X}_0$ ,  $\mathbf{V}_0$  and  $\mathbf{C}$  vectors have the same values as in section 6.1. The initial estimate of the target state and the camera parameter vectors  $\hat{\mathbf{X}}_0$ ,  $\hat{\mathbf{V}}_0$ ,  $\hat{\mathbf{C}}_0$  are also provided to the RB-EKF algorithm as in section 6.1.

The original and re-projected points are shown in Figure 6.3 and the convergence curves of the RB-EKF algorithm are plot in Figure 6.4. Note that RB-EKF algorithm estimates not only the camera parameters but also the initial target state. However in this experiment the initial target state is exactly provided and since trying to estimate an already given parameter may lead to some error, the resultant calibration error of the RB-EKF is slightly more than that of the R-EKF.

It takes 136.7 seconds to complete 1000 throw iterations for RB-EKF algorithm. This is nearly twice as much as it takes with the R-EKF method, because one throw iteration of RB-EKF includes both the forward and backward iteration of the target movement. The calibration



vector is estimated as

$$\hat{\mathbf{C}} = [0.0004 \ 0.0450 \ -0.0120 \ 0.0454 \ 0.0102 \ 0.2048 \ 0.0003 \ 0.0006]^T$$

which is very close to the original parameter vector  $\mathbf{C}$ . The re-projection error is computed as 0.0227 pixels for Cam1 and 0.1613 pixels for Cam2. The rotation and translation errors are obtained as 0.0658 radians and 0.0048 meters respectively.

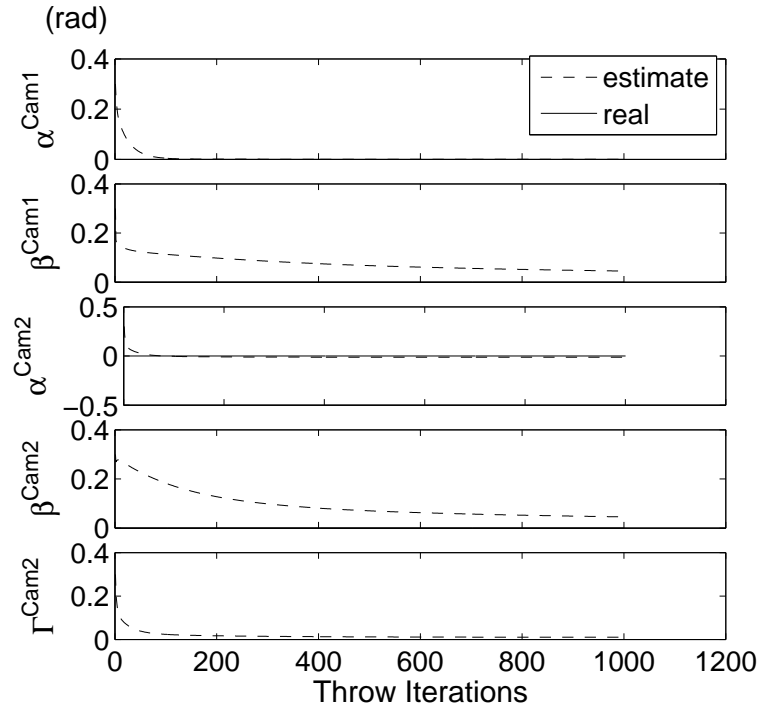
In this experiment we have shown that if the initial state of the calibration object can be measured accurately, RB-EKF algorithm can yield the parameter estimates accurately. Note that RB-EKF can also handle initial uncertainty since the initial target state is also estimated. That makes RB-EKF a practical method that may be used on the field applications because very accurate initial state measurement is quite difficult to obtain in real cases. Even when one tries to measure the position of the target by means of a ruler, the uncertainty of the initial state is unavoidable. Therefore we should further test RB-EKF method under more realistic assumptions.

### 6.3 Experiment 3: Comparison of R-EKF and RB-EKF

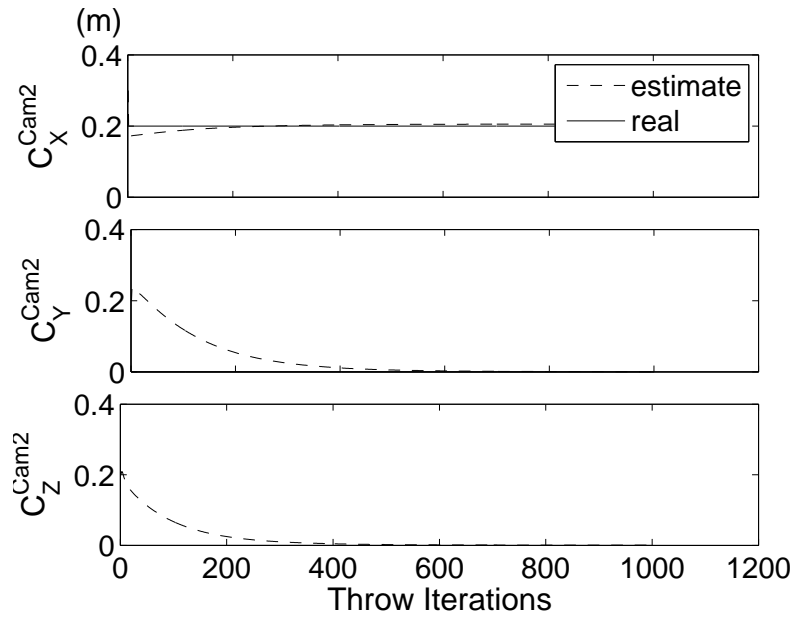
In the previous experiments we have mentioned that the initial state may not be accurately measurable and also other disturbances may affect the dynamic motion of the target object as well as its projections on to the image plane. The external disturbances may cause the actual target motion deviate from the dynamic target motion model and also the actual camera measurements deviate from the ones that would be obtained by the pinhole camera model. In this part the R-EKF and RB-EKF algorithms are compared under reasonable initial uncertainty and noise effects.

Introduction of an additional measurement system to obtain an accurate initial estimate of the target state would result in a complicated and pointless calibration procedure. Because if one already has the necessary equipment to obtain the target pose in an accurate manner, then that may also be used to obtain the camera poses. So a practical and efficient calibration method, which uses the EKF for parameter estimation, should be able to tolerate the uncertainties of the initial target state.

In this part the camera poses and the target trajectory are used as in section 6.1, but this time



(a)



(b)

Figure 6.4: Convergence of the Repeated Bidirectional EKF Algorithm w.r.t *throw* iterations. All calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters.

the initial target state and the camera poses are assumed to be guessed by the operator by only inspecting the system. Therefore the initial position uncertainty must be tolerated by at least half a meter, because that much mistake is fairly reasonable for human prediction. The initial velocity of the free falling target is assumed to be  $\vec{0}$  m/s. Hence

$$\hat{\mathbf{X}}_0 = [0 \ 0 \ 1.5]^T, \quad \hat{\mathbf{V}}_0 = [0 \ 0 \ 0]^T$$

vectors may be accepted as reasonable initial estimates for the target state. Then the initial super-state estimate is given as

$$\mathbf{s}_0 = [\hat{\mathbf{X}}_0^T \ \hat{\mathbf{V}}_0^T \ \hat{\mathbf{C}}_0^T]^T \quad (6.1)$$

where the initial camera parameter vector  $\hat{\mathbf{C}}_0$  is taken same as the first experiment.

In order to model the disturbances, process and observation noises are added while obtaining the camera views of the falling object. The target position and the camera measurements are disturbed with White Gaussian noises. The process and observation noise variances are set to 0.002 m and 0.1 pixels respectively.

The measurements and the re-projected points obtained after 1000 throw iterations with R-EKF and RB-EKF methods are shown in Figure 6.5. The convergence curves of the R-EKF and RB-EKF algorithms are illustrated in Figure 6.6 and Figure 6.7. The resultant errors for both algorithms are presented in Table 6.1.

The final estimation vector of the camera parameters are obtained as

$$\begin{aligned} \hat{\mathbf{C}}^{R-EKF} &= [0.0015 \ 0.1916 \ -0.0125 \ 0.1263 \ 0.0318 \ 0.1852 \ -0.0683 \ -0.1170]^T \\ \hat{\mathbf{C}}^{RB-EKF} &= [0.0193 \ 0.0828 \ 0.0194 \ 0.0824 \ -0.0223 \ 0.2345 \ -0.0021 \ 0.0053]^T. \end{aligned}$$

The initial target state is also estimated by RB-EKF as

$$\hat{\mathbf{X}}_0 = [-0.0045 \ 0.1173 \ 2.1923]^T, \quad \hat{\mathbf{V}}_0 = [0.0379 \ -0.1758 \ -0.7339]^T.$$

RB-EKF has clearly better performance in this more realistic example. The final camera parameters obtained by RB-EKF turned up to be closer to their actual values compared to R-EKF. The final re-projection, angle and translation errors for RB-EKF are obtained to be quite smaller than those for R-EKF as seen in Table 6.1. Also when the convergence curves in Figure 6.6 and Figure 6.7 are compared, RB-EKF seems to be the superior method between the two. Therefore it is concluded that RB-EKF should be preferred over R-EKF when the initial target state is not known accurately.

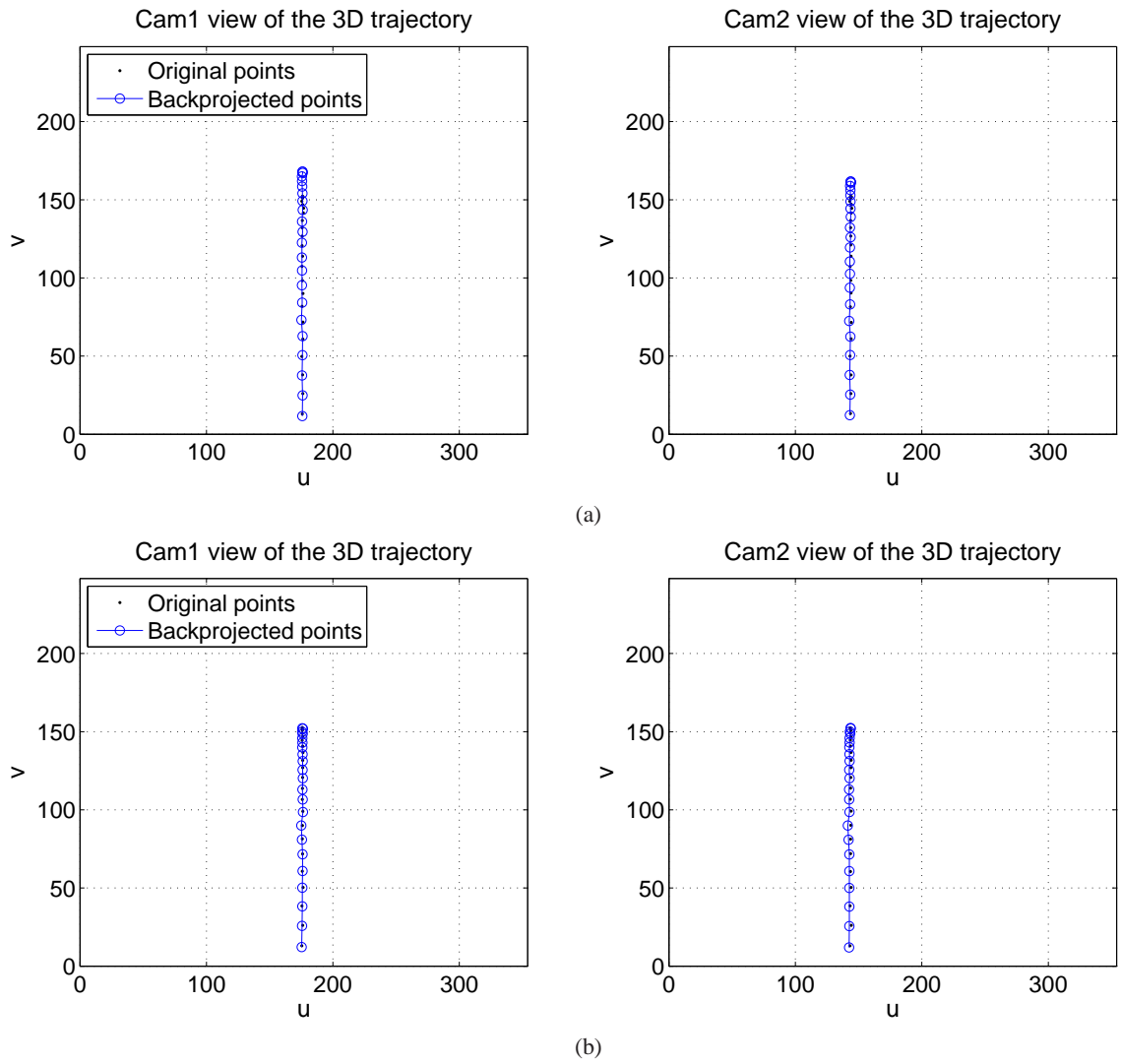
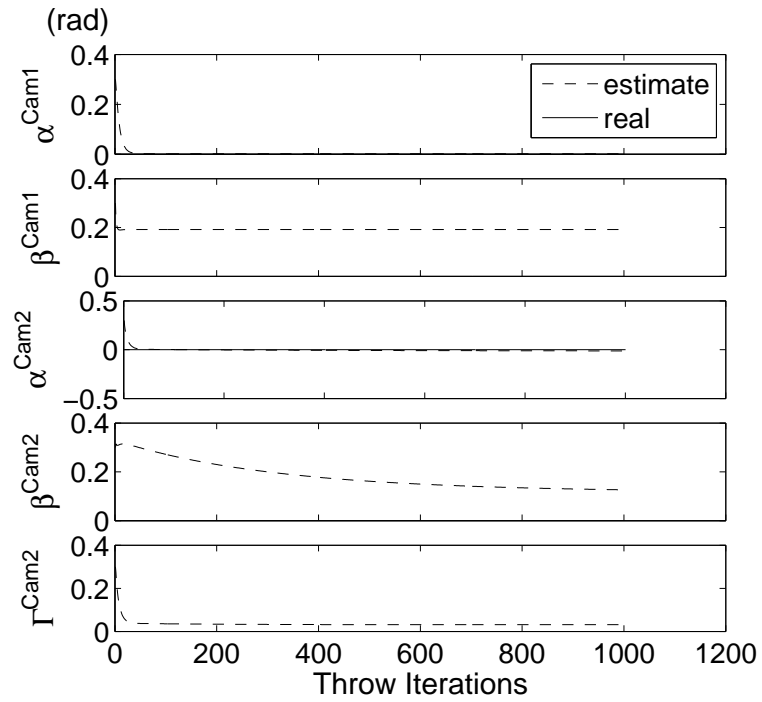


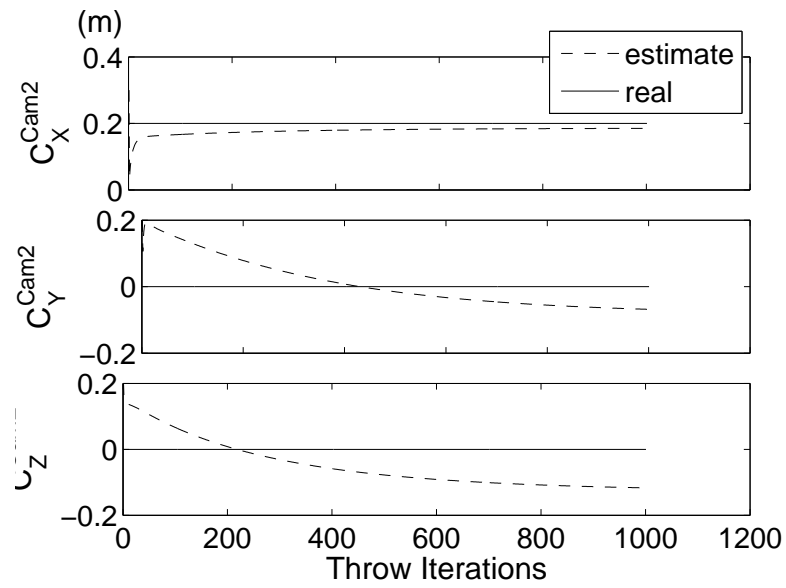
Figure 6.5: The synthetic images of a free falling target captured by a stereo camera system. The motion and observation noise variances are taken as 0.002 m and 0.1 pixels. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by (a) Repeated EKF; (b) Repeated Bidirectional EKF.

Table 6.1: The final re-projection, angle and center errors for R-EKF and RB-EKF

	$Cam1_{Reproj\_Error}$	$Cam2_{Reproj\_Error}$	Angle Error	Center Error
R-EKF	5.8282	3.5549	0.2320	0.1363
RB-EKF	0.4822	0.7799	0.1220	0.0350

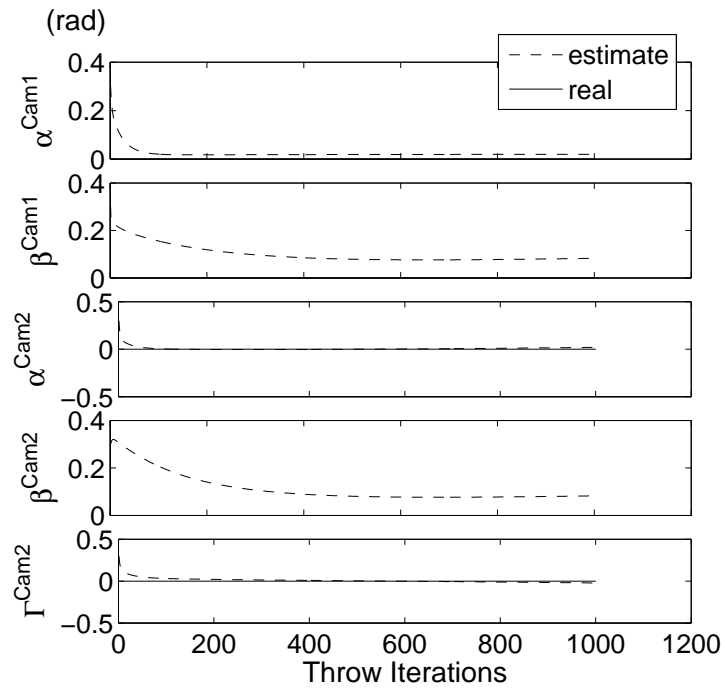


(a)

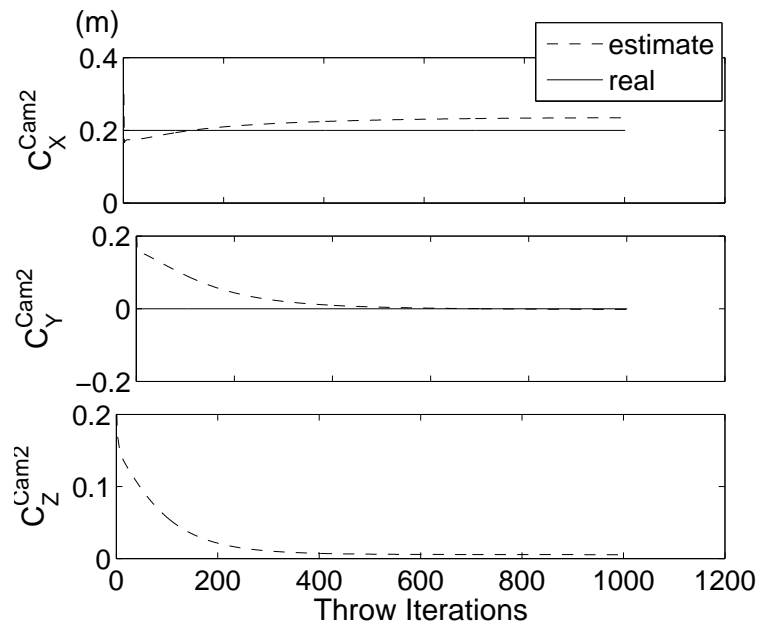


(b)

Figure 6.6: Convergence of the Repeated EKF Algorithm w.r.t *throw* iterations. Since the initial target state is not truly provided the calibration parameters are not accurately obtained (a) Rotation angles; (b) Translation parameters.



(a)



(b)

Figure 6.7: Convergence of the Repeated Bidirectional EKF Algorithm w.r.t *throw* iterations. Even though the initial target state is not truly provided the calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters.

## 6.4 Experiment 4: RB-EKF / Ballistic Motion

Until now we have focused on a free falling target scenario for calibration since the initial velocity uncertainty is minimized in this manner. However such one dimensional motion of the target may cause singularities in the estimation process. For example different orientations of the cameras may yield the same calibration frames. Assume two cameras are placed on a circle which is parallel to the ground and let principle axes of the cameras be congruent with the diameter of the circle. When the target is dropped towards the center of this circle all the cameras on the circle yields the same image sequence. So there may be more than one possible orientation for the cameras when a free falling target is used for camera pose estimation.

Another drawback of the free fall case is that only a small part of the camera image planes are covered by the vertical motion of the target. But the image planes should be covered as much as possible by the calibration object in order to estimate the camera characteristics better. To overcome these problems we move into a more information gathering motion now that RB-EKF can handle arbitrary initial conditions. So instead of dropping the calibration object, we throw it into the joint camera FOV with an initial velocity, hence the object follows a 2D ballistic trajectory. Therefore not only the singularities will be prevented but also greater regions on the image planes will be covered.

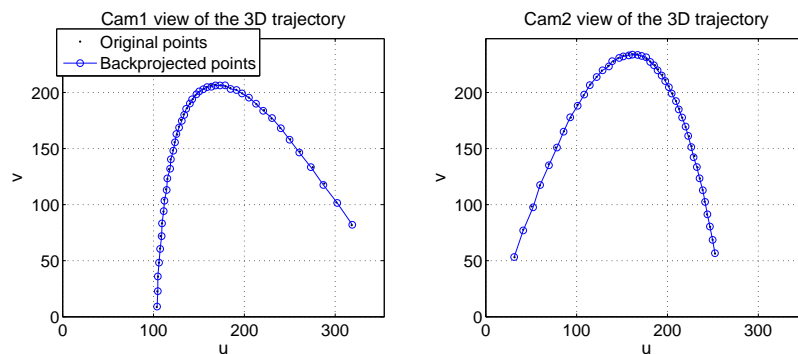


Figure 6.8: The synthetic images of the ballistic trajectory of a thrown target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm.

Since the initial velocity of the target can not be measured easily, it should be estimated

together with the calibration parameters. RB-EKF can estimate the initial target state and as shown in section 6.3, it yields more accurate results than R-EKF when the initial target state is not accurately known. So for the *ballistic throw* cases only the results obtained by RB-EKF are presented.

Suppose that two cameras are placed at the diagonal corners of a square whose edge length is equal to 1.5 m as shown in Figure 6.10. Also assume that Cam2 is placed 0.2 m above Cam1 level. Let the orientations of the cameras be arranged to make their principal axes nearly perpendicular to each other. Then assume that the operator throws a target into the joint camera FOV. An example initial target state for this setup and a calibration parameter vector can be given as

$$\mathbf{X}_0 = [0.5 \ 0 \ 1]^T, \quad \mathbf{V}_0 = [-1.2 \ 4 \ 1.2]^T,$$

$$\mathbf{C} = [-0.2 \ -0.2 \ -\pi/2 \ \pi/2 \ \pi/2 \ 1.5 \ 0.2 \ 1.5]^T.$$

and reasonable initialization vectors may be selected as

$$\hat{\mathbf{X}}_0 = [0.7 \ 0.2 \ 0.7]^T, \quad \hat{\mathbf{V}}_0 = [-1 \ 3 \ 1]^T,$$

$$\hat{\mathbf{C}}_0 = [0 \ 0 \ -1.5 \ 1.5 \ 1.5 \ 1 \ 0 \ 1]^T.$$

We disturb the target position and the camera measurements by White Gaussian noises with variances set to 0.002 m and 0.1 pixels. Starting from the super-state,  $\mathbf{s}_0 = [\hat{\mathbf{X}}_0^T \ \hat{\mathbf{V}}_0^T \ \hat{\mathbf{C}}_0^T]^T$ , after 1000 throw iterations RB-EKF yields the initial target state and the camera parameters as

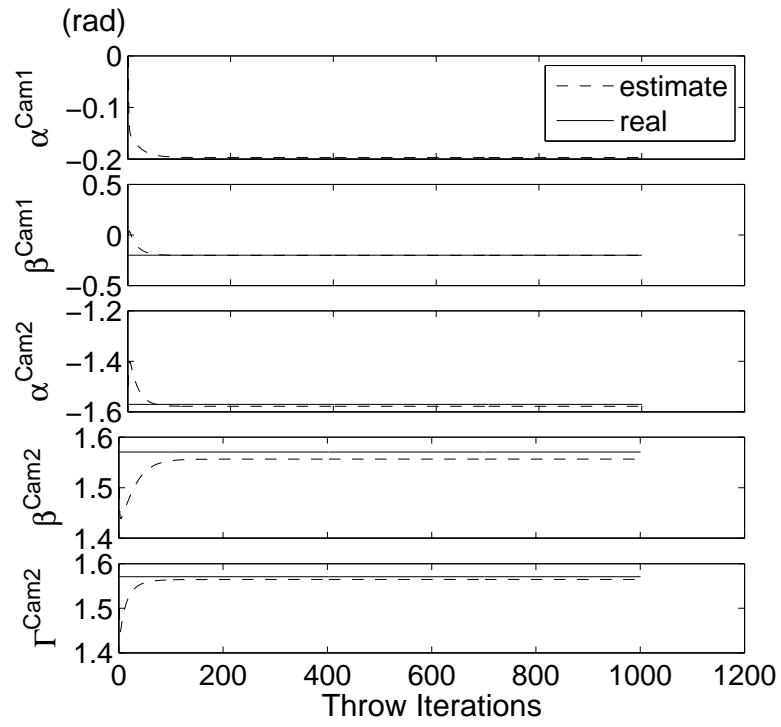
$$\hat{\mathbf{X}}_0 = [0.4960 \ 0.0002 \ 0.9931]^T, \quad \hat{\mathbf{V}}_0 = [-1.1917 \ 4.0040 \ 1.2121],$$

$$\hat{\mathbf{C}} = [-0.1972 \ -0.2024 \ 1.5784 \ 1.5566 \ 1.5645 \ 1.4982 \ 0.1913 \ 1.4750].$$

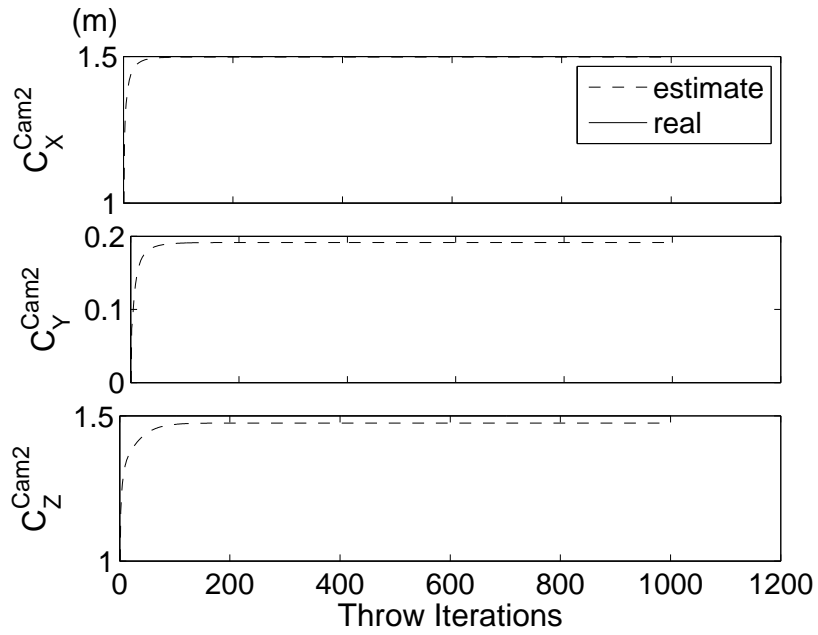
All the re-projected points are close to their original measured locations as shown in Figure 6.8. Also note that the convergence is actually achieved with less than 200 throw iterations as illustrated in Figure 6.9.

The re-projection error is computed as 0.5177 pixels for Cam1 and 0.4998 pixels for Cam2. The rotation and translation errors are obtained as 0.0177 radians and 0.0265 meters respectively.





(a)



(b)

Figure 6.9: Convergence of the Repeated EKF Algorithm w.r.t *throw* iterations. All calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters.

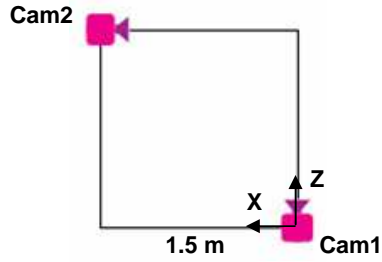


Figure 6.10: The cameras are placed perpendicular to each other and at the diagonal corners of a square with 1.5 meter edge length.

In this example we have demonstrated that RB-EKF can give successful results for both the camera pose estimates and the target trajectory even when the initial target state is not measurable. So, a ballistic motion can be used for calibration by RB-EKF method. Hence the calibration results will be improved since the singularities in the free fall case are prevented and the image plane is better covered by the target.

## 6.5 Experiment 5: RB-EKF / Multiple Trajectories

For pose estimation of a two camera setup we have defined 8 parameters to be estimated. With the additional 6 variables corresponding to the target state, a total of 14 parameters should be predicted. By using a single target trajectory however, only a small number of measurements can be obtained. Although repeated evaluation of EKF provides convergence with a limited number of measurements, more calibration data would probably improve the estimations.

Another drawback of a *single throw* scenario is that, the EKF may get stuck with a local minima, and hence the parameters may not be precisely predicted. To avoid such cases, multiple trajectories can be used instead of a single one. In other words the calibration object can be thrown a few times and various trajectories of the target can be captured. This surely provides additional calibration data that may improve the estimates. In this experiment the effect of using *multiple trajectories* during the calibration procedure is investigated.

Suppose that for the same camera orientation and target motion described in 6.4, the calibration object is thrown for a second time with the initial conditions

$$\mathbf{X}_0 = [-0.5 \ 0.25 \ 1.3]^T, \quad \mathbf{V}_0 = [1.2 \ 3.5 \ 1.2]^T.$$

Table 6.2: The final re-projection, angle and center errors for single and double throw cases

	Cam1 <sub>Reproj.Error</sub>	Cam2 <sub>Reproj.Error</sub>	Angle Error	Center Error
First Throw	0.5177	0.4998	0.0177	0.0265
Second Throw	0.5757	0.5296	0.0150	0.0350
Both Throws	0.5006	0.6152	0.0054	0.0122

Let their initial estimates be taken as

$$\hat{\mathbf{X}}_0 = [0 \ 0 \ 1]^T, \quad \hat{\mathbf{V}}_0 = [1 \ 2 \ 1]^T.$$

We disturb the target position and the camera measurements by White Gaussian noises with variances set to 0.002 m and 0.1 pixels. When the calibration is performed by using the second trajectory only, after 1000 throw iterations RB-EKF yields

$$\hat{\mathbf{X}}_0 = [-0.5002 \ 0.2690 \ 1.3062]^T, \quad \hat{\mathbf{V}}_0 = [1.2140 \ 3.5089 \ 1.2027],$$

$$\hat{\mathbf{C}} = [-0.2044 \ -0.2116 \ 1.5650 \ 1.5647 \ 1.5707 \ 1.5171 \ 0.2302 \ 1.4952].$$

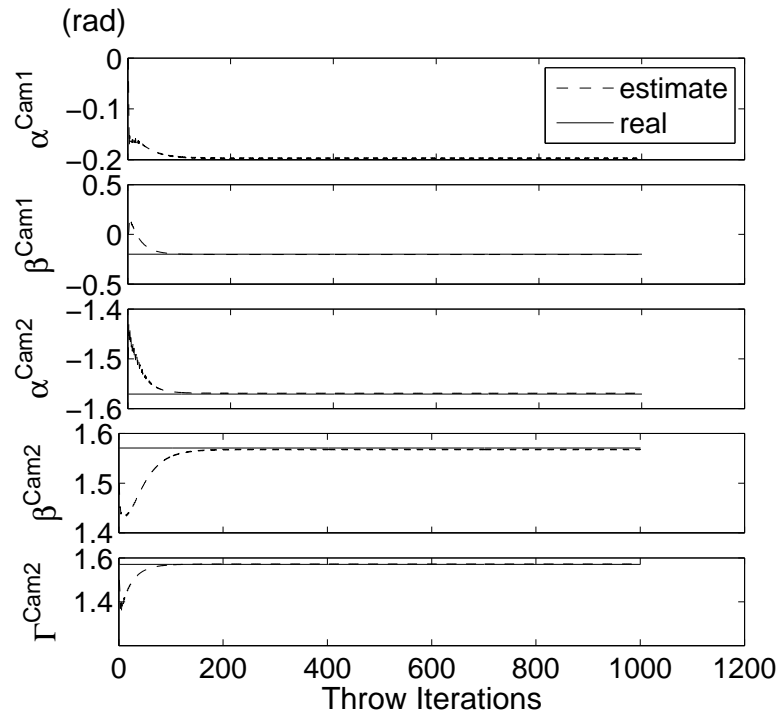
The re-projection error is computed as 0.5757 pixels for Cam1 and 0.5296 pixels for Cam2. The rotation and translation errors are obtained as 0.0150 radians and 0.0350 meters respectively.

When two of the trajectories are successively used to estimate the camera poses, after 1000 throw iterations RB-EKF yields

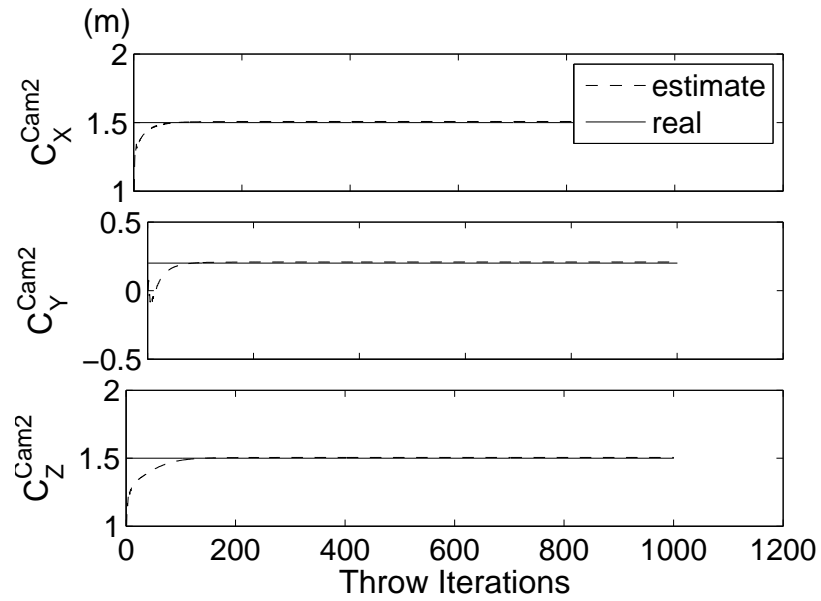
$$\hat{\mathbf{C}} = [-0.1972 \ -0.2029 \ 1.5692 \ 1.5679 \ 1.5720 \ 1.5075 \ 0.2080 \ 1.5053].$$

The re-projection error is computed as 0.5006 pixels for Cam1 and 0.6152 pixels for Cam2. The rotation and translation errors are obtained as 0.0054 radians and 0.0122 meters respectively. The convergence plots are given in Figure 6.11.

The errors corresponding to the single and double throw scenarios are compared in Table 6.2. When the pixel errors are observed, using both trajectories seem not to have any significant advancement compared to using either of the first or the second trajectory alone. However, although pixel error is useful to evaluate the real life performance, it is computed on the image plane only. In the simulation we have the actual values of the calibration parameters and hence the real rotation and translation errors. When they are compared the performance



(a)



(b)

Figure 6.11: Convergence of the Repeated EKF Algorithm w.r.t *throw* iterations when two different target trajectories are used for calibration. All calibration parameters converge to their true values. (a) Rotation angles; (b) Translation parameters.

of the multiple throw case turns out to be significantly better. Therefore the calibration data can be increased by throwing the target several times, and even when the pixel error, which is the fundamental performance measure for a real life setup, does not improve, the estimated parameters will probably be closer to their actual values. So, we are especially encouraged to try calibration by multiple throws to increase the accuracy.

## 6.6 Discussion on Simulation Experiments

In this chapter simulation experiments are performed to understand the workings and evaluate the convergence performances of the proposed methods. Experiments are realized to compare the performances of the R-EKF and RB-EKF methods. The effects of using a free falling target or a ballistically moving target and the effects of using a single or multiple trajectories for calibration are also investigated.

In the early experiments a free falling target trajectory is used since the initial velocity of such a motion is a well defined 0 m/s vector. R-EKF has the best performance when the initial target state is exactly provided for EKF initialization. However such an exact knowledge of the initial state is not easy to obtain in real life. Further experiments show that unlike R-EKF, RB-EKF can tolerate significant initial state uncertainties. Hence RB-EKF is a much more suitable method in practical applications, especially for the field applications where the ease of the calibration method is desired most, but the uncertainty is unavoidable.

A free fall motion reduces the unknowns of the initial target state and opens the way to use R-EKF, however such a motion is not sufficiently informative since only a small part of the camera image planes are covered. Additionally different poses of the cameras may yield the same calibration frames for a free falling target. This means that there may be more than one possible solution for the same set of calibration frames in a free fall scenario, and the filter may converge to one of these solutions according to its initialization. In order to prevent such situations, using a ballistic trajectory is suggested. Although that would increase the initial state uncertainty and R-EKF can not be utilized anymore, RB-EKF can still be reliably employed as shown in section 6.4.

The accuracy also depends on the sufficiency of the available calibration data. Increasing the calibration data would probably result in better pose estimates. One way of increasing the

number of calibration points is to use more than one target trajectory. Though each additional trajectory brings 6 additional unknowns corresponding to the initial target state more accurate parameter estimates are obtained as shown in section 6.5. Although the average reprojection pixel errors stay at the same level as in a single trajectory case, the parameter estimates are much closer to their actual values when multiple target trajectories are used.

To sum up, the simulation experiments demonstrate that for practical applications the RB-EKF method is superior to R-EKF method. Furthermore using a ballistic motion and multiple trajectories are superior to using a free fall motion and a single trajectory. Therefore the camera parameters can be estimated best by using RB-EKF method together with multiple ballistic trajectories.

## CHAPTER 7

### Physical Experiments

Simulation experiments help us to understand the workings of the presented algorithms. However, although in simulations something works excellent, in real life we have to deal with several unexpected situations and difficulties. These difficulties and real life performances of our methods are discussed in this chapter.

First, the components of the multi camera setup that is used throughout the experiments are described in section 7.1. Next, the internal parameters of the cameras are obtained by MATLAB calibration toolbox in section 7.2. Then the physical experiments to test the proposed R-EKF and RB-EKF methods are presented. A free falling target is used for calibration with R-EKF and RB-EKF algorithms in sections 7.3 and 7.4. Then RB-EKF is used with a single ballistic target trajectory to obtain the camera poses for parallel, perpendicular and arbitrary placement of the cameras in sections 7.5 and 7.6 and 7.7 respectively. Section 7.8 presents the RB-EKF performance when multiple target trajectories are used for pose estimation of arbitrarily placed cameras and finally a general discussion of the physical experiments is given in section 7.9.

#### 7.1 The Multi Camera Setup Components

In this part the multi camera setup, which is prepared for evaluating the performance of our algorithm, is described. The utilized video capture card and the analog cameras are presented in sections 7.1.1 and 7.1.2 respectively. The connection cables tripods and clamps are displayed in section 7.1.3.

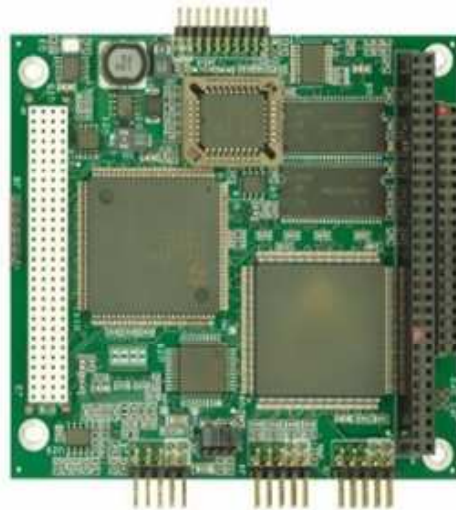


Figure 7.1: Eurotech CTR-1472 frame grabber and MPEG-4 Compressor

### 7.1.1 CTR-1472 MPEG-4 Compressor

The CTR-1472 is a high performance four channel MPEG-4 Video Compressor that supports real-time video encoding. The CTR-1472 can be used to capture analog video streams (from PAL/NTSC cameras, VCR and other video sources) up to four sources at the same time. CTR-1472 can encode frames in MPEG-4 format and send them to a host via PCI BUS. This PC/104-Plus compliant CTR-1472 MPEG-4 compressor, which is developed by Eurotech Company is shown in Figure 7.1.

### 7.1.2 Measurement Cameras

The UWC-300 camera, the product of Outland Technologies Inc., is an underwater camera designed for deep sea operation. It is illustrated in Figure 7.2. UWC-300 can serve as a general purpose camera for our experiments since its image quality in the air is also acceptable.

The camera sensor is sensitive to low light conditions (down to 0.0003 lux) and the sensor provides a 600 horizontal lines of resolution. The UWC-300 camera comes with a 3.6 mm focal length integrated lens. It gives analog video output and therefore necessitate a video frame grabber. Due to the large amount of data from multiple cameras, the captured board requires a video compressor. The cameras are therefore used together with CTR-1472 MPEG-4 compressor.

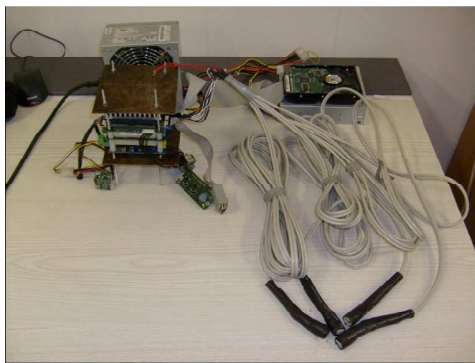




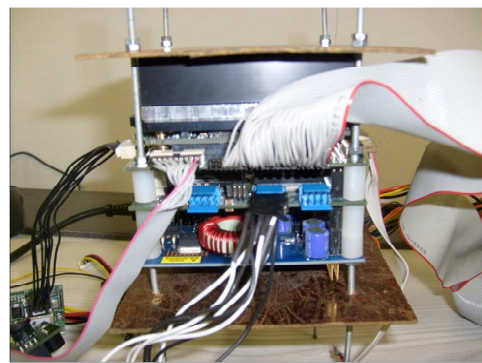
Figure 7.2: Outland Tech. UWC-300 underwater cameras (black cylinder, top)

### 7.1.3 Connection Cables, Tripods and Clips

UWC-300 and other underwater cameras have their standardized underwater connectors which are not commonly available. To work in the laboratory, connectors are manufactured by us in the laboratory environment. After some trials, these connectors are produced to minimize noise and provide data with high quality. The PC104 stack which is set in the laboratory and the attached connectors are shown in Figure 7.3(a). The connection cables are coaxial CCTV cables which also include a number of control signal lines. Another connector which was needed in the laboratory is the one that connects the video cables to the CTR-1472 MPEG-4 compressor as shown in Figure 7.3(b).



(a)



(b)

Figure 7.3: The connection cables and connectors. (a) PC104 stack which includes CTR-1472 MPEG-4 compressor and the manufactured cable set connecting the cameras to the stack. (b) CTR-1472 connector.



Figure 7.4: Tripod-clamp-camera trio and the calibration pattern. (Shown for a single camera)

To hold the cameras in a secure manner during the internal and external calibration suitable set of tripods (SLIK-ABLE 300DX) and clamps(MONFROTTO SUPER CLAMP 035) are acquired. This setup is shown in Figure 7.4 with a single camera and Figure 7.5 shows the synchronous video capture from the set of four cameras and the associated preliminary software interface.



Figure 7.5: 4 pieces of BWC-300 underwater cameras and synchronous video capturing with these cameras.

## 7.2 Internal Calibration

After all the connections are made in order to capture video synchronously from 4 cameras as a first step, the software which was provided by Eurotech is updated and used so that the video frames could be captured, compressed and saved. When the camera hardware became ready, the captured frames could be compressed and saved in the MPEG-4 format with the help of the software provided by Eurotech, the manufacturer of the frame grabber/compressor card. When MPEG-4 compressor captures in PAL format with 100 fps rate the frames are obtained in  $354 \times 248$  pixel resolution and if the frame rate is 25 fps then the resolution is doubled.

While the work on video capturing was in progress, we contacted the Eurotech Company and obtained the source code of the video capturing software. The source code is valuable as it provides a large number of examples on how the drivers of the card can be configured and used. This allows us to incorporate the functionality into our own software and modify and extend what is provided by the standard test interface of the manufacturer. As an example, in order to perform the internal calibration of the cameras, a single frame of a static scene is needed instead of a long video sequence. This problem is solved by adding a *SaveOne-Frame* function to the source code so that single frame capturing and saving in JPEG format is achieved. This allowed the capture of the necessary images which are used to obtain the internal calibration parameters of the cameras with the help of the MATLAB Calibration Toolbox [23] since the calibration software uses the photographs (or a single video frame) of a chess pattern Figure 7.6 from different angles.

Although MATLAB Calibration Toolbox inspired the main initialization step from Zhang's calibration algorithm [19], the closed form estimation of the internal parameters is slightly different. For example the distortion parameters are not estimated at the initialization step, and intrinsic camera model is inspired from Heikkilä and Silvén's work [21] which includes two extra distortion coefficients corresponding tangential distortion. Radial and tangential distortion model is taken directly from the Brown's work [56] which is one of the first introductions of the camera model.

In the simulations two cameras have been used for testing our external calibration algorithms. Although the internal parameters were perfectly known in the simulations, for the physical

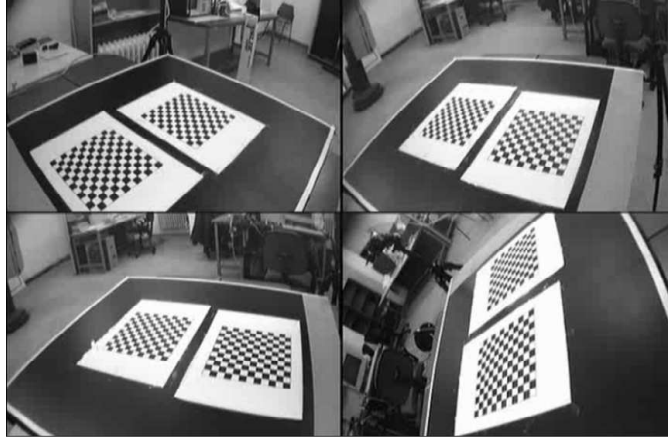


Figure 7.6: The calibration view obtained from 4 cameras synchronously. This kind of chess-board pattern shown in the figure is used for internal calibration purpose.

experiments, first of all, the internal parameters of the cameras should be obtained. We have performed the internal calibration by MATLAB calibration toolbox in order to calibrate two of the UWC-300 cameras. For the first camera focal length, principal point, skew, and distortion coefficients are obtained as

$$\begin{aligned}
 \text{Focal length :} \quad \mathbf{fc} &= [277.44915 \ 298.82608] \pm [1.19135 \ 1.27202] \\
 \text{Principal point :} \quad \mathbf{cc} &= [183.64332 \ 117.42620] \pm [1.59237 \ 1.64054] \\
 \text{Skew :} \quad \mathbf{alpha\_c} &= [-0.00084] \pm [0.00093] \\
 \text{Distortion :} \quad \mathbf{kc} &= [-0.42093 \ 0.20117 \ -0.00270] \\
 &\pm [0.01084 \ 0.02578 \ 0.00095] \\
 \text{Pixel error :} \quad \mathbf{err} &= [0.28266 \ 0.26950]
 \end{aligned}$$

and for the second camera the same parameters are obtained as

$$\begin{aligned}
 \text{Focal length :} \quad \mathbf{fc} &= [270.39002 \ 291.26888] \pm [1.42153 \ 1.55879] \\
 \text{Principal point :} \quad \mathbf{cc} &= [180.25090 \ 118.13432] \pm [1.61322 \ 1.68951] \\
 \text{Skew :} \quad \mathbf{alpha\_c} &= [-0.00026] \pm [0.00099] \\
 \text{Distortion :} \quad \mathbf{kc} &= [-0.39303 \ 0.14915 \ -0.00353] \\
 &\pm [0.01171 \ 0.02362 \ 0.00099] \\
 \text{Pixel error :} \quad \mathbf{err} &= [0.30324 \ 0.30778]
 \end{aligned}$$

In a real image formation process the lens distortion may cause great fluctuations from the pinhole camera model. Apart from the focal length, principal point and skew parameters, MATLAB calibration toolbox yields the distortion coefficients too. Luckily an additional

utility to undistort the images is also included in the toolbox. We integrated that function in our external calibration method, so that undistorted images of a calibration target can be acquired and directly used by our calibration algorithms. Otherwise the distorted images would be used in the expense of low calibration accuracy. In fact we previously tried to include the distortion model in the EKF equations to be able to use the original distorted images acquired from the cameras, but the evaluation time for the Jacobian matrix was increased too much due to the higher complexity.

Finally note that the internal calibration error is in the order of 0.3 pixel for both cameras. In the simulations no such error corresponding to the internal calibration results was taken into account. Anyway this additional disturbance can be accepted as part of the observation noise in the EKF equations.

### 7.3 Experiment 1: R-EKF / Free Fall Motion

In this experiment the simulation experiment in section 6.1 is repeated with real data. Two cameras are tried to be placed in parallel with zero rotations. The main difference from the simulation experiment is the distance between the cameras towards  $x$ -axis. Instead of a 0.2 m gap the cameras are separated by half a meter distance. A free falling target is used to calibrate the two camera system with R-EKF algorithm. The initial target state is measured as

$$\hat{\mathbf{X}}_0 = [0.25 \ 0.35 \ 1.10]^T, \quad \hat{\mathbf{V}}_0 = [0 \ 0 \ 0]^T,$$

and the camera parameter vector is initially set as

$$\hat{\mathbf{C}}_0 = [0 \ 0 \ 0 \ 0 \ 0 \ 0.5 \ 0 \ 0]^T.$$

After 1000 throw iterations, R-EKF algorithm yields the camera parameters as

$$\hat{\mathbf{C}} = [-0.0191 \ 0.0160 \ 0.0106 \ 0.0145 \ 0.0083 \ 0.4916 \ -0.0054 \ -0.0162]^T.$$

As discussed in section 6.1 the target position is measured by a ruler and the initial velocity is taken as  $\vec{0}$  m/s since the calibration object falls freely. However, the acquired initial position value is not exact due to uncontrollable measurement errors. The initial velocity is not also absolutely equal to  $\vec{0}$  m/s, because the moment that the target is dropped can not be synchronized with the camera capturing instant. Furthermore while releasing the calibration object,

Table 7.1: The final parameter estimates for different initial positions of the calibration object and the corresponding re-projection pixel errors after 1000 throw iterations with R-EKF.

	$\alpha^{c1}$	$\beta^{c1}$	$\alpha^{c2}$	$\beta^{c2}$	$\Gamma^{c2}$	$C_x^{c2}$	$C_y^{c2}$	$C_z^{c2}$	$err^{c1}$	$err^{c2}$
$ex^1$	-0.02	0.02	0.01	0.01	0.01	0.50	0.00	-0.02	5.16	4.97
$ex^2$	-0.01	-0.05	0.02	0.01	0.02	0.48	0.08	0.00	7.92	7.73
$ex^3$	-0.01	0.07	0.03	0.08	0.18	0.53	0.04	-0.07	8.12	7.02
$ex^4$	-0.03	0.08	0.02	0.05	0.01	0.51	-0.05	-0.08	8.76	7.14
$ex^5$	0.02	-0.05	0.00	0.05	0.00	0.49	0.13	0.04	7.60	5.38

undesired movements in  $x$  and  $z$  directions may also occur. Still taking the initial velocity of a free falling target as  $\vec{0}$  m/s is a reasonable assumption.

The final re-projection pixel errors for Cam1 and Cam2 are 5.1617 and 4.9735 pixels respectively. Although our algorithm is not an optimization method that minimizes the pixel error, this re-projection error is a measure that reflects the accuracy of our estimation. Figure 7.7 illustrates the original and re-projected points for both cameras. The convergence graphs for the calibration parameters are shown in Figure 7.8.

In four additional examples the initial position of the calibration object is changed and the same experiment is repeated. In other words the calibration target is freely dropped from different locations but the cameras are not moved. R-EKF results for 1000 throw iterations are presented in Table 7.1. Although all of the parameters are consistent with our expectations the pixel errors are changing from 5 to 9 pixels. This much error is mainly caused by the inaccurate measurement of the initial target state.

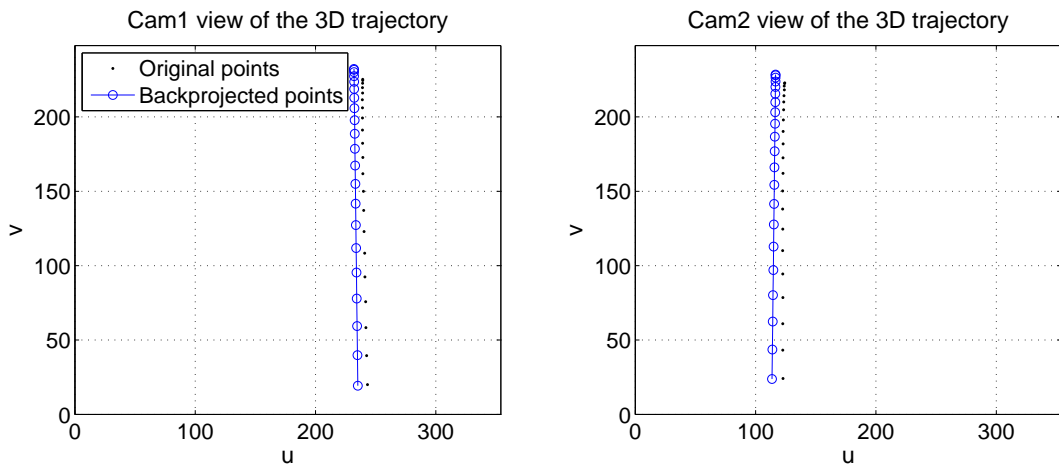
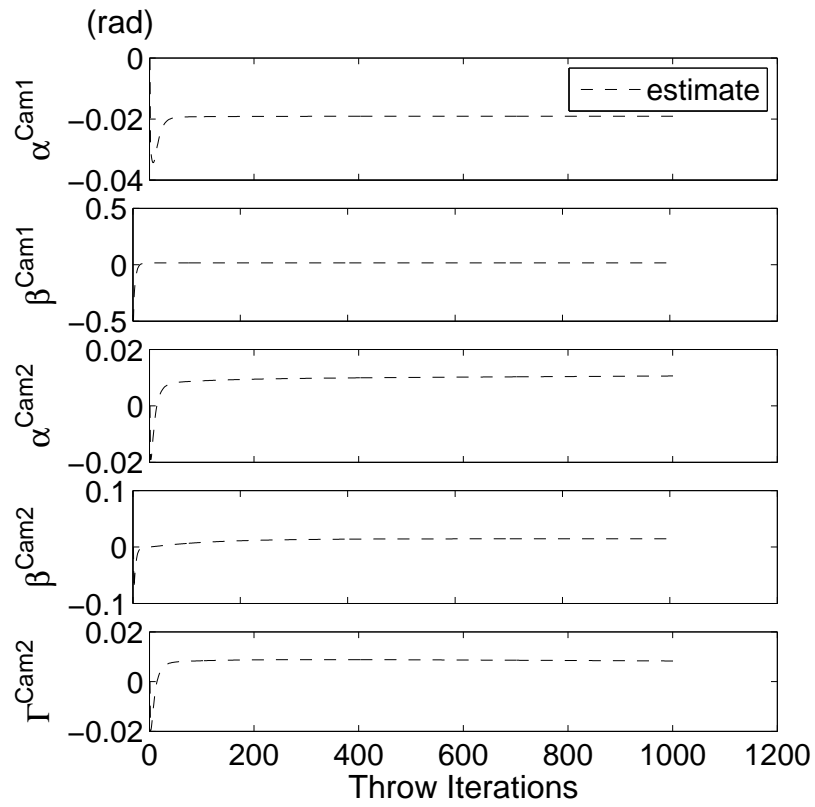
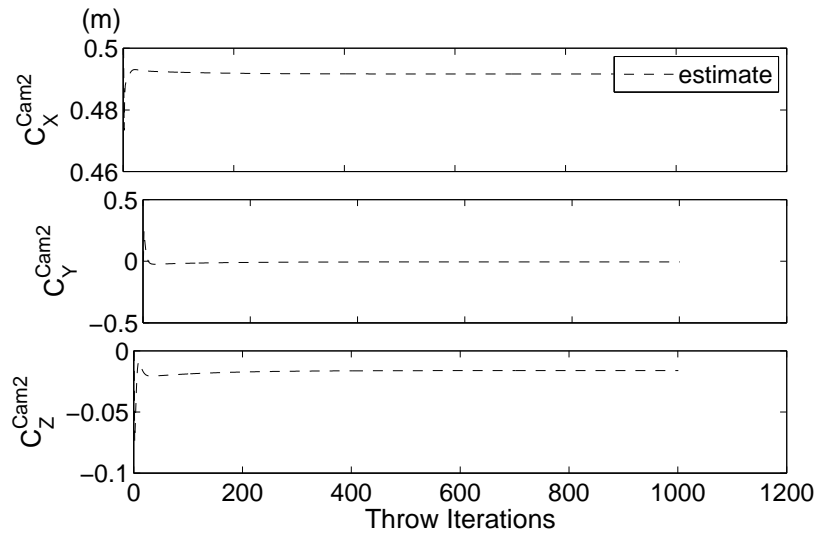


Figure 7.7: The extracted 2-D positions of a free falling target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated EKF algorithm.



(a)



(b)

Figure 7.8: Convergence of the Repeated EKF Algorithm w.r.t *throw* iterations, (a) Rotation angles; (b) Translation parameters.

## 7.4 Experiment 2: RB-EKF / Free Fall Motion

In this section the first experiment is repeated for the RB-EKF algorithm. The initial target state and the initial camera parameter vector are provided to the RB-EKF algorithm as in section 7.3 since the same observation data (i.e. captured videos of the free falling target) is used. At the end of 1000 throw iterations RB-EKF yields

$$\hat{\mathbf{C}} = [-0.0459 \ 0.0265 \ -0.0015 \ 0.0220 \ -0.0105 \ 0.5142 \ -0.0266 \ -0.0071]^T$$

and also the initial target state is also estimated as

$$\hat{\mathbf{X}} = [0.3040 \ 0.3238 \ 1.1714]^T, \quad \hat{\mathbf{V}} = [-0.0446 \ -0.0441 \ -0.0233]^T$$

with only 0.1448 and 0.2693 re-projection pixel errors for Cam1 and Cam2 respectively. The camera parameters are similar to the ones obtained by the R-EKF, however the final re-projection errors are much more smaller. The main reason for such an error reduction with the RB-EKF is that the initial target state is also estimated in addition to the camera parameters. Remember that in the simulations since the initial target state was exactly known and provided, R-EKF had a better performance. Unfortunately in real life such an exact measurement seems not possible to be acquired, so in practice the RB-EKF algorithm works much more efficiently.

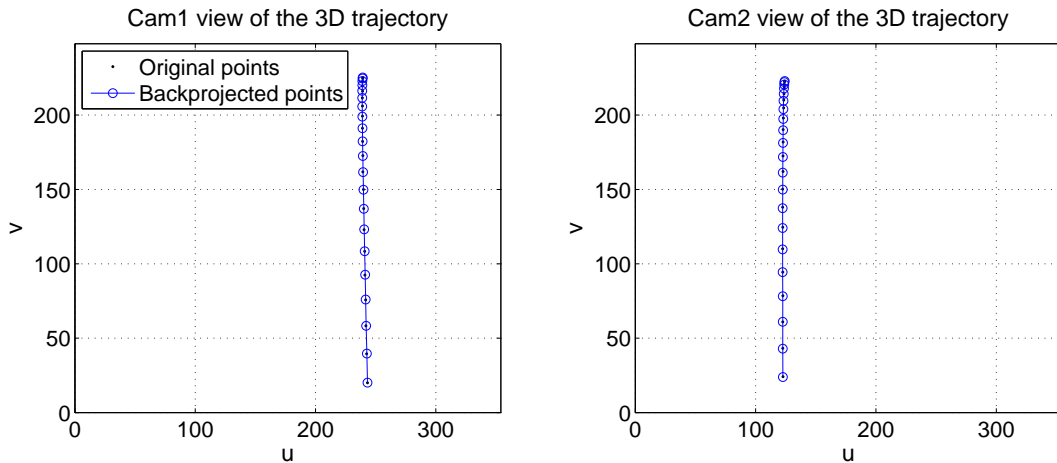
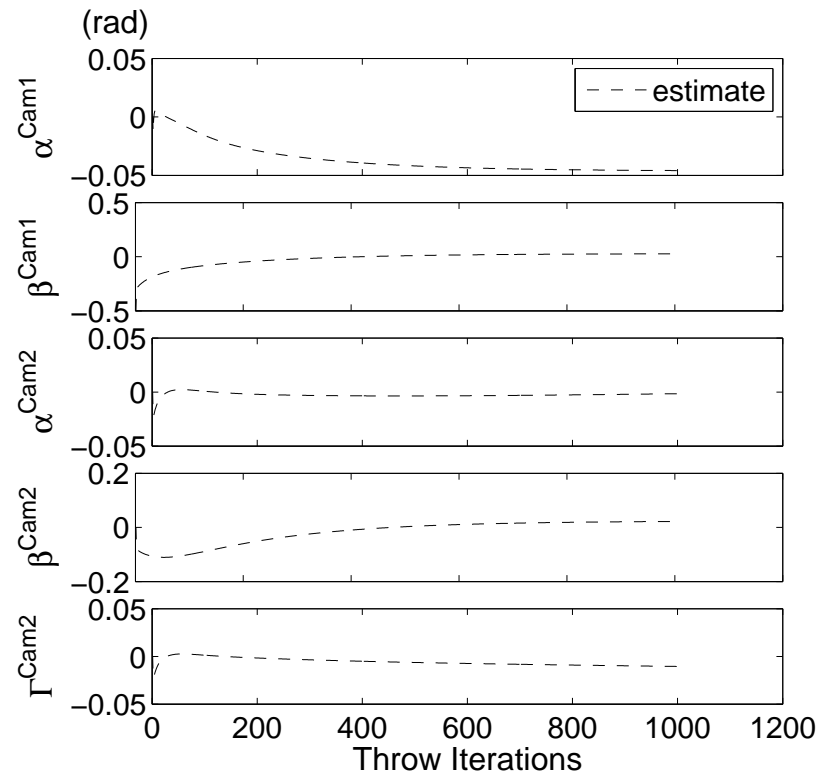
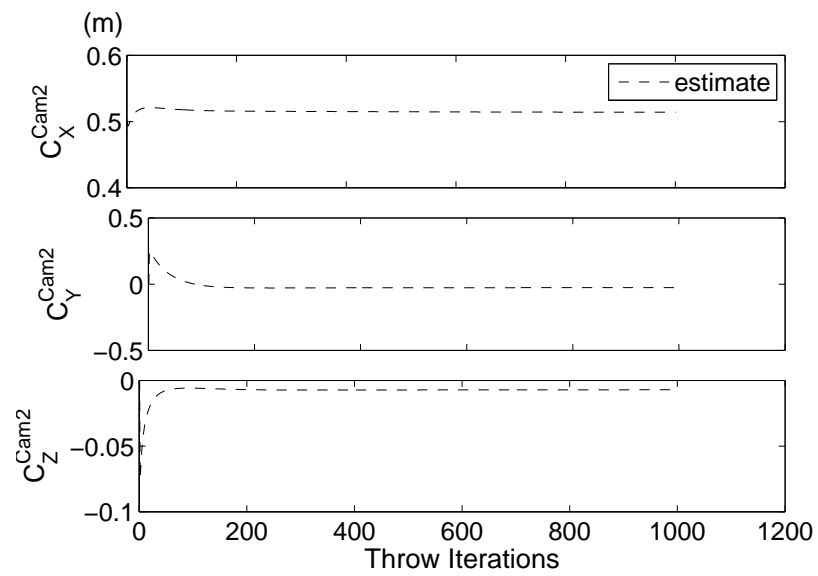


Figure 7.9: The extracted 2-D positions of a free falling target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm.





(a)



(b)

Figure 7.10: Convergence of the Repeated Bidirectional EKF Algorithm w.r.t *throw* iterations, (a) Rotation angles; (b) Translation parameters.

Table 7.2: The final parameter estimates for different initial positions of the calibration object and the corresponding re-projection pixel errors after 1000 throw iterations with RB-EKF.

	$\alpha^{c1}$	$\beta^{c1}$	$\alpha^{c2}$	$\beta^{c2}$	$\Gamma^{c2}$	$C_x^{c2}$	$C_y^{c2}$	$C_z^{c2}$	$err^{c1}$	$err^{c2}$
$ex^1$	-0.05	0.03	0.00	0.02	-0.01	0.51	-0.03	-0.01	0.14	0.27
$ex^2$	-0.02	0.05	0.02	0.07	0.02	0.49	0.03	-0.02	0.15	0.31
$ex^3$	-0.02	0.00	0.03	0.05	-0.01	0.52	0.08	-0.02	0.18	0.46
$ex^4$	-0.03	0.02	0.01	0.04	0.00	0.52	0.00	-0.02	0.21	0.42
$ex^5$	-0.02	-0.02	0.00	-0.01	-0.02	0.53	0.00	-0.02	0.37	0.67

The original and re-projected points are shown in Figure 7.9 and the convergence curves of the RB-EKF algorithm are plot in Figure 7.10. For the additional four examples in which the target is initiated from different places in the joint-FOV of the cameras while the cameras are kept fixed, the RB-EKF results after 1000 throw iterations are summarized in Table 7.2. The re-projection pixel errors are on the order of 0.2 – 0.5 pixels and it should be noted that part of it is caused by the internal calibration error. The calibration parameters are obtained as expected and the parameter estimations in different examples are nearly equal to each other.

### 7.5 Experiment 3: RB-EKF / Ballistic Motion

In this experiment a ballistic trajectory is used for calibrating the two camera system instead of a free fall motion. The cameras are kept fixed at their parallel poses as described in section 7.3. Since the calibration target is thrown into the joint camera FOV, the initial position and velocity of the target are not known approximately as in the previous cases, but they can only be roughly guessed by the operator.

Since the cameras are placed in a parallel position with half a meter distance between them, the calibration parameters are approximately known in our example. However when the cameras are placed in an obscure manner with rotation and translation in all directions, the operator may not be able to provide the initial estimation for the camera poses as accurate as we have done until now. In this experiment we observe the performance of the RB-EKF algorithm when the initial estimates for the target state and camera parameters are not provided precisely.

The initial estimate for the camera orientations and target state is assumed to be provided by the operator by only looking at the camera poses and calibration frames. We have estimated

the initial target state roughly by looking at the calibration frames as

$$\hat{\mathbf{X}}_0 = [-0.25 \ 0 \ 1]^T, \quad \hat{\mathbf{V}}_0 = [1 \ 2 \ 1]^T.$$

It is a reasonable assumption that human eye may predict the rotation and translation parameters with less than 0.5 rad, and 0.5 m errors. So the camera parameters are distorted by that amount and the initial parameter estimate vector is provided as

$$\hat{\mathbf{C}}_0 = [0.5 \ 0.5 \ 0.5 \ 0.5 \ 0.5 \ 0 \ 0.5 \ 0.5]^T.$$

After 1000 throw iterations with RB-EKF method the complete state is obtained as

$$\begin{aligned} \hat{\mathbf{X}} &= [0.0355 \ 0.0733 \ 0.8864]^T, \quad \hat{\mathbf{V}} = [1.1149 \ 1.8869 \ 0.7419]^T, \\ \hat{\mathbf{C}} &= [-0.0091 \ 0.0120 \ -0.1212 \ 0.0503 \ 0.1260 \ 0.5158 \ 0.0408 \ -0.0126]^T \end{aligned}$$

with 0.4009 and 1.3721 re-projection pixel errors for Cam1 and Cam2. Note that all of the camera parameters are obtained as expected. One can argue that  $-0.1212$  and  $0.1260$  values are too high because their expected value is approximately zero. However since  $z$ - $x$ - $z$  rotation is used,  $\alpha^{c2}$  and  $\Gamma^{c2}$  terms both correspond to the rotations around  $z$ -axis. Since  $\beta^{c2}$  is also approximately zero, these  $-0.1212$  and  $0.1260$  terms nearly cancel each other validating the parallelism of the cameras.

The original and re-projected points are shown in Figure 7.11 and the convergence curves of the RB-EKF algorithm are plot in Figure 7.12.

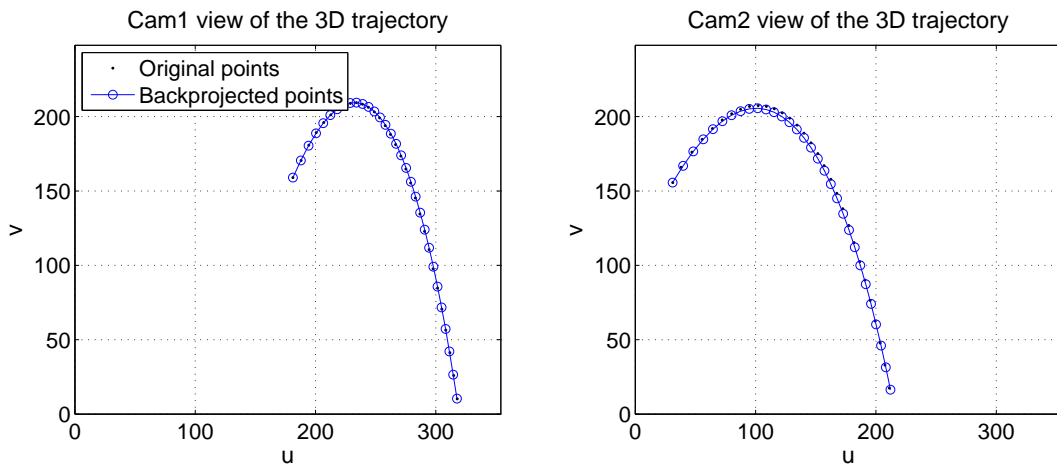
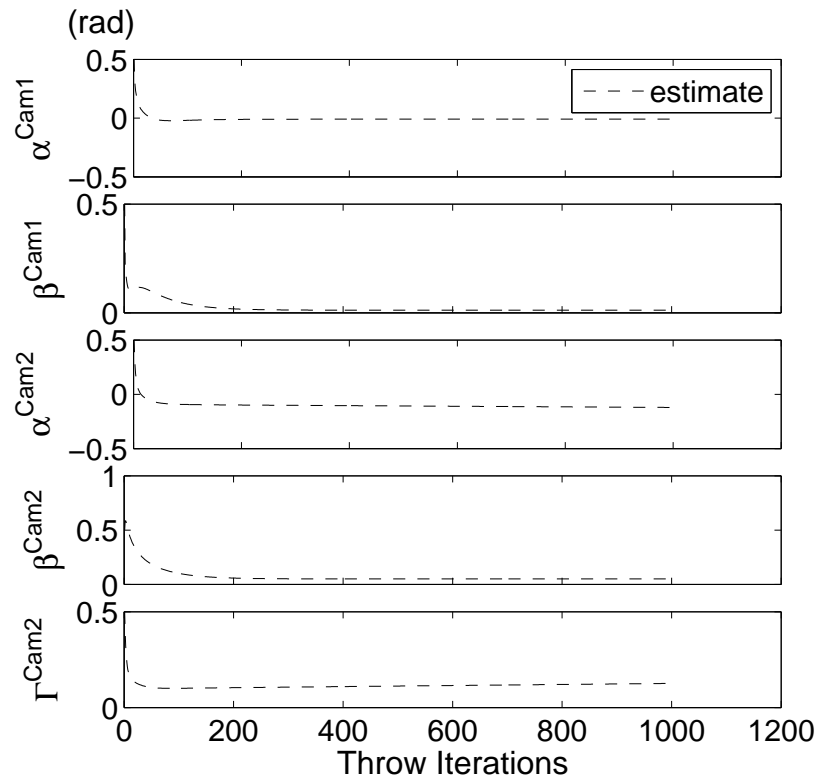
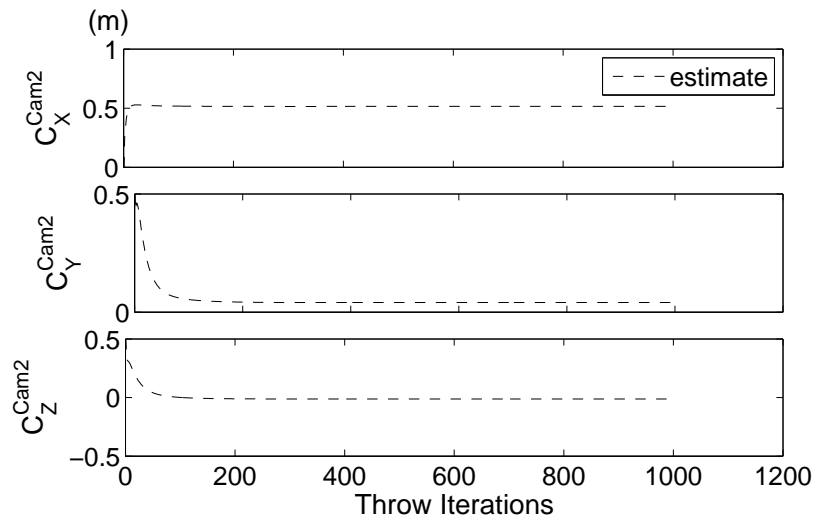


Figure 7.11: The extracted 2-D positions of a ballistically moving target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm.



(a)



(b)

Figure 7.12: Convergence of the Repeated Bidirectional EKF Algorithm w.r.t *throw* iterations, (a) Rotation angles; (b) Translation parameters.

## 7.6 Experiment 4: Perpendicular Camera Placement

Up to now the experiments are performed to calibrate two cameras which are placed in a parallel configuration. Such a placement is not only quite useful for stereo imaging, but it also helps us to verify the trustiness of our algorithm since the actual camera poses are approximately known. However the proposed method is not limited to calibrate the cameras that are placed in parallel orientation. In this part we calibrate a two camera setup when the cameras are placed on the diagonal corners of a square with 1 meter edge length and Cam1 is rotated 90 degrees around y-axis with respect to the basis coordinate frame as shown in Figure 7.13.

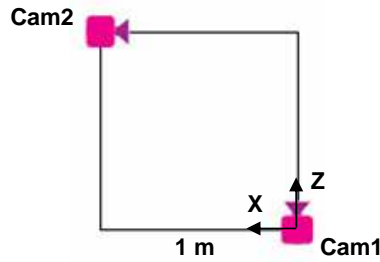


Figure 7.13: The cameras are placed at the diagonal corners of a square with 1 meter edge length.

We have estimated the initial target state roughly as

$$\hat{\mathbf{X}}_0 = [-1 \ 0.5 \ 1]^T, \quad \hat{\mathbf{V}}_0 = [1 \ 2 \ 1]^T.$$

Since  $z$ - $x$ - $z$  rotation is used, Cam2 rotation parameters should initially be set as  $\alpha^{c2} = -\pi/2$ ,  $\beta^{c2} = \pi/2$ ,  $\Gamma^{c2} = \pi/2$ . So the initial parameter estimate vector is provided as

$$\hat{\mathbf{C}}_0 = [0 \ 0 \ -\frac{\pi}{2} \ \frac{\pi}{2} \ \frac{\pi}{2} \ 1 \ 0 \ 1]^T.$$

After 1000 throw iterations with RB-EKF method the complete state is obtained as

$$\begin{aligned} \hat{\mathbf{X}} &= [-0.3618 \ -0.1101 \ 0.8088]^T, \quad \hat{\mathbf{V}} = [1.4191 \ 2.5723 \ 0.5942]^T, \\ \hat{\mathbf{C}} &= [-0.0115 \ -0.0010 \ -1.4995 \ 1.6111 \ 1.5774 \ 1.0100 \ 0.0662 \ 1.0005]^T \end{aligned}$$

with 0.3181 and 0.5495 re-projection pixel errors for Cam1 and Cam2 respectively. The first two elements of the  $\hat{\mathbf{C}}$  vector,  $\alpha^{c2}$  and  $\beta^{c2}$ , are nearly zero since Cam1 is placed approximately parallel to the ground. The orientation of Cam2 is also obtained as expected. Only the 0.0662

value for  $C_y^{c2}$  seems to be a little bit away from its expected value of zero, because the cameras are placed nearly at the same height. Still under the disturbance effects such as air friction, lens

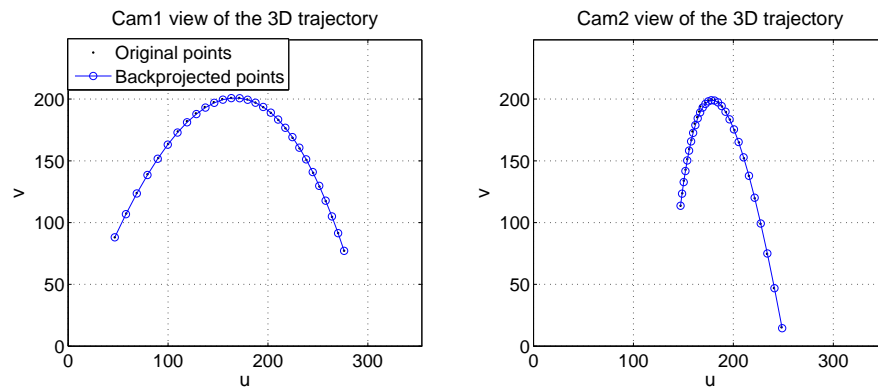
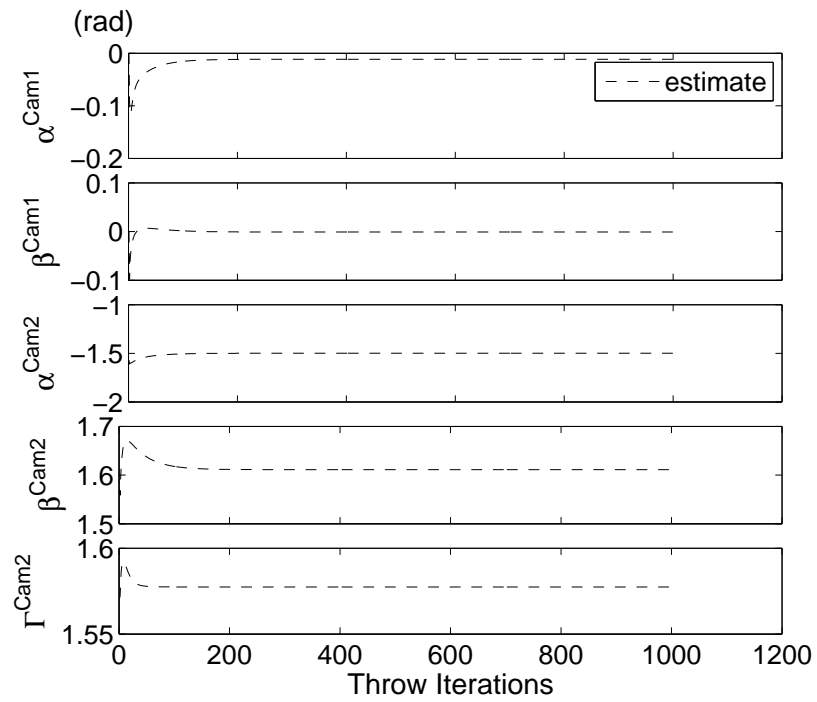


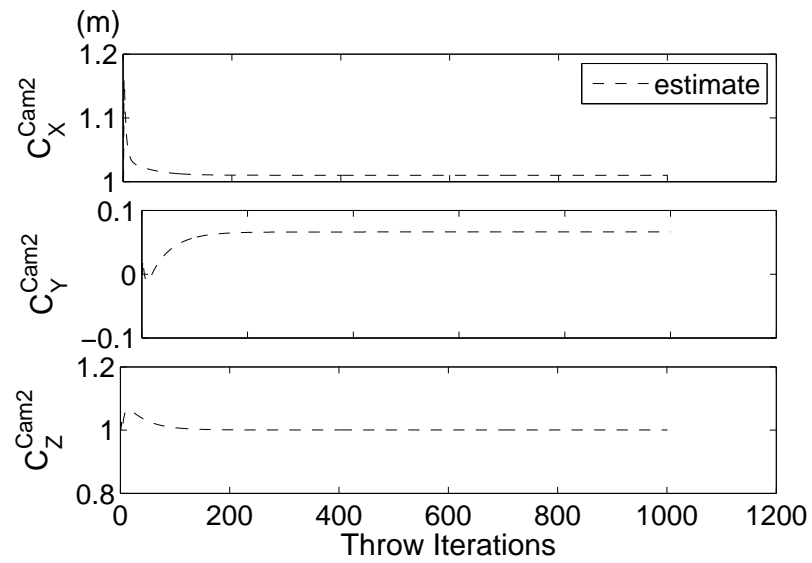
Figure 7.14: The extracted 2-D positions of a ballistically moving target captured by a stereo camera system. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm.

distortion and pixel errors the obtained results are successful especially in the re-projection pixel error sense.

The original and re-projected points are shown in Figure 7.14 and the convergence curves of the RB-EKF algorithm are plot in Figure 7.15.



(a)



(b)

Figure 7.15: Convergence of the Repeated Bidirectional EKF Algorithm w.r.t *throw* iterations, (a) Rotation angles; (b) Translation parameters.

## 7.7 Experiment 5: Arbitrary Camera Placement

In many practical applications the cameras may be placed arbitrarily and it may be quite difficult to provide the initializing parameters of the EKF accurately. Even though the initial state is not known it should be predicted by the operator to prevent the RB-EKF to converge to a local minimum solution which may be a wrong result. If the initial estimates for the target state and camera parameters are given in a wrong manner, the filter may even diverge. Therefore at least a rough estimate of the super-state should be provided initially.

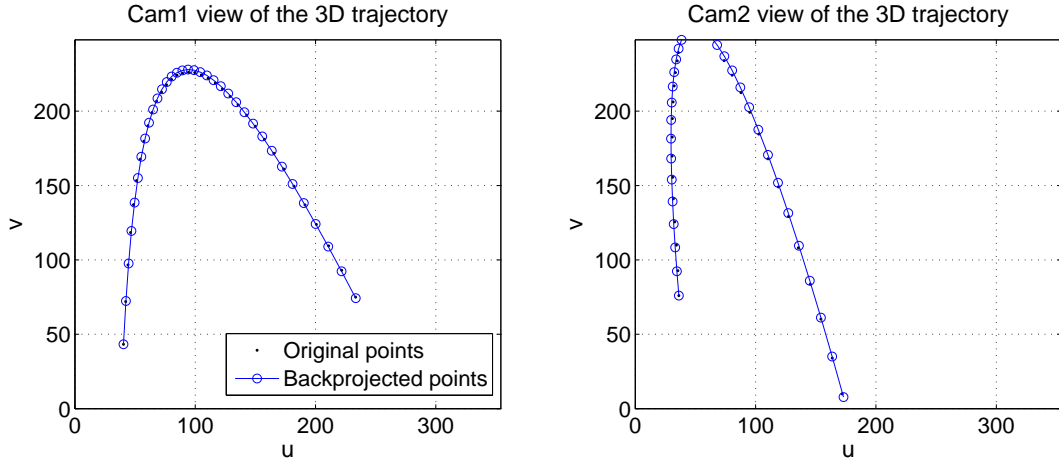


Figure 7.16: The extracted and re-projected 2-D positions of a ballistically moving target captured by two arbitrarily placed cameras. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm.

In this part we place the cameras arbitrarily and give the initializing state estimate roughly to the RB-EKF. No measuring device is used to obtain the initial state estimate. Just by observing the thrown target and the camera system, the initial state estimate is predicted as

$$\hat{\mathbf{X}}_0 = [-1 \ 0.5 \ 1]^T, \quad \hat{\mathbf{V}}_0 = [1 \ 2 \ 1]^T.$$

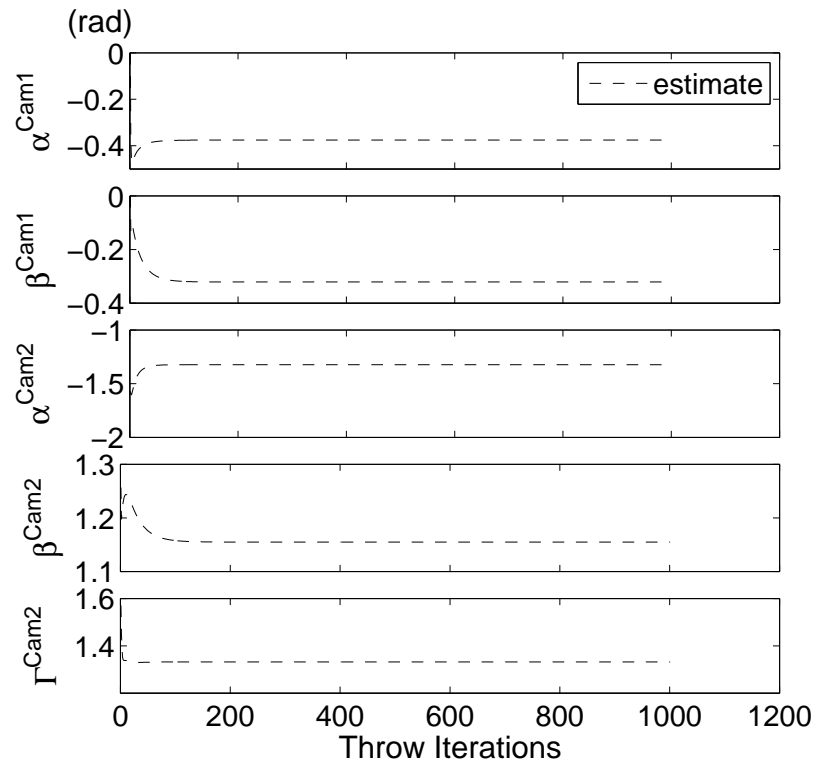
$$\hat{\mathbf{C}}_0 = [0 \ 0 \ -\frac{\pi}{2} \ \frac{\pi}{2} \ \frac{\pi}{2} \ 1 \ 0 \ 1]^T.$$

After 1000 throw iterations the RB-EKF algorithm yields the initial target state and the calibration parameter estimates as

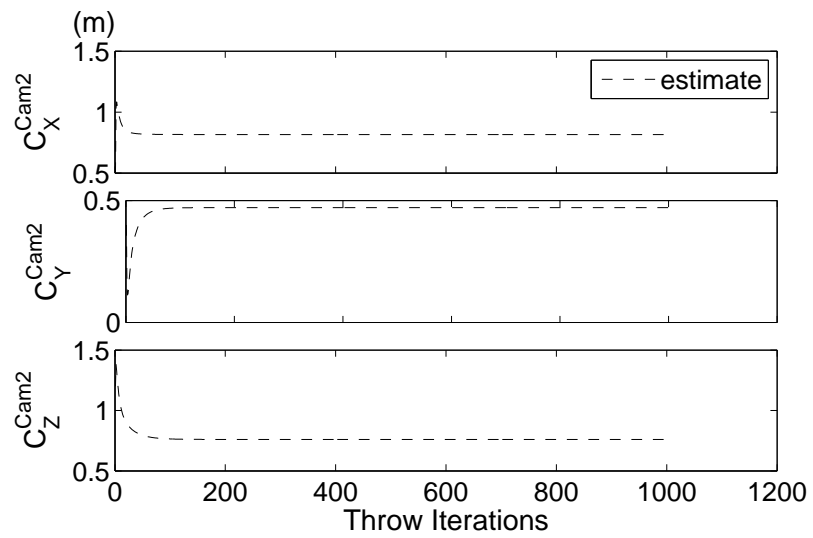
$$\hat{\mathbf{X}} = [-0.2721 \ 0.1341 \ 0.6719]^T, \quad \hat{\mathbf{V}} = [0.7714 \ 3.2010 \ 0.4417]^T,$$

$$\hat{\mathbf{C}} = [-0.3754 \ -0.3214 \ -1.3230 \ 1.1549 \ 1.3321 \ 0.8160 \ 0.4710 \ 0.7604]^T.$$





(a)



(b)

Figure 7.17: Convergence of the Repeated Bidirectional EKF Algorithm w.r.t *throw* iterations, (a) Rotation angles; (b) Translation parameters.

The re-projection pixel errors are 1.0963 and 1.2239 for Cam1 and Cam2 respectively. The original and re-projected points are shown in Figure 7.16 and the convergence curves of the RB-EKF algorithm are plot in Figure 7.17. Note that the target is thrown with more thrust than expected, so at the highest position it stays out of the FOV of Cam2. Hence the trajectory is not fully captured by Cam2. Anyway, those unobserved frames are handled by only executing the *prediction* steps but not the *update* steps of the EKF algorithm given in Table 3.1. So even when the calibration target is unobservable in some of the frames, the proposed method can still be effectively used.

In this experiment we have shown that when the initial target state and the camera orientation vector for an arbitrary placement of the cameras are initialized by only observing the system (without using any measurement device), RB-EKF method can yield the calibration parameters as well as the initial target state. Additionally even when the calibration object is unobservable in some of the frames, our method can still be used for the estimation of the parameters by only executing the *prediction* step of the EKF algorithm for those unobserved frames.

## 7.8 Experiment 6: Using Multiple Trajectories for Arbitrary Camera Placement

The physical experiments with a multi camera setup are finally concluded with this final calibration performance of the RB-EKF by using multiple target trajectories. The same camera configuration as in section 7.7 is kept and two additional target trajectories are captured for calibration. The initial estimate for the first target and the camera parameter vector are set as in section 7.7. For the initial states of the additional two trajectories, the following estimates are used to initialize the EKF:

$$\begin{aligned}\hat{\mathbf{X}}_0^{t_2} &= [1 \ 0.5 \ 1]^T, \quad \hat{\mathbf{V}}_0^{t_2} = [-1 \ 2 \ -1]^T, \\ \hat{\mathbf{X}}_0^{t_3} &= [-1 \ 0.5 \ 1]^T, \quad \hat{\mathbf{V}}_0^{t_3} = [1 \ 2 \ 1]^T.\end{aligned}$$

The target trajectories are alternately used for calibration and after 300 throw iterations for each trajectory (i.e. a total of 900 throw iterations) the RB-EKF algorithm yields the camera parameters as

$$\hat{\mathbf{C}} = [-0.3795 \ -0.3144 \ -1.3250 \ 1.1665 \ 1.3336 \ 0.8276 \ 0.4712 \ 0.7828]^T$$

and also the initial target state estimates for each throw as

$$\begin{aligned}\hat{\mathbf{X}}^{t_1} &= [-0.2755 \ 0.1320 \ 0.6776]^T, \quad \hat{\mathbf{V}}^{t_1} = [0.7798 \ 3.1949 \ 0.4577]^T, \\ \hat{\mathbf{X}}^{t_2} &= [0.4228 \ 0.1714 \ 0.9674]^T, \quad \hat{\mathbf{V}}^{t_2} = [-1.2062 \ 3.1798 \ -0.2982]^T, \\ \hat{\mathbf{X}}^{t_3} &= [-0.0309 \ 0.5113 \ 0.7266]^T, \quad \hat{\mathbf{V}}^{t_3} = [0.5964 \ 0.8585 \ 1.1182]^T.\end{aligned}$$

The average re-projection pixel errors are 0.6943 and 1.5296 for Cam1 and Cam2 respectively. The original and re-projected points for all three trajectories are shown in Figure 7.18 and the convergence curves of the RB-EKF algorithm are plot in Figure 7.19. When the re-projection pixel errors are compared with the ones in section 7.7 there seems to be no improvement over using a single trajectory. However as shown in section 6.5 even if the pixel errors are not reduced by using multiple trajectories, the obtained estimates for the camera parameters can be much more accurate than a single throw case, because much more calibration points are used for pose estimation.

As a final comment, note that only 9 observation points are provided by the second trajectory as shown in Figure 7.18(b). Such less data alone can normally not be sufficient for accurate parameter estimation. However many other data points are also provided when multiple trajectories are used for calibration, hence the trajectories for which very few data points are acquired may also be used to increase accuracy.

## 7.9 Discussion on Physical Experiments

In this chapter after the components of our multi-camera setup are described and internal parameters of the cameras are obtained by MATLAB Calibration Toolbox, physical experiments are performed to demonstrate the practical usability of the proposed algorithms. After performances of the R-EKF and RB-EKF are compared by using a free falling target, RB-EKF method is tested for various positions of the cameras by either using a single or multiple ballistic trajectories.

First two experiments show the superiority of RB-EKF over R-EKF in the sense of re-projection pixel errors. R-EKF performance relies on the accurate initialization of the target state. Unfortunately the initial target state can not be measured exactly even for a free falling target,

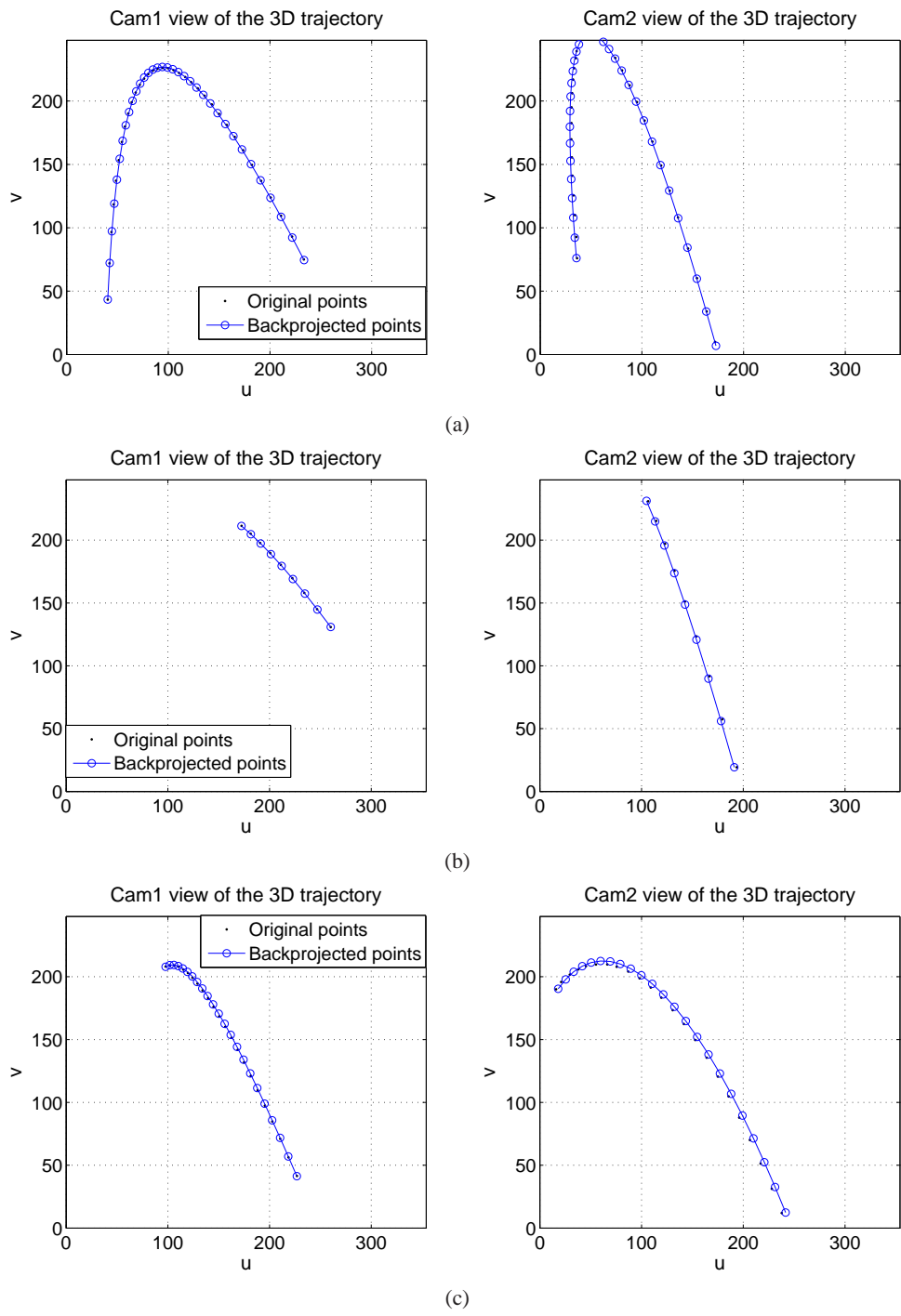
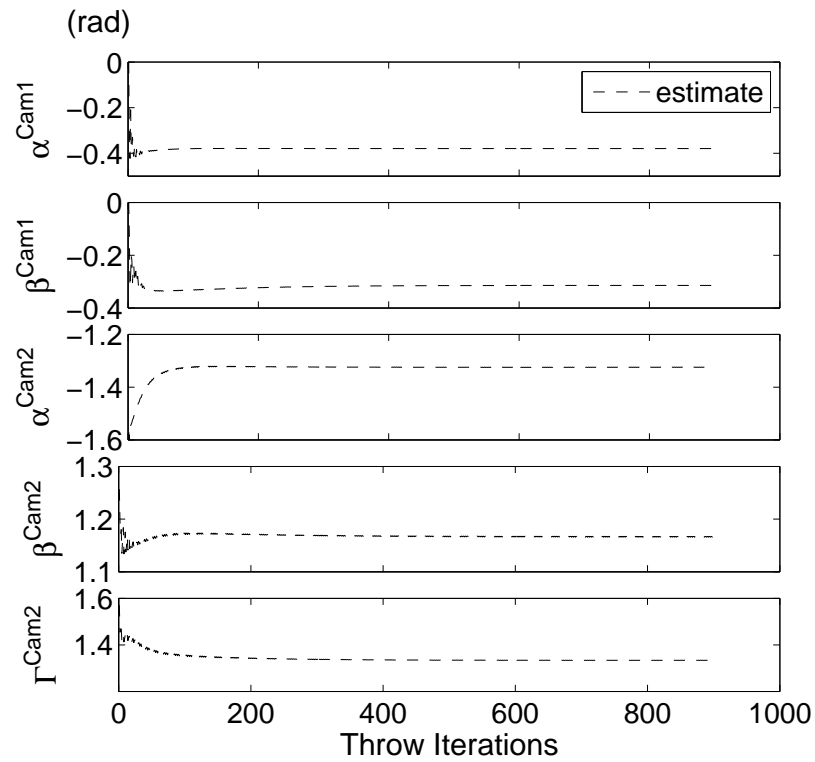
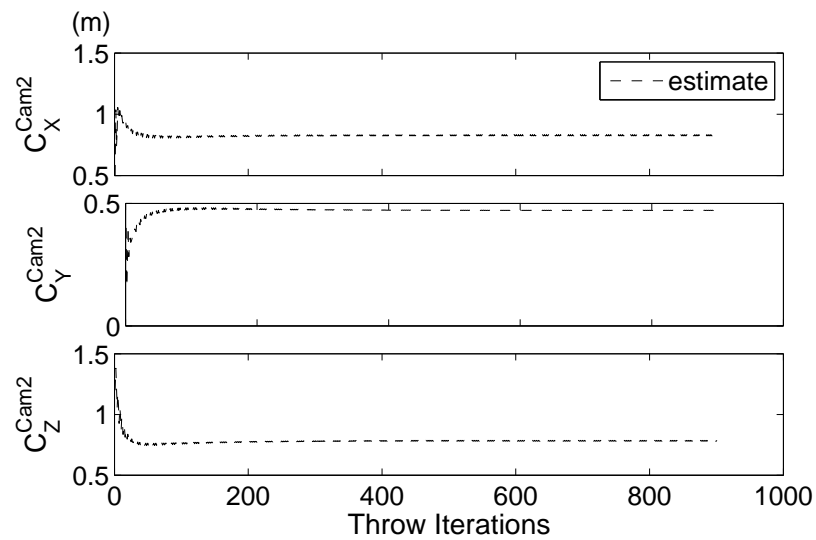


Figure 7.18: The extracted and re-projected 2-D positions of three ballistically moving targets captured by two arbitrarily placed cameras. The black dots denote the originally captured points and the small circles denote the re-projected points that are obtained after 1000 throw iterations by Repeated Bidirectional EKF algorithm for the (a) first; (b) second and (c) third trajectories.



(a)



(b)

Figure 7.19: Convergence of the Repeated Bidirectional EKF Algorithm w.r.t *throw* iterations, (a) Rotation angles; (b) Translation parameters.

because the camera capturing and target initiation instants can not be synchronized and hence the velocity of the target is not exactly equal to  $\vec{0}$  m/s when the first frame is captured. Additionally the metric measurements for acquiring the target position w.r.t the center of the first camera can only be performed from the outer core of the camera bulk. Since the initial target state can be estimated accurately by RB-EKF method, calibration parameters are also estimated more accurately.

The RB-EKF is tested by using a single ballistic trajectory for parallel, perpendicular and arbitrary placement of the cameras in sections 7.5, 7.6 and 7.7. Thus the capability of the proposed method is presented by investigating the calibration performance of the algorithm with various camera poses. In all of these experiments final estimates of the calibration parameters satisfy our expectations. Especially section 7.7 proves the practical applicability of the algorithm since the poses of arbitrarily placed cameras are estimated successfully by using partial observation of the target trajectory.

Finally multiple target trajectories are used for increasing the calibration accuracy in section 7.8. Although the re-projection pixel error is not reduced compared to a single trajectory case, the estimations are expected to be more accurate as shown in 6.5.

In conclusion physical experiments demonstrate that the ease of applicability and the successful calibration performance of the RB-EKF method makes it particularly beneficial for any application where frequent re-calibration is required such as outdoor field measurement applications using multiple-cameras.

## CHAPTER 8

### Conclusions and Future Work

We propose a procedurally effective and high performance method of calibrating a multiple-camera measurement system by making use of the known dynamics of a target object. The proposed method promises to considerably facilitate the field installation of a camera based ground truth measurement system. In fact it needs only a ballistically thrown simple calibration target in the multi-camera joint-FOV to calibrate the complete system. Extension of the two camera case considered for the formulation to a multi-camera system is straightforward in two ways: first, we can either insert the calibration parameters and measurements taken from the additional cameras into the state and measurement vectors and use the algorithm as it is, or we can calibrate the cameras two by two, i.e., Cam1 – 2, Cam1 – 3, Cam1 – 4, etc. The latter approach may be more feasible since it does not increase the dimensionality of the problem.

The simulation results validate that both R-EKF and RB-EKF algorithms can converge rapidly and asymptotically yield the desired camera parameters. However for successful calibration R-EKF needs an accurate estimate of the initial target state which is quite difficult to obtain in a real case. On the other hand RB-EKF can tolerate significant initial state uncertainty when sufficient number of *throw iterations* are performed. This removes the need for manual geometric measurements performed by the experimenter (such as 3D-2D correspondence measurements) and hence results in the removal of the human related measurement errors that can affect the calibration process. Although our primary motivation is to generate an easily applicable and fast calibration algorithm but not to surpass the performances of the standard methods we feel that the eliminated human factor may enable us to get superior calibration performance in practice compared to the methods that need 3D-2D correspondence measurements.

The testing of the algorithms on a real imaging system shows that RB-EKF method can be successfully used for determining the external poses of the cameras in a multi-camera setup. The cameras are placed in parallel, perpendicular and arbitrary positions in order to test the RB-EKF method for free fall and ballistic motion cases by either using a single or multiple target trajectories. In all of the physical experiments the calibration parameters are estimated around their expected values.

Although re-projection pixel error is used to demonstrate the success rate of our method one should not forget that pixel error is computed on the 2D image plane, and it is not always proportional with the actual 3D position errors. Our method tries to balance the information coming from the dynamic motion model with the information due measurements. Some of the present methods only use the measurements obtained from a static scene and minimize the re-projection pixel error in order to calibrate a camera setup. Hence even when RB-EKF yields higher re-projection pixel errors compared to these algorithms, the actual poses of the cameras may still be estimated better. Moreover the presence of process and measurement noise terms in the EKF algorithm compensate for modeling errors and unknown inputs to the system, reducing the sensitivity to such factors as wind disturbance, image noise, ball center detection errors etc.

An interesting extension to our study is to provide a full calibration (both internal and external) by means of the proposed method. Even though internal calibration does not have the same speed requirements of the external calibration and hence is not a critical candidate for this method, these static internal parameters can also be made part of the state vector and obtained similar to the external parameters. But it is clear that with the increasing dimensionality of the problem, the convergence properties of EKF may change for the worse in terms of speed, accuracy and stability.



## REFERENCES

- [1] I. Kakadiaris and D. Metaxas, “Model-Based Estimation of 3D Human Motion,” 2000.
- [2] DM Gavrilu and LS Davis, “3-D model-based tracking of humans in action: a multi-view approach,” *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR’96, 1996 IEEE Computer Society Conference on*, pp. 73–80, 1996.
- [3] J.M. Rehg and T. Kanade, “Visual Tracking of High DOF Articulated Structures: an Application to Human Hand Tracking,” *ECCV(2)*, pp. 35–46, 1994.
- [4] R. Stolkin, A. Greig, and J. Gilby, “A calibration system for measuring 3D ground truth for validation and error analysis of robot vision algorithms,” *Measurement Science and Technology*, vol. 17, no. 10, pp. 2721–2730, 2006.
- [5] V. Lippiello, B. Siciliano, and L. Villani, “3D pose estimation for robotic applications based on a multi-camera hybrid visual system,” *Proceedings of 2006 IEEE International Conference on Robotics and Automation*, pp. 2732–2737, 2006.
- [6] Vicon MX Motion Capture Systems, “<http://www.vicon.com/products/viconmx.html>,” (accessed August 19, 2008).
- [7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2003.
- [8] F. Remondino and C. Fraser, “Digital camera calibration methods: considerations and comparisons,” *Proc. ISPRS Commission V Symposium*, pp. 266–272, 2006.
- [9] U. Saranlı, M. Buehler, and D.E. Koditschek, “RHex: A Simple and Highly Mobile Hexapod Robot,” *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 616, 2001.
- [10] C. Everitt, “Projective Texture Mapping,” *White paper, NVidia Corporation*, 2001.
- [11] J. Bloomenthal and J. Rokne, “Homogeneous Coordinates,” 1993.
- [12] YI Abdel-Aziz and HM Karara, “Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry,” *Proceedings of the Symposium on Close-Range Photogrammetry*, pp. 1–18, 1971.
- [13] KW Wong, “Mathematical formulation and digital analysis in close-range photogrammetry,” *Photogrammetric Eng. Remote Sensing*, vol. 41, no. 11, pp. 1355–1373, 1975.
- [14] W. Faig, “Calibration of close-range photogrammetry systems: Mathematical formulation,” *Photogrammetric Engineering and Remote Sensing*, vol. 41, no. 12, pp. 1479–1486, 1975.

- [15] J. Weng, P. Cohen, and M. Herniou, "Calibration of stereo cameras using a non-linear distortion model [CCD sensory]," *Pattern Recognition, 1990. Proceedings., 10th International Conference on*, vol. 1, 1990.
- [16] R.Y. TSAI, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *Radiometry*, 1992.
- [17] J. Weng, P. Cohen, and M. Herniou, "Camera Calibration with Distortion Models and Accuracy Evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 965–980, 1992.
- [18] G.Q. Wei and S. De Ma, "Implicit and explicit camera calibration: theory and experiments," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 5, pp. 469–480, 1994.
- [19] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [20] J. Bastian and A. van den Hengel, "Computing Image-Based Reprojection Error on Graphics Hardware," *Proc. VIIth Digital Image Computing: Techniques and Applications*, 2006.
- [21] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit imagecorrection," *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 1106–1112, 1997.
- [22] P. Sturm and S. Ramalingam, "A generic concept for camera calibration," *European Conference on Computer Vision*, vol. 2, pp. 1–13, 2004.
- [23] Camera Calibration Toolbox for MATLAB, "[http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/)," (accessed August 19, 2008).
- [24] F. Pedersini, A. Sarti, and S. Tubaro, "Multi-camera systems," *Signal Processing Magazine, IEEE*, vol. 16, no. 3, pp. 55–65, 1999.
- [25] S. Ganapathy, "Decomposition of transformation matrices for robot vision," *Robotics and Automation. Proceedings. 1984 IEEE International Conference on*, vol. 1, 1984.
- [26] C. Wiles and A. Davison, "Calibrating a Multi-Camera System for 3D Modelling," *Proceedings of the IEEE Workshop on Multi-View Modelling and Analysis of Visual Scenes*, pp. 29–36, 1999.
- [27] I.O. Sebe and G.Q. Chen, "Multi-camera Calibration," Tech. Rep., STMicroelectronics Technical Report, 2002.
- [28] M. Han and T. Kanade, "Creating 3D models with uncalibrated cameras," *Applications of Computer Vision, 2000, Fifth IEEE Workshop on.*, pp. 178–185, 2000.
- [29] T. Svoboda, D. Martinec, and T. Pajdla, "A Convenient Multicamera Self-Calibration for Virtual Environments," *Presence: Teleoperators & Virtual Environments*, vol. 14, no. 4, pp. 407–422, 2005.
- [30] H.G. Maas, "Image sequence based automatic multi-camera system calibration techniques," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 54, no. 5-6, pp. 352–359, 1999.

- [31] P. Baker and Y. Aloimonos, "Complete calibration of a multi-camera network," *Omnidirectional Vision, 2000. Proceedings. IEEE Workshop on*, pp. 134–141, 2000.
- [32] B. Moghaddam et al., *Probabilistic visual learning for object detection*, Ph.D. thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 1997.
- [33] CP Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," *Computer Vision, 1998. Sixth International Conference on*, pp. 555–562, 1998.
- [34] C. Papageorgiou and T. Poggio, "A Trainable System for Object Detection," *International Journal of Computer Vision*, vol. 38, no. 1, pp. 15–33, 2000.
- [35] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-Based Object Detection in Images by Components," 2001.
- [36] S. Agarwal and D. Roth, "Learning a sparse representation for object detection," *Proc. ECCV*, vol. 4, pp. 113–130, 2002.
- [37] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proc. CVPR*, vol. 1, pp. 511–518, 2001.
- [38] P. Viola and M. Jones, "Robust real-time object detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2002.
- [39] R. Lienhart, A. Kuranov, and V. Pisarevsky, "Empirical analysis of detection cascades of boosted classifiers for rapid object detection," *DAGM 25th Pattern Recognition Symposium*, pp. 297–304, 2003.
- [40] K. Levi and Y. Weiss, "Learning object detection from a small number of examples: The importance of good features," *Proc. CVPR*, vol. 2, pp. 53–60, 2004.
- [41] H. Schneiderman and T. Kanade, "Statistical method for 3 D object detection applied to faces and cars," *Proc IEEE Comput Soc Conf Comput Vision Pattern Recognition*, vol. 1, pp. 746–751, 2000.
- [42] H. Durrant-Whyte and T. Bailey, "Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms," *Robotics and Automation Magazine*, vol. 13, pp. 99–110, 2006.
- [43] T. Bailey and H. Durrant-Whyte, "Simultaneous Localisation and Mapping (SLAM): Part II State of the Art," *Robotics and Automation Magazine*, vol. 13, pp. 108–117, 2006.
- [44] L. Lee, R. Romano, and G. Stein, "Monitoring activities from multiple video streams: establishing a common coordinate frame," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 758–767, 2000.
- [45] A. Rahimi, B. Dunagan, and T. Darrell, "Simultaneous calibration and tracking with a network of non-overlapping sensors," *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1.
- [46] B. Bose and E. Grimson, "Ground plane rectification by tracking moving objects," *IEEE International Workshop on Visual Surveillance and PETS*, 2003.

- [47] C. Stauffer and K. Tieu, “Automated multi-camera planar tracking correspondence modeling,” *Computer Vision and Pattern Recognition*, pp. 259–266, 2003.
- [48] A. Azarbayejani and A.P. Pentland, “Recursive estimation of motion, structure, and focal length,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, pp. 562–575, 1995.
- [49] F.M. Mirzaei and S.I. Roumeliotis, “A Kalman filter-based algorithm for IMU-camera calibration,” *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 2427–2434, 2007.
- [50] E. Stringa and C.S. Regazzoni, “A novel camera calibration algorithm based on Kalman filter,” *Proceedings of 15th International Conference on Pattern Recognition*, vol. 1, pp. 872–875, 2000.
- [51] Rudolph Emil Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of the ASME—Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.
- [52] T. PAEK and D.M. CHICKERING, “Evaluating the Markov assumption in Markov Decision Processes for spoken dialogue management,” *Language resources and evaluation*, vol. 40, no. 1, pp. 47–66, 2006.
- [53] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*, Wiley New York, 2001.
- [54] S. Thrun, W. Burgard, and D. Fox, “Probabilistic Robotics (Intelligent Robotics and Autonomous Agents),” 2005.
- [55] G. Welch and G. Bishop, “An Introduction to the Kalman Filter,” *ACM SIGGRAPH 2001 Course Notes*, 2001.
- [56] D.C. Brown, “Close-range camera calibration,” *Photogrammetric Engineering*, vol. 37, no. 8, pp. 855–866, 1971.

# Bir İnsansız Sualtı Aracının Hidrodinamik ve Isıl Analizi ULİSAR

---

Prof. Dr. M. Haluk Aksel  
Necmettin Cevheri

# Sunum İeriđi

---

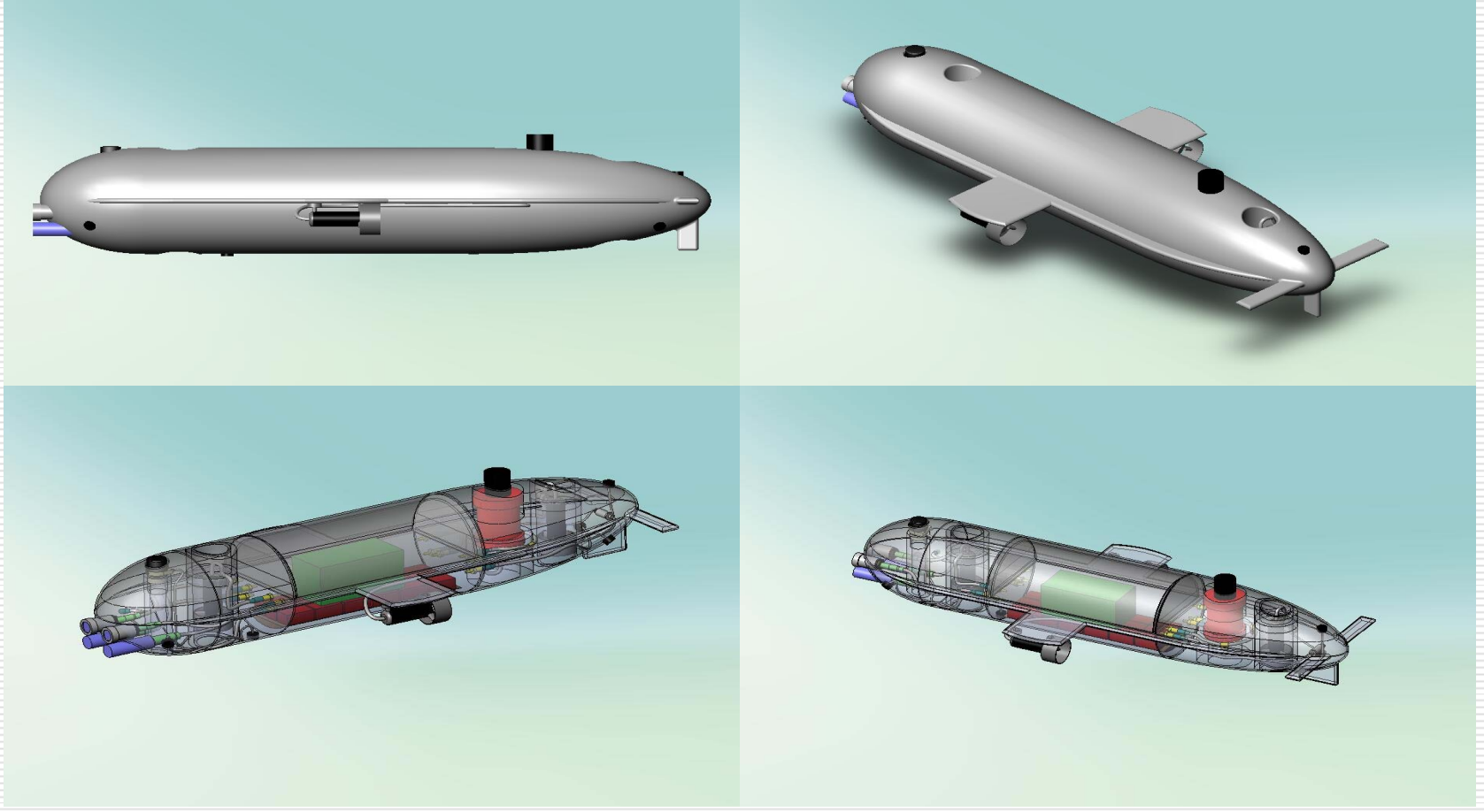
- Giriş
- Hidrodinamik Analizler
- Isıl Analizler
- Deđerlendirme

# Giriş

---

- Bir ürünün performansının tasarım sürecindeyken tahmin edilmesi önemlidir:
  - Performans isterlerini karşılaması
  - Olumsuz durumlarda tasarımın düzeltilmesi
  - Yeni tasarımın performans tahmini
  - Tasarım sürecinin hızlandırılması
  - Test ve prototip imalatı masraflarının azaltılması

# Giriş





# Analizler


---

Hidrodinamik analizler

Isıl analizler


# Hidrodinamik Analizler

---

- Aracın su altında çalışması esnasında maruz kalacağı ortamın simülasyonu
- Elde edilen önemli parametreler:
  - Maruz kalınacak sürtünme ve sürüklenme kuvvetleri 
  - Araç üzerinde oluşacak basınç dağılımı

# Hidrodinamik Analizler

---

- Tahrik elemanlarının seçimi ve yerleşimi 
- Gövde et kalınlıklarının belirlenmesi
- Aktif ve pasif kontrol yüzeylerinin tasarımı

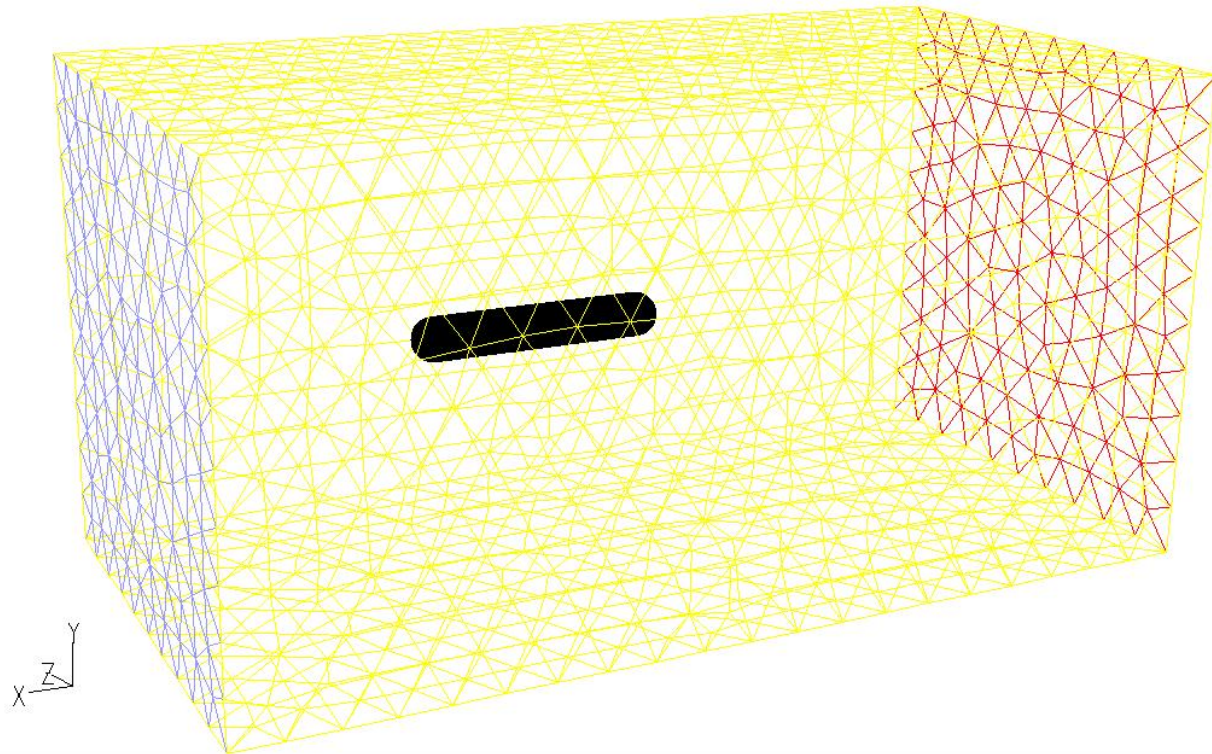
# Hidrodinamik Analizler

---

- Basit 3 boyutlu çeşitli şekiller karşılaştırıldı.
- Geometri ve ağ oluşturmak için Gambit, 3 boyutlu sınır tabaka oluşturmak için Tgrid, çözümlleme için ise Fluent yazılımları kullanıldı.
- Geometriye kanatçıklar eklenerek model geliştirildi.

# Hidrodinamik Analizler

S



Grid

Oct 07, 2008  
FLUENT 6.3 (3d, dp, pbns, rke)

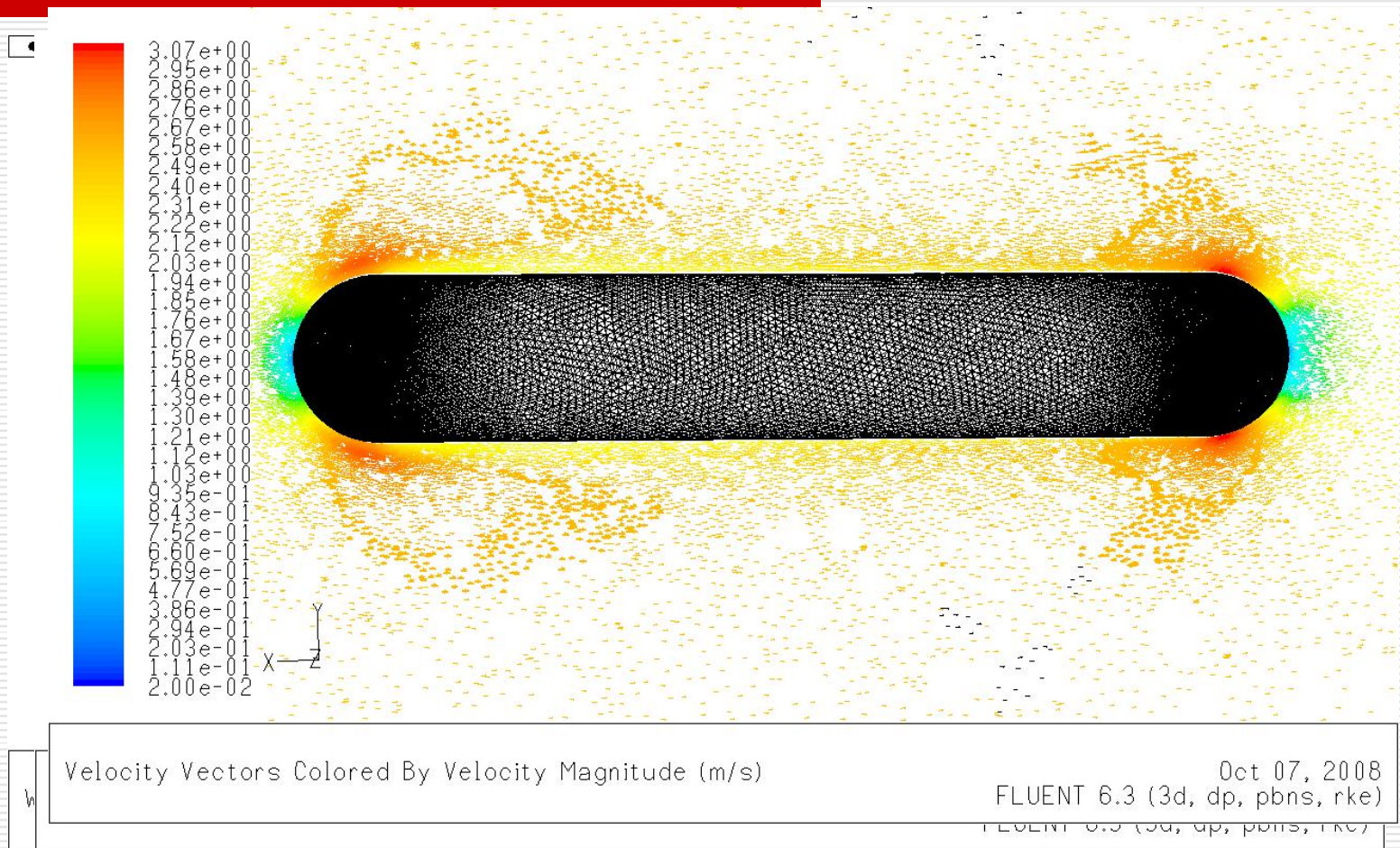
# Hidrodinamik Analizler

---

## Çözüm Parametreleri:

- 1544597 tetrahedron ve hexahedron hücre
- Solver: Pressure Based, 3D, Steady, Implicit
- Gradient Option: Green-Gauss Node Based
- Viscous Formulation: k- $\epsilon$  realizable, standard wf
- Madde: sabit yoğunluklu su
- Sınır koşulları: 2.5 m/s giriş hızı  
Turbulence intensity: 1%  
Hidrolik çap: 3.5 m

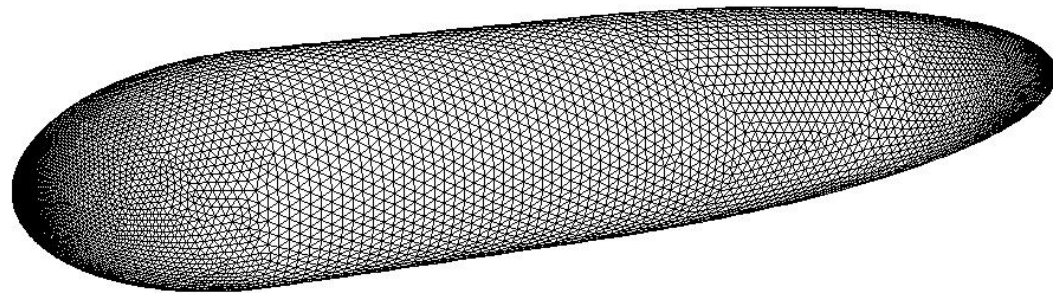
# Hidrodinamik Analizler



# Hidrodinamik Analizler

---

E



Grid

Grid

Oct 07, 2008  
FLUENT 6.3 (3d, dp, pbns, rke)

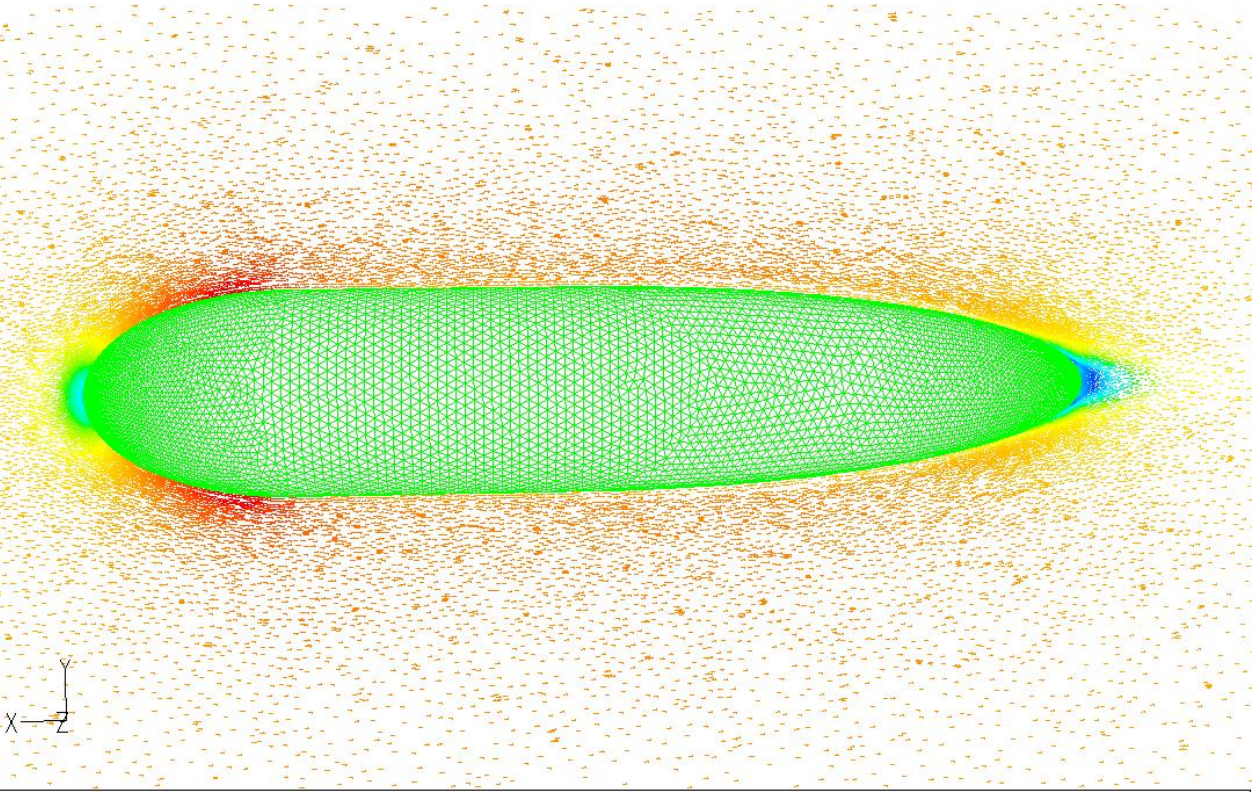


# Hidrodinamik Analizler

Eliq



2.86e+00  
2.75e+00  
2.66e+00  
2.58e+00  
2.49e+00  
2.41e+00  
2.33e+00  
2.23e+00  
2.15e+00  
2.00e+00  
1.98e+00  
1.89e+00  
1.81e+00  
1.72e+00  
1.64e+00  
1.55e+00  
1.46e+00  
1.38e+00  
1.29e+00  
1.21e+00  
1.12e+00  
1.04e+00  
9.51e-01  
8.65e-01  
7.80e-01  
6.94e-01  
6.09e-01  
5.23e-01  
4.37e-01  
3.52e-01  
2.66e-01  
1.81e-01  
9.50e-02  
3.38e-02



X  
Y  
Z

Velocity Vectors Colored By Velocity Magnitude (m/s)

Oct 07, 2008  
FLUENT 6.3 (3d, dp, pbns, rke)

# Hidrodinamik Analizler

## □ Elipsoid tasarım:

"Force Report"

zone name	pressure force n	viscous force n	total force n	pressure coefficient	viscous coefficient	total coefficient
hull	7.4327311	13.230595	20.663326	0.026480448	0.047136386	0.073616834
net	7.4327311	13.230595	20.663326	0.026480448	0.047136386	0.073616834

## □ Silindirik tasarım:

"Force Report"

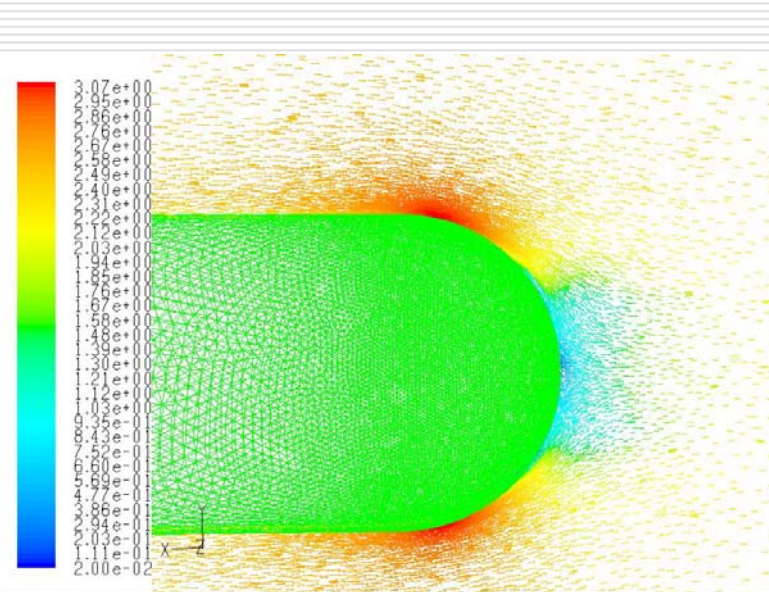
zone name	pressure force n	viscous force n	total force n	pressure coefficient	viscous coefficient	total coefficient
hull	25.029807	17.953371	42.983178	0.089173212	0.063962129	0.15313534
net	25.029807	17.953371	42.983178	0.089173212	0.063962129	0.15313534

# Hidrodinamik Analizler

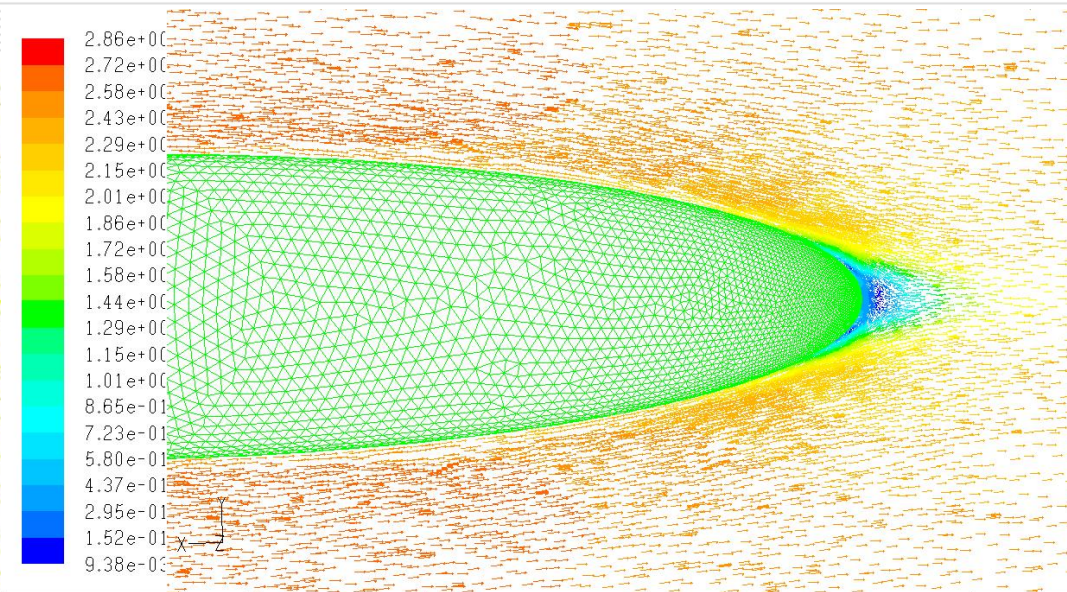
---

- Elipsoid tasarımda akış yüzeyden daha geç ayrılır.
- Aracın ön ve arka yüzleri arasındaki basınç farkı daha az olur ve sürüklenme kuvveti azalır.
- Düşük sürüklenme kuvveti nedeniyle ön ve arka tarafta parabolik, orta kısımda ise silindirik bir tasarım tercih edilmiştir.

# Hidrodinamik Analizler



Velocity Vectors Colored By Velocity Magnitude (m/s)  
FLUEN

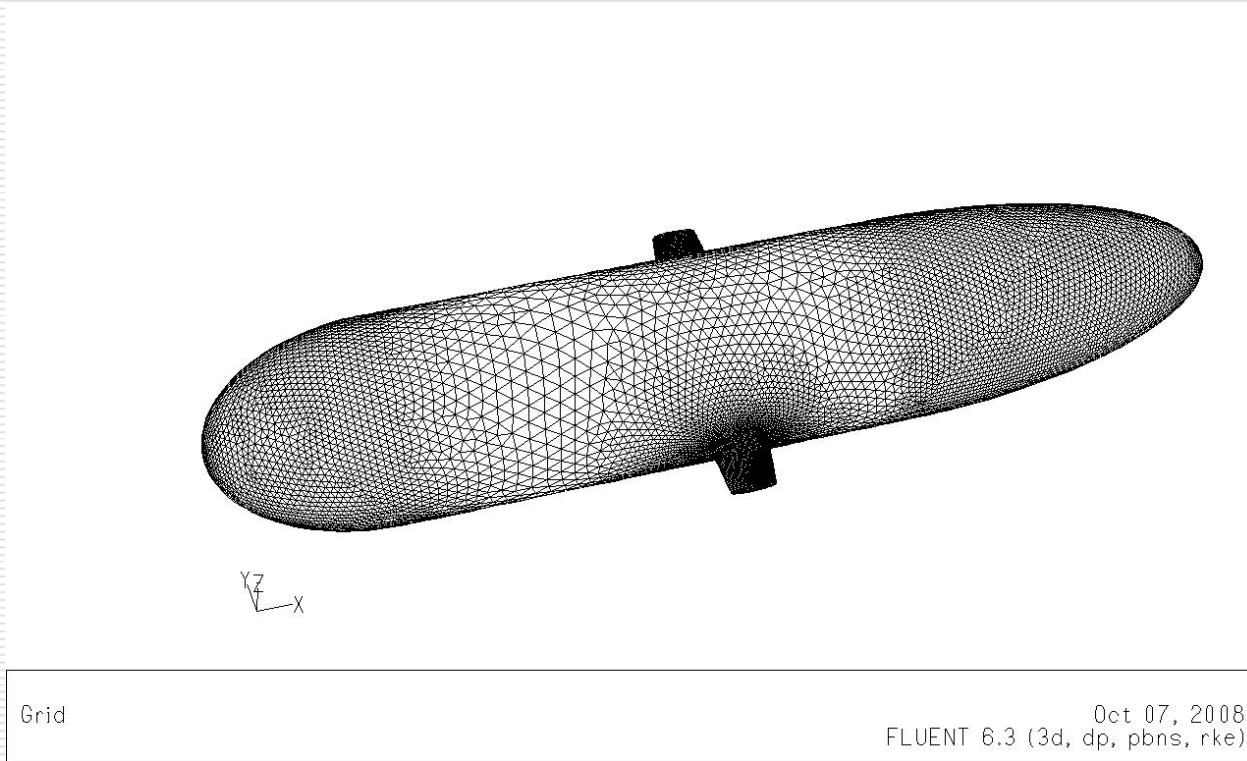


Velocity Vectors Colored By Velocity Magnitude (m/s)  
FLUENT 6.3 (3d, dp, pbns, r

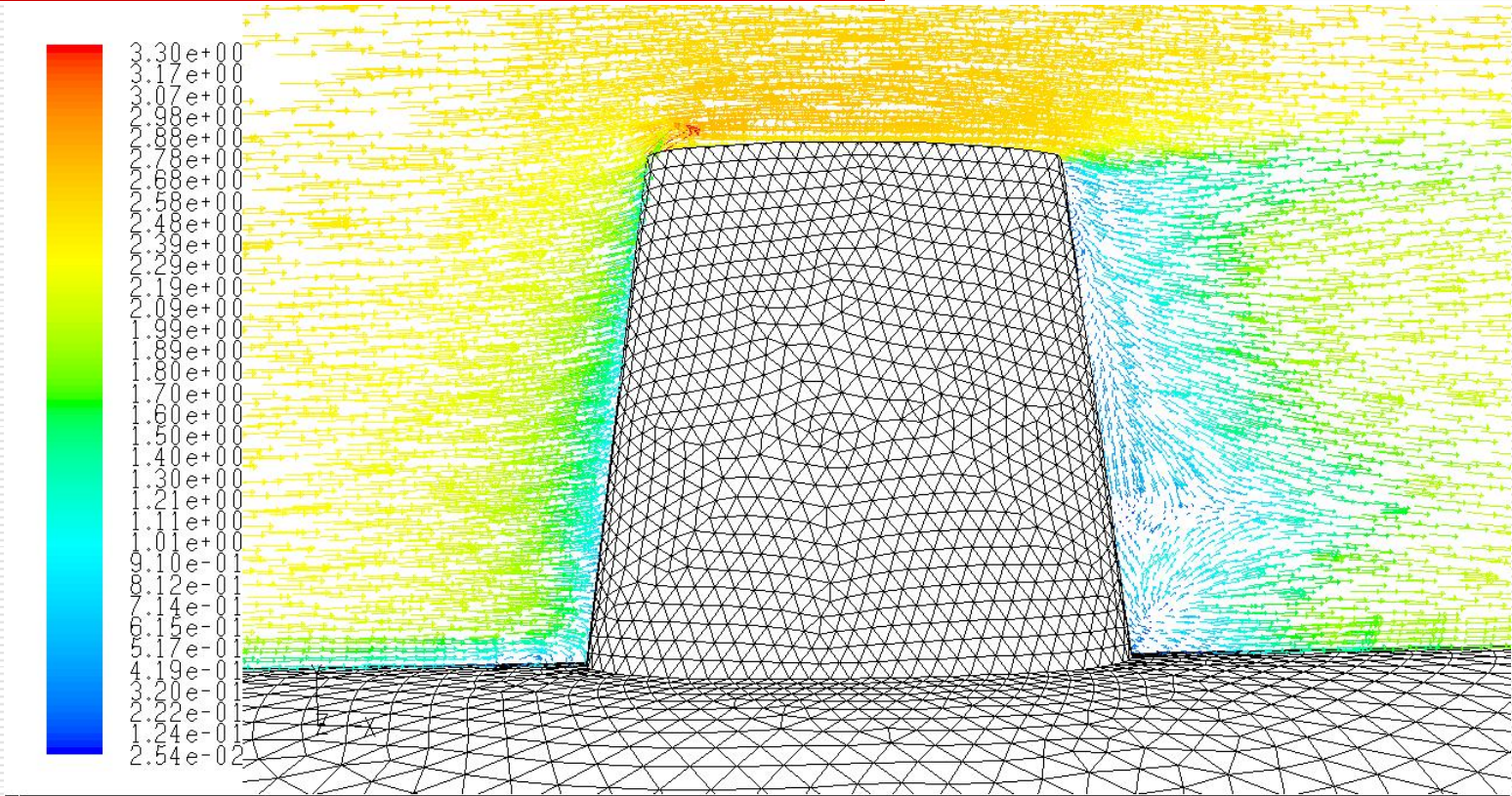
# Hidrodinamik Analizler

---

## Kanatların Eklenmesi:



# Hidrodinamik Analizler



Velocity Vectors Colored By Velocity Magnitude (m/s) Oct 07, 2008  
FLUENT 6.3 (3d, dp, pbns, rke)

# Hidrodinamik Analizler

## □ Elipsoid Tasarım:

"Force Report"

zone name	pressure force n	viscous force n	total force n	pressure coefficient	viscous coefficient	total coefficient
hull	7.4327311	13.230595	20.663326	0.026480448	0.047136386	0.073616834
net	7.4327311	13.230595	20.663326	0.026480448	0.047136386	0.073616834

## □ Elipsoid tasarım+Kanatçıklar:

"Force Report"

zone name	pressure force n	viscous force n	total force n	pressure coefficient	viscous coefficient	total coefficient
hull	10.434351	14.677407	25.111758	0.033456837	0.047061826	0.080518663
net	10.434351	14.677407	25.111758	0.033456837	0.047061826	0.080518663

# Hidrodinamik Analizler

---

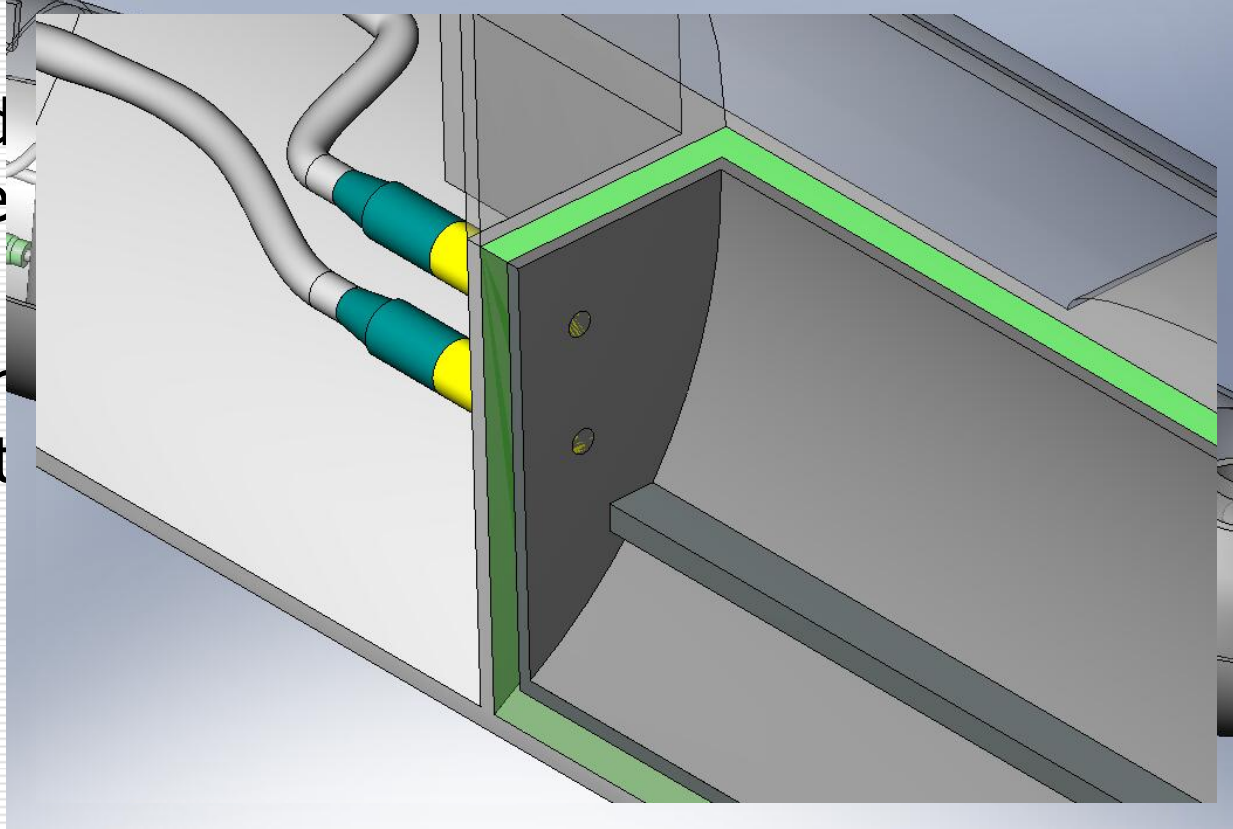
## Değerlendirme:

- ❑ Model karmaşıklıkça sürüklenme kuvvetlerinin artması beklenebilir.
- ❑ Elde edilen veriler yaklaşık olarak 25 N( $\sim 2.55$  kgf) değerinde bir sürüklenme kuvveti öngörmektedir.
- ❑ Bu nedenle her biri 2.2 kgf itki verebilen 2 adet motorun kullanılmasına karar verilmiştir.



# Isıl Analizler

- Sızdıran pille
- Komplet yalıtım



ve  
ar.

ar

# Isıl Analizler

---

- ❑ Yayılan ısı yalıtkanlık nedeniyle ortamdan atılamayıp sıcaklığı yükseltebilir.
- ❑ Artan ortam sıcaklığı ekipmanın çalışmasını olumsuz etkileyebilir.
- ❑ Isınma probleminin üretim sonrası çözümü zordur.
- ❑ Kritik sıcaklık noktalarının belirlenmesi önemlidir.

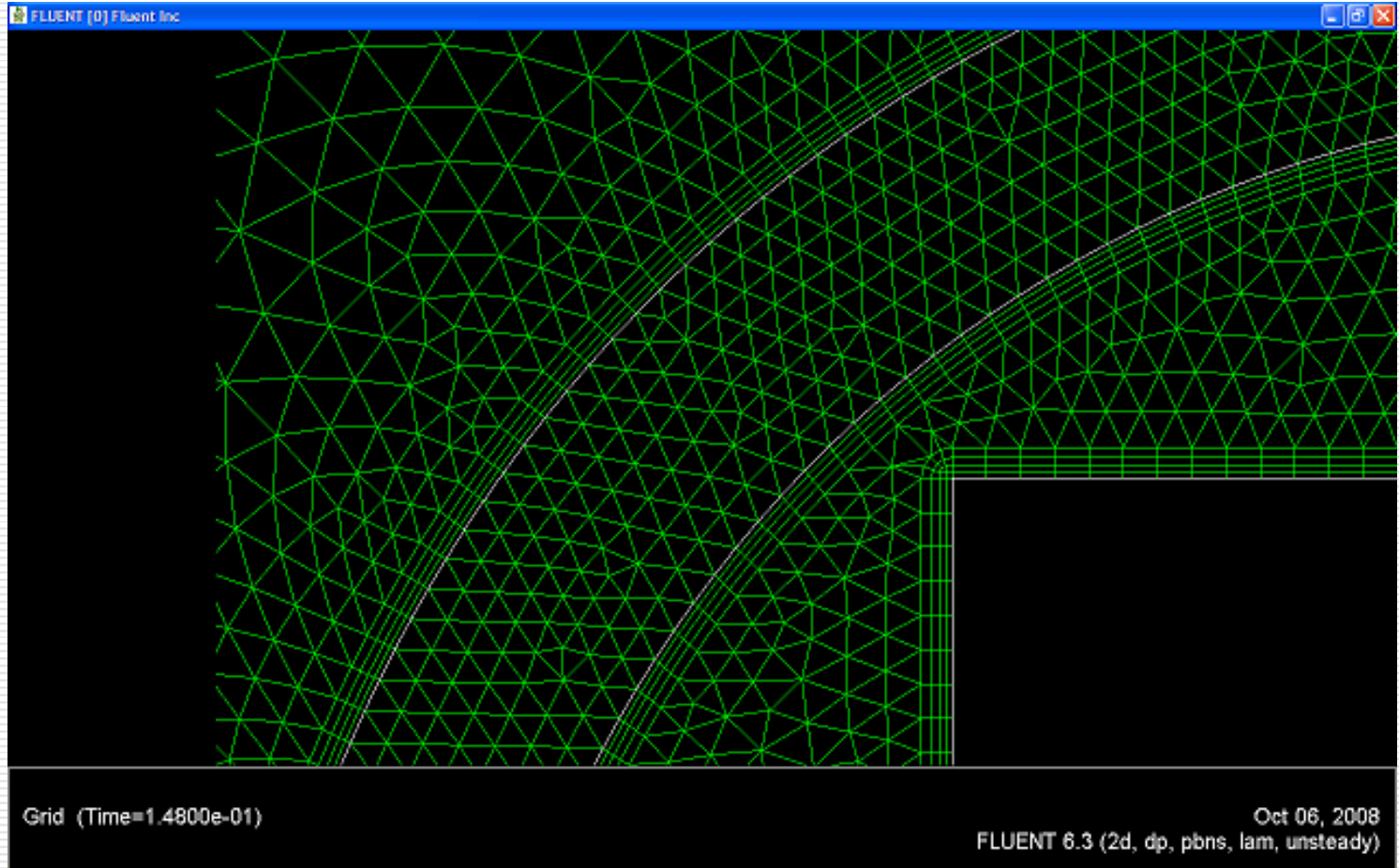
# Isıl Analizler

---

- Zamana bağlı çözümlene gerekmektedir.
- Öncelikli tercih 2 boyutlu çözümlenmedir.
- Elde edilen verilere göre 3 boyutlu analizlere başvurulabilir.

# Isıl Analizler

İlk



# Isıl Analizler

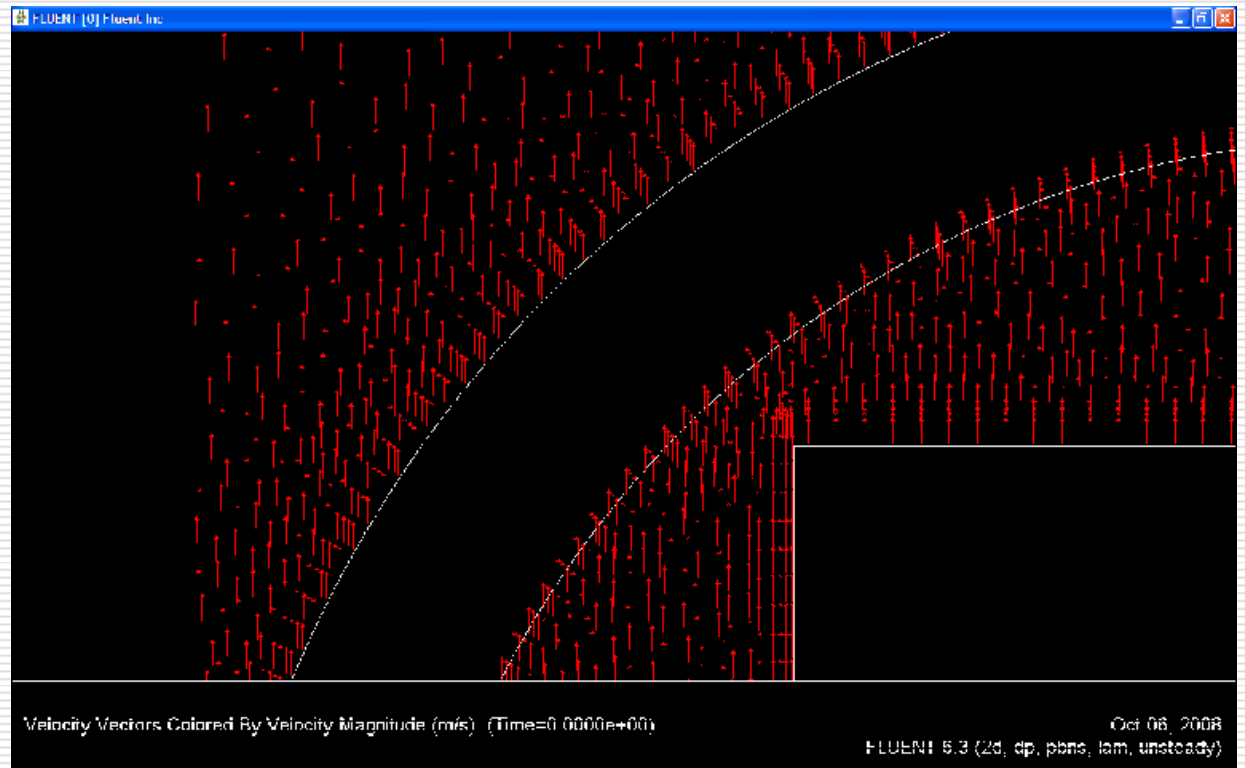
---

- Piller hesaba katılmamıştır.
- 2 saatlik süre (7200 s) için çözümlenmiştir.
- İlk sıcaklıklar  $27^{\circ}\text{C}$  olarak alınmıştır.
- Piller 5 W, kutu ise 50 W ısı yaymaktadır.
- Aracın hareketsiz olduğu kabul edilmiştir.
  
- Solver: Pressure-based, 2D, unsteady
- Gradient Option: Green-Gauss Node Based
- Viscous Model: Laminar
  
- Hava: ideal gaz, Su: sabit yoğunluklu

# Isıl Analizler

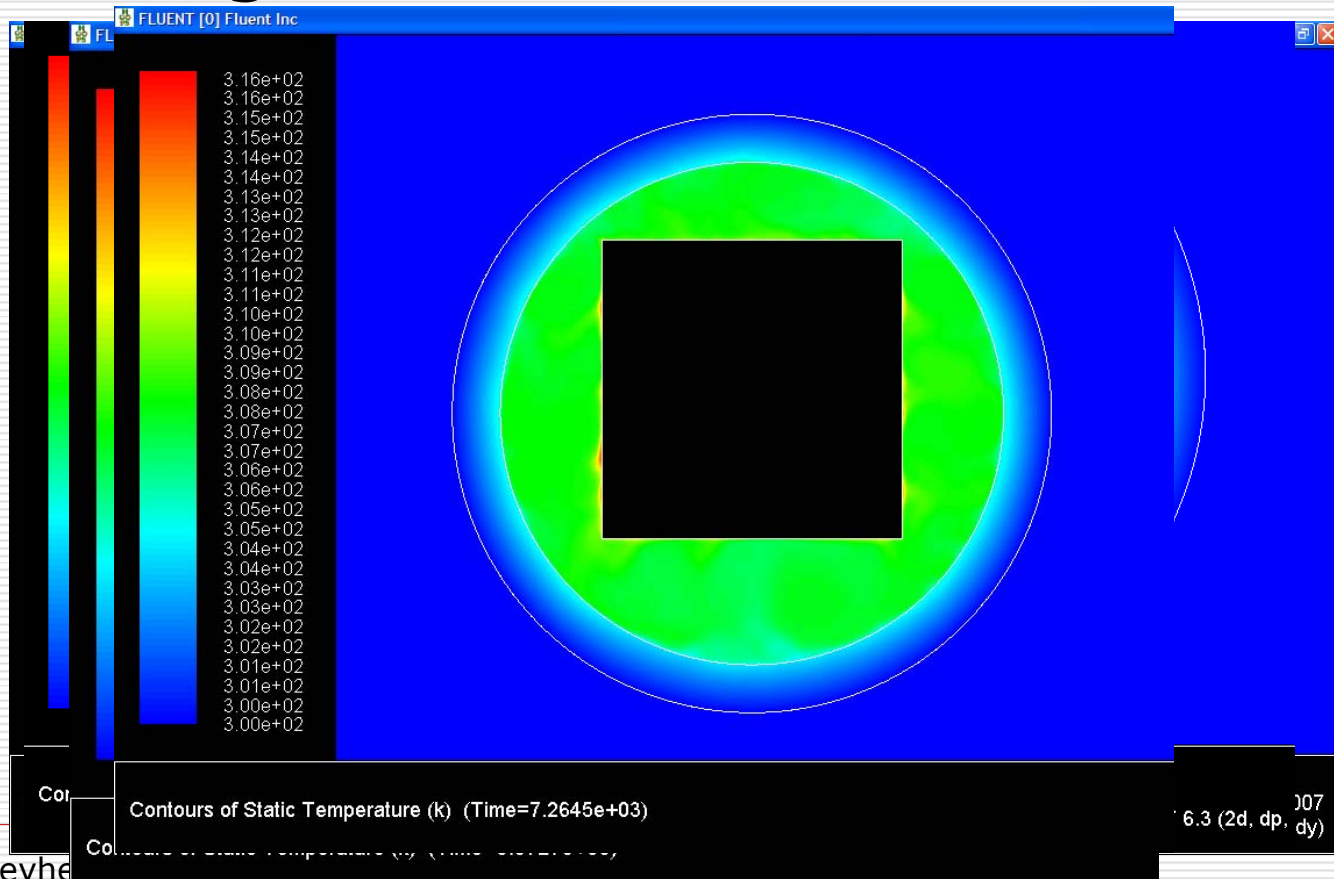
## □ Initialization:

□  $V_y = 0.02 \text{ m/s}$



# Isıl Analizler

## Sıcaklık dağılımı:



# Isıl Analizler

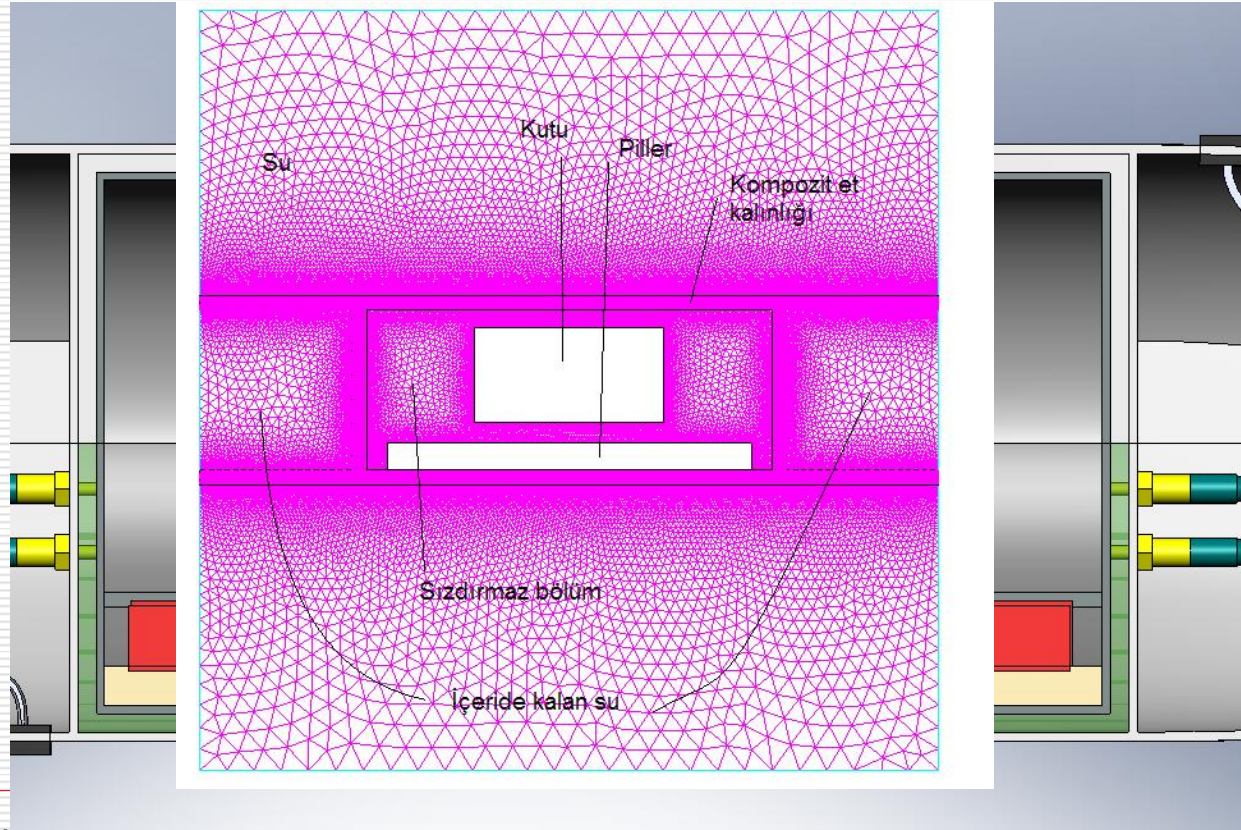
---

- Sıcaklığın en yüksek olduğu noktalar kutu üzerindedir.
- Azami sıcaklık 316 K ( $43^{\circ}\text{C}$ )' e yakınsamaktadır.
- Bu sıcaklık kabul edilir limitlerdedir. ( $<55^{\circ}\text{C}$ )



# Isıl Analizler

## □ İkinci Analiz:



# Isıl Analizler

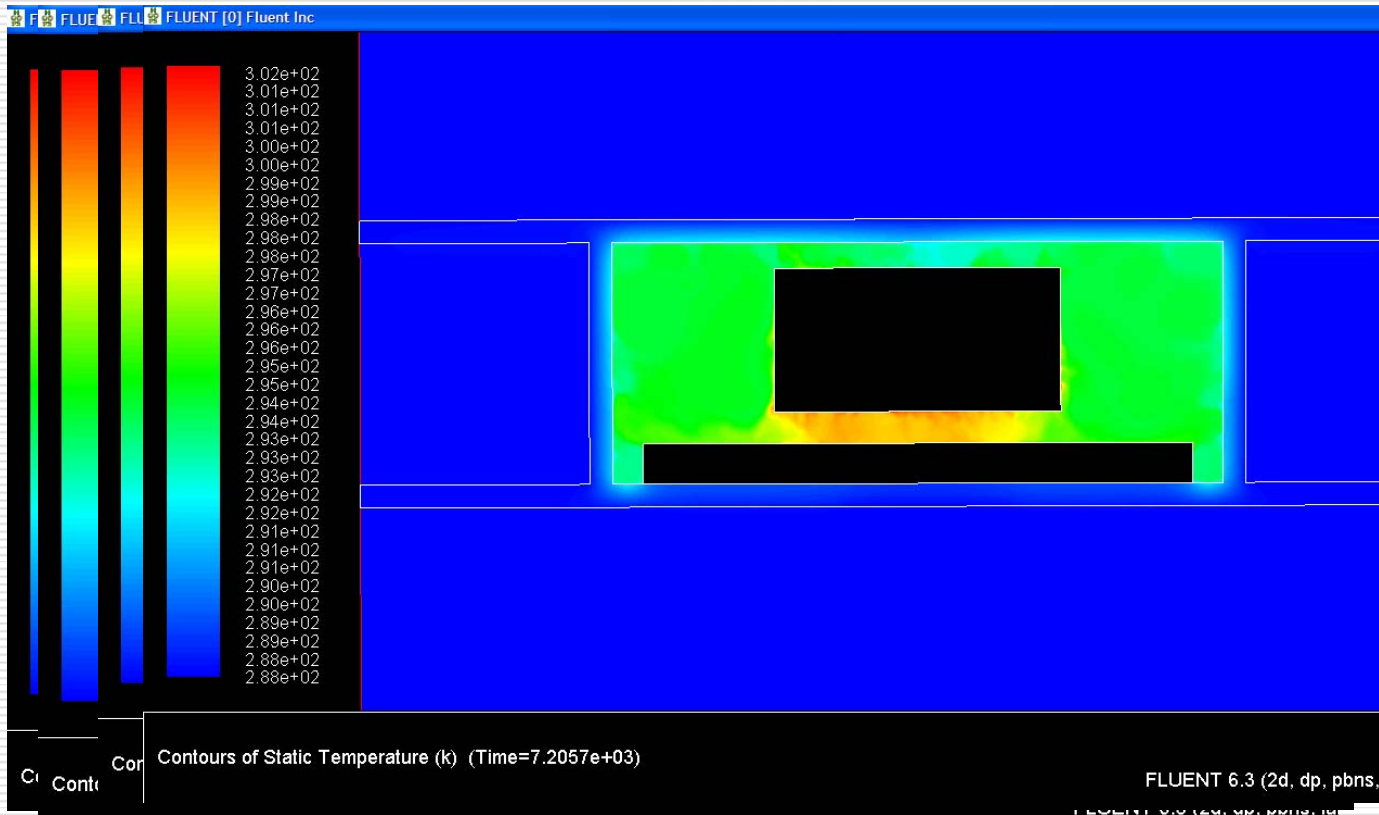
---

İkinci analiz:

- Gövdenin ilk sıcaklığı:  $35^{\circ}\text{C}$   
Suyun ilk sıcaklığı :  $15^{\circ}\text{C}$
- Aynı çözüm parametreleri

# Isıl Analizler

## Sıcaklık dağılımı:



# Isıl Analizler

---

## Sıcaklık Dağılımı:

- Azami sıcaklık:

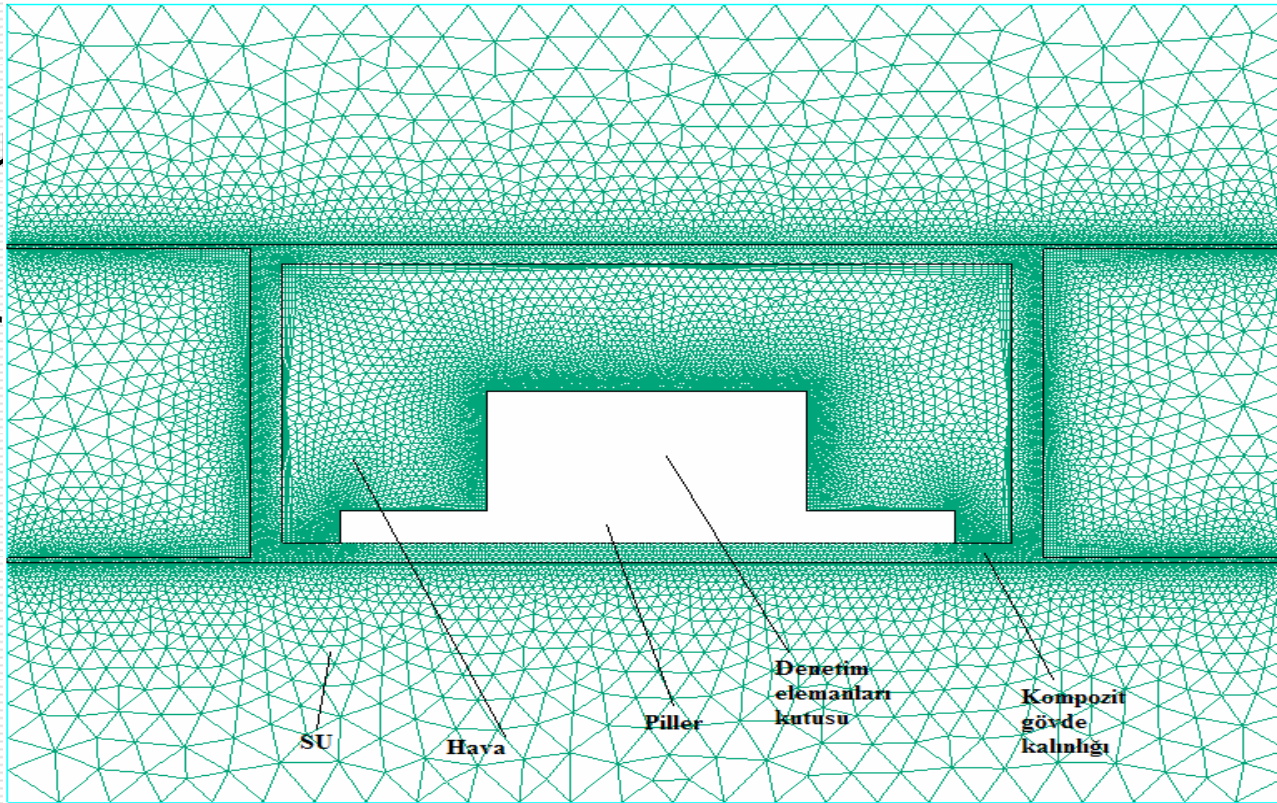
46°C ---> 45°C ---> 33°C ---> 30°C ---> 29°C

- Düşük su sıcaklığı azami sıcaklığı azaltmıştır.
- En kritik nokta piller ile kutu arasındaki bölümdür.

# Isıl Analizler

## Üçüncü Analiz:

- PİL
- SU
- PİL
- AR

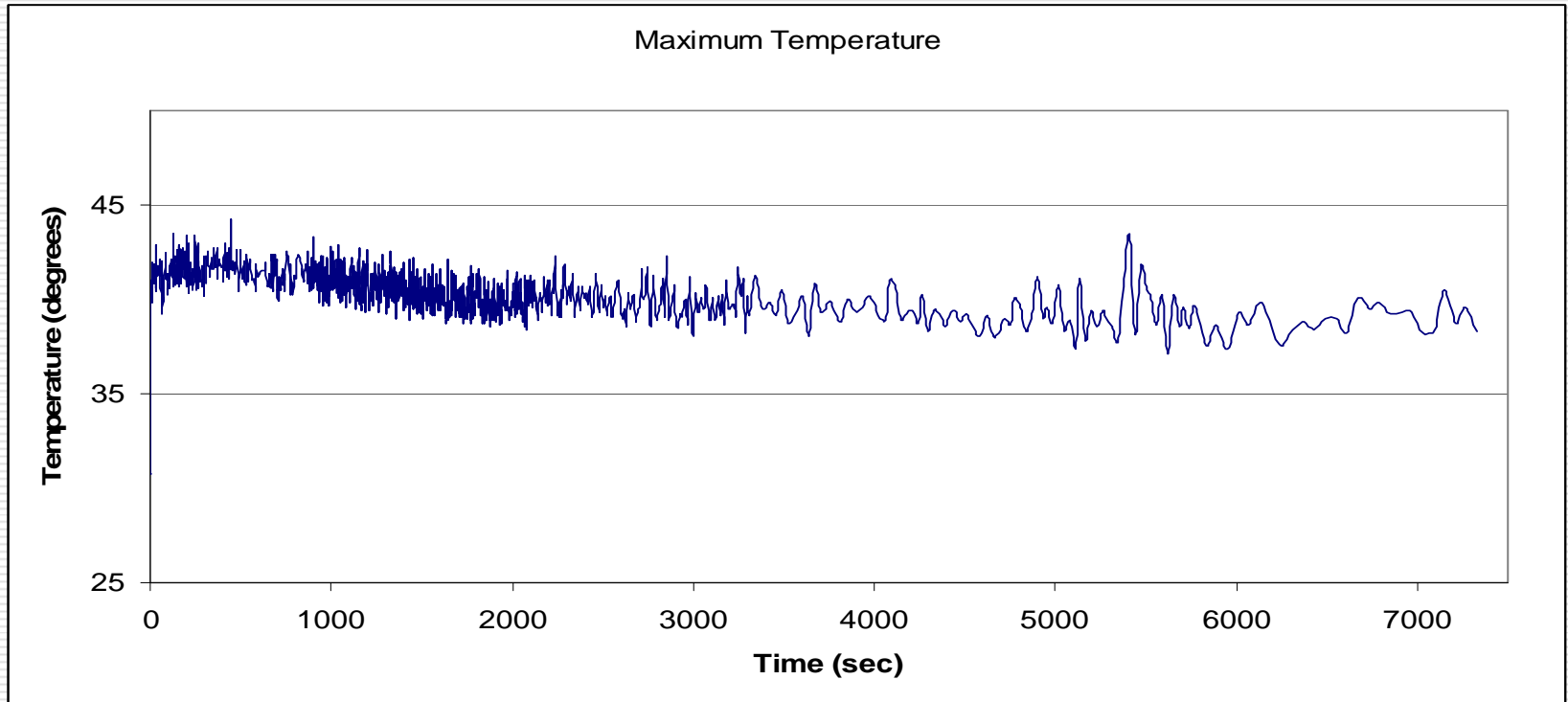


ştür.

# Isıl Analizler

---

## Sıcaklık dağılımı:



# Isıl Analizler

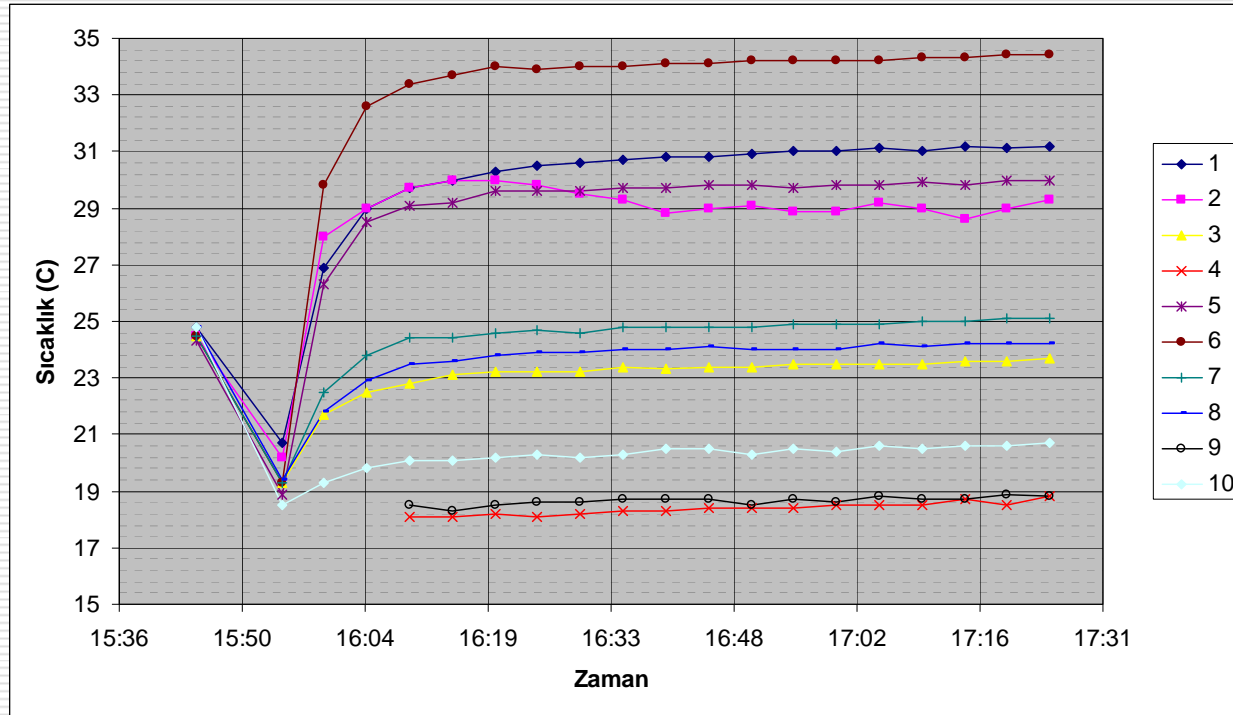
---

## Doğrulama Çalışmaları:

- Üretici firma (Barış Elektrik AŞ) tarafından test yapılmıştır.
- Sızdırmaz bölümün benzeri kompozit bir tüp ve 50 W ısı yayan bir ampul kullanılmıştır.
- Suyu batırılan düzenekten 2 saat boyunca sıcaklık değerleri alınmıştır.

# Isıl Analizler

## Doğrulama Çalışmaları:





# Isıl Analizler

---

## Değerlendirme:

- Analizler ve test koşullarında bazı farklılıklar vardır. (İlk sıcaklıklar, sınır koşullar)
- Elde edilen sıcaklık dağılımları benzer olup, azami sıcaklıklar çalışma koşullarını etkilemeyecek durumdadır.
- Nihai sonuç üretim sonrası testlerde ortaya çıkacaktır.

# Değerlendirmeler

---

- ❑ Hidrodinamik analizler sayesinde aracın şekline karar verilmiştir.
- ❑ Bu durumda beklenen karşı kuvvetler hesaplanmış ve motor seçimi yapılmıştır.
- ❑ Isıl analizler sayesinde sızdırmaz bölümde oluşabilecek aşırı ısınma durumu kontrol edilmiştir.
- ❑ Analizlerin geçerli olup olmadığı üretimden sonra yapılacak testlerde kontrol edilecektir.
- ❑ Daha gelişmiş araçların tasarım sürecinde daha detaylı ve daha çok durumu test eden analizler gerekecektir.

---

# TEŐEKKÜRLER

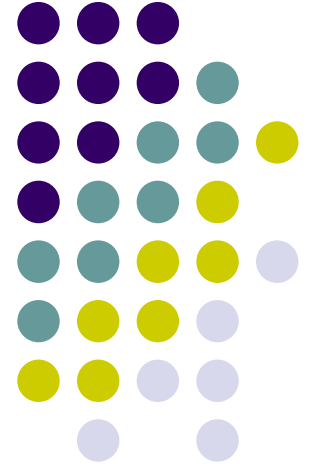
# ULİSAR

## Ulusal İnsansız Sualtı Aracı

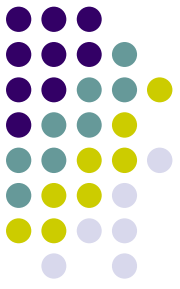
**Prof. Dr. M. Kemal LEBLEBİCİOĞLU**

Elektrik ve Elektronik Mühendisliği Bölümü

**ODTÜ**



# İçerik

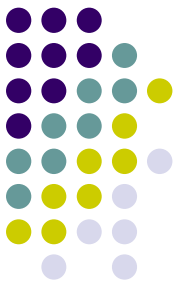


- Proje tanımı
- Mekanik tasarımı ve üretimi
- Donanım tasarımı ve üretimi
- Yazılım tasarımı ve geliştirilmesi
- Sualtında yapılan çalışmalardan örnekler
- Gelecek çalışmalar

# Proje Ekibi



- Prof. Dr. M. Kemal Leblebiciođlu (İnsansız Deniz Araçları Birimi, Biltir, ODTÜ) (Proje koordinatörü),
- Y. Doç. Dr. Afşar Saranlı (ODTÜ, Elektrik ve Elektronik Müh. Bölümü),
- Y. Doç. Dr. İlkey Ulusoy (ODTÜ, Elektrik ve Elektronik Müh. Bölümü),
- Yük. Müh. Hüseyin Yiğitler (ODTÜ, Elektrik ve Elektronik Müh. Bölümü),
- Yük. Müh. Emre Ege (ODTÜ, Elektrik ve Elektronik Müh. Bölümü),
- Üst. Kadir Isıyel (ODTÜ, Elektrik ve Elektronik Müh. Bölümü),
- Binbaşı İsmail Çalışkan (ODTÜ, Elektrik ve Elektronik Müh. Bölümü),
- Doç. Dr. Halit Oğuztüzün (ODTÜ, Bilgisayar Müh. Bölümü),
- Yük. Müh. Alper Kılıç (ODTÜ, Bilgisayar Müh. Bölümü),
- Haldun Mıdođlu (Yenel Ltd.),
- Fikret Şenel, (Barış Elektrik Endüstrisi A.Ş.),
- Prof. Dr. S. Kemal İder, (ODTÜ, Makina Müh. Bölümü),
- Prof. Dr. M. Haluk Aksel (ODTÜ, Makina Müh. Bölümü),
- Prof. Dr. Levend Parnas (ODTÜ, Makina Müh. Bölümü),
- Prof. Dr. Kemal Özgören (ODTÜ, Makina Müh. Bölümü),
- Yük. Müh. Necmettin Cevheri (ODTÜ, Makina Müh. Bölümü),
- Mine Cüneyitođlu (ODTÜ, Makina Müh. Bölümü),

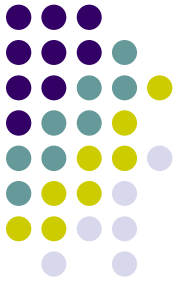


# Proje Grubu

Projede görev alan kurum ve kuruluşlar:

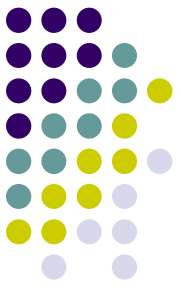
- ODTÜ Elektrik ve Elektronik Mühendisliği Bölümü
- ODTÜ Makina Mühendisliği Bölümü
- ODTÜ Bilgisayar Mühendisliği Bölümü
- YeneL LTD. (Elektronik kart üretimi)
- Barış Elektrik Endüstrisi A.Ş. (Gövde üretimi)
- ODTÜ Biltir

# İnsansız Sualtı Araçları



- Denizaltıları andıran sualtı robotları
- Otonom ve uzaktan yönetilen araçlar
- Bağlantı kablosuz/kablolu ve mürettebat gereksinimleri yok

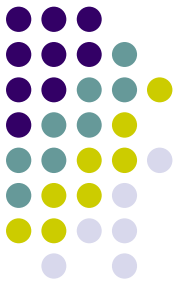




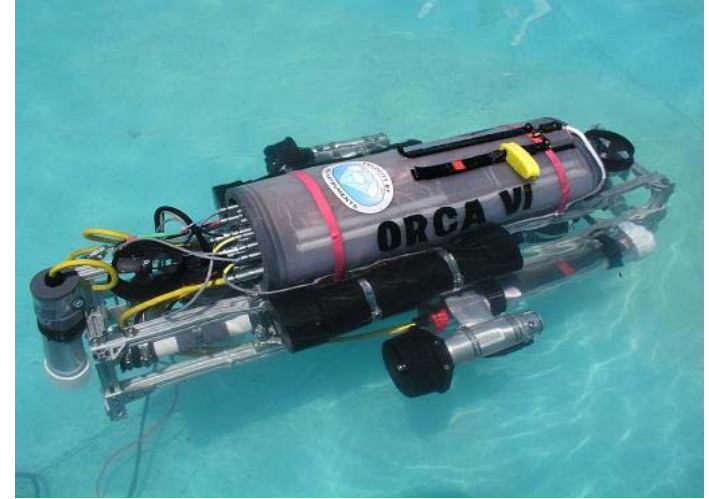
# Kullanım Alanları

- Mayın imha
- Keşif
- Sualtı kablolarının döşenmesi
- Sualtı hedeflerine ekipman taşınması
- Deniz ve okyanus suyu incelemeleri
- Petrol ve doğalgaz boruları gözlem ve bakımı
- Sualtı arkeoloji çalışmaları
- Deprem araştırmaları

# Dünyadaki Örnekleri

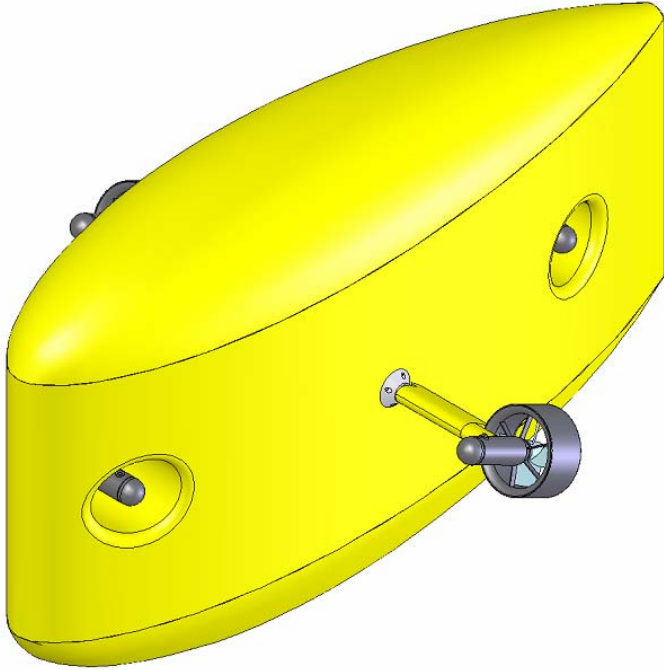
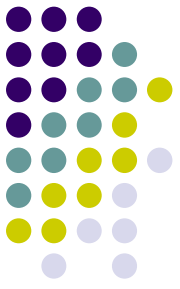


Cornell University (sea Monkey)

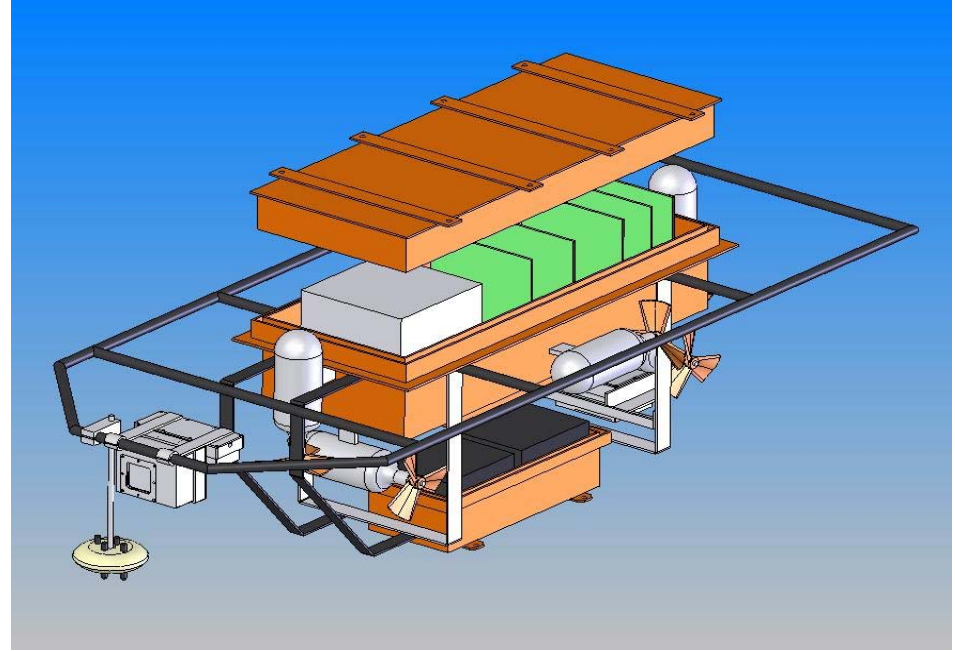


MIT AUV Tasarımı (ORCA VI)

# Dünyadaki Örnekleri

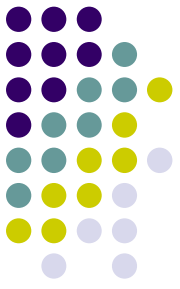


Odyssey IV (MIT)

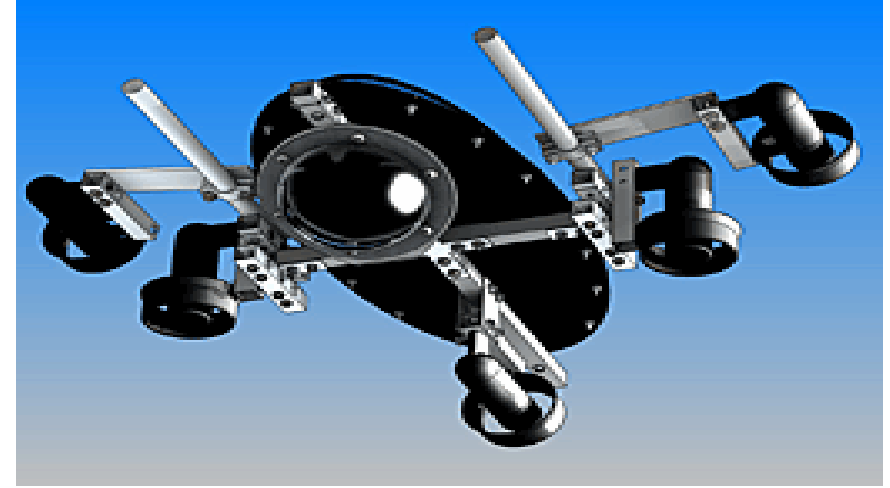


Quebec University Submarine

# Dünyadaki Örnekleri

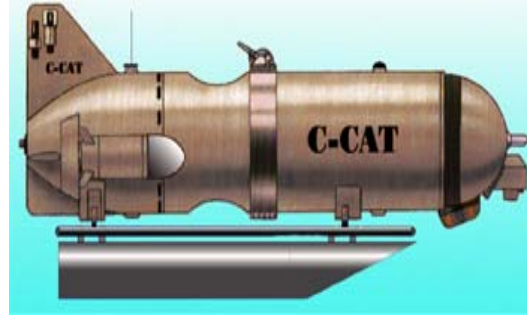
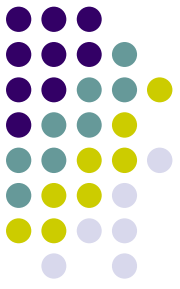


North Carolina State Uni. AUV Tasarımı  
(Seawolf II)

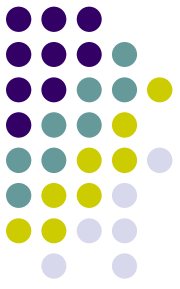


University of Toronto AUV Tasarımı

# Dünyadaki Örnekleri



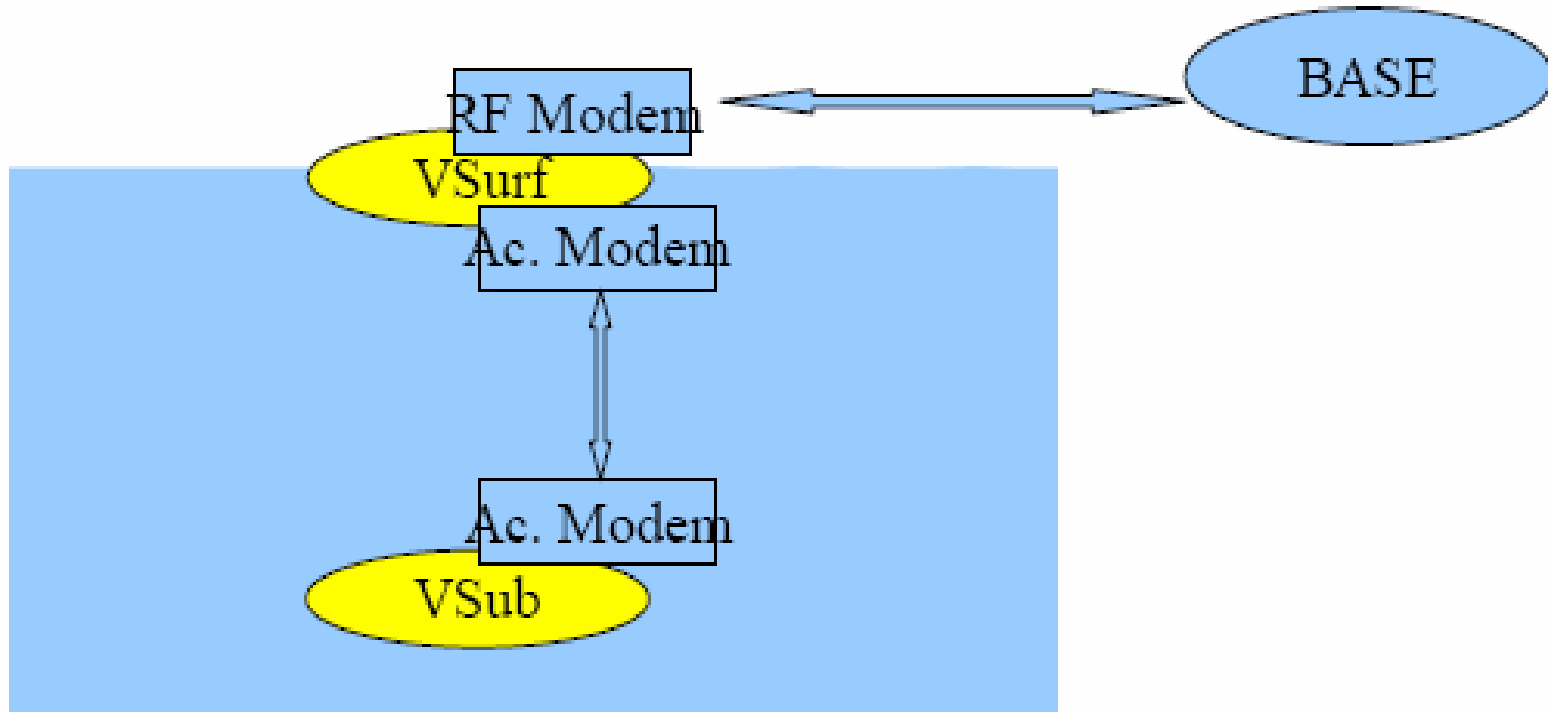
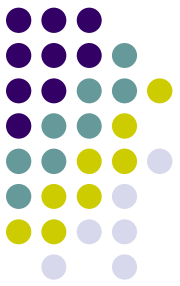
# ULİSAR



## Temel Fikir:

- Sualtı aracı ile beraber uyumlu olarak çalışan bir suüstü aracının varlığı,
- Aradaki haberleşme: Akustik modem kullanarak düşük rezolüsyonlu görüntü ve navigasyon bilgilerinin aktarımı,
- Suüstü haberleşmesi: RF kullanarak.

# ULİSAR



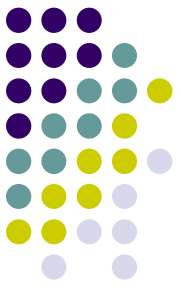
Önerilen insansız sualtı aracı sistemi



# Mekanik Tasarım ve Üretim

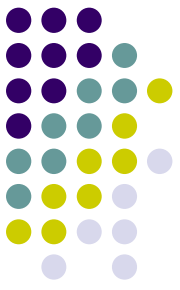
- Su sızdırmaz bölme tasarımı
- Dış gövde tasarımı
- İtki ve manevra sistemleri
- Denge ve yüzerlik
- Kablolama
- İmalat yöntemi





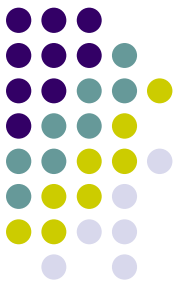
# Dış Gövde Tasarımı

- Dış gövde: aracı en dışta saran yapı
- Ekipmanın korunması, hidrodinamik yapı ve aracın şekli
- Denizaltıların aksine çeşitli şekillerde olabilirler



# Dış Gvde Tasarımı

- Dnme kararlılıđı aısından simetrik yapıda
- Ekipmanı kapsayacak lde uzun ve byk apta
- Srklenme ve srtnme kuvvetlerini azaltacak lde kısa ve kk apta
- Basınca dayanacak ve ađırlıđı ok arttırmayacak lde bir et kalınlıđında
- Elektronik ekipmanın fazla ısınmasını engelleyecek lde ısı transferine izin veren yapıda
- Modler yapıda

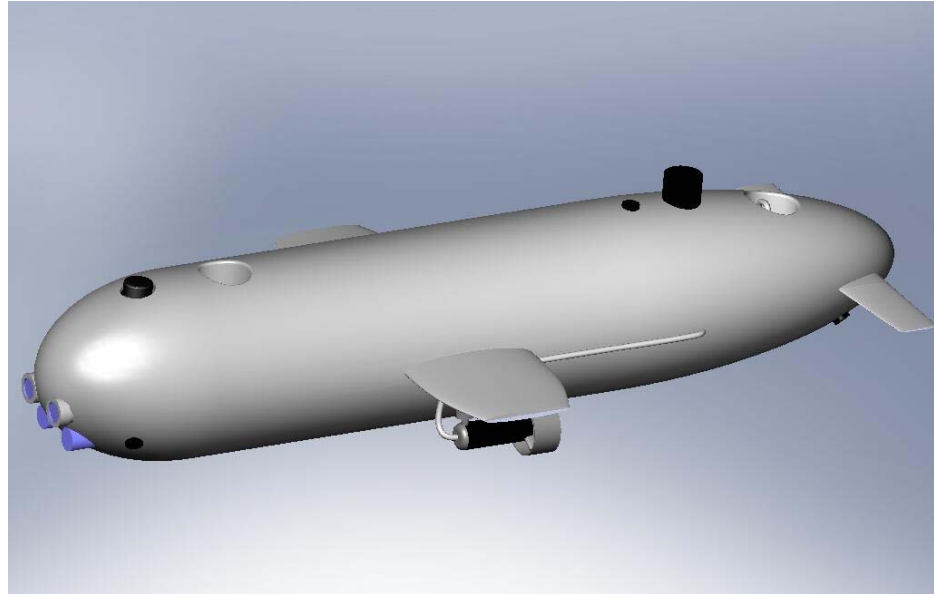


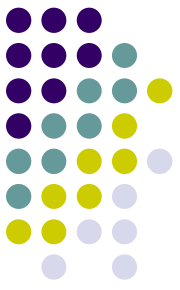
# Dış Gövde Tasarımı

- Düşük karşı kuvvet
- Üretim kolaylığı
- Geniş veritabanı



Silindirik yapı

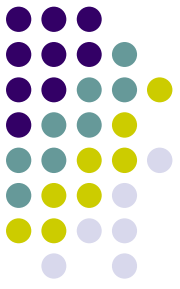




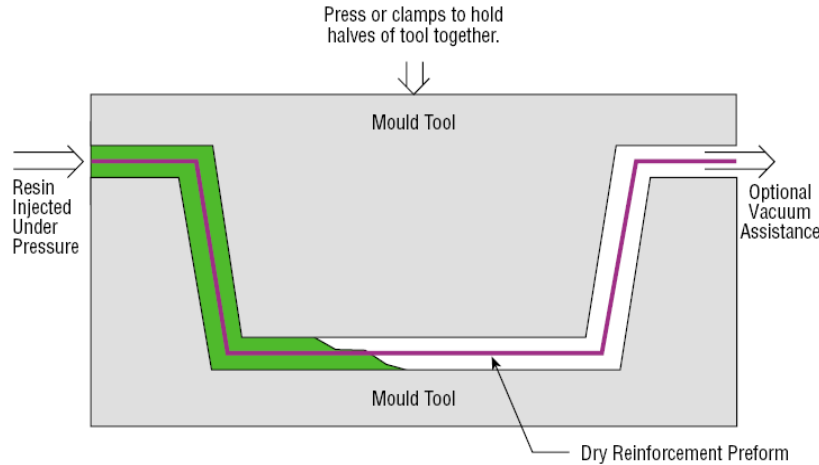
# Üretim Malzemesi

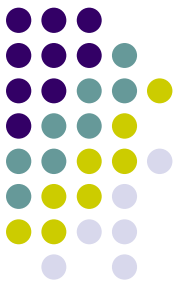
- Gövde kompozit malzemedен yapılmıştır:
  - Düşük yoğunluklarda yüksek mukavemet
  - Tasarım esnekliği
  - Kalıplama kolaylığı
  - Çeşitli yüzey uygulamalarına izin vermesi
  - Gövde içine metal parçalar yerleştirilebilmesi
  - Yapıştırma işlemleri uygulanabilmesi
- Malzeme: Cam dokuma kumaş – epoksi reçine

# Üretim Tekniği



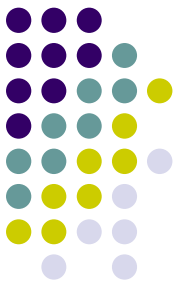
- Reçine Transfer Kalıplama (RTM)
  - Kalıp parçaları arasına elyaf yatırma
  - Basınç ve/veya vakum ile kontrollü reçine besleme
  - Isıl işlem ile polimerizasyon



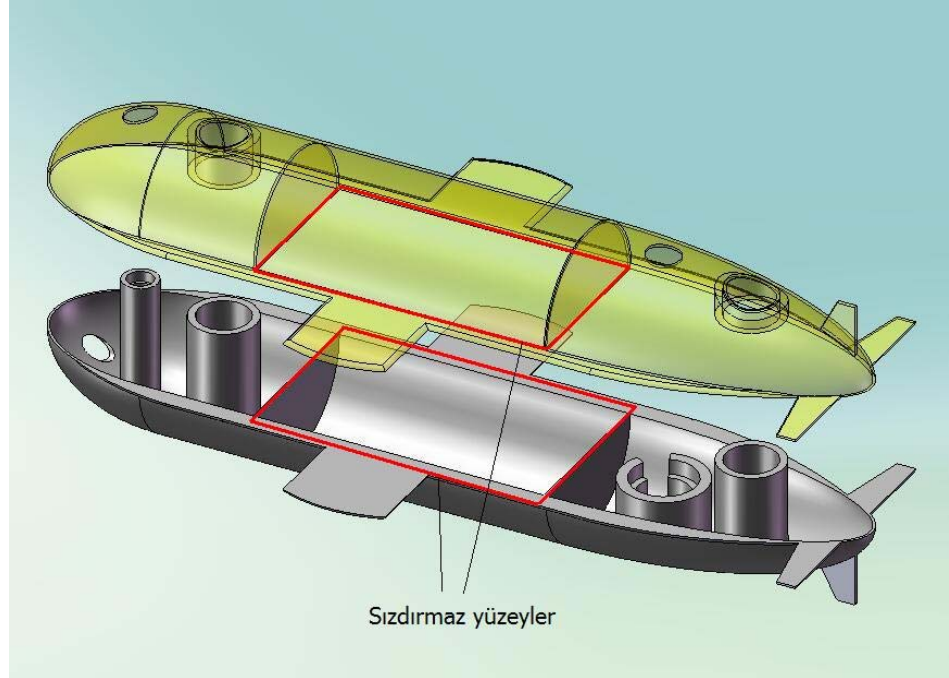


# Üretim Tekniđi - RTM

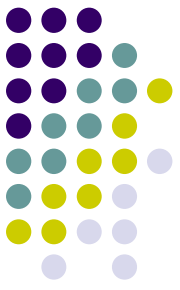
- Avantajlar:
  - - Yüksek elyaf-reçine oranı
  - - Düşük hava kabarcığı oranı
  - - Düzgün yüzey kalitesi
  - - Tasarım esnekliđi
- Dezavantajlar:
  - -Artan kalıplama masrafı
  - Büyük parçalara uygulama zorluđu



# Dış Gvde Tasarımı



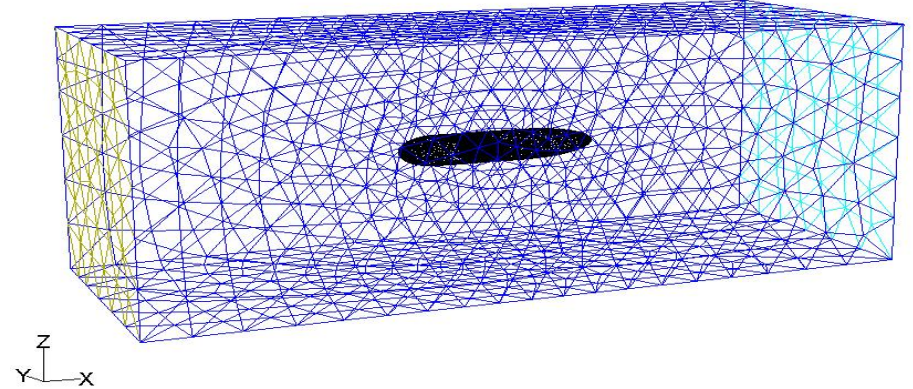
- Maliyeti dşrmek
- Yksek mukavim yapı elde etmek



# Dış Gövdenin Tasarımı

## Hidrodinamik analizler:

- Gambit-Fluent
- Basitleştirilmiş model
- 3B, zamandan bağımsız
- k- $\epsilon$  türbülans modeli
- Sıcaklık ve özellikler sabit



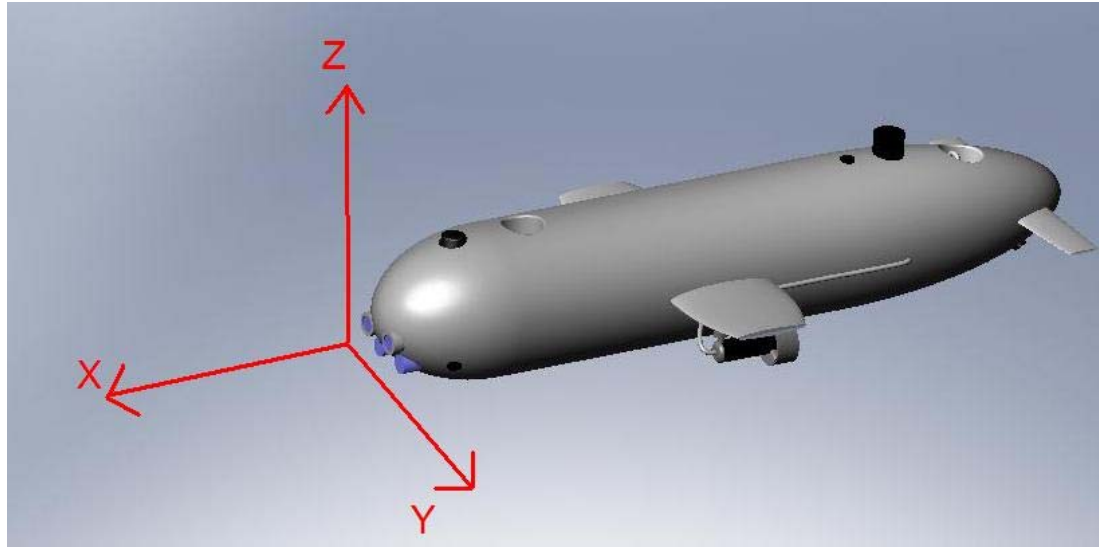
- Analizler sonucu araca etkiyen sürüklenme kuvvetleri elde edilmiş ve bu değerlerin seçilen motorların sağladığı itki değerlerini geçmediği görülmüştür.

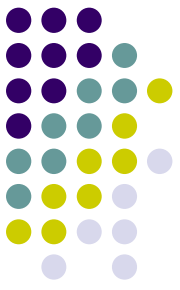




# İtki ve Manevra

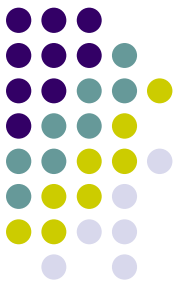
- İleri geri hareket (x eksenini boyunca)
- Yukarı aşağı hareket (z eksenini boyunca)
- Sağa sola dönüş (z eksenini etrafında)
- Yukarı aşağı dönüş (y eksenini etrafında)





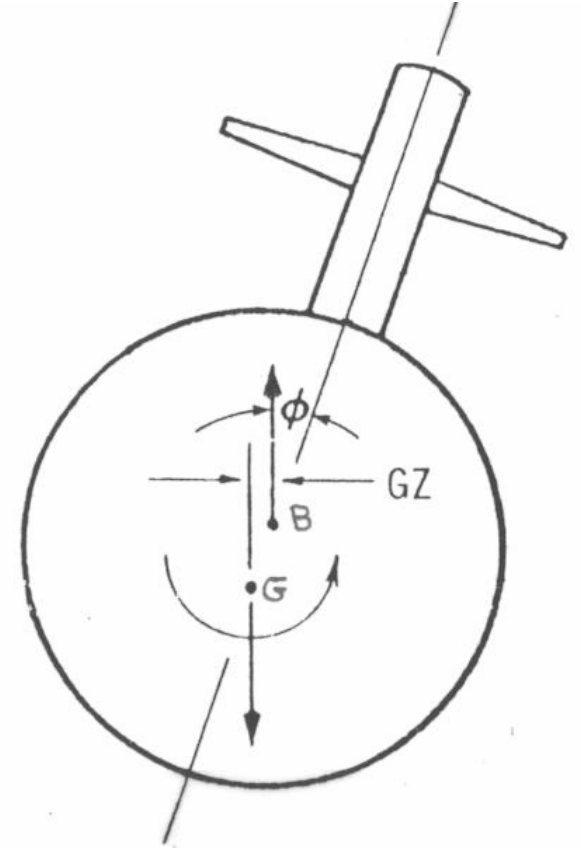
# Denge, Kararlılık ve Yüzerlik

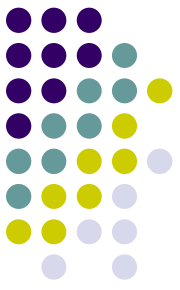
- Yüzerlik özelliğine sahip araçlar suda asılı kalabilmektedir. Bu sayede hareketsizken güç harcanması yoktur.
- Olası sistem problemlerinde aracı kaybetmemek için araç 0.5-1 kg arasında kaldırma kuvvetine sahip olacak şekilde tasarlanabilir.
- Denge ve kararlılık ise aracın istenilen şekilde durabilmesi ve hareket edebilmesi açısından önemlidir.
- Aracın denge ve kararlılığında ağırlık ve kaldırma kuvveti merkezlerinin konumları belirleyicidir.



# Denge, Kararlılık ve Yüzerlik

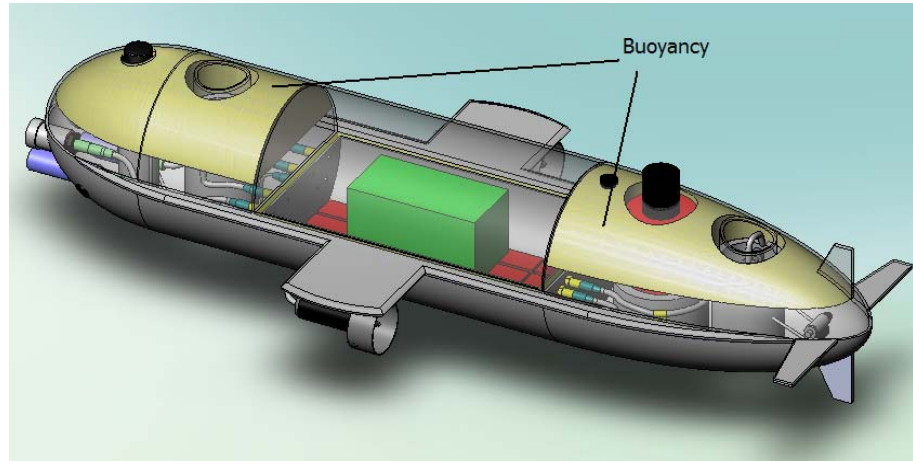
- Denge açısından iki merkezin x ve y konumları aynı olmalıdır.
- Kararlılık açısından ise kaldırma kuvveti merkezi, ağırlık merkezinin mümkün olduğunca üstünde olmalıdır
- İki merkez arasındaki bu mesafe aracın maruz kaldığı dönme etkilerine karşı düzeltici etkide bulunup kararlılık sağlamaktadır



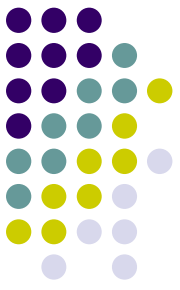


# Denge, Kararlılık ve Yüzerlik

- Araca belirtilen özellikleri sağlamak amacıyla alt bölümlere metal ağırlıklar, üst kısımlara ise yüksek yoğunluklu su emmeyen köpük eklenmektedir.



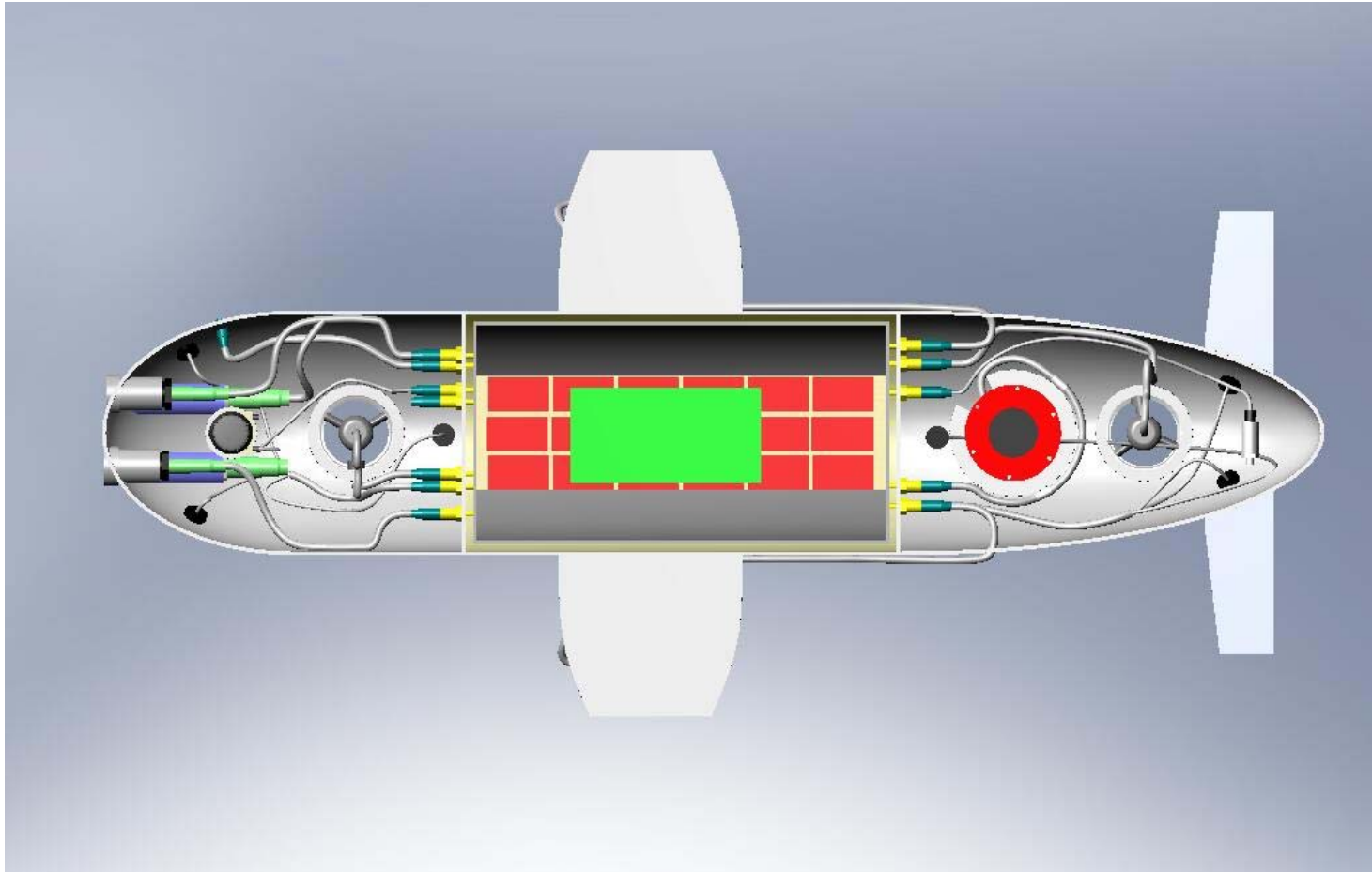
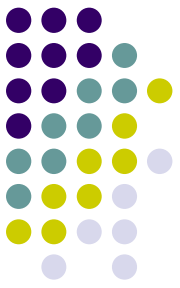
- Bu işlemler sonucu ağırlık ve kaldırma kuvveti merkezleri yaklaşık 3 cm kadar ayrılmışlardır.



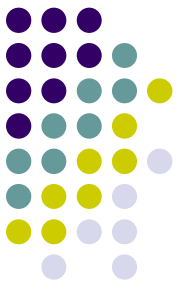
# Kablolama

- Denetim elemanları ve bataryalar su geçirmez bölümde saklanırken diğer cihazlar su geçirmez olduklarından su alan bölümde bulunmaktadır.
- Su geçirmez kablolar, aracın su geçirmez orta bölümü ile su alan bölümündeki cihazlar arasındaki bağlantıyı sağlamaktadır.
- Su geçirmez konektörler ise su geçirmez bölüm ile su alan bölümü ayıran duvardan geçişi sağlamaktadır.

# Kablolama



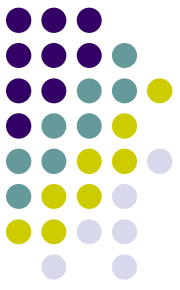
# Üretim Aşamalarından Görüntüler



## Model



# Üretim Aşamalarından Görüntüler

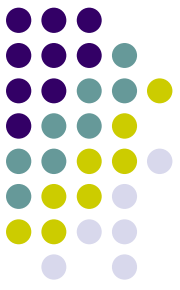


## Kalıp

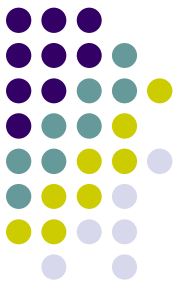




# Gövde Üretimi



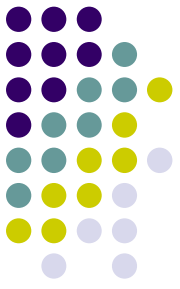
# Gövde Üretimi



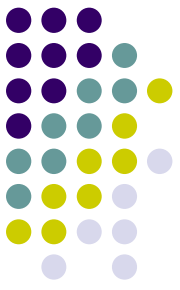
# Gövde Üretimi



# Donanım

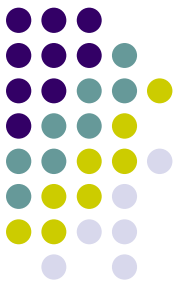


- Denizaltı aracında bulunan çevresel donanım birimleri:
  - 4 adet itici
  - 2 adet kamera ve 2 adet aydınlatma
  - 1 adet sonar
  - 1 adet akustik modem
  - 6 adet yaklaşma almacı(ultrasonik transducer)
  - 1 adet basınç almacı
  - 2 adet sıcaklık ölçer
  - 1 adet ataletsel ölçüm birimi(IMU)
  - 1 adet uzaktan kumandalı güç-açma kapama anahtarı



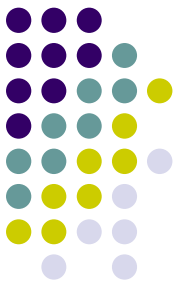
# Donanım

- Denizaltı aracını kontrol etmek amacı ile 2 adet PC/104+ form faktöründe bilgisayar ve bu bilgisayarlara bağlı değişik amaçlarda kartlar kullanılmıştır.
  - 2 adet bilgisayar(kontrol ve görüntü)
  - 1 adet MPEG-4 kodlayıcı kartı
  - 2 adet DC/DC çevirici
  - 1 adet PCMCIA-Ethernet kartı
  - 1 adet Compact Flash kartı



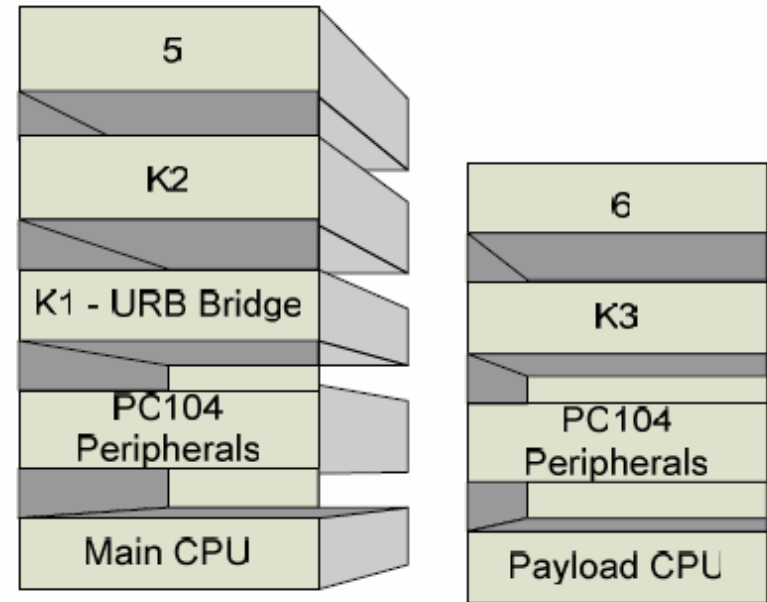
# Donanım

- Denizaltı aracındaki çevre birimlerle haberleşmek ve güç dağıtımını sağlamak amacıyla çeşitli kartlar tasarlanıp üretilmiştir.
  - URB Köprü Kartı
  - Güç – Kamera- Işık – Motor – Analog Almaç Kartı
  - Ultrasonik Almaç Düğümü
  - Denetim bilgisayarına bağlı yığın arayüz kartı
  - Faydalı yük bilgisayarına bağlı yığın arayüz kartı
  - Güç durumu ölçme ve güç dağıtım kartı

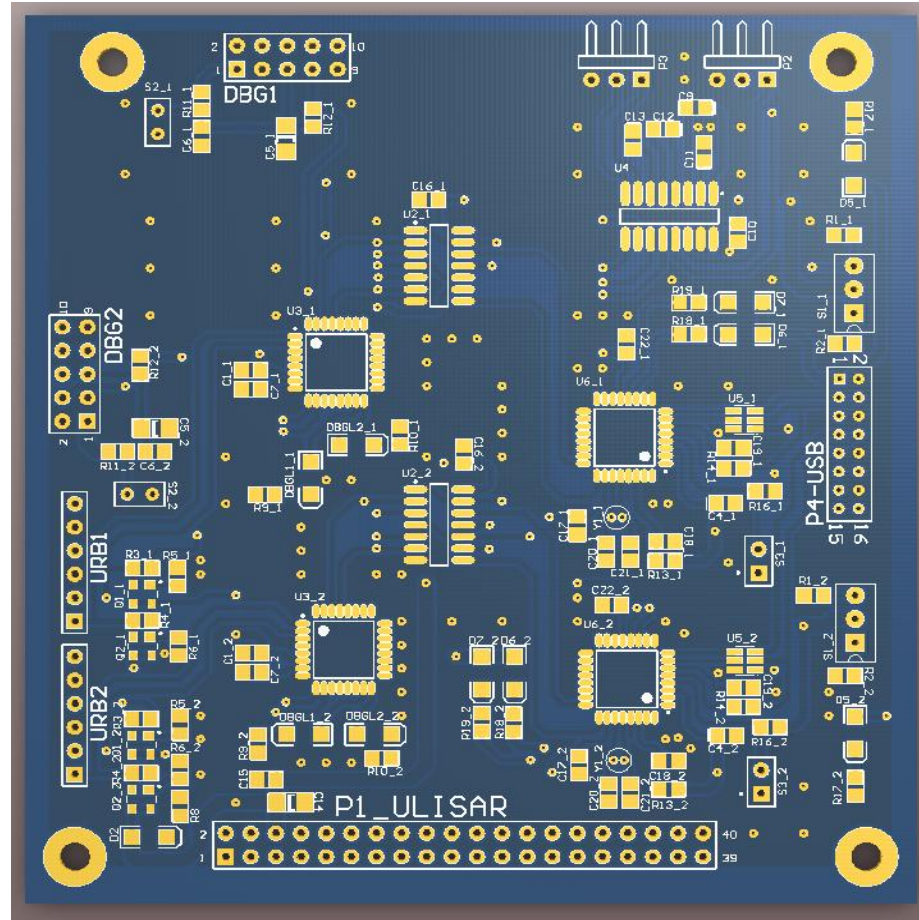


# Donanım

- Ön ultrasonik almaç seçme ve kamera-ışık denetleme kartı
- Arka ultrasonik almaç seçme kartı
- Yönelim denetleme kartı

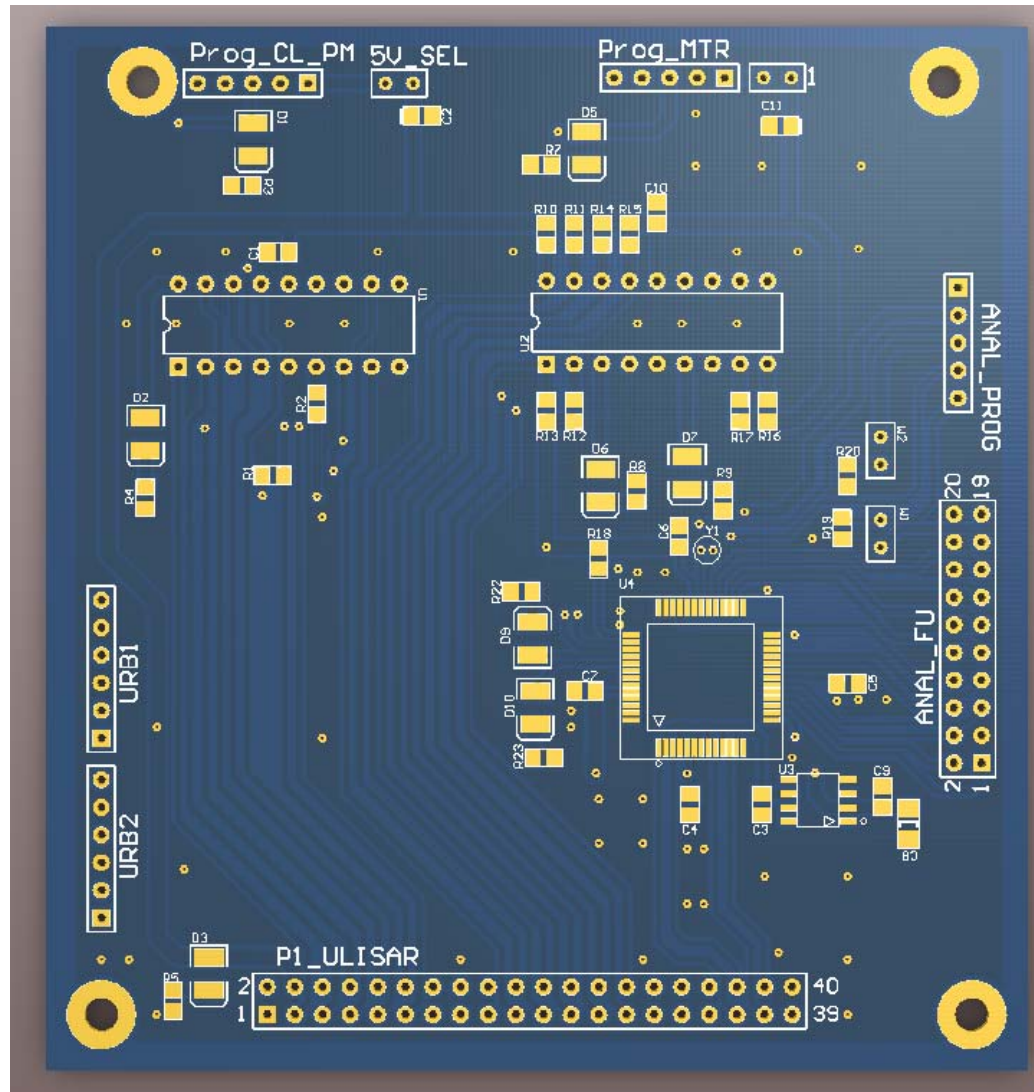
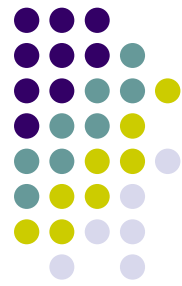


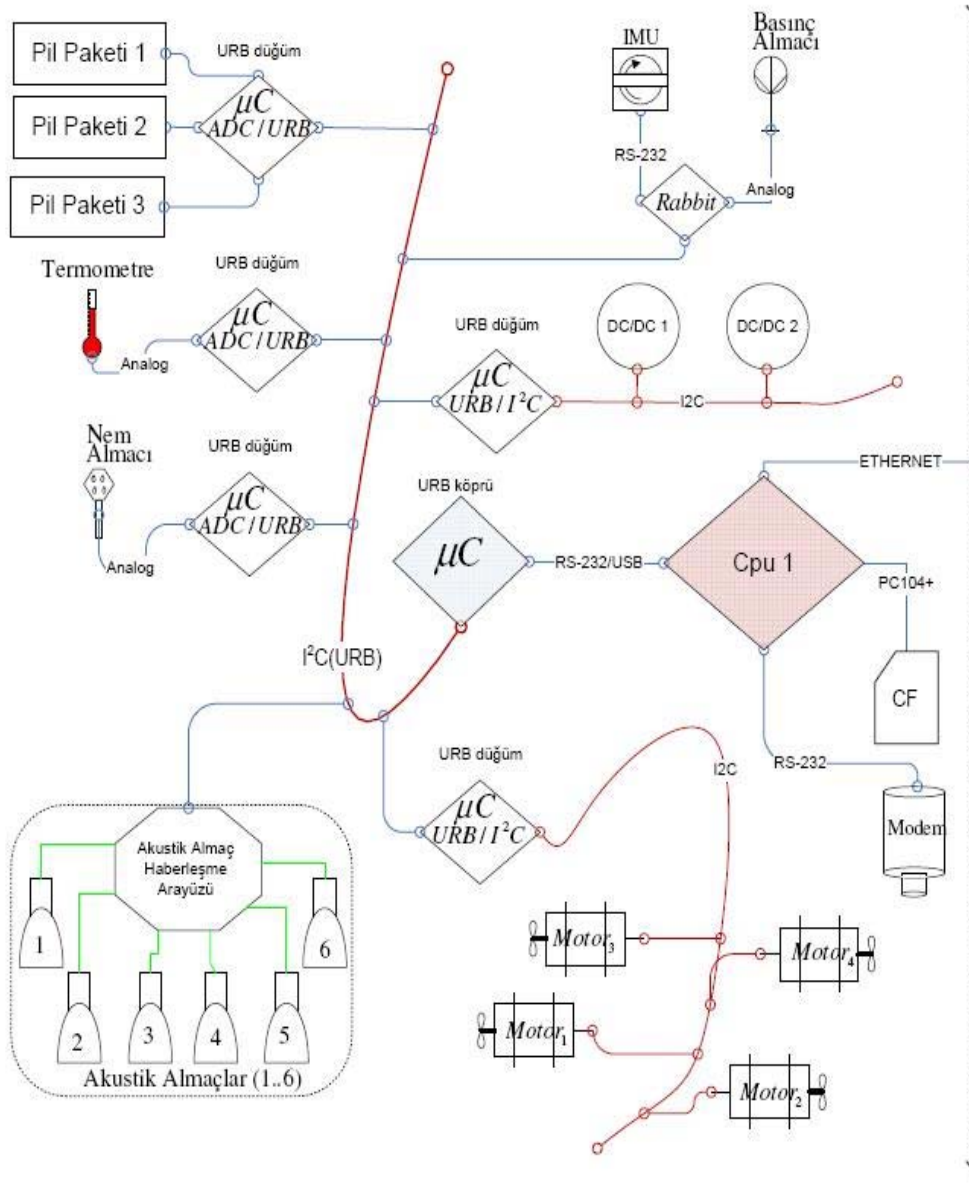
# URB Köprü Kartı



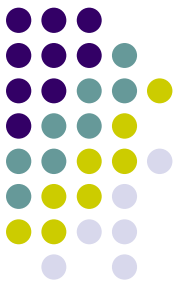


# Güç – Kamera- Işık – Motor – Analog Almaç Kartı



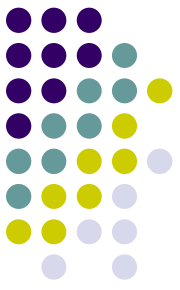


CPU 1: Ana denetim bilgisayarı  
 CPU 2: Faydalı yük denetim bilgisayarı  
 URB/I<sup>2</sup>C: URB'den I<sup>2</sup>C veri yolu çeviricisi  
 URB/ADC: Analogdan URB'ye çevirici  
 URB DÜĞÜM: URB Arayüz birimi (URB NODE)



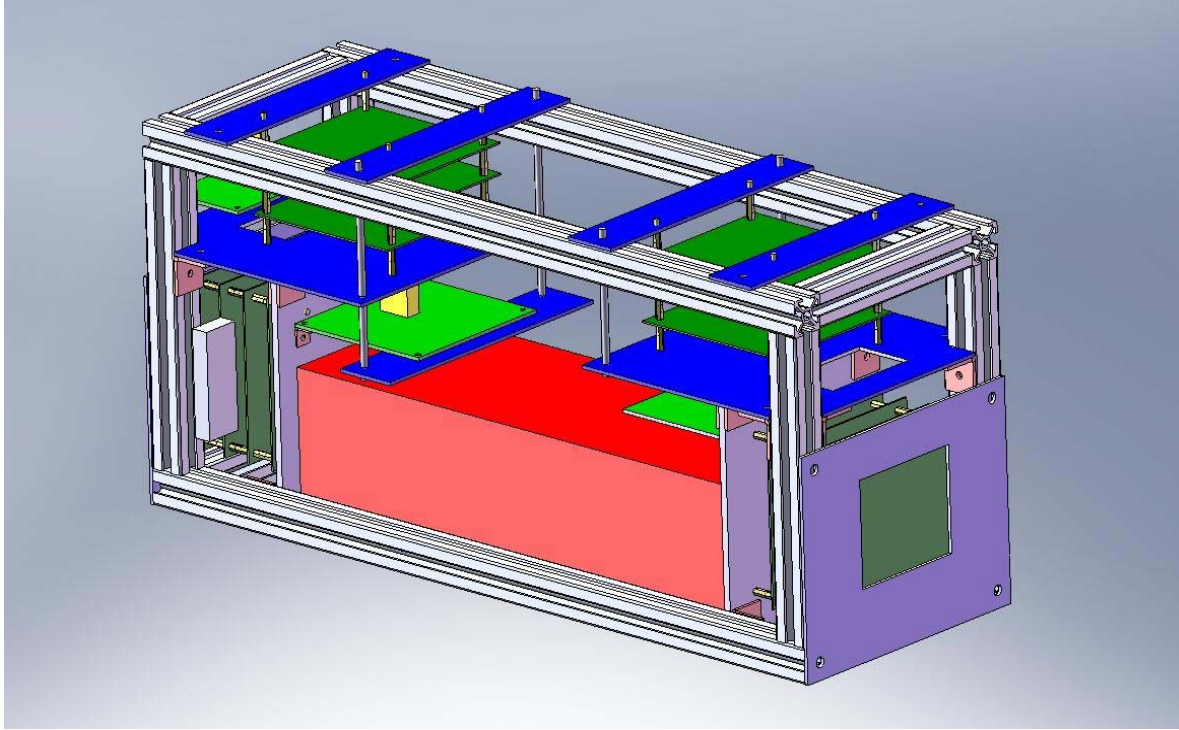
# Bataryalar

- Denizaltı aracı yaklaşık olarak sualtında 2 saat süreyle operasyon yapacak şekilde batarya yerleşimi yapılmıştır.
- Yüksek kapasitesi ve boyutlarının küçüklüğü bakımından LiPo bataryalar kullanılmıştır.
- 3 ayrı batarya grubu kullanılmıştır.
  - 4 hücreli 6 paralel batarya (14.4V, Elektronik)
  - 8 hücreli 4 paralel batarya (28.8V, 2 motor)
  - 8 hücreli 4 paralel batarya (28.8V, 2 motor)

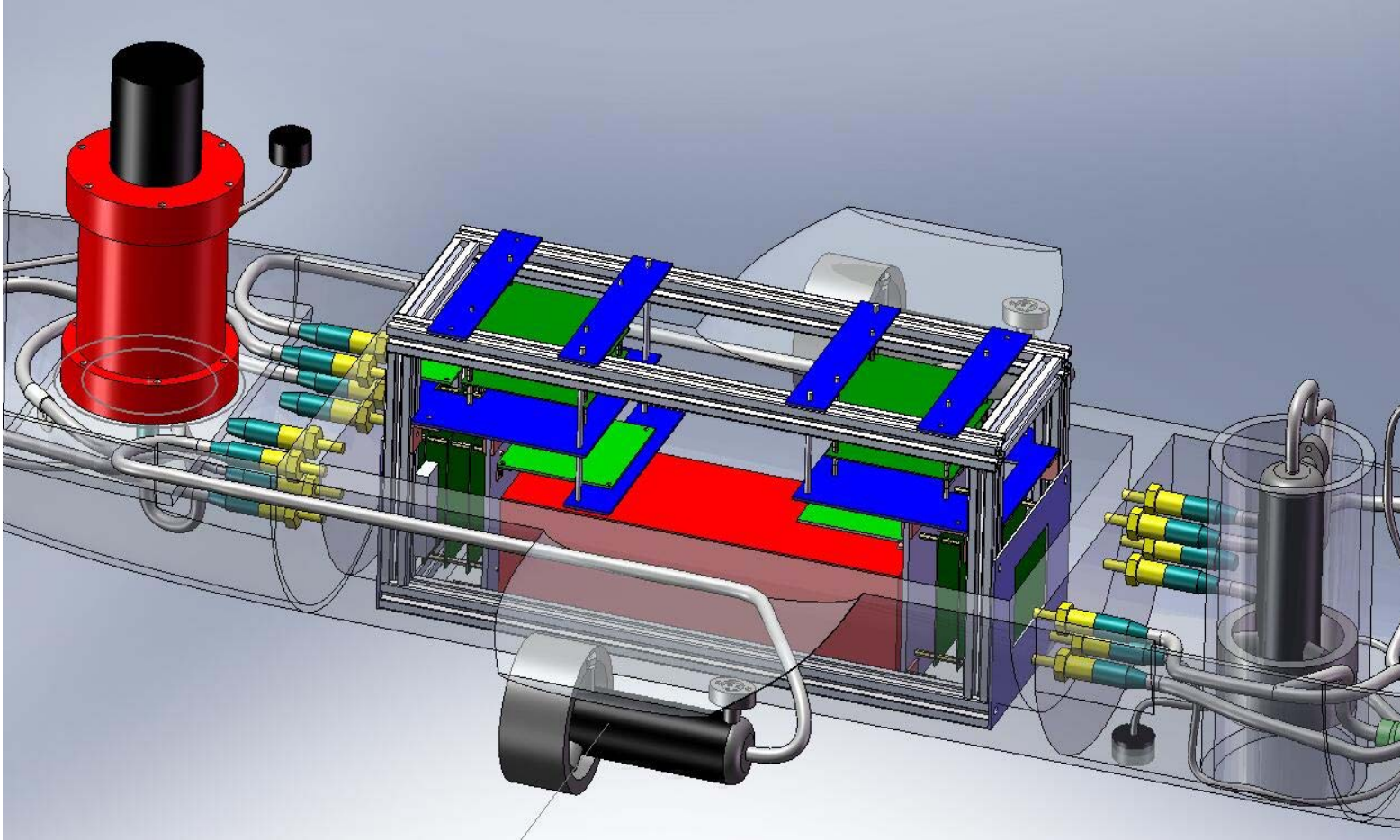
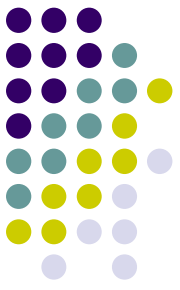


# Elektronik Kutu

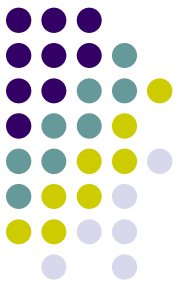
- Gvde ierisinde bulunan kartlar ve bataryaları sabit tutmakta kullanılmaktadır.



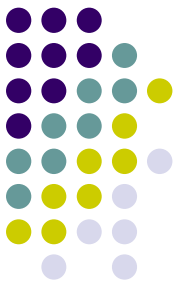
# İç Yerleşim



# Yazılım



- Denizaltı aracı için bir çok deęişik donanımda kořacak çeřitli yazılımlar geliştirilmiştir.
- Kontrol bilgisayarında QNX, görüntü (faydalı yük) bilgisayarında ise Linux işletim sistemi çalışmaktadır.
- Sistemde bulunan mikroişlemcilerde farklı amaçlarda yazılımlar geliştirilmiştir.
- Kamera ve dięer almaçlardan alınan veriler operatör bilgisayarına gönderilmekte, ve operatör bilgisayarından gelen emirler motorlara ve dięer birimlere aktarılmaktadır.



# Operatör Arayüzü

- Denizaltı aracı bir dizüstü bilgisayar aracılığı ve bir adet kumanda ile kontrol edilmektedir.
- Denizaltı aracı ve operatör bilgisayarı arasındaki haberleşme fiberoptik kablo ile sağlanmaktadır.

# Operatör Arayüzü



UlisarGUI

**Sensor Degerleri**

Basinc Sensörü: 0 Mesafe Sensörü 2 0  
Nem Sensörü: 0 Mesafe Sensörü 4 0  
Mesafe Sensörü 1 0 Mesafe Sensörü 5 0  
Mesafe Sensörü 2 0 Mesafe Sensörü 6 0

**Denizalti Durumu**

ACIL DURUM

**Sonar Kontrol Ekranı**

Sonar Aktif

**Motorlar**

0 On Motor M.D: 0  
0 Sol Motor M.D: 0  
0 Sag Motor M.D: 0  
0 Arka Motor M.D: 0

**Batarya Voltaj Seviyeleri**

1. Batarya Grubu 0 3. Batarya Grubu 0  
2. Batarya Grubu 0 4. Batarya Grubu 0

**Ag Durumu**

**IMU Bilgileri**

**Kamera ve Isik Kontrol**

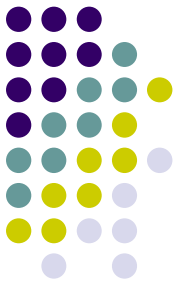
Isiklar Acik  
 Kamera 1 Acik  
 Kamera 2 Acik

Kamera 1 bulunamadi

Kamera 2 bulunamadi

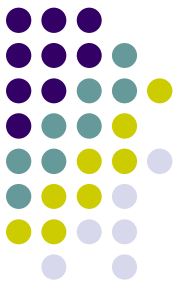


# Kumanda kolu

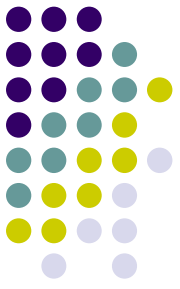


Numara	İşlev
1	Kamera 1 açma-kapama
2	Kamera 2 açma-kapama
3	Işıkları açma-kapama
4	Sonar açma-kapama
5	Denizaltı aracının yönlendirilemesi. İleri-geri,sağ,sol.
6	Denizaltı aracının yunuslama(pitch) kontrolü
7	Kullanılmıyor
8	Denizaltı aracının dik olarak yukarı çıkması
9	Denizaltı aracının dik olarak aşağı inmesi
10,11	10 ve 11 numaralı düğmelere aynı anda basıldığında Acil Durum konumuna geçirmek

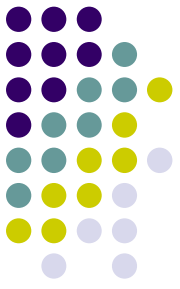
# Testlerden Görüntüler



# Testlerden Görüntüler

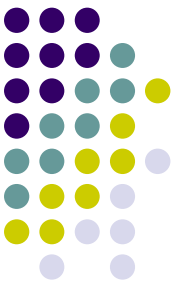


# Testlerden Görüntüler



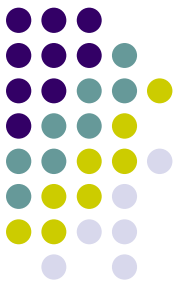
Yüzerlilik denge testi sualtı görüntüsü





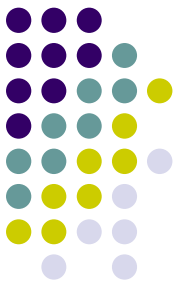
# Sonuç

- Mekanik tasarım ve üretim evresi projenin başarısı açısından son derece önemlidir.
- Bu evrede çeşitli tasarım öğeleri göz önünde bulundurulmuş ve mühendislik çalışmaları gerçekleştirilmiştir.
- Proje çalışmalarında makina, elektronik ve bilgisayar mühendislerinden oluşan bir disiplinlerarası proje grubu görev almıştır.
- Gereken noktalarda analizler ve deneyler gerçekleştirilmiştir.



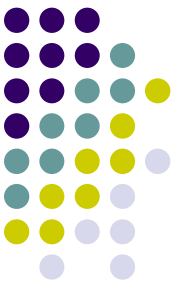
# Sonuç

- Oldukça yoğun çalışmalar daha gerekmektedir
- Navigasyon çalışmaları konusunda bilinenlerin ötesini zorlamak gerekiyor,
- Çalışmaların hiçbir zaman sonlandırılmaması gerekli,
- Bu konuda ülkemizin çok eksikliği var,
- Aynı zamanda bu konulara ihtiyaç çok fazla,
- Sadece askeri amaçlar değil sivil ihtiyaçlar düşünüldüğünde pazar çok büyük.



# Sonuç

- Projede elde edilen tecrübe ve bilgi birikiminin önümüzdeki yıllarda gerçekleşebilecek benzer projelerde faydalı olması beklenmektedir.
- Böyle bir çalışmanın sonunda görüntü işleme, sonar/kamera görüntülerinin birleştirilmesi, yol planlama, vs., gibi takip eden çalışmalar yapılabilir.



# *Dinlediğiniz için teşekkürler*



AUTOPILOT DESIGN AND GUIDANCE CONTROL OF ULISAR UUV (UNMANNED  
UNDERWATER VEHICLE)

KADİR ISİYEL

AUGUST 2007

AUTOPILOT DESIGN AND GUIDANCE CONTROL OF ULISAR UUV (UNMANNED  
UNDERWATER VEHICLE)

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

KADİR ISİYEL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF MASTER OF SCIENCE  
IN  
ELECTRICAL AND ELECTRONICS ENGINEERING

AUGUST 2007

Approval of the Graduate School of Natural and Applied Sciences

---

Prof. Dr. Canan Özgen  
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

---

Prof. Dr. İsmet Erkmen  
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

---

Prof. Dr. M. Kemal Leblebicioğlu  
Supervisor

Examining Committee Members

Prof. Dr. Mübeccel Demirekler	(METU,EE)	_____
Prof. Dr. M. Kemal Özgören	(METU,EE)	_____
Prof. Dr. M. Kemal Leblebicioğlu	(METU,ME)	_____
Dr. Afşar Saranlı	(METU,EE)	_____
Dr. İlkay Ulusoy	(METU,EE)	_____

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Last name : Kadir Isiyel

Signature :

# ABSTRACT

## AUTOPILOT DESIGN AND GUIDANCE CONTROL OF ULISAR UUV (UNMANNED UNDERWATER VEHICLE)

Isiyel, Kadir

M.Sc., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Kemal Leblebicioğlu

August 2007, 74 pages

Unmanned Underwater Vehicles (UUV) in open-seas are highly nonlinear with coupled motions, because of the complex interaction of the body with environment it is difficult to control them efficiently. At this point linearization is applied to system in order to use robust control methods for linear systems. Hence to overcome those disruptive effects and gain influence on system, a definite mathematical model which will compensate all disturbances is required for control. In this study first a mathematical model is formed wherein the linear and nonlinear hydrodynamic coefficients are calculated with strip theory by integrating two-dimensional damping formula according the shape of the vehicle.

After retrieving the mathematical model for the facility of control, system is decoupled into speed, steering and diving subsystems and consecutively PID (Proportional Derivative Integral), SMC (Sliding Mode Control) and LQR (Linear Quadratic Regulator)/LQG (Linear Quadratic Gaussian) control methods are applied to subsystems. Then as parameter estimation methods linear sequential estimation and genetic algorithm is implemented. The results of the control and estimation methods are also presented and compared with each other to assess the effective one. As a part of study, for on-line navigation an obstacle avoidance algorithm which avoids local optimum by surpassing the practicality of Boolean operators, solved with different optimization methods. Lastly a basic guidance algorithm is formed for our vehicle to follow the waypoints.

Owing to the fact that ULISAR UUV is still on construction phase, we were not able test our algorithms in a real system. But for the future work, it is certainly on plan to test these works researched in this thesis.

Keywords: mathematical modeling, PID, SMC, LQR/LQG, parameter estimation, guidance

# ÖZ

## ULİSAR (ÇOK MAKSATLI ULUSAL İNSANSIZ SU ALTI ARACI) İNSANSIZ SU ALTI ARACININ OTOPILOT DİZAYNI VE GÜDÜM KONTROLU

Isıyel, Kadir

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. M. Kemal Leblebicioğlu

Ağustos 2007, 74 sayfa

Açık deniz koşullarında bağımsız hareketleriyle insansız su altı araçları yüksek seviyede doğrusal olmayan özellik göstermelerinin yanında, araç gövdesinin çevresi ile karmaşık etkileşimlerde bulunması, aracın etkin olarak denetimini oldukça güçleştirir. Bu noktada doğrusal sistemler için kararlılığı ve sağlamlığı bilinen denetim yöntemlerinin kullanılabilmesi için sistem doğrusallaştırılır. Denetim açısından, tüm bu bozucu etkilerin üstesinden gelebilmek ve sistem üzerinde denetim sağlayabilmek için bu etkileri karşılayabilecek sağlıklı bir matematik model gerekmektedir. Bu çalışmada öncelikle bir matematiksel model oluşturulmuş ve modelde yer alan doğrusal ve doğrusal olmayan hidrodinamik katsayılar, aracın şekline bağlı olarak çıkarılan iki boyutlu sönüm formülünün integrali alınarak hesaplanmıştır.

Matematiksel modelin elde edilmesinden sonra denetimin kolaylığı açısından sistem; sürat, dönüş ve dalma alt sistemlerine ayrıştırılmış ve akabinde sisteme sırası ile PID, SMC ve LQR/LQG denetim sistemleri uygulanmıştır. Daha sonra katsayı kestirimi yöntemi olarak doğrusal ardışık kestirim ve genetik algoritma yöntemleri uygulanmıştır. En etkin yöntemi değerlendirmek üzere tüm bu çalışmalar sonunda elde edilen sonuçlar karşılaştırılmalı olarak gösterilmiştir. Çalışmanın bir parçası olarak, çevrimiçi seyirde kullanılmak üzere, Bool algortimalarını kullanarak yerel minimum noktalarından kaçınan engelden sakınma algorit-

maları çeşitli optimizasyon yöntemleri ile çözdürülmüştür. Son olarak, aracın belirlenen noktaları takip edebilmesi için temel bir güdüm algoritması oluşturulmuştur.

ULİSAR projesi hâlihazırda üretim aşamasında olduğundan dolayı oluşturulan algoritmalar gerçek bir sistemde denenememiştir. Ancak gelecekte bu tezde oluşturulan tüm algoritmaların gerçek sistem üzerinde denenmesi planlanmaktadır.

Anahtar Kelimeler: matematiksel modelleme, PID, SMC, LQR/LQG, katsayı kestirimi, güdüm



To My Mother

With Whom Life Had Meant A Lot

# ACKNOWLEDGMENTS

I would like to thank Prof. Dr. M. Kemal Leblebiciođlu for his guidance, patience and understanding in every phase of my study. He became more than a supervisor in every aspect. I would like to thank to Hüseyin Yiđitler for his invaluable advices and support by any means and to Emre Ege with other members of Robotics Laboratory. With his motivation and logistic support to laboratory I would like to thank to Dr. Afşar Saranlı.

I would like to express my respect to Lieutenant İsmail Çalıřkan for leading me in every way and both for sharing all his resources and his invaluable time. I can never forget his sacrifices in many that I still face.

Introducing the  $\text{\LaTeX} 2_{\epsilon}$  style to me I would like to thank Tolga İnan and with other members of Computer Vision Laboratory to Erdem Akagündüz for their helps.

Giving permission to me for getting a graduate program and letting me to have the best national education opportunities available, I want to impart all my gratitude to my sequent commanders in Turkish Naval Forces.

Finally I would like to thank my parents for their perpetual support.

# CONTENTS

ABSTRACT .....	iv
ÖZ .....	vi
ACKNOWLEDGMENTS .....	ix
CONTENTS .....	x
LIST OF TABLES .....	xiii
LIST OF FIGURES .....	xiv
LIST OF SYMBOLS .....	xvi

## CHAPTER

1 INTRODUCTION .....	1
1.1 Motivation and Introduction to Underwater Vehicles .....	1
1.2 Literature on Control and Guidance of UUVs .....	3
1.3 Organization .....	4
2 MATHEMATICAL MODELING .....	1
2.1 Introduction .....	1

2.2	Kinematics . . . . .	3
2.2.1	Coordinate Frames . . . . .	4
2.2.2	Euler Angles . . . . .	6
2.3	Rigid-Body Dynamics . . . . .	6
2.4	Added Mass . . . . .	8
2.5	Damping . . . . .	10
2.6	Gravitational and Buoyant Forces . . . . .	13
2.7	Hydrodynamic Coefficients . . . . .	14
2.8	Equations of Motion . . . . .	19
3	CONTROL PROCEDURES . . . . .	22
3.1	LQG Design . . . . .	46
3.2	Kalman Filter . . . . .	48
4	PARAMETER ESTIMATION . . . . .	55
5	OBSTACLE AVOIDANCE AND GUIDANCE . . . . .	62
6	CONCLUSION . . . . .	68
6.1	Summary of the Results . . . . .	68
 <b>APPENDICES</b>		
A	NONLINEAR EQUATIONS OF MOTION . . . . .	69
B	REYNOLDS NUMBER . . . . .	71

BIBLIOGRAPHY ..... 73

# LIST OF TABLES

2.1	Notation used for marine vehicles . . . . .	3
3.1	Continuous Time Kalman Filter . . . . .	51
4.1	Linear Sequential Estimation . . . . .	59
4.2	Parameter Estimation Comparison Table . . . . .	61

# LIST OF FIGURES

2.1	ULISAR UUV and main parts . . . . .	2
2.2	Earth-fixed and Body-fixed reference frames . . . . .	5
2.3	Prolate Ellipsoid and Dimensions . . . . .	15
2.4	Two-dimensional Added Mass Coefficients . . . . .	16
2.5	Cross-sectional view of our Vehicle . . . . .	18
3.1	Simulink PID Model . . . . .	24
3.2	Commanded and Real Output Velocities (m/s) for PID . . . . .	25
3.3	Model of Sliding Mode Control for Steer . . . . .	31
3.4	Input for Sliding Mode Control-Steer . . . . .	31
3.5	Steering for Sliding Mode Control-Steer . . . . .	32
3.6	Control Input for Optimal Control-Steer . . . . .	37
3.7	Steering for Optimal Control-Steer . . . . .	37
3.8	Model of Simulink PID Control for Steer . . . . .	38
3.9	Input for PID Control . . . . .	39
3.10	Steering for PID Control-Steer . . . . .	39
3.11	Model of Simulink PID Control for Depth . . . . .	40
3.12	Input Value of Simulink PID Control for Depth . . . . .	41
3.13	Desired Depth of PID for Depth Control . . . . .	41
3.14	Model of Sliding Mode Control for Depth . . . . .	44

3.15	Input for Sliding Mode Control-Depth . . . . .	44
3.16	Depth Control for Sliding Mode Control-Depth . . . . .	45
3.17	Control Input for Optimal Control-Depth . . . . .	46
3.18	Depth Control with Optimal Control-Depth . . . . .	47
3.19	LQG Design . . . . .	52
3.20	Simulink LQG Steer Subblock . . . . .	53
3.21	Simulink LQR Steer Subblock . . . . .	53
3.22	LQG Output . . . . .	54
5.1	Obstacle Avoidance for first case with start point [-4,-4] . . . . .	63
5.2	Obstacle Avoidance for second case with start point [-4,-4] . . . . .	64
5.3	Obstacle Avoidance for third case with start point [-7,-8] . . . . .	65
5.4	Obstacle Avoidance for third case with start point [-2,-8] . . . . .	66
5.5	Obstacle Avoidance for fifth case with start point [-8,-8] . . . . .	67
B.1	The drag coefficient for a sphere. . . . .	72



# LIST OF SYMBOLS

$\mathbf{x}$	A 2D homogeneous point ( $3 \times 1$ )
$\mathbf{X}$	A 3D homogeneous point ( $4 \times 1$ )
$\mathbf{l}$	A 2D line ( $3 \times 1$ )
$\mathbf{■}$	A 3D plane ( $4 \times 1$ )
$\mathbf{C}$	A conic in 2D ( $3 \times 3$ )
$\mathbf{C}^*$	A dual conic in 2D ( $3 \times 3$ )
$\mathbf{Q}$	A quadric in 3D ( $4 \times 4$ )
$\mathbf{Q}^*$	A dual quadric in 3D ( $4 \times 4$ )
$\mathbf{l}_\infty$	The Line at Infinity ( $3 \times 1$ )
$\mathbf{■}_\infty$	The Plane at Infinity ( $4 \times 1$ )
$\Omega_\infty$	The Absolute Conic
$\mathbf{Q}_\infty^*$	The Absolute Dual Quadric ( $4 \times 4$ )
$\omega$	Image of the Absolute Conic ( $3 \times 3$ )
$\omega^*$	Dual Image of the Absolute Conic ( $3 \times 3$ )
$\mathbf{F}$	Fundamental Matrix ( $3 \times 3$ )
$\mathbf{H}$	Homography Matrix ( $3 \times 3$ )
$\mathbf{P}$	Camera Projection Matrix ( $3 \times 4$ )
$\mathbf{K}$	Calibration Matrix ( $3 \times 3$ )
$f_x$	Focal length in $x$ direction
$f_y$	Focal length in $y$ direction
$s$	Skew
$u_0$	$x$ coordinate of the principal point
$v_0$	$y$ coordinate of the principal point
$\kappa_1$	First radial distortion coefficient
$\kappa_2$	Second radial distortion coefficient
$\mathbf{A}^{-T}$	Transposed inverse of the matrix $\mathbf{A}$

# CHAPTER 1

## INTRODUCTION

### 1.1 Motivation and Introduction to Underwater Vehicles

In this thesis, our goal is to simulate the control and guidance procedure of ULISAR unmanned underwater vehicle.

In this chapter a brief information about underwater vehicles, their applications and importance and lastly our objectives to be achieved for this thesis are mentioned.

Underwater vehicles are classified in two main groups as manned underwater vehicles (MUVs) and unmanned underwater vehicles (UUVs). Today, because of high operational costs, operator weariness and the painful experiences in history, which gave rise to the improvements in the UUVs, employment of the MUVs are highly limited. From operational aspects UUVs are grouped in two main categories as remotely operated vehicles (ROVS) and autonomous underwater vehicles (AUVs). While ROVs give chance of intervention to the operator in any phase of operation, with their highly operational costs and their hulk values in case of lost, in recent years they have been disfavored. Nowadays, research on fully autonomous systems has increased and lessened the necessity of a human operator. In the 1990s, about 30 new AUVs are built worldwide [19]. A self-contained, intelligent and self-decisive AUV is the goal for the current underwater vehicle research .

ULISAR is a TUBITAK (Türkiye Bilimsel ve Teknolojik Araştırma Kurumu) supported project . Being a small UUV compared with their coevals, ULISAR will be a novel ROV vehicle. ROVs are small, efficient and tethered vehicles to collect underwater data and fulfill onboard commanded operations. Online communication with vehicle is achieved by generally fiber optic cable because of its variable huge bandwidths. Instead of utilizing cables, a

few disadvantages of which can be stated as drag in water, risk of disjunction and reduction in speed, ULISAR's communication will be maintained via acoustic link which will be satisfied by acoustic modems. Project will comprise two vehicles, one on the sea surface maintaining the RF (Radio Frequency) communication with control center and the other one in the sea which will gather underwater information and achieve main task. The surface vehicle will just relay the information she takes from bottom vehicle and vice versa.

For many year ROVs proved their efficiency in many situations like underwater pipe inspections, rescuing goods from sunken, oceanographic data collection and different mine counter-measure operations.

Constructing an UUV is an exhaustive and time-consuming job where most of the tests are executed in laboratory environment. Testing the parts and works done in the the project in water environment is not available and logical because of the risk of losing the valuable equipments and most important of all risk of damage to the healths of project personnel hence an effective and not expensive choice to be implemented for simulation of system for tests. Simulation with computer aid is a practical and quick method of finding failures that may be confronted at sea. A good working simulator needs an actual model of the system which will then imitate the outputs as similar as the real outputs when the same inputs are applied. The coefficients of the system have to be accurately found in order to simulate the system efficiently otherwise simulation will fail and unpredicted situations may occur at real system tests.

In this study, we started the simulation first by forming the mathematical model. We generated the model forming the kinematic and rigid-body dynamics. Then we found the linear and nonlinear hydrodynamic coefficients by strip theory and boundary integral method as stated in [5] and [12]. Forming a mathematical model, exploiting the fact that our vehicle is not a fast varying system we linearized it around an equilibrium point, which is a one degree of freedom (DOF) known motion in x-coordinate in our case. Linearized system should be controlled more easily where most of the robust control methods are for linear systems. We started control design first with PID because of its simplicity and applicability to the most of the linear systems. We then tried SMC, trusting the compensating success of it. Lastly we designed a controller using LQR/LQG methods. In last phase we used the benefit of separation principle, where we first designed a regulator and an observer using Kalman Filter, and put them together to form a compensator for the plant [15].

For the efficiency of the simulation the unknown parameters of the model outranks any other effects therefore we should have the nearly correct values of these parameters hence we ran

two parameter estimation algorithms. We started with linear sequential estimation method and tried to estimate the coefficients then we attempt to find them by running a genetic algorithm.

As the next step underwater vehicles in real time operations need obstacle avoidance algorithms. Thinking the needs we worked on a problem such that the generated path would avoid the vehicle to end up in a local minimum, in a narrow path between two rocks where the vehicle can not pass successfully. Using the Boolean algorithms, the local minimum point is avoided [16]. Sometimes the new path may be far from the optimum path but when the safety conditions are prior to any other function.

Last of all our vehicle needs a guidance system where we used a waypoint guidance system based on line of sight. The details about the guidance will be given on the related chapter.

## **1.2 Literature on Control and Guidance of UUVs**

The main factors that make control process difficult can be stated as: highly nonlinear, time-varying dynamic behaviour of the vehicle, uncertainties in hydrodynamic coefficients, disturbances by ocean vehicles (especially high frequency waves near surface), unpredicted underwater currents, for our case changes in the gravity and buoyancy. Considering the difficulty to fine-tune the control gains during operations, it will be advantageous to have a control system that will tune itself if the control performance decreases [19].

Different control techniques have been applied to underwater vehicles in recent years. Jalving used classical PID control methods for Norwegian Defence Research Establishment-AUV. He decoupled the system into three lightly interacting subsystems and designed three autopilots for steering, diving and speed control. The design of the each controller was based on PID techniques [7]. Yoerger and Slotine designed a sliding mode controller for an underwater vehicle. In their study they neglected cross-coupling terms and investigated the uncertainties of the hydrodynamic coefficients [18]. Meanwhile, preferring SMC for controlling their vehicle, Healey and Lienard were the ones who decoupled the system into three subsystems first time. Each autopilot was again designed using SMC with exploiting the advantage and ease of decoupled system [6]. Nakamura and Savant urged a nonlinear tracking control of an AUV pondering kinematic motion [19], [11]. They achieved the control by thinking the nonholonomic nature of the system without considering the dynamics of the system. Cristi, Papoulias and Healey designed a robust adaptive SMC such that in the presence of dynamical uncertainties, controllers can adjust to the changing dynamics and

operating conditions [4]. A hybrid adaptive controller using both continuous and discrete operations was mentioned by Tabaii et al [14].

In guidance of UUVs not so many studies have been performed. Healey et al. worked on the waypoint guidance by line of sight principle where the guidance is accomplished by a heading command to the vehicle's steering system to approach the line of sight between the present position of the vehicle and the waypoint to be reached. In missile guidance this is related to "proportional navigation" [6]. Caccia et al. introduced a PI-type task functions which enable a Lyapunov-based guidance system to compensate the effects of both unmodelled interactions between vehicle and environment and of bias in velocity measurements [2].

### **1.3 Organization**

The organization of the thesis is as follows:

- Chapter 1 mentions what is planned to achieve with this thesis and some studies done by other authors.
- Chapter 2 gives some mathematical formulation and transformations forming the mathematical model.
- Chapter 3 informs about the control methods used to control our vehicle. Comparison between the methods are also mentioned.
- Chapter 4 shows the efforts in estimating the linear hydrodynamic coefficients. Linear sequential estimation method and genetic algorithm are the methods used for estimation.
- Chapter 5 acquaints about guidance system for underwater vehicle and obstacle avoidance method.
- Chapter 6 shows a summary of the obtained results in this study. Then a discussion and possible future enhancements concluded in this the chapter.

# CHAPTER 2

## MATHEMATICAL MODELING

### 2.1 Introduction

In this chapter we will retrieve the equations of motion of an underwater vehicle. First we will inform the body-fixed reference frame linear and angular velocities and inertial reference frame positions and Euler rates. Next, we will show the vehicle kinematics which will be the relation of body-fixed velocities with inertial frame positions. Then we will derive the rigid body dynamics which is expanded from the Newton's second law. Lastly we will look over dynamics as the study of forces and moments to the moving objects.

ULISAR is a small and modular UUV which brings a novel approach to underwater operations. She is equipped with the equipments that will carry out basic underwater operations successfully and fulfill the requirements of an underwater inspection. ULISAR will comprise an imaging sonar, two B/W cameras, lights, an acoustic modem to communicate with surface vehicle, acoustic transducers, pressure sensor, PC-104 stack and video grabber as main equipments. All power requirement will be satisfied by Lithium-Polymer battery packs. Her average speed is predicted to be about 1,5 knots. She will have stabilizers and fins to enhance the stability. Since she has no roll and sway control directly those particles will aid in satisfying passive roll control. Also in order to have passive roll stability, center of gravity must be below the center of buoyancy which will be performed by placing the heavy parts near the bottom of the vehicle. This method proved its success in many different designs [10]. She will be capable of diving to the depths of 100 meters but for the first tries 50 memetersill be a fair depth.

General planned design of ULISAR with her main components are shown in Figure 2.1.

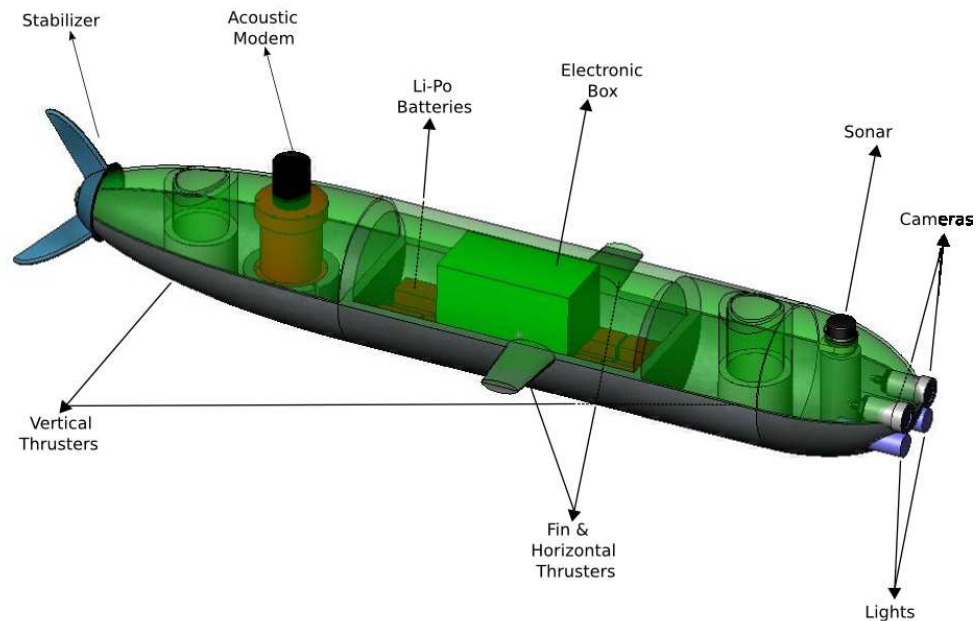


Figure 2.1: ULISAR UUV and main parts

The design of a practical underwater vehicle guidance and control systems requires knowledge of a extensive field of disciplines. Some of these are vectorial kinematics and dynamics, hydrodynamics, navigation systems and lastly control theory [5]. To able to design a high performance control system it is obvious that a good mathematical model of the vehicle is needed for both simulation and verification of the design.

First of all modeling of underwater vehicle is based on the study of statics and dynamics. Statics is the analysis of the forces and moments on physical systems in static equilibrium, while dynamics is concerned with the effects of forces on the motion of objects.

The motion of underwater vehicles is studied in 6 degrees of freedom (DOF) where 6 independent coordinates are necessary to determine the position and orientation of a rigid body. The first three coordinates and their time derivatives correspond to the position and translational motion along the  $x$ -,  $y$ -, and  $z$ - axes, whereas the last 3 coordinates and their time derivatives are used to describe orientation and rotational motion. For underwater vehicles these 6 degree of freedom are explained as:

- surge* : motion in the  $x$ -direction
- sway* : motion in the  $y$ -direction

- heave* : motion in the z-direction
- roll* : rotation about the x-axis
- pitch* : rotation about the y-axis
- yaw* : rotation about the z-axis

DOF	Motions & Rotations	Forces & Moments	Linear & Angular Velocities	Positions & Euler Angles
1	Motions in the x-direction (surge)	X	u	x
2	Motions in the y-direction (sway)	Y	v	y
3	Motions in the z-direction (heave)	Z	w	z
4	Rotation in the x-axis (roll)	K	p	$\phi$
5	Rotation in the y-axis (pitch)	M	q	$\theta$
6	Rotation in the z-axis (yaw)	N	r	$\psi$

Table 2.1: Notation used for marine vehicles

## 2.2 Kinematics

In this thesis we will use the following assumptions:

- Our vehicle is a rigid-body with a constant mass (Our vehicle's mass will change in time with proportional to the amount of water she will let in, but this amount is predicted to be small because of the slow velocity hence this mass change can be assumed to negligible.)
- Vehicle is not affected by the surface high frequency waves (operation condition is assumed to be deep waters).
- The effect of the rotating world to the accelerations of a point on the surface of the Earth is negligible (Indeed for slow vehicles this is a practical and tolerable assumption).
- Hydrodynamic coefficients are not variable (Though stated in [8] nonlinear damping terms do not affect maneuverability of the underwater vehicles, changes in the speed



and accelerations will differ the hydrodynamic coefficients, but since these coefficients are very small their changes will be much smaller where they can be assumed to negligible).

- We have the port-starboard (xz-plane) and bottom-top (xy-plane) symmetry. (Our heavy main parts are located on the middle of the xz-plane axis hence we have gained automatically a symmetry. Due to the installation of sonar near the head and acoustic modem at the aft, fore-aft symmetry can not be satisfied but in order to navigate in a horizontal plane regardless of the positions of the components pitch stability has to be satisfied. Since the distance between center of gravity and center of buoyancy in the body-fixed frame we will assume that we have a bottom-top symmetry).

### 2.2.1 Coordinate Frames

Defining the motions of the underwater vehicles in 6-DOF, two coordinate reference frames are used. The moving coordinate frame  $X_0Y_0Z_0$  is fixed to the vehicle and called the "Body-fixed reference frame" and other one according to the ground (earth) is called "Earth-fixed reference frame". Selecting the origin of the body-fixed coordinate frame as the *center of gravity (CG)* is a logical solution.

For underwater vehicles, body axes  $X_0$ ,  $Y_0$  and  $Z_0$  coincide with the principal axes of inertia and are usually defined as [5]:

- $X_0$  - longitudinal axis (directed from aft to fore)
- $Y_0$  - transverse axis (directed to starboard)
- $Z_0$  - normal axis (directed from top to bottom)

Based on the The Society of Naval Architects and Marine Engineers (SNAME) notation, general motion of a vehicle in 6-DOF can be shown by the below vectors,

$$\eta = [\eta_1^T, \eta_2^T]^T; \quad \eta_1 = [x, y, z]^T; \quad \eta_2 = [\phi, \theta, \psi]^T \quad (2.1)$$

$$v = [v_1^T, v_2^T]^T; \quad v_1 = [u, v, \omega]^T; \quad v_2 = [p, q, r]^T \quad (2.2)$$

$$\tau = [\tau_1^T, \tau_2^T]^T; \quad \tau_1 = [X, Y, Z]^T; \quad \tau_2 = [K, M, N]^T \quad (2.3)$$

Above  $\eta$  denotes the position and orientation vector with coordinates in the earth-fixed frame,  $v$  denotes the linear and angular velocity vector with coordinates in the body-fixed coordinate frame and  $\tau$  describes the forces and moments acting on the vehicle in the body-fixed frame. In a guidance and control system, orientation is usually represented by means of Euler angles or quaternions. Generally Euler angles are preferred for their simplicity but because tangent  $90^\circ$  is not defined for pitch angle, quaternions are used. In our case  $90^\circ$  pitch angle is an extreme case hence using Euler angles brings no problem to us.

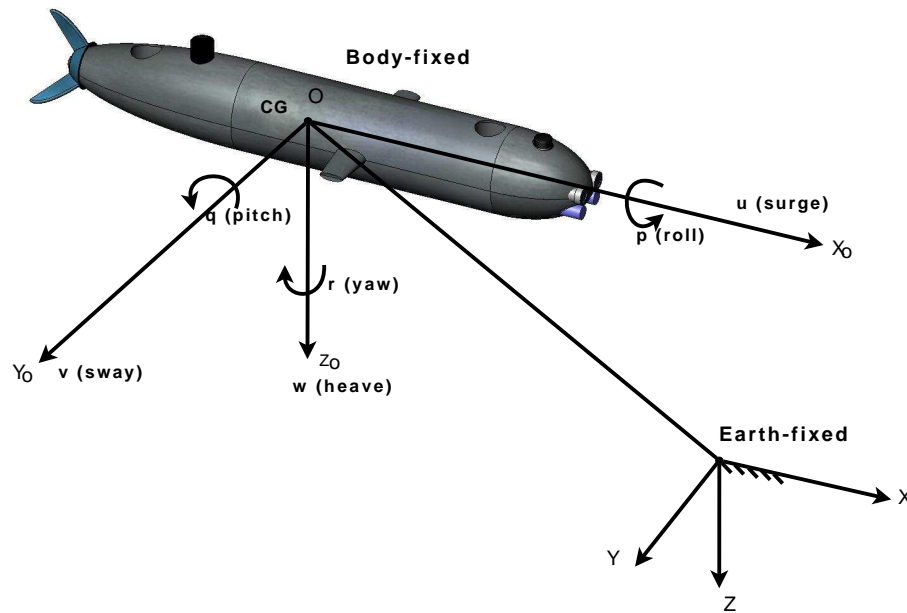


Figure 2.2: Earth-fixed and Body-fixed reference frames

All the motions of our vehicle in the body-fixed frame have to be represented relative to an inertial reference frame. For underwater vehicles we can assume that the effect of the rotating world to the accelerations of a point on the surface of the Earth is negligible. Therefore we do not need a star-fixed reference frame and we can select earth-fixed reference frame  $XYZ$  as inertial. In all our calculations the position and the orientation of our vehicle should be explained according to the inertial reference frame where the linear and angular velocities should be expressed in the body-fixed reference frame.

### 2.2.2 Euler Angles

As mentioned above for transformation from body-fixed frame to earth-fixed frame and vice versa, Euler angles are used. In all our transformations *xyz-convention* will be used. First transforming translational motion we will utilize the following equation:

$$\dot{\eta}_1 = T_1(\eta_2) v_1 \quad (2.4)$$

Writing above equation according to (2.1) and (2.2) we get

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = T_1(\eta_2) \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.5)$$

where  $T_1$  in (2.5) is defined as [5]

$$T_1(\eta_2) = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\psi s\theta s\phi & -c\psi s\phi + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (2.6)$$

Rotational transformations are achieved by the body-fixed angular velocity vector  $v_2 = [p, q, r]^T$  and Euler rate vector  $\dot{\eta}_2 = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$  related formula as

$$\dot{\eta}_2 = T_2(\eta_2) v_2 \quad (2.7)$$

Mentioning the vectors in open form we get

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T_2(\eta_2) \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.8)$$

Angular velocity transformation matrix in (2.8) is defined as

$$T_2(\eta_2) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \quad (2.9)$$

## 2.3 Rigid-Body Dynamics

In a general form the nonlinear dynamic equations of motion in 6 DOF can be written as:

$$M\dot{v} + C(v)v + D(v)v + \mathbf{g}(\eta) = \tau \quad (2.10)$$

Using the Euler's first and second axioms which were built on Newton's second law we can write the 6 DOF Rigid body equations of motion as:

$$\begin{aligned}
X &= m [\dot{u} - vr + wq - x_G(q^2 + r^2) + y_G(pq - \dot{r}) + z_G(pr + \dot{q})] \\
Y &= m [\dot{v} - wp + ur - y_G(r^2 + p^2) + z_G(qr - \dot{p}) + x_G(qp + \dot{r})] \\
Z &= m [\dot{w} - uq + vp - z_G(p^2 + q^2) + x_G(rp - \dot{q}) + y_G(rq + \dot{p})] \\
K &= I_x \dot{p} + (I_z - I_y)qr - (\dot{r} + pq)I_{xz} + (r^2 - q^2)I_{yz} + (pr - \dot{q})I_{xy} \\
&\quad + m [y_G(\dot{w} - uq + vp) - z_G(\dot{v} - wp + ur)] \\
M &= I_y \dot{q} + (I_x - I_z)pr - (\dot{p} + qr)I_{xy} + (p^2 - r^2)I_{zx} + (qp - \dot{r})I_{yz} \\
&\quad + m [z_G(\dot{u} - vr + wq) - x_G(\dot{w} - uq + vp)] \\
N &= I_z \dot{r} + (I_y - I_x)pq - (\dot{q} + rp)I_{yz} + (q^2 - p^2)I_{xy} + (rq - \dot{p})I_{zx} \\
&\quad + m [x_G(\dot{v} - wp + ur) - y_G(\dot{u} - vr + wq)]
\end{aligned} \tag{2.11}$$

In mathematical formulation for the ease of calculations (2.11) can be represented by vectorial form as

$$M_{RB}\dot{v} + C_{RB}(v)v = \tau_{RB} \tag{2.12}$$

Here the elements of the equations of motion relating to  $\dot{v}$  can be written in  $M_{RB}$  as

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mz_G & -my_G \\ 0 & m & 0 & -mz_G & 0 & mx_G \\ 0 & 0 & m & my_G & -mx_G & 0 \\ 0 & -mz_G & my_G & I_x & -I_{xy} & -I_{zx} \\ mz_G & 0 & -mx_G & -I_{yx} & I_y & -I_{yz} \\ -my_G & mx_G & 0 & -I_{zx} & -I_{zy} & I_z \end{bmatrix} \tag{2.13}$$

and the remainder elements the Coriolis term  $\omega \times v$  and centripetal term  $\omega \times (\omega \times r_G)$  can be written in  $C_{RB}$ . The Coriolis effect is the apparent deflection of objects from a straight path if the objects are viewed from a rotating frame of reference and the centripetal force is the external force required to make a body follow a circular path at constant speed. The force is directed inward, toward the center of the circle. Hence we can write our matrix as

$$C_{RB} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -m(y_Gq + z_Gr) & m(y_Gp + w) & m(z_Gp - v) \\ m(x_Gq - w) & -m(z_Gr + x_Gp) & m(z_Gq + u) \\ m(x_Gr + v) & m(y_Gr - u) & -m(x_Gp + y_Gq) \end{bmatrix}$$

$$\begin{bmatrix} m(y_Gq + z_Gr) & -m(x_Gq - w) & -m(x_Gr + v) \\ -m(y_Gp + w) & m(z_Gr + x_Gp) & -m(y_Gr - u) \\ -m(z_Gp - v) & -m(z_Gq + u) & m(x_Gp + y_Gq) \\ 0 & -I_{yz}q - I_{xz}p + I_zr & I_{yz}r + I_{xy}p - I_yq \\ I_{yz}q + I_{xz}p - I_zr & 0 & -I_{xz}r - I_{xy}q + I_xp \\ -I_{yz}r - I_{xy}p + I_yq & I_{xz}r + I_{xy}q - I_xp & 0 \end{bmatrix} \quad (2.14)$$

## 2.4 Added Mass

Added mass is the inertia added to system because of the accelerating body which will move some liquid surrounding its body. But the vehicle will force the surrounding fluid with proportional to forced harmonic motion due to acceleration of body where the particles which are far from the body will be induced less. In order to generate the added mass forces and moments Kirchoff's equations related to the fluid kinetic energy will be used.

Kinetic energy of the ideal fluid can be written as

$$T_A = \frac{1}{2} \mathbf{v}^T M_A \mathbf{v} \quad (2.15)$$

where  $M_A$  is the  $6 \times 6$  added mass inertia matrix which is comprised of 36 distinct added mass coefficients.

Hence we can write the added mass inertia matrix as

$$M_A \cong - \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix} \quad (2.16)$$

In the added mass inertia matrix the elements of the matrix are the derivatives of the forces and moments in the stated axis with respect to the accelerations. In other words,

$$M_{\dot{w}} = \frac{\partial M}{\partial \dot{w}} \quad (2.17)$$

Expressing the body-fixed velocity vectors as  $v_1 = [u, v, \omega]^T$  and  $v_2 = [p, q, r]^T$ , relation of the force  $\tau_1$  and moment  $\tau_2$  is achieved with Kirchhoff's equations in vector form.

$$\frac{d}{dt} \left( \frac{\partial T}{\partial v_1} \right) + v_2 \times \frac{\partial T}{\partial v_1} = \tau_1 \quad (2.18)$$

$$\frac{d}{dt} \left( \frac{\partial T}{\partial v_2} \right) + v_2 \times \frac{\partial T}{\partial v_2} + v_1 \times \frac{\partial T}{\partial v_1} = \tau_2 \quad (2.19)$$

Hence for the totally submerged vehicle we will find the added mass terms by using Kirchhoff's equations. Here we will utilize the fluid kinetic energy principle and take into consideration that by the motion of the vehicle in any direction, it will bring forth a kinetic energy for surrounding fluid [5]. Expanding the equations (2.18) and (2.19) yields

$$\frac{d}{dt} \begin{bmatrix} \frac{\partial T_A}{\partial u} \\ \frac{\partial T_A}{\partial v} \\ \frac{\partial T_A}{\partial \omega} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} \frac{\partial T_A}{\partial u} \\ \frac{\partial T_A}{\partial v} \\ \frac{\partial T_A}{\partial \omega} \end{bmatrix} = \begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix} \Rightarrow \frac{d}{dt} \begin{bmatrix} \frac{\partial T_A}{\partial u} \\ \frac{\partial T_A}{\partial v} \\ \frac{\partial T_A}{\partial \omega} \end{bmatrix} + \begin{bmatrix} q \frac{\partial T_A}{\partial \omega} - r \frac{\partial T_A}{\partial v} \\ r \frac{\partial T_A}{\partial u} - p \frac{\partial T_A}{\partial \omega} \\ p \frac{\partial T_A}{\partial v} - q \frac{\partial T_A}{\partial u} \end{bmatrix} = \begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix} \quad (2.20)$$

For the moments from the added mass,

$$\frac{d}{dt} \begin{bmatrix} \frac{\partial T_A}{\partial p} \\ \frac{\partial T_A}{\partial q} \\ \frac{\partial T_A}{\partial r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} \frac{\partial T_A}{\partial p} \\ \frac{\partial T_A}{\partial q} \\ \frac{\partial T_A}{\partial r} \end{bmatrix} + \begin{bmatrix} u \\ v \\ \omega \end{bmatrix} \times \begin{bmatrix} \frac{\partial T_A}{\partial u} \\ \frac{\partial T_A}{\partial v} \\ \frac{\partial T_A}{\partial \omega} \end{bmatrix} = \begin{bmatrix} K_A \\ M_A \\ N_A \end{bmatrix} \quad (2.21)$$

Applying vectorial products in (2.21) gives

$$\frac{d}{dt} \begin{bmatrix} q \frac{\partial T_A}{\partial r} - r \frac{\partial T_A}{\partial q} \\ r \frac{\partial T_A}{\partial p} - p \frac{\partial T_A}{\partial r} \\ p \frac{\partial T_A}{\partial q} - q \frac{\partial T_A}{\partial p} \end{bmatrix} + \begin{bmatrix} v \frac{\partial T_A}{\partial \omega} - \omega \frac{\partial T_A}{\partial v} \\ \omega \frac{\partial T_A}{\partial u} - u \frac{\partial T_A}{\partial \omega} \\ u \frac{\partial T_A}{\partial v} - v \frac{\partial T_A}{\partial u} \end{bmatrix} = \begin{bmatrix} K_A \\ M_A \\ N_A \end{bmatrix} \quad (2.22)$$

When we take the partial derivatives of  $T_A$  with respect to our linear and angular velocity

vectors and substitute them in (2.20) and (2.22) yields us the main added mass terms [5].

$$\begin{aligned}
X_A &= X_{\dot{u}}\dot{u} + X_{\dot{\omega}}(\dot{\omega} + uq) + X_{\dot{q}}\dot{q} + Z_{\dot{\omega}}\omega q + Z_{\dot{q}}q^2 \\
&\quad + X_{\dot{v}}v + X_{\dot{p}}\dot{p} + X_{\dot{r}}\dot{r} - Y_{\dot{v}}vr - Y_{\dot{p}}pr - Y_{\dot{r}}r^2 \\
&\quad - X_{\dot{v}}ur - Y_{\dot{\omega}}\omega r \\
&\quad + Y_{\dot{\omega}}vq + Z_{\dot{p}}pq - (Y_{\dot{q}} - Z_{\dot{r}})qr \\
Y_A &= X_{\dot{v}}\dot{u} + Y_{\dot{\omega}}\dot{\omega} + Y_{\dot{q}}\dot{q} \\
&\quad + Y_{\dot{v}}\dot{v} + Y_{\dot{p}}\dot{p} + Y_{\dot{r}}\dot{r} + X_{\dot{v}}rv - Y_{\dot{\omega}}vp + X_{\dot{r}}r^2 + (X_{\dot{p}} - Z_{\dot{r}})rp - Z_{\dot{p}}p^2 \\
&\quad - X_{\dot{\omega}}(up - \omega r) + X_{\dot{u}}ur - Z_{\dot{\omega}}\omega p \\
&\quad - Z_{\dot{q}}pq + X_{\dot{q}}qr \\
Z_A &= X_{\dot{\omega}}(\dot{u} - \omega q) + Z_{\dot{\omega}}\dot{\omega} + Z_{\dot{q}}\dot{q} - X_{\dot{u}}uq - X_{\dot{q}}q^2 \\
&\quad + Y_{\dot{\omega}}\dot{v} + Z_{\dot{p}}\dot{p} + Z_{\dot{r}}\dot{r} + Y_{\dot{v}}vp + Y_{\dot{r}}rp + Y_{\dot{p}}p^2 \\
&\quad + X_{\dot{v}}up + Y_{\dot{\omega}}\omega p \\
&\quad - X_{\dot{v}}vq - (X_{\dot{p}} - Y_{\dot{q}})pq - X_{\dot{r}}qr \\
K_A &= X_{\dot{p}}\dot{u} + Z_{\dot{p}}\dot{\omega} + K_{\dot{q}}\dot{q} - X_{\dot{v}}\omega u + X_{\dot{r}}uq - Y_{\dot{\omega}}\omega^2 - (Y_{\dot{q}} - Z_{\dot{r}})\omega q + M_{\dot{r}}q^2 \\
&\quad + Y_{\dot{p}}\dot{v} + K_{\dot{p}}\dot{p} + K_{\dot{r}}\dot{r} + Y_{\dot{\omega}}v^2 - (Y_{\dot{q}} - Z_{\dot{r}})vr + Z_{\dot{p}}vp - M_{\dot{r}}r^2 - K_{\dot{q}}rp \\
&\quad + X_{\dot{\omega}}uv - (Y_{\dot{v}} - Z_{\dot{\omega}})v\omega - (Y_{\dot{r}} + Z_{\dot{q}})\omega r - Y_{\dot{p}}\omega p - X_{\dot{q}}ur \\
&\quad + (Y_{\dot{r}} + Z_{\dot{q}})vq + K_{\dot{r}}pq - (M_{\dot{q}} - N_{\dot{r}})qr \\
M_A &= X_{\dot{q}}(\dot{u} + \omega q) + Z_{\dot{q}}(\dot{\omega} - uq) + M_{\dot{q}}\dot{q} - X_{\dot{\omega}}(u^2 - \omega^2) - (Z_{\dot{\omega}} - X_{\dot{u}})\omega u \\
&\quad + Y_{\dot{q}}\dot{v} + K_{\dot{q}}\dot{p} + M_{\dot{r}}\dot{r} + Y_{\dot{p}}vr - Y_{\dot{r}}vp - K_{\dot{r}}(p^2 - r^2) + (K_{\dot{p}} - N_{\dot{r}})rp \\
&\quad - Y_{\dot{\omega}}uv + X_{\dot{v}}v\omega - (X_{\dot{r}} + Z_{\dot{p}})(up - \omega r) + (X_{\dot{p}} - Z_{\dot{r}})(\omega p + ur) \\
&\quad - M_{\dot{r}}pq + K_{\dot{q}}qr \\
N_A &= X_{\dot{r}}\dot{u} + Z_{\dot{r}}\dot{\omega} + M_{\dot{r}}\dot{q} + X_{\dot{v}}u^2 + Y_{\dot{\omega}}\omega u - (X_{\dot{p}} - Y_{\dot{q}})uq - Z_{\dot{p}}\omega q - K_{\dot{q}}q^2 \\
&\quad + Y_{\dot{r}}\dot{v} + K_{\dot{r}}\dot{p} + N_{\dot{r}}\dot{r} - X_{\dot{v}}v^2 - X_{\dot{r}}vr - (X_{\dot{p}} - Y_{\dot{q}})vp + M_{\dot{r}}rp + K_{\dot{q}}p^2 \\
&\quad - (X_{\dot{u}} - Y_{\dot{v}})uv - X_{\dot{\omega}}v\omega + (X_{\dot{q}} + Y_{\dot{p}})up + Y_{\dot{r}}ur + Z_{\dot{q}}\omega p \\
&\quad - (X_{\dot{q}} + Y_{\dot{p}})vq - (K_{\dot{p}} - M_{\dot{q}})pq - K_{\dot{r}}qr
\end{aligned} \tag{2.23}$$

## 2.5 Damping

Hydrodynamic damping for the underwater vehicles occurs because of the following effects.

- Potential Damping: Damping caused by the surface waves. These waves are generally

high frequency waves with small wave lengths. By our assumption at the beginning of the mathematical model stating that our vehicle works near the sea bottom gives us the right to neglect this effect.

- **Skin Friction:** This is the damping occurring because of the flow of water around the boundary of vehicle. While vehicle advancing with a constant speed, water near the bow achieves laminar flow (streamline flow) with no disruption to the surface. Going forward on its flow after passing the bow, skin friction decelerates the liquid that is why turbulent flow starts at this point. This process of passing from laminar flow to turbulent flow is known as boundary layer transition. Skin friction is represented with linear skin friction because of laminar boundary layer and quadratic skin friction due to turbulent boundary layers. Non-dimensional Reynolds number assigns the type of the flow.
- **Wave Damping:** This is the damping due to the waves while vehicles try to advance on the surface of the water. Again with our assumption that the operations will be near the sea bottom, we can neglect this damping.
- **Damping of Vortex Shedding:** Vortex shedding occurs because of the pressure differences on the flow path of water. Liquid after passing the first meet surface of the object creates the low pressure vortices, which ends with a turbulent and unsteady flow. The size of the vortices and the effect of damping due to vortex shedding is directly proportional to cross-sectional area of the vehicle and with square of velocity. Trying to increase the operational speed of underwater vehicle brings damping disadvantage with it. At this point, the outer body design and the vehicle's production material take an important role.

From the aspect of losses, effect of the damping will mostly occur due to skin friction and vortex shedding. Skin friction is an important effect on the damping of the vehicle but the details of it is far beyond the scope of this thesis. More details about damping can be found in [9] and [12]. The design of the body to decrease the damping is an important problem for the construction of vehicle. Here eccentricity plays an important role, which shows how much the head shape of vehicle deviates from circular mode. Also the ratio of the total length of the vehicle to the diameter directly affects the speed of the vehicle where the torpedoes are good examples for this kind of design. Now let us formulate the damping which are important for us.



The damping force due to the vortex shedding can be modeled as

$$D_f(U) = -\frac{1}{2} \rho c_d (Rn) A_{cs} |U| U \quad (2.24)$$

where  $c_d$  is the non-dimensional drag coefficient directly related with Reynolds number.  $A_{cs}$  is the projected cross sectional area of the vehicle facing with water which is  $\pi d^2/4$  for a sphere ( $d$  is diameter).  $U$  stands for the velocity of the vehicle and  $\rho$  is the density of the water. Reynolds number is a function of velocity ( $U$ ), physical length ( $l$ ) and kinematic viscosity ( $\nu$ ). Reynolds number can be calculated by the following formula [12]. In our calculations we assumed  $c_d = 0.20$ .

$$R_n = \frac{\rho U l}{\mu} = \frac{U l}{\nu} \quad (2.25)$$

Here  $\mu$  stands for the fluid viscosity and  $\rho$  for the density of water ( $\nu = 1.05 \times 10^{-6}$  for sea water with 20°C and salinity of 3.5 %). Appendix B shows the drag coefficient of a sphere for different Reynolds numbers.

Damping due to the skin friction will be modeled as linear and quadratic damping. Hence our damping will be as

$$D(\nu) \nu + |\nu| D(\nu) \nu \quad (2.26)$$

Though some approximations and simplifications will be achieved on the damping matrix in following steps, now it can be written as

$$D_M(\nu) = \begin{bmatrix} X_u + X_{|u|u}|u| & X_v + X_{|v|v}|v| & X_w + X_{|w|w}|w| \\ Y_u + Y_{|u|u}|u| & Y_v + Y_{|v|v}|v| & Y_w + Y_{|w|w}|w| \\ Z_u + Z_{|u|u}|u| & Z_v + Z_{|v|v}|v| & Z_w + Z_{|w|w}|w| \\ K_u + K_{|u|u}|u| & K_v + K_{|v|v}|v| & K_w + K_{|w|w}|w| \\ M_u + M_{|u|u}|u| & M_v + M_{|v|v}|v| & M_w + M_{|w|w}|w| \\ N_u + N_{|u|u}|u| & N_v + N_{|v|v}|v| & N_w + N_{|w|w}|w| \end{bmatrix} \quad (2.27)$$

$$\begin{bmatrix} X_p + X_{|p|p}|p| & X_q + X_{|q|q}|q| & X_r + X_{|r|r}|r| \\ Y_p + Y_{|p|p}|p| & Y_q + Y_{|q|q}|q| & Y_r + Y_{|r|r}|r| \\ Z_p + Z_{|p|p}|p| & Z_q + Z_{|q|q}|q| & Z_r + Z_{|r|r}|r| \\ K_p + K_{|p|p}|p| & K_q + K_{|q|q}|q| & K_r + K_{|r|r}|r| \\ M_p + M_{|p|p}|p| & M_q + M_{|q|q}|q| & M_r + M_{|r|r}|r| \\ N_p + N_{|p|p}|p| & N_q + N_{|q|q}|q| & N_r + N_{|r|r}|r| \end{bmatrix}$$

## 2.6 Gravitational and Buoyant Forces

In our vehicle center of gravity will be defined with  $r_G = [x_G, y_G, z_G]^T$  and the center of buoyancy will be expressed with  $r_B = [x_B, y_B, z_B]^T$ . The gravitational force  $f_G$  will act on center of gravity and buoyant force will act on center of buoyancy  $f_B$  where both forces act in inertial frame but they are defined in body-fixed frame.

Defining the  $m$  as the mass of the vehicle,  $V$  as the volume of fluid displaced,  $g$  as the acceleration of gravity downwards,  $\rho$  density of the fluid, weight of vehicle and buoyancy force can be written as  $W = m g$ ,  $B = \rho g V$ .

Then the gravitational and buoyant forces in body-fixed frame can be defined by using Euler transformations

$$f_G = T_1^{-1}(\eta_2) \begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix} \quad (2.28)$$

and

$$f_B = -T_1^{-1}(\eta_2) \begin{bmatrix} 0 \\ 0 \\ B \end{bmatrix} \quad (2.29)$$

Finally gravitational and buoyant forces and moments can be written as [5]

$$\mathbf{g} = - \begin{bmatrix} f_G(\eta) + f_B(\eta) \\ r_G \times f_G(\eta) + r_B \times f_B(\eta) \end{bmatrix} \quad (2.30)$$

Substituting the center of gravity and buoyancy with forces in (2.30) we get

$$\mathbf{g} = \begin{bmatrix} (W - B)s\theta \\ -(W - B)c\theta s\phi \\ -(W - B)c\theta c\phi \\ -(y_G W - y_B B)c\theta c\phi + (z_G W - z_B B)c\theta s\phi \\ (z_G W - z_B B)s\theta + (x_G W - x_B B)c\theta c\phi \\ -(x_G W - x_B B)c\theta s\phi - (y_G W - y_B B)s\theta \end{bmatrix} \quad (2.31)$$

Till that point we showed the path to generate the mathematical model of an underwater vehicle. But because of the nonlinear and coupled attitude of the model there will be too many unknowns with different weighted values which is an undesired conclusion. Now we will use some simplifications and assumptions to reduce the complicated model which proved itself in many designed underwater vehicles.

First because of the position of the center of gravity and center of buoyancy,  $x_G$  and  $y_G$  will be equal to zero. On the other hand, since  $I_{xy} = 0$  (top/bottom symmetry) and  $I_{xz} = 0$  (port/starboard symmetry) our new inertia . matrix becomes.

$$I_0 \cong \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & I_{yz} \\ 0 & I_{zy} & I_z \end{bmatrix} \quad (2.32)$$

Also because of the symmetry properties, our mass inertia matrix can be simplified in an important manner which is also applicable to damping matrix. In Section 2.2 we assumed that we have xz- and xy- plane symmetries (port/starboard and bottom/top symmetries) by which we acquire the following simplified mass matrix

$$M = \begin{bmatrix} M_{11} & 0 & 0 & 0 & M_{15} & 0 \\ 0 & M_{22} & 0 & M_{24} & 0 & M_{26} \\ 0 & 0 & M_{33} & 0 & M_{35} & 0 \\ 0 & 0 & 0 & M_{44} & 0 & 0 \\ 0 & 0 & M_{53} & 0 & M_{55} & 0 \\ 0 & M_{62} & 0 & 0 & 0 & M_{66} \end{bmatrix} \quad (2.33)$$

The same simplification method can be applied to the damping matrix and the same coefficients on the stated positions will be left but in most of the underwater applications a rough approximation is done where the damping matrix with its linear and quadratic terms is assumed as diagonal. But for our case we will not take such a rough approximation and leave the coefficients at their positions after symmetry simplifications.

## 2.7 Hydrodynamic Coefficients

In this section we will retrieve the numerical values of hydrodynamical coefficients. We will derive the added mass coefficients and damping coefficients which are the representations of the derivation of forces and moments with respect to the linear and angular velocities and accelerations. As we evinced in Section 2.5 axial drag can be calculated by

$$X_{|u|u} = -\frac{1}{2}\rho c_d (Rn) A_{cs} |U|U \quad (2.34)$$

The remaining crossflow drag will be found by strip theory according to [5]. After simplification in the damping matrix the following coefficients remain, which are found by the

following formulas:  $Y_{|v|v}, Z_{|w|w}, K_{|p|p}, M_{|q|q}, N_{|r|r}, Y_{|r|r}, Z_{|q|q}, M_{|w|w}, N_{|v|v}$ .

$$Y_{|v|v} = Z_{|w|w} = -\frac{1}{2}\rho c_{dc} \int_{-L/2}^{L/2} 2b(x) dx \quad (2.35)$$

$$M_{|w|w} = -N_{|v|v} = \frac{1}{2}\rho c_{dc} \int_{-L/2}^{L/2} 2b(x)x dx \quad (2.36)$$

$$Y_{|r|r} = -Z_{|q|q} = -\frac{1}{2}\rho c_{dc} \int_{-L/2}^{L/2} 2b(x)x|x| dx \quad (2.37)$$

$$M_{|q|q} = -N_{|r|r} = -\frac{1}{2}\rho c_{dc} \int_{-L/2}^{L/2} 2b(x)x^3 dx \quad (2.38)$$

$$K_{|p|p} = 0 \quad (2.39)$$

Here  $\rho$  is the water density,  $c_{dc}$  crossflow drag coefficient,  $b(x)$  is the half-width of the vehicle with respect to the total length,  $l$  is the length of the vehicle.

After simplifications because of symmetry our added mass matrix (2.16) will transform into

$$M = \begin{bmatrix} X_{\dot{u}} & 0 & 0 & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 & 0 & 0 & N_{\dot{v}} \\ 0 & 0 & Z_{\dot{w}} & 0 & M_{\dot{w}} & 0 \\ 0 & 0 & 0 & K_{\dot{p}} & 0 & 0 \\ 0 & 0 & Z_{\dot{q}} & 0 & M_{\dot{q}} & 0 \\ 0 & Y_{\dot{r}} & 0 & 0 & 0 & N_{\dot{r}} \end{bmatrix} \quad (2.40)$$

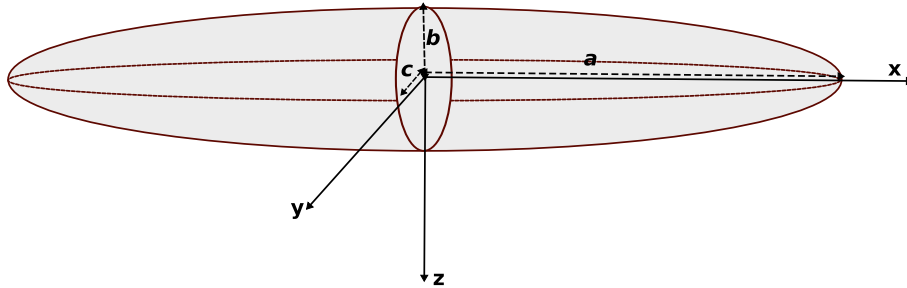


Figure 2.3: Prolate Ellipsoid and Dimensions

Hence we have to derive the remainders from the added mass matrix terms which are;  $X_{\dot{u}}$ ,  $Y_{\dot{v}}$ ,  $Z_{\dot{w}}$ ,  $K_{\dot{p}}$ ,  $M_{\dot{q}}$ ,  $N_{\dot{r}}$ ,  $Y_{\dot{i}}$ ,  $Z_{\dot{q}}$ ,  $M_{\dot{w}}$ ,  $N_{\dot{v}}$ . For slender bodies these coefficients can be derived by strip theory. In our case our vehicle shows similarities with a prolate ellipsoid as shown in Figure (2.3).

Therefore using the strip theory added mass coefficients will be found according to [5] as

$$-X_{\dot{u}} = \int_{-L/2}^{L/2} A_{11}(y,z)dx \simeq 0.10m \quad (2.41)$$

$$-Y_{\dot{v}} = \int_{-L/2}^{L/2} A_{22}(y,z)dx \quad (2.42)$$

$$-Z_{\dot{w}} = \int_{-L/2}^{L/2} A_{33}(y,z)dx \quad (2.43)$$

$$-K_{\dot{p}} = \int_{-L/2}^{L/2} A_{44}(y,z)dx \triangleq \int_{-B/2}^{B/2} y^2 A_{33}(x,z)dy + \int_{-H/2}^{H/2} z^2 A_{22}(x,y)dz \quad (2.44)$$

$$-M_{\dot{q}} = \int_{-L/2}^{L/2} A_{55}(y,z)dx \triangleq \int_{-L/2}^{L/2} x^2 A_{33}(x,z)dx + \int_{-H/2}^{H/2} z^2 A_{11}(x,y)dz \quad (2.45)$$

$$-N_{\dot{p}} = \int_{-L/2}^{L/2} A_{66}(y,z)dx \triangleq \int_{-B/2}^{B/2} y^2 A_{11}(x,z)dy + \int_{-L/2}^{L/2} x^2 A_{22}(y,z)dx \quad (2.46)$$

where L,B and H are the dimensions of the vehicle. For a prolate ellipsoid, 2-dimensional coefficients in the above equations are given in Figure (2.4).

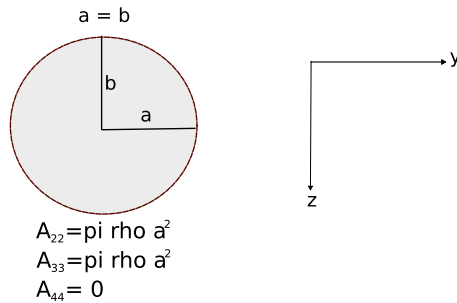


Figure 2.4: Two-dimensional Added Mass Coefficients

On the other hand those added mass coefficients can be found by theoretical formulations

stated by Lamb in [9].

$$X_{\dot{u}} = -\frac{\alpha_0}{2 - \alpha_0} m \quad (2.47)$$

$$Y_{\dot{v}} = \quad Z_{\dot{w}} = -\frac{\beta_0}{2 - \beta_0} m \quad (2.48)$$

$$K_{\dot{p}} = \quad 0 \quad (2.49)$$

$$N_{\dot{r}} = \quad M_{\dot{q}} = -\frac{1}{5} \frac{(b^2 - a^2)^2 (\alpha_0 - \beta_0)}{2(b^2 - a^2) + (b^2 + a^2)(\beta_0 - \alpha_0)} m \quad (2.50)$$

Here  $m$  is the mass of the vehicle which can be found by

$$m = \frac{4}{3} \pi \rho a b^2 \quad (2.51)$$

and  $\alpha_0$  and  $\beta_0$  are defined as

$$\alpha_0 = \frac{2(1 - e^2)}{e^3} \left( \frac{1}{2} \ln \frac{1 + e}{1 - e} - e \right) \quad (2.52)$$

$$\beta_0 = \frac{1}{e^2} - \frac{1 - e^2}{2e^3} \ln \frac{1 + e}{1 - e} \quad (2.53)$$

Above in the equations  $e$  stands for the eccentricity defined as

$$e = 1 - (b/a)^2 \quad (2.54)$$

Also there exists another alternative method of equations which are related with the added mass terms. We used this last mentioned method to check the coefficients found by the first explained method above, both gave the same results after calculations. This last method is similar with the one method mentioned above. In this method, Lamb expresses first k-terms as

$$k_1 = \frac{\alpha_0}{2 - \alpha_0} \quad (2.55)$$

$$k_2 = \frac{\beta_0}{2 - \beta_0} \quad (2.56)$$

$$k' = \frac{e^4 (\beta_0 - \alpha_0)}{(2 - e^2) [2e^2 - (2 - e^2)(\beta_0 - \alpha_0)]} \quad (2.57)$$

Then he mentioned the added mass coefficients as

$$X_{\dot{u}} = -k_1 m \quad (2.58)$$

$$Y_{\dot{v}} = -Z_{\dot{w}} = -k_2 m \quad (2.59)$$

$$N_{\dot{r}} = -M_{\dot{q}} = -k' I_y \quad (2.60)$$

where the moment of inertia in y- axis,  $I_y$  can be found by

$$I_y = I_z = \frac{4}{15} \pi \rho a b^2 (a^2 + b^2) \quad (2.61)$$

For a prolate ellipsoid calculation of the quadratic damping coefficients can be achieved by the equations (2.35)-(2.39) but since shape of our vehicle is a little different from an ellipsoid we have to separate it in order to find stated coefficients. It is clear in Figure (2.5) that we have

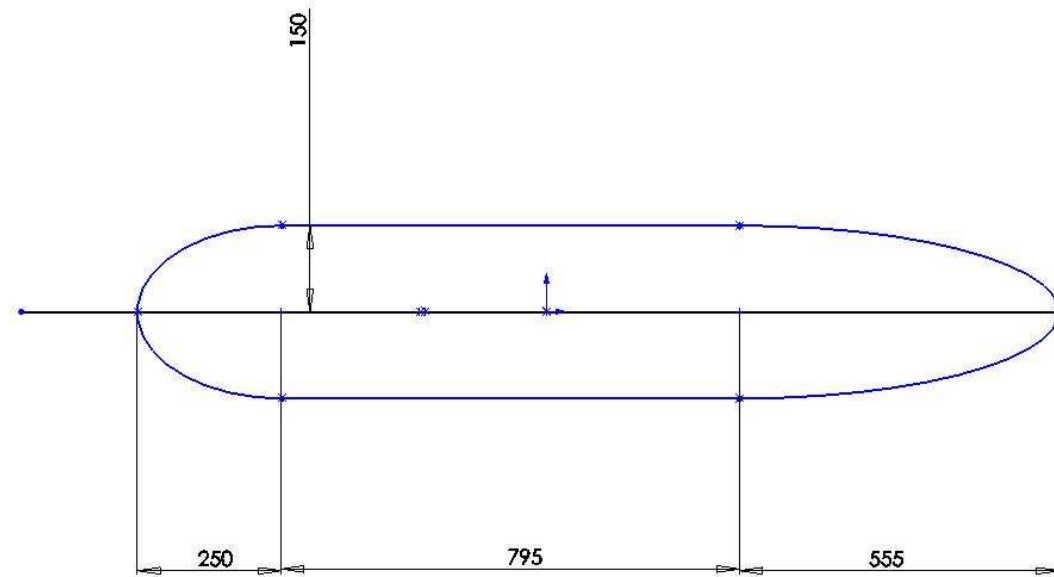


Figure 2.5: Cross-sectional view of our Vehicle

to separate the ellipsoid into three parts. We will find the equations of each part. Our vehicle's length is 1.60 meters and the distance of the separation points from nose to aft are .25 m. and 1.045 m. The width of hull with respect to total length of vehicle is required in the quadratic damping equations where we formulated the each section of the vehicle as

$$y = \sqrt{0.15^2 - \left(\frac{0.15}{0.25}\right)^2 (x - 0.25)^2} \quad 0 \leq x \leq 0.25 \quad (L_1) \quad (2.62)$$

$$y = 0.15 \quad 0.25 < x < 1.045 \quad (L_2) \quad (2.63)$$

$$y = \sqrt{0.15^2 - \left(\frac{0.15}{0.555}\right)^2 (x - 1.045)^2} \quad 1.045 \leq x \leq 1.60 \quad (L_3) \quad (2.64)$$

Hence our quadratic damping equations will be as

$$\begin{aligned}
Y_{|v|v} = Z_{|w|w} = & -\frac{1}{2}\rho c_{dc} \int_0^{P1} 2\sqrt{0.15^2 - \left(\frac{0.15}{0.25}\right)^2 (x-0.25)^2} dx \\
& -\frac{1}{2}\rho c_{dc} \int_{P1}^{P2} 0.30 dx - \frac{1}{2}\rho c_{dc} \int_{P2}^{P3} 2\sqrt{0.15^2 - \left(\frac{0.15}{0.555}\right)^2 (x-1.045)^2} dx \quad (2.65)
\end{aligned}$$

$$\begin{aligned}
M_{|w|w} = -N_{|v|v} = & -\frac{1}{2}\rho c_{dc} \int_0^{P1} 2\sqrt{0.15^2 - \left(\frac{0.15}{0.25}\right)^2 (x-0.25)^2} x dx \\
& -\frac{1}{2}\rho c_{dc} \int_{P1}^{P2} 0.30 x dx - \frac{1}{2}\rho c_{dc} \int_{P2}^{P3} 2\sqrt{0.15^2 - \left(\frac{0.15}{0.555}\right)^2 (x-1.045)^2} x dx \quad (2.66)
\end{aligned}$$

$$\begin{aligned}
Y_{|r|r} = -Z_{|q|q} = & -\frac{1}{2}\rho c_{dc} \int_0^{P1} 2\sqrt{0.15^2 - \left(\frac{0.15}{0.25}\right)^2 (x-0.25)^2} x|x| dx \\
& -\frac{1}{2}\rho c_{dc} \int_{P1}^{P2} 0.30 x|x| dx - \frac{1}{2}\rho c_{dc} \int_{P2}^{P3} 2\sqrt{0.15^2 - \left(\frac{0.15}{0.555}\right)^2 (x-1.045)^2} x|x| dx \quad (2.67)
\end{aligned}$$

$$\begin{aligned}
M_{|q|q} = -N_{|r|r} = & -\frac{1}{2}\rho c_{dc} \int_0^{P1} 2\sqrt{0.15^2 - \left(\frac{0.15}{0.25}\right)^2 (x-0.25)^2} x^3 dx \\
& -\frac{1}{2}\rho c_{dc} \int_{P1}^{P2} 0.30 x^3 dx - \frac{1}{2}\rho c_{dc} \int_{P2}^{P3} 2\sqrt{0.15^2 - \left(\frac{0.15}{0.555}\right)^2 (x-1.045)^2} x^3 dx \quad (2.68)
\end{aligned}$$

$$K_{|p|p} = 0 \quad (2.69)$$

## 2.8 Equations of Motion

In this section we will show the vectorial representation of the body-fixed equations of motion for our vehicle. In Section 2.3 we have expressed the nonlinear equations of motion in body-fixed frame as

$$M\dot{v} + C(v)v + D(v)v + \mathbf{g}(\eta) = \tau \quad (2.70)$$

$$\dot{\eta} = T(\eta)v \quad (2.71)$$



where

$$M = M_{RB} + M_A \quad C(v) = C_{RB}(v) + C_A(v) \quad (2.72)$$

$$D(v) = D_{skin}(v) + D_{vortex}(v) \quad (2.73)$$

After our simplifications we retrieve the component matrices of the equations of motion as

$$M = M_{RB} + M_A = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & m - Y_{\dot{v}} & 0 & 0 & 0 & m - N_{\dot{v}} \\ 0 & 0 & m - Z_{\dot{w}} & 0 & m - M_{\dot{w}} & 0 \\ 0 & 0 & 0 & m - K_{\dot{p}} & 0 & 0 \\ 0 & 0 & m - Z_{\dot{q}} & 0 & m - M_{\dot{q}} & 0 \\ 0 & m - Y_{\dot{r}} & 0 & 0 & 0 & m - N_{\dot{r}} \end{bmatrix} \quad (2.74)$$

$$C_A = \begin{bmatrix} 0 & 0 & 0 & 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v \\ 0 & 0 & 0 & Z_{\dot{w}}w & 0 & -X_{\dot{u}}u \\ 0 & 0 & 0 & -Y_{\dot{v}}v & X_{\dot{u}}u & 0 \\ 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v & 0 & -N_{\dot{r}}r & M_{\dot{q}}q \\ Z_{\dot{w}}w & 0 & -X_{\dot{u}}u & N_{\dot{r}}r & 0 & -K_{\dot{p}}p \\ -Y_{\dot{v}}v & X_{\dot{u}}u & 0 & -M_{\dot{q}}q & K_{\dot{p}}p & 0 \end{bmatrix} \quad (2.75)$$

$$C_{RB} = \begin{bmatrix} 0 & 0 & 0 & mz_{Gr} & mw & -mv \\ 0 & 0 & 0 & -mw & mz_{Gr} & mu \\ 0 & 0 & 0 & -m(z_{Gp} + v) & -m(z_{Gq} + u) & 0 \\ -mz_{Gr} & mw & m(z_{Gp} + v) & 0 & I_{yz}q + I_zr & I_{yz}r - I_yq \\ -mw & -mz_{Gr} & m(z_{Gq} + u) & I_{yz}q + I_zr & 0 & I_xp \\ mv & -mu & 0 & -I_{yz}r + I_yq & -I_xp & 0 \end{bmatrix} \quad (2.76)$$

$$D_v(v) = -$$

$$\begin{bmatrix} X_u + X_{|u|u}|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_v + Y_{|v|v}|v| & 0 & 0 & 0 & Y_r + Y_{|r|r}|r| \\ 0 & 0 & Z_w + Z_{|w|w}|w| & 0 & Z_q + Z_{|q|q}|q| & 0 \\ 0 & 0 & 0 & K_p + K_{|p|p}|p| & 0 & 0 \\ 0 & 0 & M_w + M_{|w|w}|w| & 0 & M_q + M_{|q|q}|q| & 0 \\ 0 & N_v + N_{|v|v}|v| & 0 & 0 & 0 & N_r + N_{|r|r}|r| \end{bmatrix} \quad (2.77)$$

Last of all our gravitational and buoyant forces matrix will be as

$$\mathbf{g} = \begin{bmatrix} (W - B)s\theta \\ -(W - B)c\theta s\phi \\ -(W - B)c\theta c\phi \\ -(y_G W - y_B B)c\theta c\phi + (z_G W - z_B B)c\theta s\phi \\ (z_G W - z_B B)s\theta + (x_G W - x_B B)c\theta c\phi \\ -(x_G W - x_B B)c\theta s\phi - (y_G W - y_B B)s\theta \end{bmatrix} \quad (2.78)$$

# CHAPTER 3

## CONTROL PROCEDURES

### **Linearized Equations of Motion**

Underwater vehicles operating in complex environments with coupled maneuvers are known to be highly nonlinear, nevertheless to exploit the advantage of enhanced control methods we prefer to linearize our model around an equilibrium point, which is a constant speed for our case. Hence linearizing our model and achieving some simplifications a sort of well known control methods should be applied easily .

In our configuration we have 4 thrusters, 2 of them are placed vertically and other 2 horizontally. Horizontal thrusters are used both for speed and steering (yaw) and vertical thrusters are used for depth control. In order to achieve robust control, we have chosen the Decoupled Control Method hence divided the 6 DOF (Degree of Freedom) motion into three main subsystems and designed different control methods for each subsystem. We achieved separation as

1. Speed System
2. Steering System
3. Diving System

Design and analysis phase of all work is done by using Control Toolbox in *MATLAB* and *Simulink*.

## Speed Control

Today robust and reliable control stands as a first priority towards the development of efficient underwater vehicles most of which operate in strict and tough conditions. On nonlinear control systems, modeling inaccuracies may cause to undesired effects, hence to deal with model uncertainties we need robust control methods. On the other hand when we have simple systems or simplified models of complex systems we prefer simple methods like PID (Proportional Integral Differential) control method.

On the speed control due to the simplicity of the our model where we neglected the effects from sway, heave, roll, pitch and yaw, we preferred PID Control method and designed the control using Simulink. Neglecting other effects we retrieved a SISO model with one state and one input as

$$(m - X_{\dot{u}})\dot{u} = X_{|u|u}|u|u + \tau + X_{ext} \quad (3.1)$$

In our equation we omitted some effects like Coriolis and centripetal forces nevertheless we took quadratic damping as the main disturbing effect. Here  $\tau$  stands for the vertical thruster force. For linear case we know that 1<sup>st</sup> order approximation of the thrust force  $\tau$  is equal to,

$$\tau = \rho D^4 K_T (J_0) |n|n \quad (3.2)$$

where  $\rho$  states the density of sea water,  $D$  propeller diameter,  $K_T$  advance thruster coefficient which is a function of advance number ( $J_0 = V_a/nD$ ),  $V_a$  water speed passing through the propeller and lastly  $n$  for the revolution of propeller.

Hence we can write our equation as,

$$(m - X_{\dot{u}})\dot{u} = X_{|u|u}|u|u + \tau_c + X_{ext} \quad (3.3)$$

$\tau_c$  stands for the thruster coefficient, which is found by (3.2).

We applied standart PID control to one DOF model with positive coefficients of  $K_p$ ,  $K_d$  and  $K_i$  selected with respect to the response of the system. It is assumed that the state and output is directly measurable and the model parameters are retrieved according to the formulas that

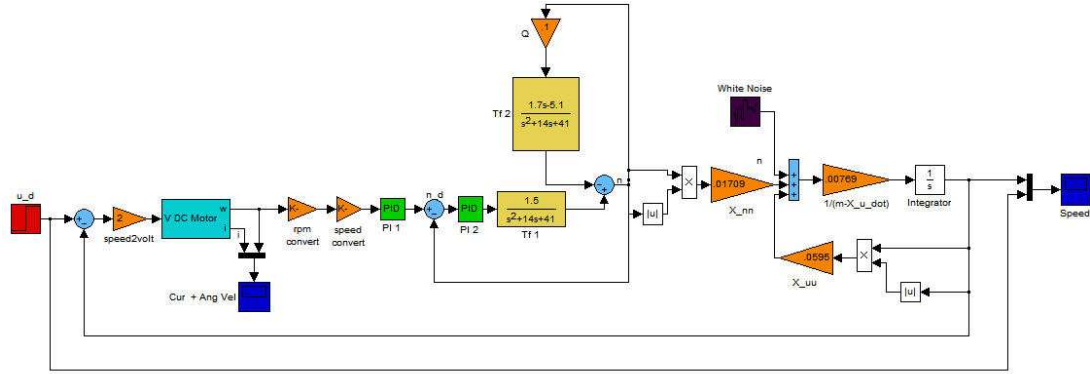


Figure 3.1: Simulink PID Model

stated on the previous chapter. A white Gaussian noise is added to the system as an external disturbance in order to raise the reality of model compared with the actual one.

$$u = K_p(x(t) - x_d(t)) + K_d(\dot{x}(t) - \dot{x}_d(t)) + K_i \int_0^t (x(\tau) - x_d(\tau))d\tau \quad (3.4)$$

There exists two loops, one inner and one outer. Inner loop feeds the system by tuning the propeller revolution to retrieve desired speed due to the differing load, whereas outer loop directly works for reducing the speed error. Like the most underwater vehicle controllers we preferred PI- control law instead of PID in order to get rid of necessity for Kalman filter which we should design for estimating the derivative of surge and angular propeller accelerations.

We can easily see the effect of the disturbance in *Figure-2* where PID controller can not successfully and quickly compensate the error.

## Steer Control

Before starting steer control it will be more appropriate to say a few things about SMC and Optimal Control because we will design our controllers on both methods beside PID.

## Sliding Mode Control

As being one of the most robust control methods, Sliding Mode Control (SMC) is based on the philosophy that it is easier to control 1<sup>st</sup> order systems compared with highly  $n^{th}$  order

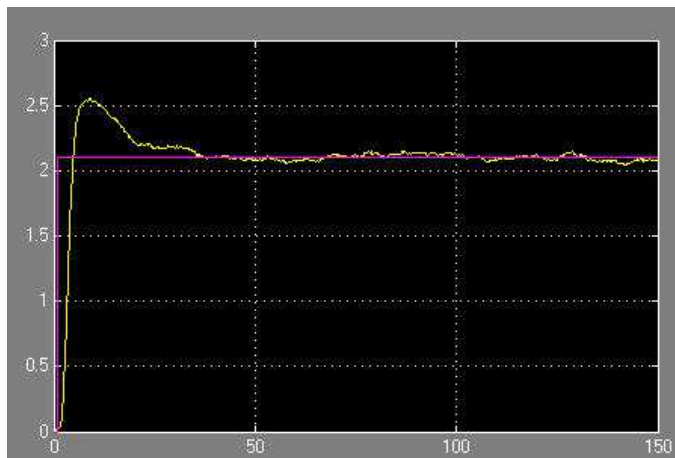


Figure 3.2: Commanded and Real Output Velocities (m/s) for PID

systems. Therefore any modeling and parameter inaccuracies can be compensated with this method though in a wearisome manner. In addition to stated assets, SMC brings us the advantage to face with strong perturbations like currents, waves and other unpredicted effects in complex sea environment. General application of SMC to underwater vehicles consists of designing a controller for the linearized part of the system and considering the nonlinearities as the parametric uncertainties. In design step we face with two different sliding surface selections. In the first method we select a scalar function of form  $s = \dot{e} + \lambda e$ , which is the sum of the position error and the velocity vector. For  $s = 0$  this functions defines a sliding surface ensuring that the tracking error  $e$  converges to zero. In the second method sliding surface is based on the state variable errors depicted as:  $\sigma(e) = s^T e$ .

In the first method we start design by defining the tracking error vector with  $e = x - x_d$  where  $x$  stands for the state vector,  $x_d$  for the desired state vector. Then we define a scalar time-varying surface  $s(t)$  in  $\mathfrak{R}^n$  by the scalar equation  $s(x, t) = 0$ , where

$$s(x, t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} e \quad (3.5)$$

with  $\lambda$  being a strictly positive constant. In a general manner, choosing  $n = 2$  we get a weighted sum of the position error and the velocity vector. For  $s = 0$ , our surface defines a sliding surface with dynamics:

$$e(t) = e^{-\lambda(t-t_0)} e(t_0) \quad (3.6)$$

guaranteeing that tracking error  $e(t)$  will converge to zero exponentially in finite time whatever the initial condition is.

As a second method when the coupled movements considered, using the sliding surface based on the state variable errors instead of the output errors seems to be more logical and more useful especially in underwater environment. In that manner sliding surface is defined as  $\sigma(e) = s^T e$  where  $e = x - x_d$  is the state tracking error,  $s \in \mathfrak{R}^n$  is an arbitrary vector to be evaluated in the end. It will be a sufficient condition to lead the sliding surface to zero ( $\sigma(e) \rightarrow 0$ ) for the convergence of the state tracking error to zero ( $e \rightarrow 0$ ).

Assuming our model as:

$$\dot{x} = Ax + Bu + f(x) \quad (3.7)$$

where for our case  $x \in \mathfrak{R}^n$ ,  $u \in \mathfrak{R}^n$ ,  $A \in \mathfrak{R}^{n \times n}$ ,  $B \in \mathfrak{R}^{n \times n}$ ,  $f(x)$  acting as the deviation from linearity because of modeling errors and environmental disturbances. Feedback control input can be taken as:

$$u = \bar{u} + \hat{u} \quad (3.8)$$

where  $\hat{u}$  is the linear feedback part of model and  $\bar{u}$  is the nonlinear feedback control that has a compensating effect. Nominal part of control is chosen as:

$$\hat{u} = -k^T x \quad (3.9)$$

where  $k$  stating the feedback gain vector. Applying this input into our linear model we retrieve the closed loop dynamics:

$$\dot{x} = (A - Bk^T)x + B\bar{u} + f(x) = A_c x + B\bar{u} + f(x) \quad (3.10)$$

Here feedback gain vector can be determined by pole placement or optimal control methods. To find the compensating part of the control input, we have to keep (3.10) satisfying that  $\sigma(e) \rightarrow 0$ , which requires  $\dot{\sigma}(e) < 0$ . From the definition of sliding surface we know that  $\dot{\sigma}(e) = s^T(\dot{x} - \dot{x}_d)$  hence multiplying (3.10) with  $s^T$  from left and subtracting  $s^T \dot{x}_d$  from both side yields:

$$\dot{\sigma}(e) = s^T A_c x + s^T B \bar{u} + s^T f(x) - s^T \dot{x}_d \quad (3.11)$$

Assuming that  $s^T B \neq 0$ , we choose compensating part of control,  $\bar{u}$  as:

$$\bar{u} = (s^T B)^{-1} [s^T \dot{x}_d - s^T \hat{f}(x) - \eta \operatorname{sgn}(\sigma)] \quad (3.12)$$

and applying to the equation yields:

$$\dot{\sigma}(e) = s^T A_c x - \eta \operatorname{sgn}(\sigma(e)) + s^T \Delta f(x) \quad (3.13)$$

Now we work on  $s$ . We know that if  $\lambda$  stating the eigenvalue of an arbitrary matrix  $M$ , following equation satisfies with a nonzero vector  $\bar{v}$ ,

$$M \bar{v} = \lambda \bar{v} \quad (3.14)$$

Then assigning one of the eigenvalues of  $A_c$  as zero, the term  $s^T A_c x$  in (3.13) can be made zero by taking vector  $s$  as the right eigenvector of  $A_c^T$  corresponding to the eigenvalue with zero value.

Eliminating the term  $s^T A_c x$  in (3.13) yields:

$$\dot{\sigma}(e) = -\eta \operatorname{sgn}(\sigma(e)) + s^T \Delta f(x) \quad (3.15)$$

This terms is global asymptotically stable in case of,

$$\eta > \|s\| \cdot \|\Delta f(x)\| \quad (3.16)$$

which can be shown by the *Barbälät's* lemma first by selecting a candidate Lyapunov function as:

$$V(\sigma) = \frac{1}{2} \sigma^2 \quad (3.17)$$

which ensures that  $V(\sigma) > 0$  then differentiating  $V$ , we get:



$$\dot{V}(\sigma) = \sigma \dot{\sigma} = -\eta \sigma \operatorname{sgn}(\sigma) + \sigma s^T \Delta f(x) = -\eta |\sigma| + \sigma s^T \Delta f(x) \quad (3.18)$$

from that equation it is clear that selecting  $\eta$  as stated in (3.16),  $\dot{V}(\sigma)$  becomes negative semi definite ( $\dot{V}(\sigma) \leq 0$ ). Lastly taking second derivative of  $V$  yields:

$$\ddot{V}(\sigma) = \eta^2 \operatorname{sgn}^2(\sigma) - \eta s^T \Delta f(x) \operatorname{sgn}(\sigma) - \eta \operatorname{sgn}(\sigma) s^T \Delta f(x) + (s^T \Delta f(x))^2 + \sigma s^T \Delta \dot{f}(x) \quad (3.19)$$

It can be easily seen that  $\ddot{V}(\sigma)$  is bounded. Hence (3.17), (3.18) and (3.19) satisfies *Barbălat's* lemma asserting that:

if,

1.  $V(\sigma, t)$  is lower bounded. ( $V(\sigma) \geq 0$ )
2.  $\dot{V}(\sigma, t)$  is negative semi definite. ( $\dot{V}(\sigma) \leq 0$ )
3.  $\dot{V}(\sigma, t)$  is uniformly continuous in time. ( $\ddot{V}(\sigma)$  is bounded  $\forall t \geq t_0$ )

then  $V(\sigma, t) \rightarrow 0$  as  $t \rightarrow \infty$ .

That fact in accordance with (3.16) brings the consequence of  $\sigma$  to converge to zero in finite time.

$$V(\sigma, t) \rightarrow 0 \implies -\eta |\sigma| + \sigma s^T \Delta f(x) \rightarrow 0 \implies \sigma \rightarrow 0 \quad (3.20)$$

To conclude our calculations, combining two parts of control effort as stated in (3.8), our control law becomes,

$$u = -k^T x + (s^T B)^{-1} [s^T \dot{x}_d - s^T \hat{f}(x) - \eta \operatorname{sgn}(\sigma)] \quad (3.21)$$

In practice to reduce the chattering effect instead of  $\operatorname{sgn}(\sigma)$  function,  $\operatorname{sat}(\sigma/\phi)$  or  $\tanh(\sigma/\phi)$  functions are used with  $\phi$  appears as the sliding surface boundary layer thickness.

Finally we retrieve the modified control law as:

$$u = -k^T x + (s^T B)^{-1} [s^T \dot{x}_d - s^T \hat{f}(x) - \eta \tanh(\sigma/\phi)] \quad (3.22)$$

### Steer Control with SMC

First we started our calculations by linearizing the system around an equilibrium point and substituted the coefficients in the linearized matrix. Decoupling the system for an efficient control we retrieved the following matrix for steer control.

Here  $v$  stands for sway,  $r$  for yaw and  $\psi$  for the Euler Angle for steer.

$$\begin{bmatrix} m - Y_{\dot{v}} & -Y_{\dot{r}} & 0 \\ -N_{\dot{v}} & (I_{xz} - N_{\dot{r}}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} Y_v & mu - Y_r & 0 \\ N_v & N_r & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ r \\ \psi \end{bmatrix} = \begin{bmatrix} 0 \\ \tau_N \\ 0 \end{bmatrix} \quad (3.23)$$

The sliding surface is taken as

$$\sigma = s_1 v + s_2 r + s_3 (\psi - \psi_d) \quad (3.24)$$

and our model transformed into

$$\dot{x} = Ax + Bu \quad (3.25)$$

form via MATLAB yielding

$$\dot{x} = \begin{bmatrix} -0.0943 & -0.0322 & 0 \\ -0.9537 & -1.1733 & 0 \\ 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} -1.6627 \\ 4.5960 \\ 0 \end{bmatrix} u \quad (3.26)$$

Then we find  $A_c$  matrix

$$A_c = (A - Bk^T) = \begin{bmatrix} -0.3653 & -0.2994 & 0 \\ -0.2047 & -0.4347 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (3.27)$$

where the vector  $k$  is found by the pole-placement method. Choosing the poles as

$$p = [-0.15; -0.65; 0]^T \quad (3.28)$$

we retrieve the  $k$  vector as

$$k = [-0.1630; -0.1607; 0]^T \quad (3.29)$$

Equating  $k_3$  to 0 shows us that feedback from  $\psi$  has no effect to stabilize the sway-yaw dynamics. On the other hand, in order to simplify our equation we find the right eigenvector  $s$  of  $A_c$  corresponding to  $\lambda_3$ , which is equal to 0.

Then the right eigenvector is found as

$$s = [-0.4762; 0.8496; 0.2268]^T \implies A_c^T s = 0 \quad (3.30)$$

hence our sliding surface  $\sigma$ , takes the form

$$\sigma = -0.4762v + 0.8496r + 0.2268(\psi - \psi_d) \quad (3.31)$$

Lastly after finding all the coefficients, our control input turns into

$$u = 0.1630v + 0.1607r + \frac{1}{4.6965} (-0.23 \tanh(-0.4762v + 0.8496r + 0.2268(\psi - \psi_d)/0.05)) \quad (3.32)$$

Here the sliding surface boundary layer thickness is selected as 0.05 which seems to be a proper value and  $\eta$  is selected as 0.23 in accordance with the balance between robustness and performance.

## Optimal Control

Like the behaviour of nature which is designed in a manner of always selecting the minimum effort with maximum performance, every system can be controlled with a minimum effort. Therefore in order to find the minimum effort we use Optimal Control.

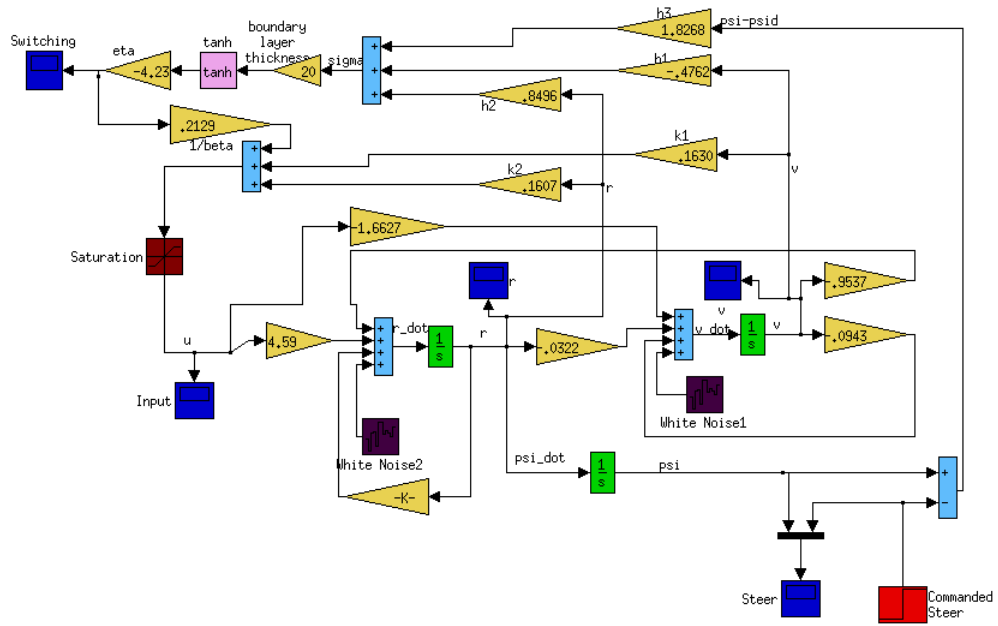


Figure 3.3: Model of Sliding Mode Control for Steer

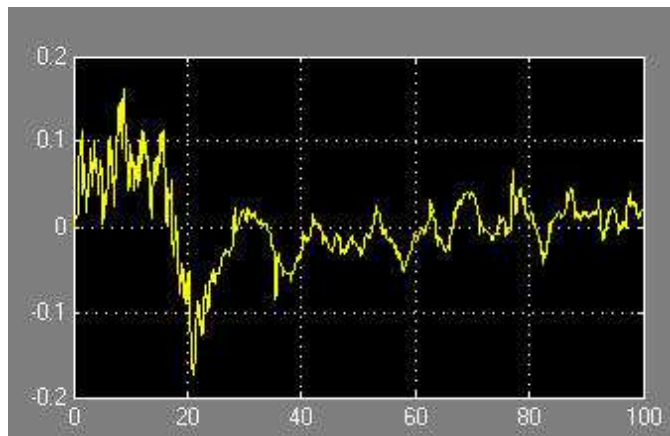


Figure 3.4: Input for Sliding Mode Control-Steer

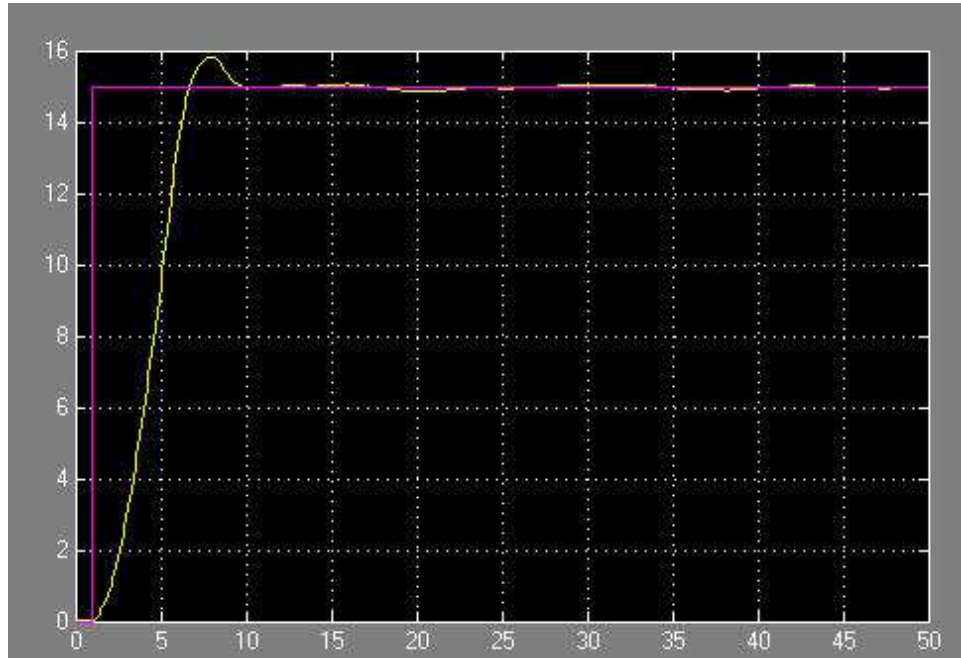


Figure 3.5: Steering for Sliding Mode Control-Steer

Before starting work for optimal control, we know that our system is linear hence selecting a quadratic performance function we can apply the Linear Quadratic Regulator (LQR) rules, which requires a linear system and quadratic performance index to minimize in order to get minimum control effort. Here our aim is to find a control input which will guide the system to follow a desired state variable and meanwhile minimize a performance index which is chosen for minimum energy control system in conformity with our situation.

Considering a linear time-invariant system stated as

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (3.33)$$

with a cost functional

$$\begin{aligned}
J(u) &= J(x(t_0), u(t), t_0) & (3.34) \\
&= \frac{1}{2} x^T(t_f) F x(t_f) \\
&\quad + \frac{1}{2} \int_{t_0}^{t_f} [x^T(t) Q x(t) + u^T(t) R u(t)] dt \\
&= \frac{1}{2} x^T(t_f) F x(t_f) \\
&\quad + \frac{1}{2} \int_{t_0}^{t_f} [x^T(t) \ u^T(t)] \begin{bmatrix} Q(t) & 0 \\ 0 & R(t) \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt
\end{aligned}$$

where error  $e(t)$ , states the difference between the reference (desired state)  $x_d$  and state as  $e(t) = x - x_d$ . For infinite time interval case *final cost function* in the cost functional  $J(u)$  does not have any meaning for the system and the final time  $t_f$ , is taken as infinite ( $t_f \rightarrow \infty$ ). Hence our cost functional can be written in a more simplified way as

$$J(u) = + \frac{1}{2} \int_{t_0}^{\infty} [x^T(t) Q x(t) + u^T(t) R u(t)] dt \quad (3.35)$$

We follow the standart procedure, first building with the *Hamiltonian* as

$$\begin{aligned}
H(x(t), u(t), \lambda(t)) &= \frac{1}{2} x^T(t) Q x(t) + \frac{1}{2} u^T(t) R u(t) & (3.36) \\
&\quad + \lambda^T(t) [A x(t) + B u(t)]
\end{aligned}$$

where  $\lambda(t)$  is the costate vector.

Then we obtain the optimal control  $u(t)$  as

$$\frac{\partial H}{\partial u} = 0 \implies R u(t) + B^T \lambda(t) = 0 \quad (3.37)$$

from which we retrieve the control input as

$$u(t) = -R^{-1} B^T \lambda(t) \quad (3.38)$$

after that we obtain the state and costate equations as

$$\dot{x}(t) = + \left( \frac{\partial H}{\partial \lambda} \right) \implies \dot{x}(t) = A x(t) + B u(t) \quad (3.39)$$

$$\dot{\lambda}(t) = - \left( \frac{\partial H}{\partial x} \right) \implies \dot{\lambda}(t) = -Qx(t) - A^T \lambda(t) \quad (3.40)$$

with using (3.38) in (3.39) we get

$$\dot{x}(t) = Ax(t) - BR^{-1}B^T \lambda(t) \quad (3.41)$$

In the infinite final-time interval optimal control, we have to satisfy that the system is completely controllable, which requires that the controllability matrix,

$$C = [B \ AB \ \dots \ A^{n-1}B] \quad (3.42)$$

must be *nonsingular* (*n linearly independent columns*). Having a controllable system guarantees that the optimal cost is *finite*.

Assuming a transformation

$$\lambda(t) = Px(t) \quad (3.43)$$

where  $P$  is not known yet, our new optimal control becomes

$$u(t) = -R^{-1}B^T \hat{P}x(t) \quad (3.44)$$

now which stands as a negative feedback of our error vector  $x(t)$ . Here,

$$\hat{P} = \lim_{t_f \rightarrow \infty} \{P\} \quad (3.45)$$

Continuing our replacements with differentiating (3.43) w.r.t. time yields,

$$\dot{\lambda}(t) = \dot{\hat{P}}x(t) + \hat{P}\dot{x}(t) \quad (3.46)$$

and using it in the state and costate equations we get,

$$\dot{x}(t) = A x(t) - B R^{-1} B^T \hat{P} x(t) \quad (3.47)$$

$$\dot{\lambda}(t) = -Q x(t) - A^T \hat{P} x(t) \quad (3.48)$$

Lastly, substituting our new equations (3.47) and (3.48) in (3.46) results,

$$\begin{aligned} -Q x(t) - A^T \hat{P} x(t) &= \dot{\hat{P}} x(t) + \hat{P} [A x(t) - B R^{-1} B^T \hat{P} x(t)] \\ 0 &= \left[ \dot{\hat{P}} + \hat{P} A + A^T \hat{P} + Q - \hat{P} B R^{-1} B^T \hat{P} \right] x(t) \end{aligned} \quad (3.49)$$

$$(3.50)$$

The above equation is called *differential Riccati equation (DRE)*, where the matrix  $\hat{P}$  is often called *Riccati matrix*.  $\hat{P}$  is an  $n \times n$  symmetric, positive definite matrix found by the solution of the *DRE*, satisfying the final condition

$$\hat{P}(t_f \rightarrow \infty) = 0 \quad (3.51)$$

Our optimal state is the found as

$$\dot{x}(t) = [A - B R^{-1} B^T \hat{P}] \quad (3.52)$$

and the optimal cost is retrieved by

$$J = \frac{1}{2} x^T(t) \hat{P} x(t) \quad (3.53)$$

## Steer Control with Optimal Control

We applied Optimal Control to our linearized steer matrix, which is stated on the previous section. Hence substituting the coefficients we retrieved the following matrix



$$\dot{x} = \begin{bmatrix} -0.0943 & -0.0322 & 0 \\ -0.9537 & -1.1733 & 0 \\ 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} -1.6627 \\ 4.5960 \\ 0 \end{bmatrix} u \quad (3.54)$$

Then we used the optimal control equations stated above and built a MATLAB m-file to retrieve the desired output. The following figures shows the outputs obtained by the m-files which are given on the Appendix A.

Here  $Q$  and  $R$  matrices in cost functional are taken as

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.55)$$

$$R = 1/4$$

Also we preferred to find the Riccati matrix  $\hat{P}$  from MATLAB with  $lqr$  command as

$$[K, P, Eig] = lqr(A, B, Q, R) \quad (3.56)$$

which yielded the  $K, P, Eig$  matrices as

$$K = [-0.59491.95052.0000] \quad Eig = \begin{bmatrix} -9.7858 \\ -1.0377 \\ -0.3977 \end{bmatrix}$$

$$P = \begin{bmatrix} 1.1077 & 0.3684 & 0.4409 \\ 0.3684 & 0.2394 & 0.2683 \\ 0.4409 & 0.2683 & 1.3042 \end{bmatrix}$$

## Steer Control with PID

Lastly for steer, we selected PID method for basic control and compared the efficiency with the other methods. Again we designed a simple Simulink diagram, figure of which is shown below. Then we found the control effort and the output of the system for PID.

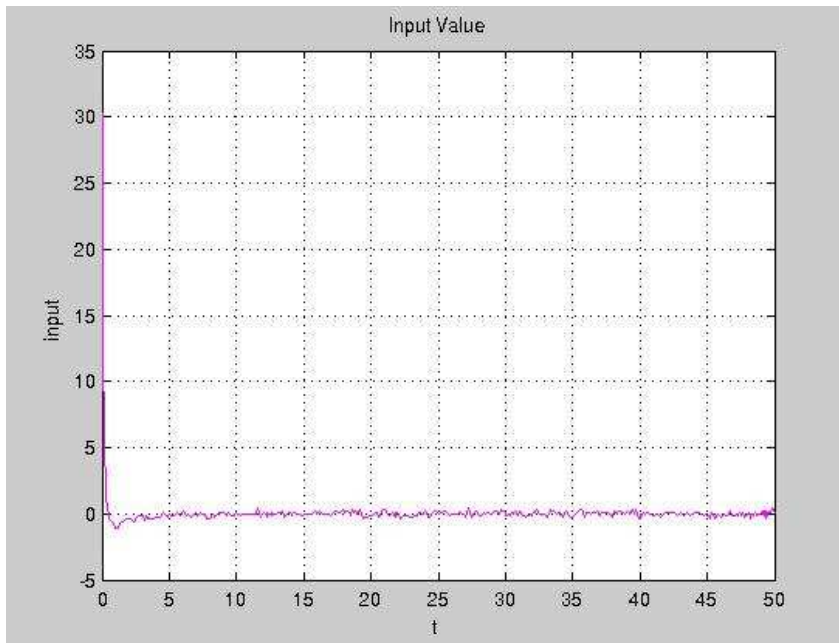


Figure 3.6: Control Input for Optimal Control-Steer



Figure 3.7: Steering for Optimal Control-Steer

The fact, that PID is not strong as much as SMC in compensating the noise and other disturbing effects, can be easily seen from Figure 10. Especially in harsh conditions like sea water, because of the inefficiency of our control module, undesired circumstances may occur. However simplicity of the control method comes forth with an advantage of less effort which means less energy consumption for our vehicle.

Here again, the same amount of white Gaussian noise is injected to the system.

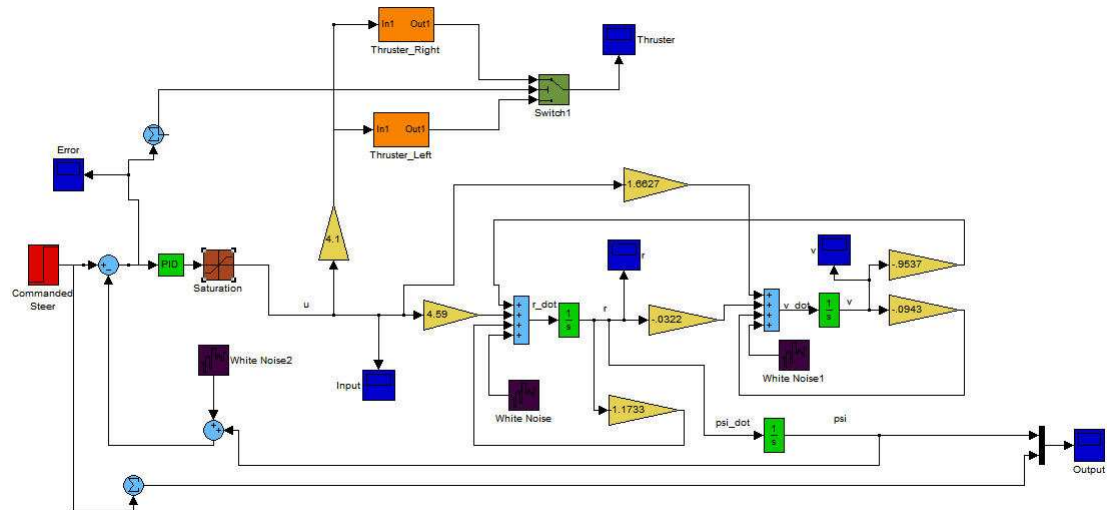


Figure 3.8: Model of Simulink PID Control for Steer

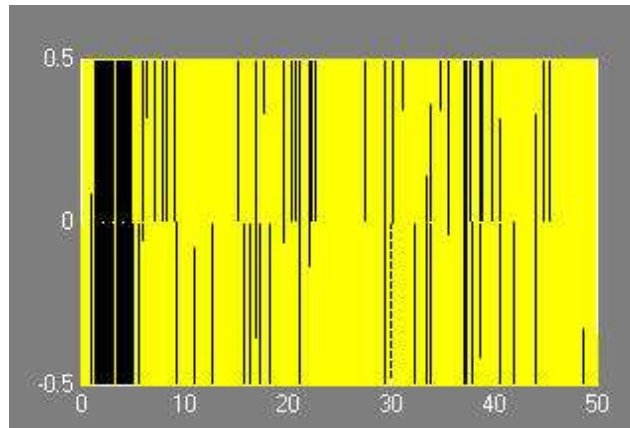


Figure 3.9: Input for PID Control

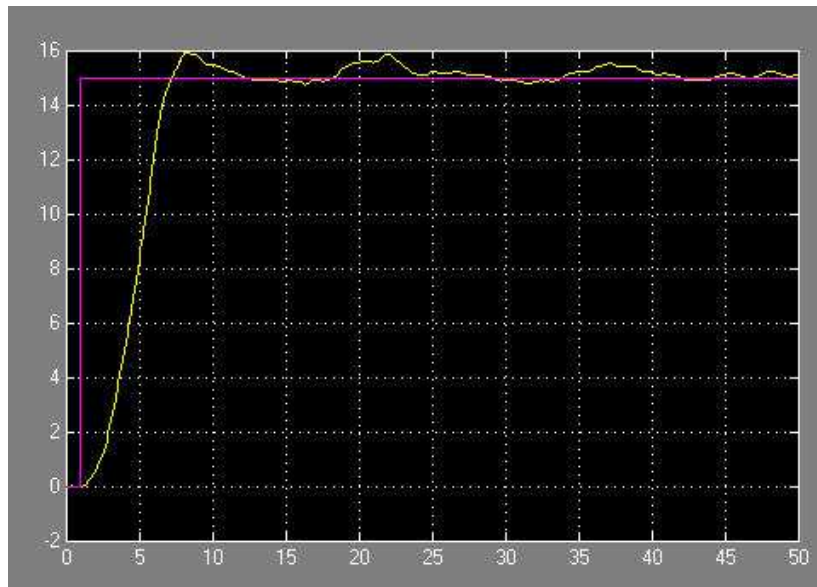


Figure 3.10: Steering for PID Control-Steer

## Depth Control

On the depth control we followed the same route as we chased in steer control. We used the similar formulas with different coefficients hence retrieved different outputs. But the main idea stays steady. For depth control, first we explained PID method and showed figures, then SMC and lastly optimal control. Like in the steering design, control methods responded in the same manner. SMC confirmed its robustness and stability in compensating noise more than other methods. Optimal control followed SMC, but evinced less efficiency. Lastly PID came into scene when the goal is robustness, but whatever the outputs are, we realised that also when the PID coefficients are adjusted correctly desired outputs are not afar. Depth is not taken a high value due to the fact that the system will response in the same manner, no matter how deep the vehicle dives.

### Depth Control with PID

This time we started first with PID method because of its direct approach. Coefficients are selected according to the response of the system. Simulink diagram of depth control design is built as

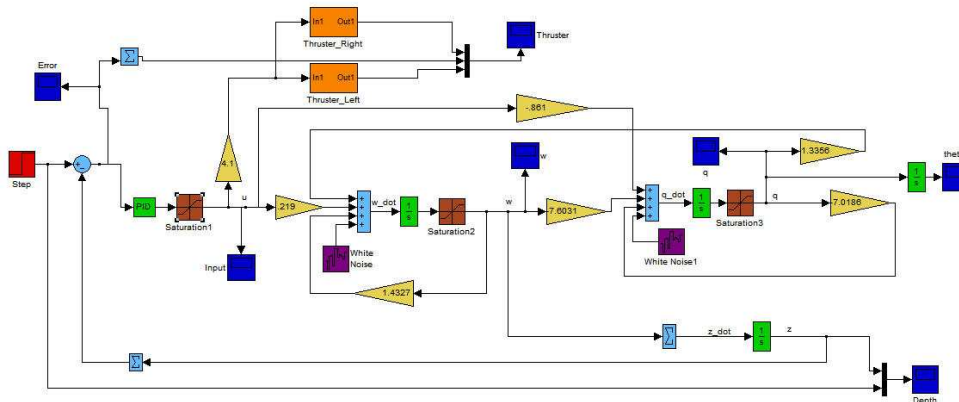


Figure 3.11: Model of Simulink PID Control for Depth

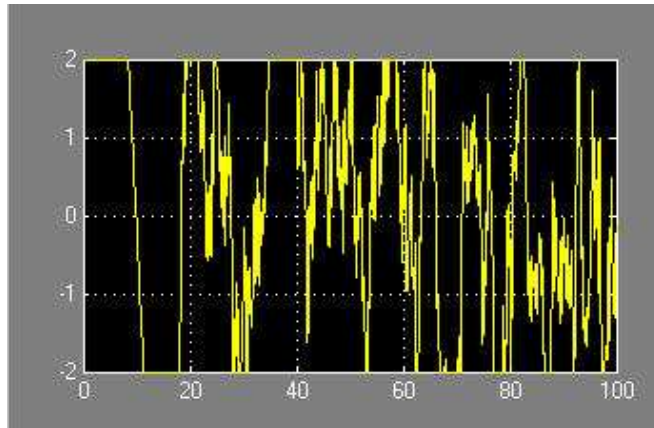


Figure 3.12: Input Value of Simulink PID Control for Depth

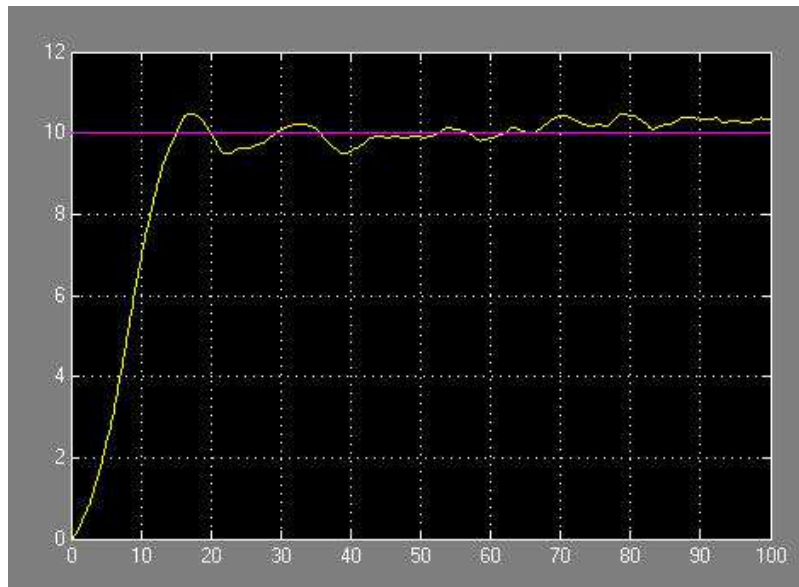


Figure 3.13: Desired Depth of PID for Depth Control

## Depth Control with SMC

In depth control with SMC, we assumed that we achieve diving operation vertically without adding the pitch effect which means, we do not use our thrusters to maintain a desired pitch angle in order to dive, where we obtain a direct thrust vector perpendicular to sea surface. On the other hand, by resultant thrust vector which is the sum of the vertical thrust and horizontal thrust, our vehicle will decline on a slope.

After linearizing our system we get the following matrix for depth. Here  $w$  stands for heave,  $q$  for pitch and  $z$  for the depth.

$$\begin{bmatrix} m - Z_{\dot{w}} & mx_G - Z_{\dot{q}} & 0 \\ mx_G - W_{\dot{w}} & (I_y - M_{\dot{q}}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{w} \\ \dot{q} \\ \dot{z} \end{bmatrix} + \begin{bmatrix} Z_w & -mu + Z_q & 0 \\ M_w & mx_G u + M_q & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} w \\ q \\ z \end{bmatrix} = \begin{bmatrix} \tau_z \\ 0 \\ 0 \end{bmatrix} \quad (3.57)$$

The sliding surface is taken as

$$\sigma = s_1 w + s_2 q + s_3 (z - z_d) \quad (3.58)$$

and our model transformed into

$$\dot{x} = Ax + Bu \quad (3.59)$$

form via MATLAB yielding

$$\dot{x} = \begin{bmatrix} 1.4327 & 1.3356 & 0 \\ -7.6031 & -7.0186 & 0 \\ 1 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 2.1937 \\ -8.6137 \\ 0 \end{bmatrix} u \quad (3.60)$$

Then we find  $A_c$  matrix

$$A_c = (A - Bk^T) = \begin{bmatrix} -0.5061 & -0.3541 & 0 \\ -0.0099 & -0.3839 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (3.61)$$

where the vector  $k$  is found by the pole-placement method. Choosing the poles as

$$p = [0; -5.5680; -0.0179]^T \quad (3.62)$$

we retrieve the  $k$  vector as

$$k = [-0.8838; 0.7703; 0]^T \quad (3.63)$$

Equating  $k_3$  to 0 shows us that feedback from  $z$  has no effect to stabilize the heave-pitch dynamics. On the other hand, in order to simplify our equation we find the right eigenvector  $s$  of  $A_c$  corresponding to  $\lambda_3$ , which is equal to 0.

Then the right eigenvector is found as

$$s = [0.6874; -0.6340; 0.3542]^T \implies A_c^T s = 0 \quad (3.64)$$

hence our sliding surface  $\sigma$ , takes the form

$$\sigma = 0.6874w - 0.6340q + 0.3542(z - z_d) \quad (3.65)$$

Lastly after finding all the coefficients, our control input turns into

$$u = -0.8838w - 0.7703q \quad (3.66) \\ + \frac{1}{6.9693} (-0.11 \tanh(0.6874w - 0.6340q + 0.3542(z - z_d)/0.05))$$

Here the sliding surface boundary layer thickness is selected as 0.05 which seems to be a proper value and  $\eta$  is selected as 0.11 in accordance with the balance between robustness and performance.



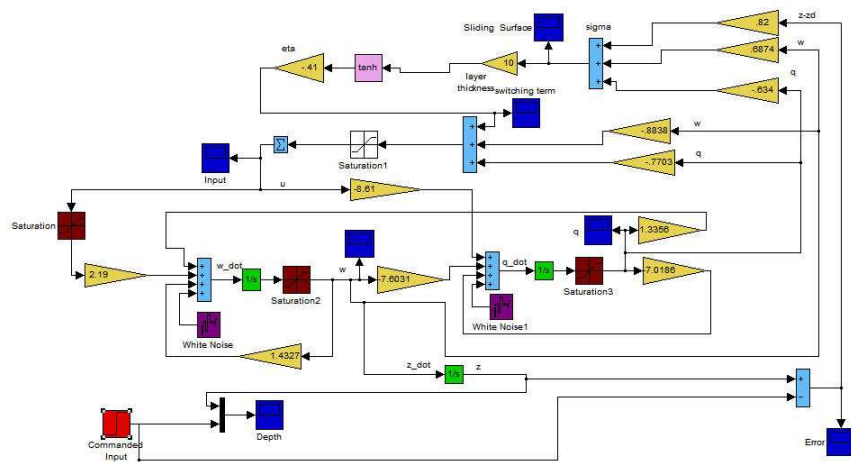


Figure 3.14: Model of Sliding Mode Control for Depth

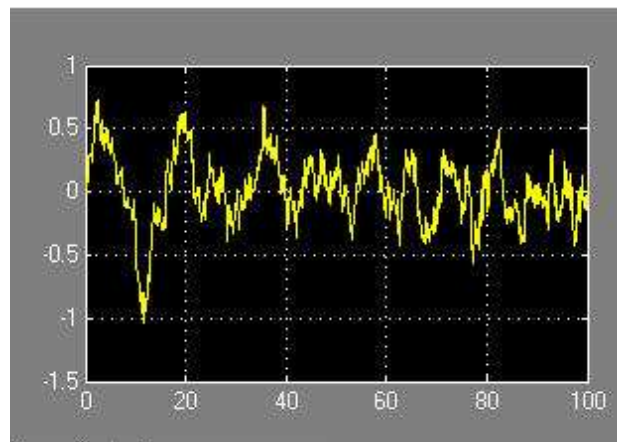


Figure 3.15: Input for Sliding Mode Control-Depth

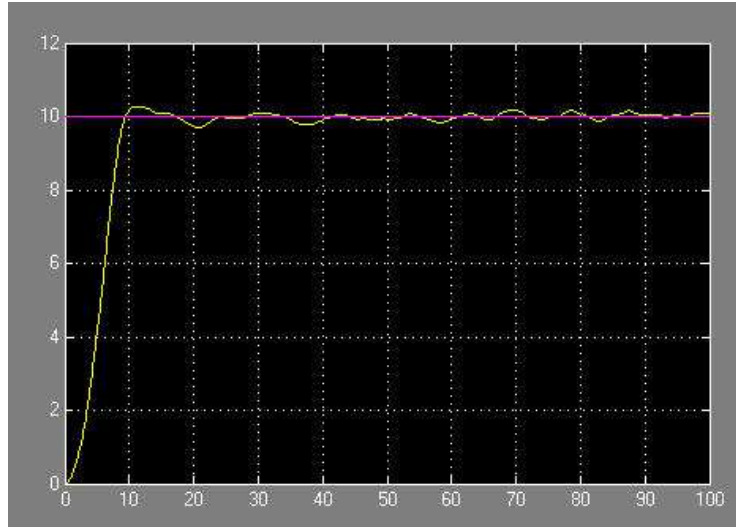


Figure 3.16: Depth Control for Sliding Mode Control-Depth

### Depth Control with Optimal Control

Optimal Control is applied to the same matrix used in SMC which is retrieved by linearizing the system for depth control. Only the coefficients of the linearized matrix are different from the linear quadratic regulator. Hence substituting the coefficients we retrieved the following matrix

$$\dot{x} = \begin{bmatrix} 1.4327 & 1.3356 & 0 \\ -7.6031 & -7.0186 & 0 \\ 1 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 2.1937 \\ -8.6137 \\ 0 \end{bmatrix} u \quad (3.67)$$

Then we used the optimal control equations stated above and built a MATLAB m-file to retrieve the desired output.

Here again  $Q$  and  $R$  matrices in cost functional are taken as the same with matrices in steer control

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.68)$$

$$R = 1/4$$

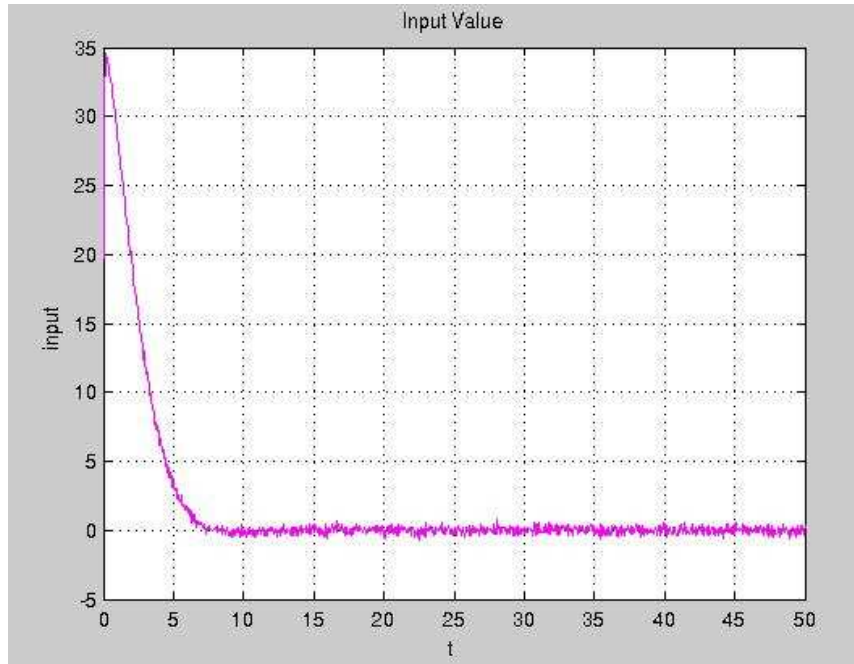


Figure 3.17: Control Input for Optimal Control-Depth

Also we preferred to find the Riccati matrix  $\hat{P}$  from MATLAB with *lqr* command as

$$[K, P, Eig] = lqr(A, B, Q, R) \quad (3.69)$$

which yielded the  $K, P, Eig$  matrices as

$$K = [-0.5949 \ 1.9505 \ 2.0000]$$

$$P = \begin{bmatrix} 2.8419 & 0.6208 & 1.7216 \\ 0.6208 & 0.2394 & 0.3804 \\ 1.7216 & 0.3804 & 2.1995 \end{bmatrix}$$

### 3.1 LQG Design

In LQR design we have supposed that we have all the state information at each step but in real time design we will not be so lucky. Environmental disturbances like currents, wind,



Figure 3.18: Depth Control with Optimal Control-Depth

waves, etc. will cause to system noise and also state information acquired from sensors will not be so definite because of the measurement noise hence system will not be controlled efficiently as desired. Generally noise will set in too much oscillation which will end with too much chattering at the input for compensation. Therefore we need a filter in order to obtain optimum state estimates. In that manner Kalman filter is selected, where in case of white noise it is the optimal filter. In our design we assumed that the all noises are white otherwise we would design a Colored-Noise Kalman filter. Combining the LQR feedback design with an estimator forms the optimal system known as "Separation Principle". In this theorem, design is achieved in two stages. First, states are retrieved from Kalman filter as optimal estimates then optimal control problem is solved with these known states.

In LQG design we followed the Separation Theorem hence first we designed an optimal regulator assuming full-state feedback for linear system then we designed a continuous-time Kalman filter with white noises and known power spectral densities. Lastly we combined optimal regulator and Kalman filter into an optimal compensator which yields an input from estimated state and whence output.

## 3.2 Kalman Filter

We start designing the Kalman filter first with a truth model:

$$\dot{x}(t) = A(t) x(t) + B(t) u(t) + G(t) w(t) \quad (3.70)$$

$$\tilde{y}(t) = C(t) x(t) + v(t) \quad (3.71)$$

where

$x(t)$  : State vector

$y(t)$  : Measurement vector

$A(t)$  : State matrix

$C(t)$  : Measurement matrix

$w(t)$  : System noise and model uncertainty

$v(t)$  : Measurement noise

Both noises are assumed to be white noises with zero-mean Gaussian and uncorrelated with each other and with state.

$$E \{w(t) w^T(t)\} = Q(t) \quad (3.72)$$

$$E \{v(t) v^T(t)\} = R(t) \quad (3.73)$$

$$E \{v(t) w^T(t)\} = 0 \quad (3.74)$$

Kalman filter for state and output estimate is build as

$$\dot{\hat{x}}(t) = A(t) \hat{x}(t) + B(t) u(t) + L(t) [\tilde{y} - C(t) \hat{x}] \quad (3.75)$$

$$\tilde{y}(t) = C(t) \hat{x}(t) \quad (3.76)$$

Let's define the state error as  $e = \hat{x} - x$ . Combining the truth model with Kalman form we get

$$\begin{aligned} \dot{e} &= (A(t) \hat{x}(t) + B(t) u(t) + L(t) [\tilde{y} - C(t) \hat{x}]) - (A(t) x(t) + B(t) u(t) + G(t) w(t)) \\ &= A(t) e + L(t) C(t) x + L(t) v(t) - L(t) C(t) \hat{x} - G(t) w(t) \\ &= (A(t) - L(t) C(t)) e - G(t) w(t) + L(t) v(t) \end{aligned} \quad (3.77)$$

We define the state error-covariance matrix as

$$P(t) \equiv E \{e(t) e^T(t)\} \quad (3.78)$$

Here we need the solution of the linear time invariant state equation, which we reach the solution by the following way. Considering the linear time-invariant state-space equation

$$\dot{x}(t) = A(t) x(t) + B(t)u(t) \quad (3.79)$$

$$y(t) = C(t)x(t) + Du(t) \quad (3.80)$$

We know that,

$$\frac{d}{dt} e^{A(t)t} = A(t) e^{A(t)t} = e^{A(t)t} A(t)$$

By multiplying the both sides of equation (3.79) with  $e^{-A(t)t}$  from left gives

$$e^{-A(t)t} \dot{x}(t) - e^{-A(t)t} A(t) x(t) = e^{-A(t)t} B(t) u(t)$$

which is equal to

$$\frac{d}{dt} \left( e^{-A(t)t} x(t) \right) = e^{-A(t)t} B(t) u(t)$$

Integrating both sides yields

$$e^{-A(t)t} x(t) \Big|_{\tau=0}^t = \int_0^t e^{-A(t)\tau} B(\tau) u(\tau) d\tau$$

Solving the integral at given bounds yields

$$e^{-A(t)t} x(t) - e^0 x(t_0) = \int_{t_0}^t e^{-A(t)\tau} B(\tau) u(\tau) d\tau \quad (3.81)$$

Lastly multiplying both sides with  $e^{A(t)t}$  to leave  $x(t)$  alone at left-hand side, we find the state equation solution

$$x(t) = e^{A(t)t} x(t_0) + \int_{t_0}^t e^{A(t)(t-\tau)} B(\tau) u(\tau) d\tau \quad (3.82)$$

In our case defining  $\alpha(t) = -G(t)w(t) + L(t)v(t)$  and  $\beta(t) = A(t) - L(t)C(t)$  we find our solution as

$$e = e^{\beta t} e(t_0) + \int_{t_0}^t e^{\beta(t)(t-\tau)} \alpha(\tau) d\tau \quad (3.83)$$

Using the fact that  $v(t)$  and  $w(t)$  are uncorrelated we get

$$\begin{aligned} E \{ \alpha(t) \alpha^T(t) \} &= (-G(t)w(t) + L(t)v(t)) * (-G(t)w(t) + L(t)v(t))^T \\ &= G(t) \underbrace{w(t)w^T(t)}_{Q(t)} G(t)^T - G(t) \underbrace{w(t)v^T(t)}_0 L(t)^T - L(t) \underbrace{v(t)w^T(t)}_0 G(t)^T \\ &\quad + L(t) \underbrace{v(t)v^T(t)}_R L(t)^T \\ &= [G(t)Q(t)G(t)^T + L(t)R(t)L(t)^T] \end{aligned} \quad (3.84)$$

Using our solution we find the state error-covariance matrix as

$$\begin{aligned}
P(t) &= E \left\{ \left( e^{\beta(t)t} e(t_0) + \int_{t_0}^t e^{\beta(t)(t-\tau)} \alpha(\tau) d\tau \right) * \left( e^{\beta(t)t} e(t_0) + \int_{t_0}^t e^{\beta(t)(t-\tau)} \alpha(\tau) d\tau \right)^T \right\} \\
&= e^{\beta(t)t} \underbrace{e(t_0) e^T(t_0)}_{P(t_0)} e^{\beta(t)Tt} + \int_{t_0}^t e^{\beta(t)t} e^{\beta(t)T(t-\tau)} \underbrace{e(t_0) \alpha^T(\tau)}_0 d\tau \\
&+ \int_{t_0}^t e^{\beta(t)Tt} e^{\beta(t)(t-\tau)} \underbrace{\alpha(\tau) e^T(t_0)}_0 d\tau + \int_{t_0}^t e^{\beta(t)(t-\tau)} \underbrace{\alpha(\tau) \alpha^T(\tau)}_{G(t)Q(t)G(t)^T + L(t)RL(t)^T} e^{\beta(t)T(t-\tau)} d\tau \\
&= e^{\beta(t)t} P(t_0) e^{\beta(t)Tt} + \int_{t_0}^t e^{\beta(t)(t-\tau)} [G(t)Q(t)G(t)^T + L(t)RL(t)^T] e^{\beta(t)T(t-\tau)} d\tau
\end{aligned} \tag{3.85}$$

Above we use the fact that

$$\left( e^{\beta(t)t} \right)^T = e^{\beta(t)Tt}$$

Derivative of state error-covariance yields

$$\begin{aligned}
\dot{P}(t) &= \frac{\partial e^{\beta(t)t}}{\partial t} P(t_0) e^{\beta(t)Tt} + e^{\beta(t)t} P(t_0) \frac{\partial e^{\beta(t)Tt}}{\partial t} \\
&+ \int_{t_0}^t \frac{\partial e^{\beta(t)(t-\tau)}}{\partial t} (G(t)Q(t)G(t)^T + L(t)Q(t)L(t)^T) e^{\beta(t)T(t-\tau)} d\tau \\
&+ \int_{t_0}^t e^{\beta(t)(t-\tau)} (G(t)Q(t)G(t)^T + L(t)Q(t)L(t)^T) \frac{\partial e^{\beta(t)T(t-\tau)}}{\partial t} d\tau \\
&+ e^{\beta(t)(t-\tau)} (G(t)Q(t)G(t)^T + L(t)Q(t)L(t)^T) e^{\beta(t)T(t-\tau)}
\end{aligned} \tag{3.86}$$

Taking the derivatives inside the error-covariance we get

$$\begin{aligned}
\dot{P}(t) &= \beta(t) e^{\beta(t)t} P(t_0) e^{\beta(t)Tt} + e^{\beta(t)t} P(t_0) e^{\beta(t)Tt} \beta(t)^T + \beta(t) P(t_0) + P(t_0) \beta(t)^T \\
&+ \beta(t) \int_{t_0}^t e^{\beta(t)(t-\tau)} (G(t)Q(t)G(t)^T + L(t)Q(t)L(t)^T) e^{\beta(t)Tt} d\tau \\
&+ \int_{t_0}^t e^{\beta(t)(t-\tau)} (G(t)Q(t)G(t)^T + L(t)Q(t)L(t)^T) e^{\beta(t)Tt} \beta(t)^T d\tau \\
&+ (G(t)Q(t)G(t)^T + L(t)Q(t)L(t)^T)
\end{aligned} \tag{3.87}$$

Lastly simplifying the equation we retrieve

$$\dot{P}(t) = [A(t) - L(t)C(t)]P(t) + P(t)[A(t) - L(t)C(t)]^T + G(t)Q(t)G(t)^T + L(t)R(t)L(t)^T \tag{3.88}$$

Now we have to calculate the gain. We find it by minimizing the cost function, which is the derivative of the state error-covariance matrix

$$\frac{\partial J}{\partial L(t)} = 2L(t)R(t) - 2P(t)C(t)^T = 0 \tag{3.89}$$

Solving this equation we find our gain as

$$L(t) = P(t)C^T(t)R^{-1}(t) \quad (3.90)$$

Substituting the gain (3.90), in equation (3.88) yields the below formula, which is known as *continuous Riccati equation*

$$\begin{aligned} \dot{P}(t) = & A(t)P(t) + P(t)A^T(t) \\ & - P(t)C^T(t)R^{-1}(t)C(t)P(t) + G(t)Q(t)G^T(t) \end{aligned} \quad (3.91)$$

The algorithm for continuous time Kalman filter is stated in the below table. Assuming the linear time varying system with white noise sequences and zero-mean Gaussian distributions, we have to first assign initial values for state and error-covariance matrix. Then we find the Kalman gain matrix and use it in updating the error-covariance matrix in next step. Here we face with a differential equation which we solved in Simulink toolbox of Matlab®.

Model	$\dot{x}(t) = A(t)x(t) + B(t)u(t) + G(t)w(t), w(t) \sim N(\mathbf{0}, Q(t))$ $\tilde{y}(t) = C(t)x(t) + v(t), v(t) \sim N(\mathbf{0}, R(t))$
Initialize	$\hat{x}(t_0) = \hat{x}_0$ $P_0 = E \{e(t_0)e^T(t_0)\}$
Gain	$K(t) = P(t)C^T(t)R^{-1}(t)$
Covariance	$\dot{P}(t) = A(t)P(t) + P(t)A^T(t)$ $- P(t)C^T(t)R^{-1}(t)C(t)P(t) + G(t)Q(t)G^T(t)$
Estimate	$\dot{\hat{x}}(t) = A(t)\hat{x}(t) + B(t)u(t) + L(t)[\tilde{y} - C(t)\hat{x}(t)]$

Table 3.1: Continuous Time Kalman Filter

Combining the optimal regulator with Kalman filter we retrieve a compensator. Regulator's weighting matrices  $\mathbf{Q}$  and  $\mathbf{R}$  and Kalman filter's spectral noise densities  $\mathbf{V}$  and  $\mathbf{Z}$  play important role in designing compensator. Using the state information retrieved from Kalman filter we get the following LQG equation which we built in Simulink.

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + Bu + L(\tilde{y} - C\hat{x}) \\ &= (A - BK - LC)\hat{x} + L\tilde{y} \end{aligned} \quad (3.92)$$

with the input value



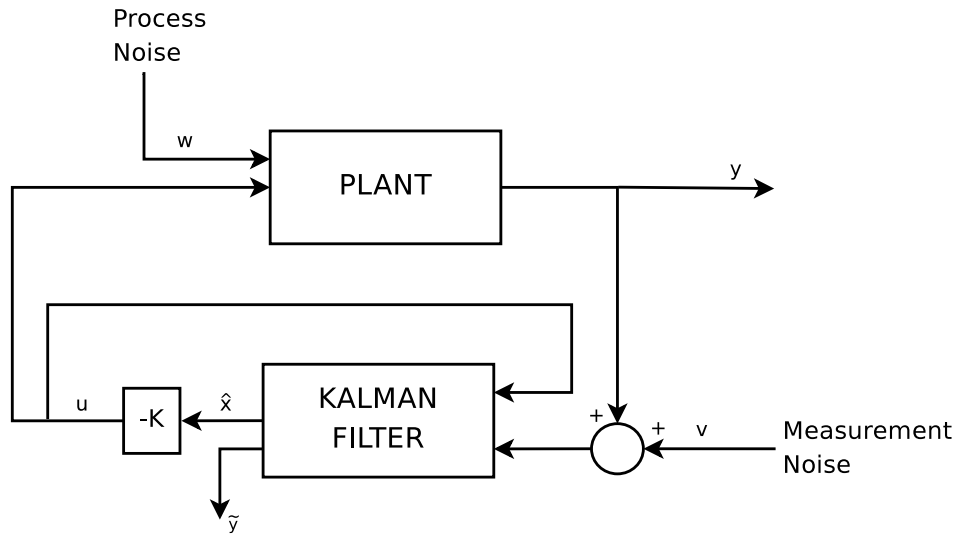


Figure 3.19: LQG Design

$$u = -K\hat{x} \quad (3.93)$$

where  $K = PBR^{-1}$ .

Our design for LQR/LQG Control is based on separation principle therefore combining each controller on the same Simulink diagram would be more appropriate in order to compare their efficiencies.

Here

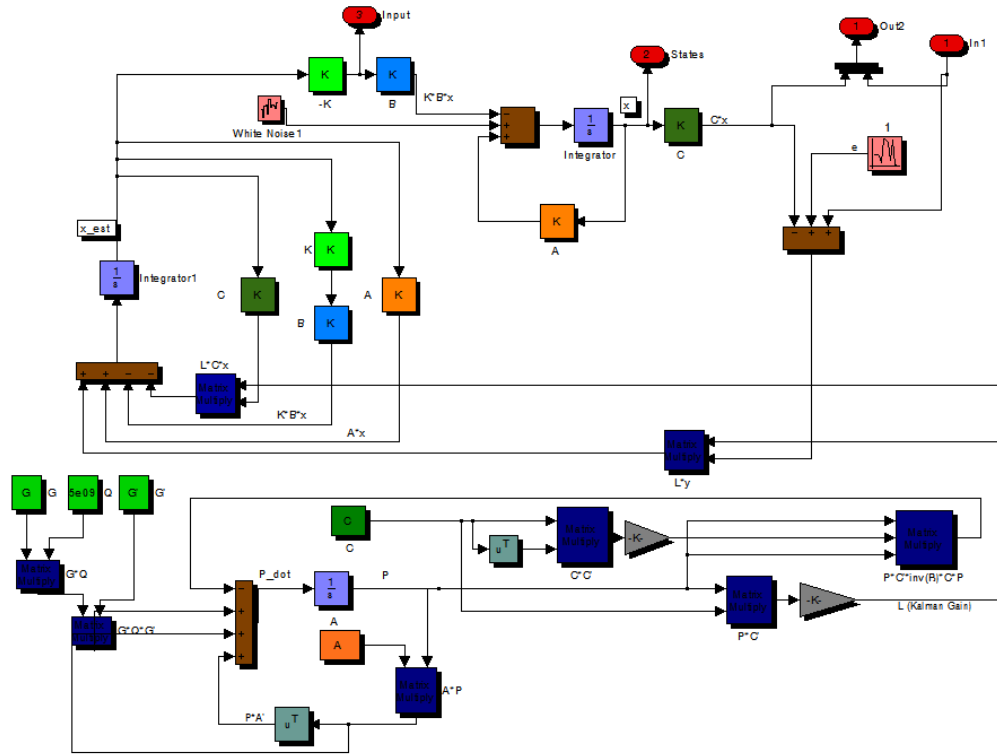


Figure 3.20: Simulink LQG Steer Subblock

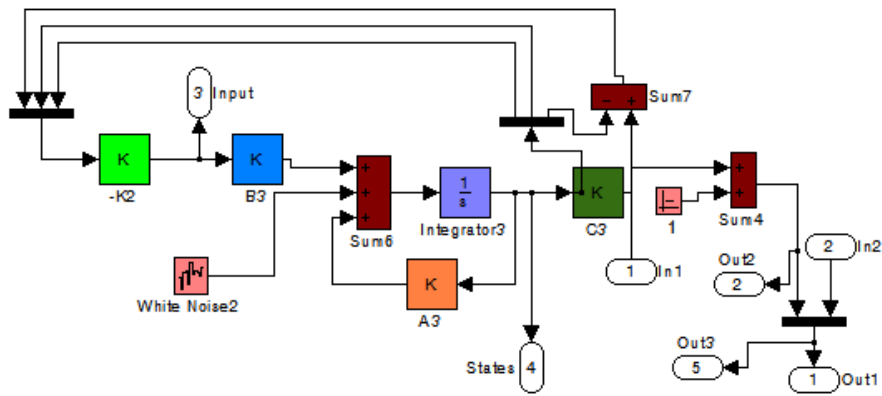


Figure 3.21: Simulink LQR Steer Subblock

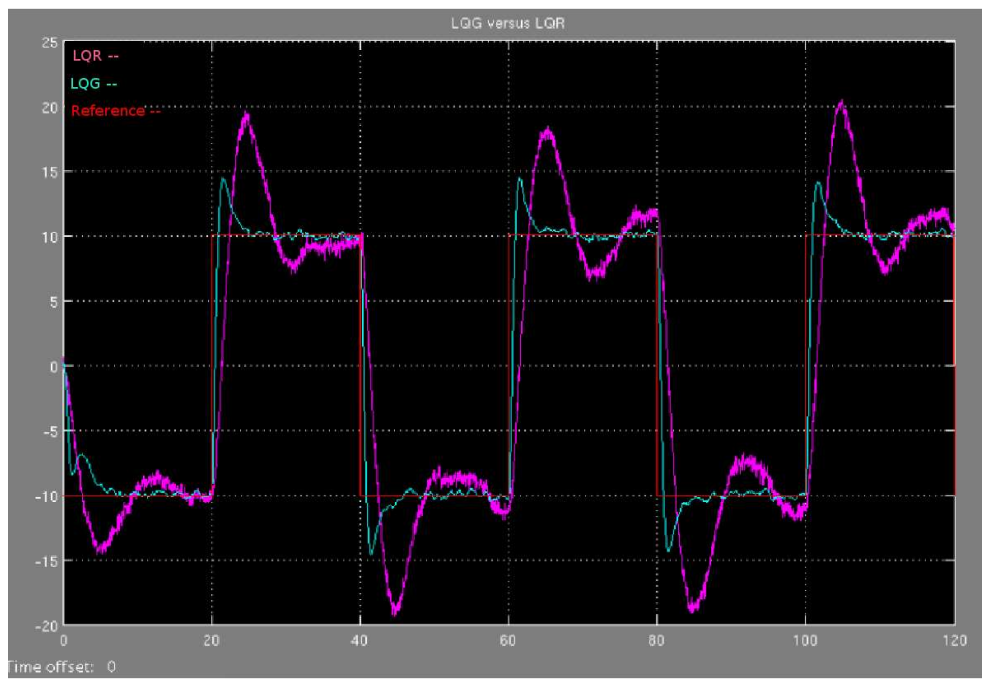


Figure 3.22: LQG Output

# CHAPTER 4

## PARAMETER ESTIMATION

### Linear Sequential Estimation

Basically linear sequential estimation method is achieved by linear least squares algorithm. In linear least squares, our goal is to estimate the unknown vector  $x$  from given measurements. Our problem can be modeled as

$$\tilde{y}_i = H_i \hat{x} + v_i \quad i = 1, \dots, k \quad (4.1)$$

where

$\tilde{y}_i$  : measurements

$H_i$  : measurement matrix

$\hat{x}$  : vector of unknowns

We try to estimate  $x$  such that the estimation error is minimum, hence we grab that solution by minimizing the square of the estimation errors with inserted weight matrix in the cost function,

$$J = \frac{1}{2} e^T W e \quad (4.2)$$

We are looking for  $\hat{x}$  that minimizes  $J$ . Here  $W$  is an  $n \times n$  symmetric matrix.  $e$  stands for error and defined as

$$e = \tilde{y}_i - H_i \hat{x}$$

our cost function can be written as

$$\begin{aligned} J(k) &= \sum_{i=1}^k [\tilde{y}_i - H_i \hat{x}]^T W_i [\tilde{y}_i - H_i \hat{x}] \\ &= \tilde{y}_k^T W_k \tilde{y}_k - \tilde{y}_k^T W_k H_k \hat{x} - \hat{x}^T H_k^T W_k \tilde{y}_k + \hat{x}^T H_k^T W_k H_k \hat{x} \end{aligned} \quad (4.3)$$

Utilizing the necessary condition for  $\hat{x}$  to be minimum, which is

$$\frac{\partial J}{\partial \hat{x}} = 0 \quad (4.4)$$

thus we find

$$\frac{\partial J}{\partial \hat{x}} = -2\tilde{y}_k^T W_k H_k + 2\hat{x}^T H_k^T W_k H_k = 0 \quad (4.5)$$

when we equate both side

$$\tilde{y}_k^T W_k H_k = \hat{x}^T H_k^T W_k H_k$$

Taking the transpose of both side

$$H_k^T W_k H_k \hat{x} = H_k^T W_k \tilde{y}_k$$

Lastly we find the least squares solution

$$\hat{x} = (H_k^T W_k H_k)^{-1} H_k^T W_k \tilde{y}_k \quad (4.6)$$

From the sufficient condition of minimum,  $\frac{\partial^2 J}{\partial \hat{x}^2} > 0$ ,  $W$  must be positive definite.

Least squares estimator is unbiased which can be defined as

$$E[e_k] = 0 \implies E[x - \hat{x}_k] = 0 \implies E[\hat{x}_k] = x \quad (4.7)$$

where  $e_k \triangleq x - \hat{x}_k$ . Selecting  $W = R^{-1}$  we get

$$\begin{aligned} E[\hat{x}_k] &= (H_k^T R_k^{-1} H_k)^{-1} H_k^T R_k^{-1} E[\tilde{y}_k] \\ &= (H_k^T R_k^{-1} H_k)^{-1} H_k^T R_k^{-1} E[H_k \hat{x} + v_k] \end{aligned} \quad (4.8)$$

Estimation error is found by

$$\begin{aligned} e_k &= x - \hat{x}_k \\ &= x - \underbrace{(H_k^T R_k^{-1} H_k)^{-1} H_k^T R_k^{-1} E[H_k \hat{x} + v_k]}_x \\ &= - (H_k^T R_k^{-1} H_k)^{-1} H_k^T R_k^{-1} v_k \end{aligned} \quad (4.9)$$

In the next step we find the information matrix as

$$\begin{aligned} P(k) &= E[e(k) e(k)^T] \\ &= (H_k^T R_k^{-1} H_k)^{-1} H_k^T R_k^{-1} \underbrace{v_k v_k^T}_{R_k} R_k^{-1} H_k (H_k^T R_k^{-1} H_k)^{-1} \\ &= H_k^{-1} R_k (H_k^T)^{-1} H_k^T R_k^{-1} R_k R_k^{-1} H_k H_k^{-1} R_k (H_k^T)^{-1} \end{aligned} \quad (4.10)$$

After cancelation we get

$$P(k) = [H_k^T R_k^{-1} H_k]^{-1} \quad (4.11)$$

Taking the inverse of both side, we can write the recursive form partitioned as

$$z_{k+1} = \begin{bmatrix} z_k \\ z(k+1) \end{bmatrix} \quad (4.12)$$

$$H_{k+1} = \begin{bmatrix} H_k \\ H(k+1) \end{bmatrix} \quad (4.13)$$

$$v_{k+1} = \begin{bmatrix} v_k \\ v(k+1) \end{bmatrix} \quad (4.14)$$

$$R_{k+1} = \begin{bmatrix} R_k & 0 \\ 0 & R(k+1) \end{bmatrix} \quad (4.15)$$

$$\begin{aligned} P_{k+1}^{-1} &= H_{k+1}^T R_{k+1}^{-1} H_{k+1} \\ &= \begin{bmatrix} H_k^T & H(k+1)^T \end{bmatrix} \begin{bmatrix} R_k & 0 \\ 0 & R(k+1) \end{bmatrix}^{-1} \begin{bmatrix} H_k \\ H(k+1) \end{bmatrix} \\ &= \begin{bmatrix} H_k^T R_k^{-1} & H(k+1)^T R_{k+1}^{-1} \end{bmatrix} \\ &= H_k^T R_k^{-1} H_k + H(k+1)^T R_{k+1}^{-1} H(k+1) \end{aligned} \quad (4.16)$$

which is equal to

$$P_{k+1}^{-1} = P_k^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1} \quad (4.17)$$

We will write the information matrix in a more compact form by using the matrix inversion lemma, where let

$$F = [A + BCD]^{-1} \quad (4.18)$$

here

$$F = n \times n \text{ matrix}$$

$$A = n \times n \text{ matrix}$$

$$B = n \times m \text{ matrix}$$

$$C = m \times m \text{ matrix}$$

$$D = m \times n \text{ matrix}$$

If we have the assumption that all parts have inverses, then we can write the above inverse as

$$F = A^{-1} - A^{-1} B (DA^{-1} B + C^{-1})^{-1} DA^{-1} \quad (4.19)$$

In our sequential algorithm selecting the matrices as

$$F = P_{k+1}$$

$$A = P_k^{-1}$$

$$B = H_{k+1}^T$$

$$C = R_{k+1}$$

$$D = H_{k+1}$$

and placing them in the matrix inverse lemma yields

$$\begin{aligned} P_{k+1} &= [P_k^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1}]^{-1} \\ &= P_k - P_k H_{k+1}^T (H_{k+1} P_k H_{k+1}^T + R_{k+1}^{-1})^{-1} H_{k+1} P_k \end{aligned} \quad (4.20)$$

In order to simply the equation we define new variables as

$$S_{k+1} = H_{k+1} P_k H_{k+1}^T \quad (4.21)$$

$$W_{k+1} = P_k H_{k+1}^T S_{k+1}^{-1} \quad (4.22)$$

hence we can write information matrix recursion

$$P_{k+1} = [I - W_{k+1} H_{k+1}] P_k \quad (4.23)$$

which is equal to

$$P_{k+1} = P_k - W_{k+1} S_{k+1} W_{k+1}^T \quad (4.24)$$

The estimation formula can be written in a recursive form

$$\hat{x}_k = \underbrace{(H_k^T R_k^{-1} H_k)}_{P_k} H_k^T R_k^{-1} \tilde{y}_k \quad (4.25)$$

for  $k+1$  it is written as

$$\begin{aligned} \hat{x}_{k+1} &= P_{k+1} H_{k+1}^T R_{k+1}^{-1} \tilde{y}_{k+1} \\ &= P_{k+1} \begin{bmatrix} H_k^T & H(k+1)^T \end{bmatrix} \begin{bmatrix} R_k & 0 \\ 0 & R(k+1) \end{bmatrix}^{-1} \begin{bmatrix} \tilde{y}_k \\ \tilde{y}(k+1) \end{bmatrix} \\ &= P_{k+1} H_k^T R_k^{-1} \tilde{y}_k + P_{k+1} H_{k+1}^T R_{k+1}^{-1} \tilde{y}_{k+1} \end{aligned} \quad (4.26)$$

when we write  $P_{k+1}$  as in (45) for the first term on the right-hand side

$$\begin{aligned}\hat{x}_{k+1} &= P_k H_k^T R_k^{-1} \tilde{y}_k - W_{k+1} H_{k+1} P_k H_k^T R_k^{-1} \tilde{y}_k + \underbrace{P_{k+1} H_{k+1}^T R_{k+1}^{-1}}_{W_{k+1}} \tilde{y}_{k+1} \\ &= [I - W_{k+1} H_{k+1}] \underbrace{P_k H_k^T R_k^{-1} \tilde{y}_k}_{\hat{x}_k} + W_{k+1} \tilde{y}_{k+1}\end{aligned}\quad (4.27)$$

in a more compact form we can write it as

$$\begin{aligned}\hat{x}_{k+1} &= \hat{x}_k - W_{k+1} H_{k+1} \hat{x}_k + W_{k+1} \tilde{y}_{k+1} \\ &= \hat{x}_k - W_{k+1} [\tilde{y}_{k+1} - H_{k+1} \hat{x}_k]\end{aligned}\quad (4.28)$$

We can write the recursion form as a list in the following table

Initialize	$P_1 = [\frac{1}{\alpha^2} I + H_1^T W_1 H_1]^{-1}$ $\hat{x}_1 = P_1 [\frac{1}{\alpha} \beta + H_1^T W_1 \tilde{y}_1]^{-1}$
Update	$\hat{x}_{k+1} = \hat{x}_k + K_{k+1} (\tilde{y}_{k+1} - H_{k+1} \hat{x}_k)$
Gain	$K_{k+1} = P_k H_{k+1}^T [H_{k+1} P_k H_{k+1}^T + W_{k+1}^{-1}]^{-1}$
Information Matrix	$P_{k+1} = [I - K_{k+1} H_{k+1}] P_k$

Table 4.1: Linear Sequential Estimation

## Steering Parameter Estimation

Applying the above sequential algorithm to the simplified linearized steering motion equations stated below, we retrieve the estimated parameters.

$$\begin{bmatrix} m - Y_{\dot{v}} & -Y_{\dot{r}} \\ -N_{\dot{v}} & (I_z - N_{\dot{r}}) \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} Y_v & mu - Y_r \\ N_v & N_r \end{bmatrix} \begin{bmatrix} v \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ \tau_N \end{bmatrix}\quad (4.29)$$

In order to reduce the coupling effect of the parameters occurring because of the multiplication we omit the parameters  $Y_{\dot{r}}$  and  $N_{\dot{v}}$ . Leaving the derivatives on the left side alone

$$(m - Y_{\dot{v}}) \dot{v} = Y_v v + (Y_r - mU) r$$

$$(I_z - N_{\dot{r}}) \dot{r} = N_v v + N_r r N_{\tau} \tau \quad (4.30)$$

$$(4.31)$$



we find the derivatives as

$$\dot{v} = \underbrace{\left(\frac{Y_v}{m - Y_{\dot{v}}}\right)}_{\delta} v + \underbrace{\left(\frac{Y_r - mU}{m - Y_{\dot{v}}}\right)}_{\varepsilon} r$$

$$\dot{r} = \underbrace{\left(\frac{N_v}{I_z - N_{\dot{r}}}\right)}_{\alpha} v + \underbrace{\left(\frac{N_r}{I_z - N_{\dot{r}}}\right)}_{\beta} r + \underbrace{\left(\frac{N_{\tau}}{I_z - N_{\dot{r}}}\right)}_{\gamma} \tau \quad (4.32)$$

$$(4.33)$$

arranging the equations to apply the algorithm we get

$$\frac{v_{t+1} - v_t}{\Delta t} = \delta v + \varepsilon r \quad \implies \quad v_{t+1} = (\delta v \Delta t + 1) v + \varepsilon \Delta t r \quad (4.34)$$

which can be written as

$$v_{t+1} = \underbrace{\begin{bmatrix} v & r \end{bmatrix}}_H \underbrace{\begin{bmatrix} (\delta v \Delta t + 1) \\ \varepsilon \Delta t \end{bmatrix}}_{\Theta} \quad (4.35)$$

## Diving Parameter Estimation

Before using the sequential parameter estimation algorithm we arrange our diving model and configure it in order to make it applicable to retrieve unknown parameters. Hence simplifying the model we get

$$\begin{bmatrix} m - Z_{\dot{\omega}} & -Z_{\dot{q}} \\ -M_{\dot{\omega}} & (I_y - M_{\dot{q}}) \end{bmatrix} \begin{bmatrix} \dot{\omega} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} Z_{\omega} & mU - Z_q \\ M_{\omega} & M_q \end{bmatrix} \begin{bmatrix} v \\ r \end{bmatrix} + \begin{bmatrix} \tau_Z \\ \tau_M \end{bmatrix} \quad (4.36)$$

then leaving the derivatives alone yields

$$\begin{aligned}\dot{\omega} &= \left( \frac{Z_{\omega}}{m - Z_{\dot{\omega}}} \right) w + \left( \frac{Z_q - mU}{m - Z_{\dot{\omega}}} \right) q + \left( \frac{\tau_Z}{m - Z_{\dot{\omega}}} \right) u \\ \dot{q} &= \left( \frac{M_{\omega}}{I_y - M_{\dot{q}}} \right) w + \left( \frac{M_q}{I_y - M_{\dot{q}}} \right) q + \left( \frac{\tau_M}{I_y - M_{\dot{q}}} \right) u\end{aligned}\quad (4.37)$$

Evaluating the sequential algorithm both for steering and diving motions we find the estimated linear hydrodynamic coefficients and errors as

Parameters	Initial Value	Estimated Value	Absolute Error	Relative Error
$Y_v$	-0.100	-0.0091	0.0909	% 90.90
$Y_r$	0.030	0.028	0.0020	% 6.66
$N_v$	-0.0074	-0.0077	0.0003	% 3.67
$N_r$	0.0160	-0.0035	0.0195	% 77.82
$Z_{\omega}$	-0.300	-0.2144	0.0856	% 28.52
$Z_q$	-0.140	0.1038	0.2438	% 174.15
$M_{\omega}$	-0.0029	-0.0027	0.0002	% 8.47
$M_q$	-0.0016	-0.0017	0.0001	% 6.44

Table 4.2: Parameter Estimation Comparison Table

# CHAPTER 5

## OBSTACLE AVOIDANCE AND GUIDANCE

### **Path Planning and Obstacle Avoidance**

Path planning is an essential requirement in different conditions especially in military operations for underwater vehicles. When time and energy come forth as important issues, path planning becomes an indispensable method in order to consume less energy because of the limited battery capacity. Selecting the environment as two dimensional space brings an important advantage where in 3D, necessity to define the position and attitude information increases the number of the variables hence problem becomes more complex. Those kind of nontrivial problems are generally solved by preferring probabilistic random search algorithms, RRT (Rapidly-exploring Random Trees) as an example.

Here we preferred online obstacle avoidance for safety of underwater vehicle hence for that reason nonlinear programming is used. Goal is to reach the global minimum point and avoiding from local minimums. Selecting the goal point as  $q_g = [4,4]$ , our objective function becomes

$$f(x,y) = (x - 4)^2 + (y - 4)^2 \quad (5.1)$$

and the goal point is the only global minimum point.

First we analyzed problem by selecting the obstacles in fixed locations on the second phase we placed obstacles at random positions with random sizes. At first step our obstacles are

placed as

$$x^2 + y^2 - 1 \leq 0 \quad (5.2)$$

$$x^2 + (y - 4)^2 - 1 \leq 0 \quad (5.3)$$

$$x^2 + (y + 4)^2 - 1 \leq 0 \quad (5.4)$$

$$(x - 4)^2 + y^2 - 1 \leq 0 \quad (5.5)$$

$$(x + 4)^2 + y^2 - 1 \leq 0 \quad (5.6)$$

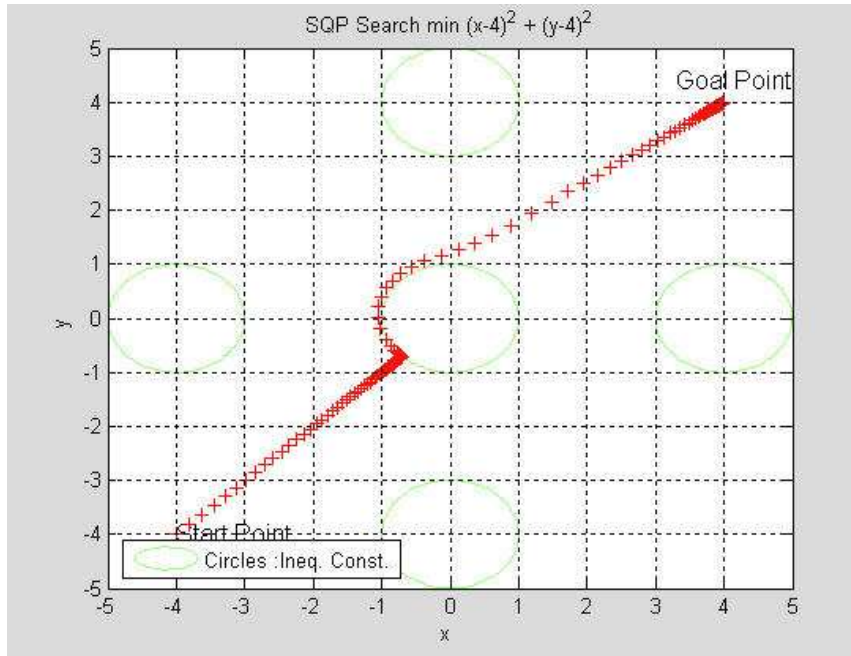


Figure 5.1: Obstacle Avoidance for first case with start point [-4,-4]

In the second case the obstacles are defined as

$$x^2 + y^2 - 8.988 \leq 0 \quad (5.7)$$

$$x^2 + (y - 4)^2 - 1 \leq 0 \quad (5.8)$$

$$x^2 + (y + 4)^2 - 1 \leq 0 \quad (5.9)$$

$$(x - 4)^2 + y^2 - 1 \leq 0 \quad (5.10)$$

$$(x + 4)^2 + y^2 - 1 \leq 0 \quad (5.11)$$

and the solution is found as

In the third case the obstacles are

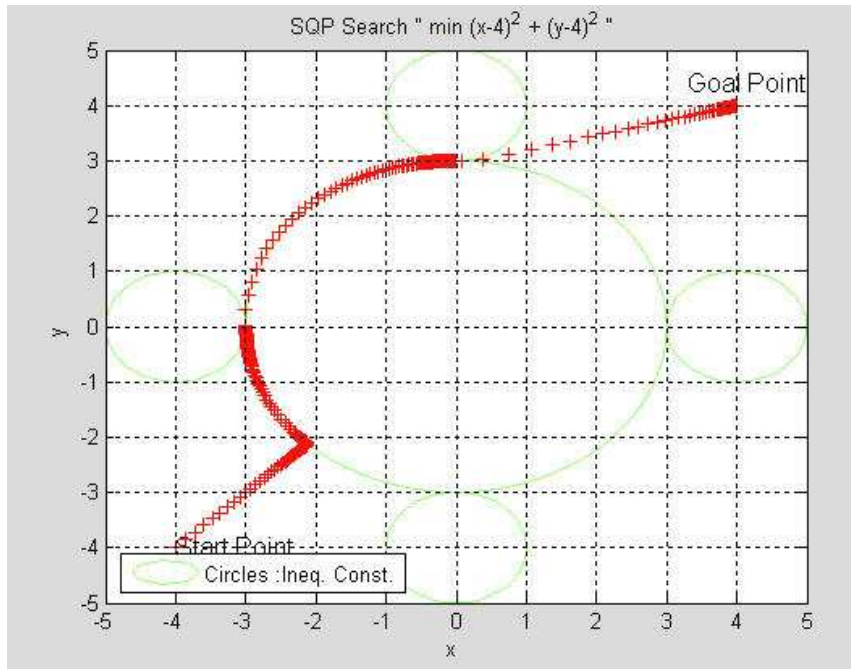


Figure 5.2: Obstacle Avoidance for second case with start point [-4,-4]

$$x^2 + y^2 - 1 \leq 0 \tag{5.12}$$

$$x^2 + (y - 4)^2 - 1 \leq 0 \tag{5.13}$$

$$x^2 + (y + 4)^2 - 9 \leq 0 \tag{5.14}$$

$$(x - 4)^2 + y^2 - 1 \leq 0 \tag{5.15}$$

$$(x + 4)^2 + y^2 - 9 \leq 0 \tag{5.16}$$

and the solution is found as

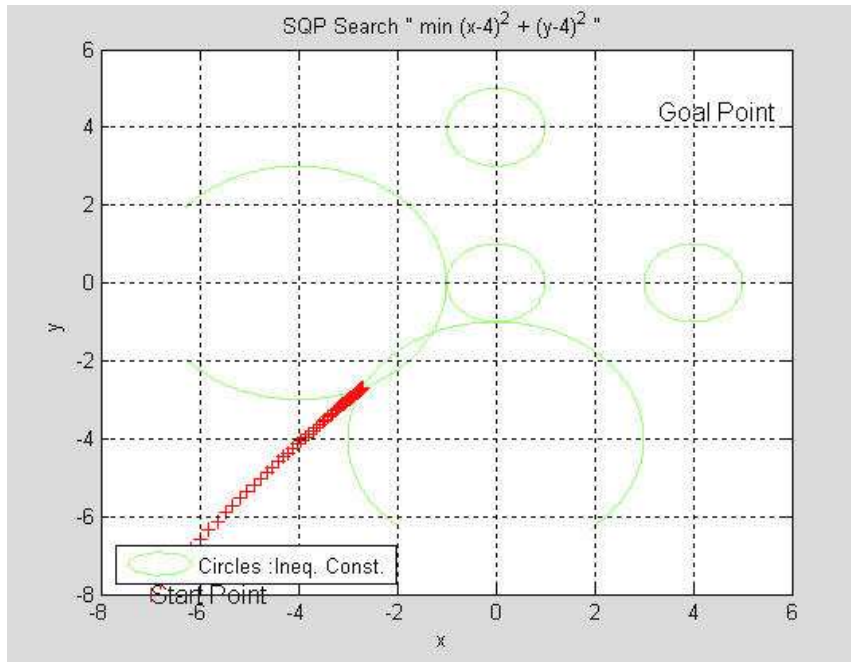


Figure 5.3: Obstacle Avoidance for third case with start point [-7,-8]

Here solution leads us to the local minima where we do not want to stuck at. After changing the start point we reach the goal point.

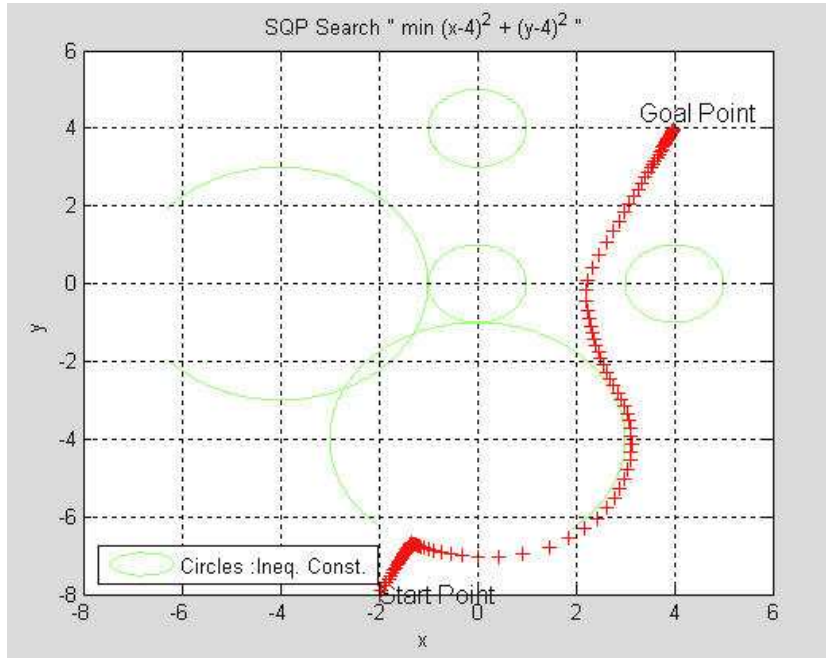


Figure 5.4: Obstacle Avoidance for third case with start point [-2,-8]

In order to enhance the algorithm with same obstacles we use boolean operations which can be defined by

$$\text{Intersection} : (f_1^m + f_2^m + \dots + f_n^m)^{1/m} = 1 \quad (5.17)$$

$$\text{Union} : (f_1^{-m} + f_2^{-m} + \dots + f_n^{-m})^{-1/m} = 1 \quad (5.18)$$

As our constraints stand as

$$g_3 = x^2 + (y+4)^2 - 9 \leq 0 \quad (5.19)$$

$$g_5 = (x+4)^2 + y^2 - 9 \leq 0 \quad (5.20)$$

applying intersection rule for our constraints for  $m = 1$

$$(g_3 + g_5)^{1/1} = 1 \quad (5.21)$$

Our new constraint becomes,

$$g_6 = 2x^2 + 2y^2 + 8x + 8y + 14 \quad (5.22)$$

gradient of our new constraint turns to

$$\nabla_{g_6} = [4x + 8; 4y + 8] \quad (5.23)$$

Adding the boolean operations to our problem we enhance solution and get rid of local minimum points, where in our example it stands at point  $(-2.707, -2.707)$ .

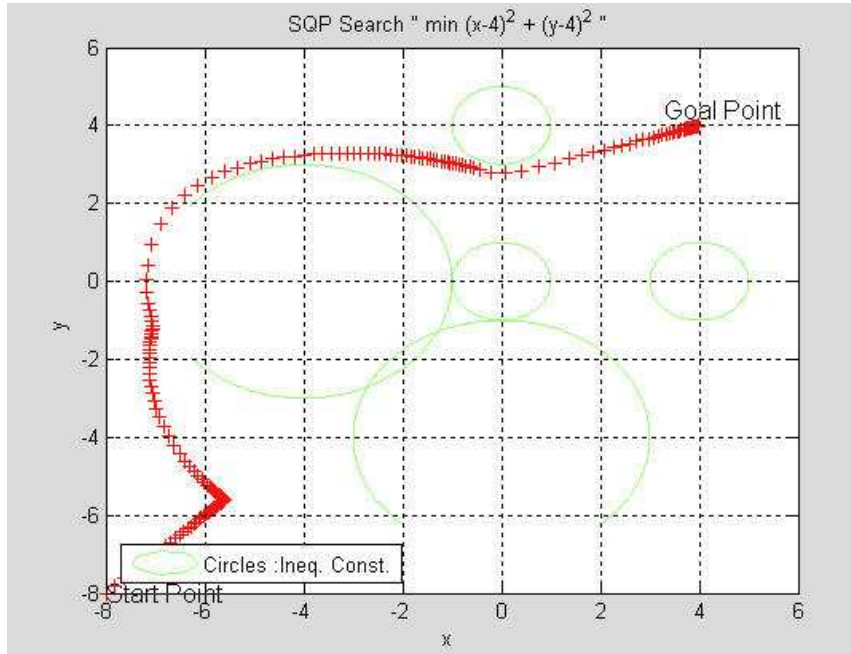


Figure 5.5: Obstacle Avoidance for fifth case with start point  $[-8, -8]$



# CHAPTER 6

## CONCLUSION

### **6.1 Summary of the Results**

# APPENDIX A

## NONLINEAR EQUATIONS OF MOTION

Nonlinear equations of motion for our vehicle:

$$\begin{aligned}
 \tau_X &= m\dot{u} - mvr + m\omega q + mz_G(pr + \dot{q}) + X_u\dot{u} + X_\omega(\dot{\omega} + uq) + X_q\dot{q} + Z_\omega\omega q \\
 &\quad + Z_qq^2 + X_v\dot{v} + X_p\dot{p} + X_r\dot{r} - Y_vvr - Y_p rp - Y_r r^2 - X_vur - Y_\omega\omega r \\
 &\quad + Y_\omega vq + Z_q pq - (Y_{\dot{q}-Z_r})qr \\
 &\quad + X_uu + X_{|u|u}|u|u + X_vv + X_\omega\omega + X_pp + X_qq + X_rr \\
 \tau_Y &= m\dot{v} - m\omega p + mur + mz_G(qr - \dot{p}) + X_v\dot{u} + Y_\omega\dot{\omega} + Y_{\dot{q}}\dot{q} + Y_v\dot{v} \\
 &\quad + Y_p\dot{p} + Y_r\dot{r} + X_vvr - Y_\omega vp + X_rr^2 + (X_p - Z_r)rp - Z_pp^2 - X_\omega(up - \omega r) \\
 &\quad + X_uur - Z_\omega\omega p - Z_q pq + X_{\dot{q}}qr \\
 &\quad + Y_vv + Y_{|v|v}|v|v + Y_uu + Y_\omega\omega + Y_pp + Y_qq + Y_rr \\
 \tau_Z &= m\dot{\omega} - muq + mvp - mz_G(p^2 + q^2) + X_\omega(\dot{u} - \omega q) + Z_\omega\dot{\omega} + Z_{\dot{q}}\dot{q} - X_uuq \\
 &\quad - X_{\dot{q}}q^2 + Y_\omega\dot{v} + Z_p\dot{p} + Z_r\dot{r} + Y_vvp + Y_r rp + Y_p p^2 + X_vup + Y_\omega\omega p \\
 &\quad - X_vvq - (X_p - Y_{\dot{q}})pq - X_rqr \\
 &\quad + Z_w w + Z_{|\omega|\omega}|\omega|\omega + Z_uu + Z_vv + Z_pp + Z_qq + Z_rr \\
 \tau_K &= I_x\dot{p} + (I_z - I_y)qr - (\dot{r} + pq)I_{xz} + (r^2 - q^2)I_{yz} + (pr - \dot{q})I_{xy} - mz_G(\dot{v} - \omega p + ur) \\
 &\quad + X_p\dot{u} + Z_p\dot{\omega} + K_{\dot{q}}\dot{q} - X_v\omega u + X_r uq - Y_\omega\omega^2 - (Y_{\dot{q}} - Z_r)\omega q + M_r q^2 \tag{A.1} \\
 &\quad + Y_p\dot{v} + K_p\dot{p} + K_r\dot{r} + Y_\omega v^2 - (Y_{\dot{q}} - Z_r)vr + Z_p vp - M_r r^2 - K_q rp + X_\omega uv \\
 &\quad - (Y_v - Z_\omega)v\omega - (Y_r + Z_{\dot{q}})\omega r - Y_p\omega p - X_{\dot{q}}ur + (Y_r + Z_{\dot{q}})vq + K_r pq - (M_{\dot{q}} - N_r)qr \\
 &\quad + K_pp + K_{|p|p}|p|p + K_uu + K_vv + K_\omega\omega + K_qq + K_rr
 \end{aligned}$$

$$\begin{aligned}
\tau_M &= I_y \dot{q} + (I_x - I_z) rp - (\dot{p} + qr) I_{xy} + (p^2 - r^2) I_{zx} + (qp - \dot{r}) I_{yz} - mz_G (\dot{u} - vr + \omega q) \\
&\quad + X_{\dot{q}} (\dot{u} + \omega q) + Z_{\dot{q}} (\dot{\omega} - uq) + M_{\dot{q}} \dot{q} - X_{\dot{\omega}} (u^2 - \omega^2) - (Z_{\dot{\omega}} - X_{\dot{u}}) \omega u + Y_{\dot{q}} \dot{u} \\
&\quad + K_{\dot{q}} \dot{p} + M_{\dot{r}} \dot{r} + Y_{\dot{p}} ur - Y_{\dot{r}} vp - K_{\dot{r}} (p^2 - r^2) + (K_{\dot{p}} - N_{\dot{r}}) rp - Y_{\dot{\omega}} uv + X_{\dot{\omega}} v\omega \\
&\quad - (X_{\dot{r}} + Z_{\dot{p}}) (up - \omega r) + (X_{\dot{p}} - Z_{\dot{r}}) (\omega p + ur) - M_{\dot{r}} pq + K_{\dot{q}} qr \\
&\quad + M_{\dot{q}} q + M_{|q|q} |q|q + M_u u + M_v v + M_{\omega} \omega + M_p p + M_r r \\
\tau_N &= I_z \dot{r} + (I_y - I_z) pq - (\dot{q} + rp) I_{yz} + (q^2 - p^2) I_{xy} + (rp - \dot{p}) I_{zx} + X_{\dot{r}} \dot{u} + Z_{\dot{r}} \dot{\omega} \\
&\quad + M_{\dot{r}} \dot{q} + X_{\dot{v}} u^2 + Y_{\dot{\omega}} \omega u - (X_{\dot{p}} - Y_{\dot{q}}) uq - Z_{\dot{q}} \omega q - K_{\dot{q}} q^2 + Y_{\dot{r}} \dot{v} + K_{\dot{r}} \dot{p} \\
&\quad + N_{\dot{r}} \dot{r} - X_{\dot{v}} v^2 - X_{\dot{r}} vr - (X_{\dot{p}} - Y_{\dot{q}}) vp + M_{\dot{r}} rp + K_{\dot{q}} p^2 - (X_{\dot{u}} - Y_{\dot{v}}) uv - X_{\dot{\omega}} v\omega \\
&\quad + (X_{\dot{q}} + Y_{\dot{p}}) up + Y_{\dot{r}} ur + Z_{\dot{q}} \omega p - (X_{\dot{q}} + Y_{\dot{p}}) vq - (K_{\dot{p}} - M_{\dot{q}}) pq - K_{\dot{r}} qr \\
&\quad + N_r r + N_{|r|r} |r|r + N_u u + N_v v + N_{\omega} \omega + N_p p + N_q q
\end{aligned}$$

## APPENDIX B

### REYNOLDS NUMBER

In the middle of the head of our vehicle stands a point called stagnation point where the pressure is in its highest value compared to the other points of vehicle and the velocity of water is zero at this point. Here we see a laminar flow till to the occurrence of separation where the turbulent flow starts. From the figure it is seen that the Reynolds number between  $10^3$  and  $3 \times 10^5$  denotes the laminar flow, after  $3 \times 10^5$  flow becomes turbulent and increase of momentum avoids separation therefore the drag decreases immediately [12].

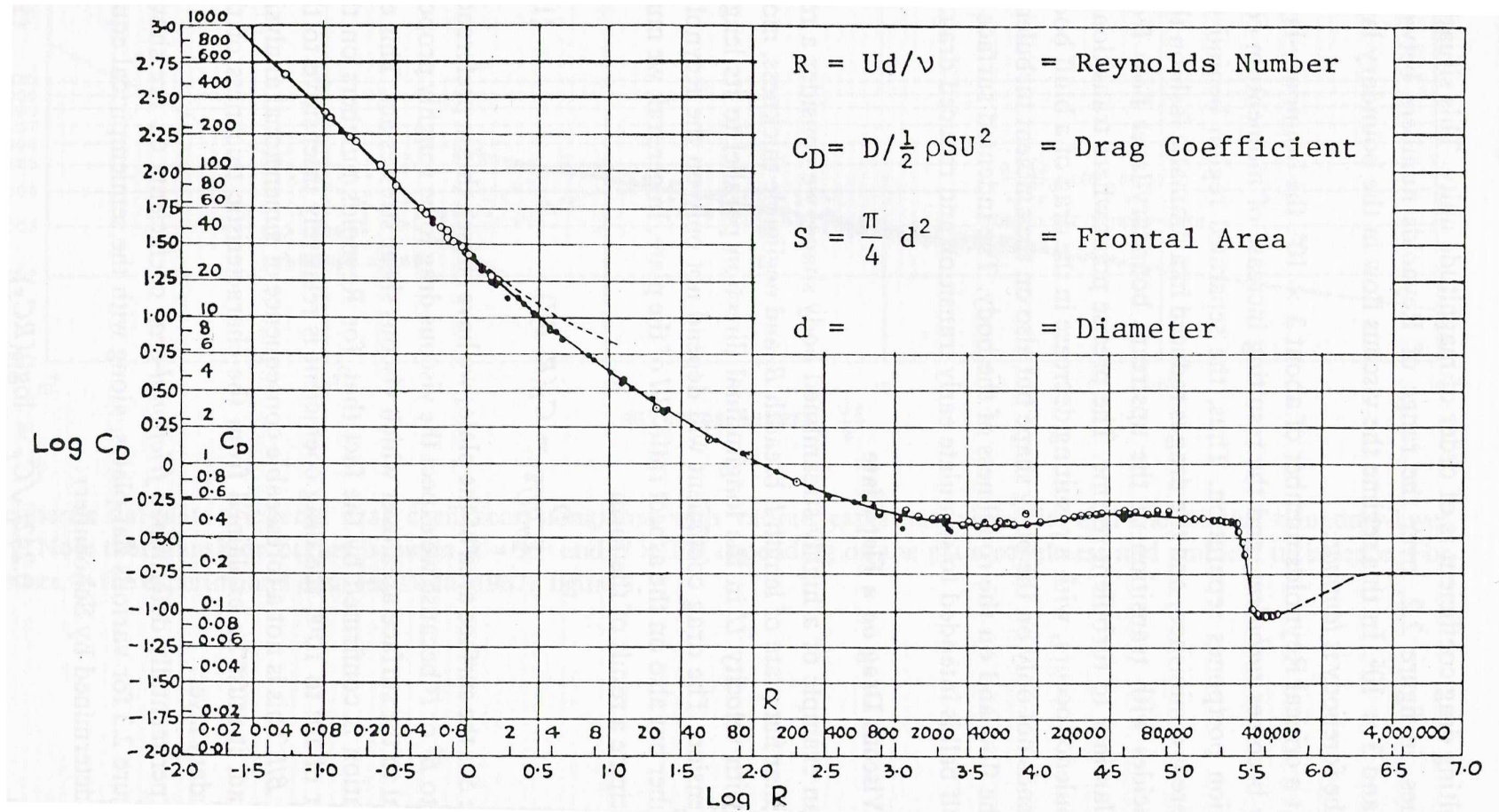


Figure B.1: The drag coefficient for a sphere.

# BIBLIOGRAPHY

- [1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications To Tracking and Navigation*. John Wiley & Sons, 2001.
- [2] M. Caccia and G. Veruggio. Guidance and control of a reconfigurable unmanned underwater vehicle. In *Control Engineering Practice* 8, pages 21–37, 2000.
- [3] J. L. Crassidis and J. L. Junkins. *Optimal Estimation of Dynamic Systems*. CRC Press, 2004.
- [4] R. Cristi, F. A. Papoulias, and A. J. Healey. Adaptive sliding mode control of autonomous underwater vehicles in the dive plane. *IEEE Journal of Oceanic Engineering*, 15(3):152–160, 1990.
- [5] T. I. Fossen. *Guidance and Control of Ocean Vehicles*. John Wiley & Sons, 1994.
- [6] A. J. Healey and D. Lienard. Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. *IEEE Journal of Oceanic Engineering*, 18(3):327–339, 1993.
- [7] B. Jalving. The ndre-auv flight control system. *IEEE Journal of Oceanic Engineering*, 19(4):497–501, 1994.
- [8] K. Kim and H. S. Choi. Navigation and control for a test bed auv-snuuv i. *IEEE*, (1):89–94, 2004.
- [9] Sir H. Lamb. *Hydrodynamics*. Dover Publications, 1945.
- [10] R. K. Lea, R. Allen, and S. L. Merry. A comparative study of control techniques for an underwater vehicle. *International Journal od Systems Science*, 30(9):347–964, 1999.
- [11] Y. Nakamura and S. Savant. Nonlinear tracking control of autonomous underwater vehicles. In *In Proceedings of IEEE Int. Conf. of Robotics and Automation*, volume 3, pages A4–A9, 1992.

- [12] J. N. Newman. *Marine Hydrodynamics*. The MIT Press, third edition, 1980.
- [13] J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, 1991.
- [14] S. S. Tabai, F. El-Hawary, and M. El-Hawary. Hybrid adaptive control of autonomous underwater vehicles. In *In Proceedings of Symposium of Autonomous Underwater Vehicle Technology*, pages 275–282, 1994.
- [15] A. Tewari. *Modern Control Design with Matlab and Simulink*. John Wiley & Sons, 2002.
- [16] Y. Wang and D. M. Lane. Subsea vehicle path planning using nonlinear programming and constructive solid geometry. In *IEE Proc. Control Theory Appl. Vol. 144*, pages 143–152, 1997.
- [17] K. Wendel. *Hydrodynamic Masses and Hydrodynamic Moments of Inertia*. Technical report, 1956.
- [18] D. N. Yoerger and J. E. Slotine. Robust trajectory control of underwater vehicles. *IEEE Journal of Oceanic Engineering*, 10(4):462–470, 1985.
- [19] J. Yuh. Design and control of autonomous underwater robots: A survey. In *Automatic Robots 8*, 7-24, *Kluwer Academic Publishers*, 2000.

COMPUTER AIDED ENGINEERING OF AN UNMANNED UNDERWATER  
VEHICLE

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

NECMETTİN CEVHERİ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF MASTER OF SCIENCE  
IN  
MECHANICAL ENGINEERING

JULY 2009



Approval of the thesis:

**COMPUTER AIDED ENGINEERING OF AN UNMANNED  
UNDERWATER VEHICLE**

submitted by **NECMETTİN CEVHERİ** in partial fulfillment of the requirements  
for the degree of **Master of Science in Mechanical Engineering Department,**  
**Middle East Technical University** by,

Prof. Dr. Canan Özgen  
Dean, Graduate School of **Natural and Applied Science**

\_\_\_\_\_

Prof. Dr. Suha Oral  
Head of Department, **Mechanical Engineering**

\_\_\_\_\_

Prof. Dr. M. Haluk Aksel  
Supervisor, **Mechanical Engineering Dept., METU**

\_\_\_\_\_

Prof. Dr. S. Kemal İder  
Co-Supervisor, **Mechanical Engineering Dept., METU**

\_\_\_\_\_

**Examining Committee Members:**

Instr. Dr. Tahsin Çetinkaya  
Mechanical Engineering Dept., METU

\_\_\_\_\_

Prof. Dr. M. Haluk Aksel  
Mechanical Engineering Dept., METU

\_\_\_\_\_

Prof. Dr. S. Kemal İder  
Mechanical Engineering Dept., METU

\_\_\_\_\_

Asst. Prof. Dr. Cüneyt Sert  
Mechanical Engineering Dept., METU

\_\_\_\_\_

Prof. Dr. M. Kemal Leblebicioğlu  
Electrical and Electronics Engineering Dept., METU

\_\_\_\_\_

**Date:**

\_\_\_\_\_

# PLAGIARISM

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced and results that are not original to this work.**

Name, Last Name : NECMETTİN CEVHERİ

Signature :



# **ABSTRACT**

## **COMPUTER AIDED ENGINEERING OF AN UNMANNED UNDERWATER VEHICLE**

Cevheri, Necmettin

M.Sc., Department of Mechanical Engineering

Supervisor: Prof. Dr. M. Haluk Aksel

Co-Supervisor: Prof. Dr. S. Kemal İder

July 2009, 150 pages

Hydrodynamic and thermal analyses performed during the conceptual design of an unmanned underwater vehicle are presented in this study. The hull shape is determined by considering alternative shapes and the dimensions are determined from the internal arrangement of components. Preliminary thermal analyses of the watertight section are performed with a commercial software called FLUENT to check the risk of over-heating due to the heat dissipation of devices.

Performance of the proposed hull design is analyzed by FLUENT. Before simulations of the vehicle, validation studies are performed. Models 4159, 4158 and 4154 of Series 58 are chosen as the experimental reference. Their total resistance coefficients are compared with the results of the validations analyses. Mesh densities, turbulence models, near wall modeling approaches and inlet turbulence intensities are varied to understand their effects on the accuracy of predictions. A suitable turbulence modeling approach is chosen to analyze forward and vertical motions of the vehicle to check whether speed requirements are fulfilled. Hull configurations

with and without appendages are used to observe their effects on total drag. It is observed that the proposed design satisfies speed requirements of the vehicle and no overheating is expected in the watertight section.

Keywords: Unmanned underwater vehicles (UUV), underwater robots, CFD, drag, FLUENT, turbulence modeling

# ÖZ

## BİR İNSANSIZ SUALTI ARACININ BİLGİSAYAR DESTEKLİ TASARIMI VE ÜRETİMİ

Cevheri, Necmettin

Yüksek Lisans, Makina Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. M. Haluk Aksel

Ortak Tez Yöneticisi: Prof. Dr. S. Kemal İder

Temmuz 2009, 150 sayfa

Bir insansız sualtı aracının kavramsal tasarımı esnasında gerçekleştirilen hidrodinamik ve ısı analizler bu tezin ana konusunu oluşturmaktadır. Gövdenin şekline çeşitli sualtı aracı şekillerinin değerlendirilmesi sonucu karar verilmektedir. Boyutlar ise cihazların gövde içine yerleşiminden sonra ortaya çıkmaktadır. Elektronik ekipmanın bulunduğu bölmenin ısınmasına dair analizler ticari bir hesaplamalı akışkanlar dinamiği yazılımı olan FLUENT yardımı ile yapılmaktadır.

Önerilen tasarımın performans analizleri de FLUENT ile yapılmaktadır. Analizlerden önce bazı doğrulama çalışmaları gerçekleştirilmiştir. Bu çalışmalarda Series 58 adı verilen bir grup gövdeden 4159, 4158 ve 4154 modelleri deneysel referans olarak alınmaktadır. Bu modellerin toplam direnç katsayıları yapılan analiz sonuçları ile kıyaslanmaktadır. Bu analizler esnasında ağ yoğunluğu, türbülans modeli, duvar yakınındaki akışı modelleme yöntemleri ve giriş türbülans yoğunluğu değiştirilerek çözümlere etkileri gözlemlenmektedir. Uygun bir türbülans modeli seçilerek aracın

ileri ve yukarı yönde hareketleri incelenmekte ve verilen tasarım kıstaslarını sağlaması şartı kontrol edilmektedir. Gövde çıkıntılarının dahil edildiđi ve edilmediđi modeller yaratılarak bu çıkıntıların toplam sürüklenme kuvvetine etkileri gözlemlenmektedir. Analizler sonucu gövdenin istenen hız kriterlerini sağladığı ve sızdırmaz orta bölümde aşırı ısınma oluşmadığı sonucuna varılmaktadır.

Anahtar Kelimeler: İnsansız sualtı aracı, sualtı robotu, Sualtı Robotu, hesaplamalı akışkanlar dinamiđi, sürüklenme, FLUENT, türbülans modelleme

*Dedicated to my family and Mehtap...*



## ACKNOWLEDGEMENTS

I would like to express my gratitude to my thesis supervisor Prof. Dr. M. Haluk Aksel and my thesis co-supervisor Prof. Dr. S. Kemal İder for their support, guidance and criticism throughout my studies.

This thesis work is based on the Unmanned Underwater Vehicle (ULISAR) Project. I would like to thank to Prof. Dr. M. Kemal Leblebiciođlu, the supervisor of the project, for this valuable opportunity and his leadership throughout the project. The contributions from project group members Prof. Dr. Levend Parnas, Assist. Prof. Dr. Afşar Saranlı, Emre Ege, Hüseyin Yiđitler, Kadir Isiyel, Alper Kılıç and Mine Cüneyitođlu were invaluable for the project and for the thesis study.

This project was funded by the Scientific and Technological Research Council of Turkey. I would like to express my appreciation for their support and motivation throughout the project.

Manufacturing was performed by Barış Elektrik Endüstrisi Inc. I would like to thank to Fikret Şenel, Bora Balya, Onur Miskbay and Ergül Soyçeken for their efforts and suggestions.

I would like to express my appreciation to Emre Öztürk and Özgün Güler for their help with the CFD simulations and Necati Karaismailođlu for his help with the format of the thesis.

I would like to thank to Mehtap Çakmak for her continuous support, faith and love. Finally I would like to express my deepest gratitude to my family for their encouragement, understanding and support during my studies.



# TABLE OF CONTENTS

PLAGIARISM .....	iii
ABSTRACT .....	v
ÖZ .....	vii
ACKNOWLEDGEMENTS .....	x
LIST OF TABLES .....	xv
LIST OF FIGURES .....	xvi
NOMENCLATURE.....	xx
CHAPTERS	
1. INTRODUCTION.....	1
1.1 Classification Of Underwater Systems .....	4
1.2 Purposes Of AUV's .....	5
1.3 History Of Uuvs .....	7
1.4 Literature Survey – UUV Examples .....	10
1.5 Basic Features Of UUV's .....	18
1.6 Information About The Research Project ULISAR.....	19
1.7 Present Study.....	20
2. CONCEPTUAL DESIGN OF THE HULL .....	21
2.1 Theoretical Background .....	22
2.1.1 Skin Friction Drag.....	22
2.1.2 Form Drag .....	24
2.1.3 Total Drag Force .....	25
2.1.4 Prismatic Coefficient.....	27
2.2 Hull Requirements .....	27

2.3	Hull Shape Selection .....	28
2.3.1	Teardrop Shape .....	29
2.3.2	Torpedo Shape .....	31
2.3.3	Rectangular Shape.....	32
2.3.4	Oblate Shape .....	33
2.3.5	Open Space Frame .....	33
2.4	Propulsion and Maneuvering .....	35
2.4.1	Propulsion Type .....	35
2.4.2	Maneuvering .....	36
2.5	Internal Arrangement .....	42
2.5.1	Internal Arrangement of Devices.....	45
2.6	Material Selection .....	49
2.6.1	Composite Materials .....	51
3.	HEAT TRANSFER ANALYSES.....	58
3.1	The Software .....	59
3.2	Analyses .....	60
3.2.1	Analysis for the First Case .....	60
3.2.2	Analysis for the Second Case.....	63
3.2.3	Heat Transfer Test.....	65
4.	VALIDATION STUDIES .....	68
4.1	Introduction to CFD .....	69
4.1.1	Governing Equations of Fluid Flow.....	70
4.1.2	Introduction to Turbulence Modeling .....	72
4.1.3	Reynolds Averaging.....	74
4.1.4	Boussinesq Approach.....	75
4.1.5	Turbulence Models .....	77
4.1.6	Near Wall Treatment.....	79
4.2	Validation Studies .....	83
4.2.1	The Software .....	83
4.2.2	Series 58 Studies .....	84

5. DRAG CALCULATIONS.....	116
5.1 Hull Geometry.....	116
5.1.1 Bare Hull.....	117
5.1.2 Hull with Wings.....	118
5.1.3 Fully Appended Hull.....	119
5.2 Solution.....	119
5.2.1 Forward Motion.....	121
5.2.2 Vertical Motion.....	132
5.3 Thruster Selection.....	136
6. MANUFACTURING AND TESTING.....	137
6.1 Manufacturing.....	137
6.2 Tests.....	141
7. CONCLUSIONS AND DISCUSSIONS.....	143
7.1 Conclusions.....	143
7.2 Future Work Recommendations.....	145
REFERENCES.....	147

# LIST OF TABLES

## TABLES

Table 1.1 Classification of unmanned underwater vehicles.....	5
Table 4.1 Simulation cases.....	90
Table 5.1 Drag components on bare hull for forward motion.....	123
Table 5.2 Drag components on forward motion for hull-wings configuration.....	126
Table 5.3 Drag components for forward motion.....	130

# LIST OF FIGURES

## FIGURES

Figure 1.1 The AUV – MAYA [11].....	10
Figure 1.2 Nereus in AUV mode [12].....	13
Figure 1.3 Nereus in ROV model [12].....	14
Figure 1.4 Odyssey 4 class HAUV [13] .....	15
Figure 1.5 Pressure hulls of HAUV [13] .....	15
Figure 1.6 Frame of HAUV [13].....	16
Figure 1.7 Aquabox II.....	17
Figure 1.8 AQUA underwater robot .....	18
Figure 2.1 Flow around a submarine [16].....	23
Figure 2.2: Vehicle drag curves [3].....	26
Figure 2.3 Alistar of ECA [19] .....	30
Figure 2.4 Odyssey IIc of MIT AUV Laboratory [19] .....	30
Figure 2.5: Autosub 6000 [19].....	31
Figure 2.6: Hugin 1000 [19] .....	31
Figure 2.7 Urashima AUV .....	32
Figure 2.8: Double Eagle MKII of SAAB Underwater Systems [19] .....	33
Figure 2.9: SeaQuest [19] .....	34
Figure 2.10: Podded propulsor – BTD150 model of Seabotix .....	36
Figure 2.11: Maneuvering capabilities of the vehicle.....	37
Figure 2.12: Seawolf II of North Carolina State University .....	38
Figure 2.13: Bluefin unmanned underwater vehicle.....	39
Figure 2.14: Thruster configuration – Bottom view .....	40
Figure 2.15: Thruster configuration – Right view .....	41
Figure 2.16: Thruster configuration – Front view.....	41
Figure 2.17 Lower and upper bodies .....	43
Figure 2.18 Hull thickness differences.....	44
Figure 2.19 3D model of the acoustic modem.....	46

Figure 2.20 Technical drawing of the acoustic modem .....	46
Figure 2.21 Internal arrangement of devices.....	47
Figure 2.22 Technical drawing of the lower body part.....	48
Figure 2.23 Technical drawing of the upper body part.....	48
Figure 2.24 Hull materials chart [19].....	50
Figure 2.25 Filament winding illustration [22].....	53
Figure 2.26 Schematic view of Resin Transfer Molding process [23] .....	54
Figure 2.27 Schematics of Vacuum Assisted RTM [23] .....	55
Figure 2.28 Schematic representation of Vacuum Bagging [23].....	56
Figure 2.29 Hull materials comparison [24] .....	57
Figure 3.1 Pressure hull .....	59
Figure 3.2 Front view cross-section .....	61
Figure 3.3 Computational model - 1 .....	61
Figure 3.4 Details of the computational model - 1 .....	62
Figure 3.5 Temperature distribution inside the pressure hull .....	63
Figure 3.6 Details of the computational model – 2.....	64
Figure 3.7 Critical temperature locations.....	64
Figure 3.8 Maximum temperature vs time.....	65
Figure 3.9 Experimental setup .....	66
Figure 3.10 Temperature sensor positions .....	67
Figure 3.11 Temperature vs time inside the hull.....	67
Figure 4.1 Near wall region .....	80
Figure 4.2 Test Apparatus for Series 58 Experiments [17].....	85
Figure 4.3 Mesh 1 for Model 4159 – boundary layer .....	87
Figure 4.4 Mesh 1 for Model 4159 - trailing edge detail.....	87
Figure 4.5 Mesh 1 for Model 4159 – flow domain and boundary conditions .....	88
Figure 4.6 Wall $y^+$ for Mesh 1 Case 2 at $Re=2 \times 10^6$ .....	91
Figure 4.7 Wall $y^+$ for Mesh 1 Case 2 at $Re=2.5 \times 10^7$ .....	91
Figure 4.8 Comparison of Mesh 1 results – total resistance coefficient.....	92
Figure 4.9 Adaptation results for Mesh 1 Case 3.....	93
Figure 4.10 Modified Mesh 1 after $y^+$ adaption.....	94



Figure 4.11 Adaptation results for Mesh 1 Case 5.....	95
Figure 4.12 Effect of turbulence intensity on total resistance coefficient.....	96
Figure 4.13 Mesh 2 for Model 4159 – flow domain.....	97
Figure 4.14 Mesh 2 for Model 4159 – boundary layer .....	97
Figure 4.15 Mesh 2 for Model 4159 – trailing edge .....	98
Figure 4.16 Comparison of Mesh 2 results – total resistance coefficient.....	99
Figure 4.17 Adaptation for Case 4 with Mesh 2 – total resistance coefficient .....	100
Figure 4.18 Adaptation for Case 7 with Mesh 2 – total resistance coefficient .....	101
Figure 4.19 Adaptation for Case 9 Mesh 2 – total resistance coefficient .....	102
Figure 4.20 Mesh 3 for Model 4159 – flow domain.....	103
Figure 4.21 Mesh 3 for Model 4159 – boundary layer .....	103
Figure 4.22 Mesh 3 for Model 4159 – trailing edge .....	104
Figure 4.23 Comparison of Mesh 3 results – total resistance coefficient.....	105
Figure 4.24 Adaptation results for Mesh 3 – total resistance coefficient.....	106
Figure 4.25 Wall $y^+$ plot for Case 4 at $Re=2 \times 10^7$ .....	107
Figure 4.26 Comparison of total resistance coefficient for $y^+$ meshes .....	108
Figure 4.27 Laminar solution of Model 4159 Meshes 1 and 2.....	110
Figure 4.28 Total resistance comparison for Model 4158 .....	111
Figure 4.29 Pressure coefficient comparison at $Re=2 \times 10^6$ .....	112
Figure 4.30 Pressure coefficient comparison at $Re=2.5 \times 10^7$ .....	113
Figure 4.31 Total resistance comparison for Model 4154 .....	114
Figure 5.1 Bare hull of the vehicle.....	117
Figure 5.2 Dimensions of the bare hull.....	118
Figure 5.3 Hull with wings .....	118
Figure 5.4 Fully Appended Hull .....	119
Figure 5.5 Flow domain for bare hull configuration in forward motion.....	121
Figure 5.6 Surface mesh for bare hull configuration in forward motion .....	122
Figure 5.7 Wall $y^+$ contours for bare hull configuration in forward motion.....	122
Figure 5.8 Pressure contours for bare hull configuration in forward motion.....	123
Figure 5.9 Surface mesh for hull-wings configuration in forward motion .....	124
Figure 5.10 Mesh details for hull-wings configuration in forward motion.....	124

Figure 5.11 Wall $y^+$ contours for hull-wings configuration in forward motion.....	125
Figure 5.12 Pressure contours for bare hull configuration in forward motion.....	125
Figure 5.13 Mesh for fully appended configuration in forward motion – 1 .....	127
Figure 5.14 Mesh for fully appended configuration in forward motion – 2 .....	127
Figure 5.15 Mesh for fully appended configuration in forward motion – 3 .....	128
Figure 5.16 Wall $y^+$ contours for fully appended configuration .....	129
Figure 5.17 Pressure contours for fully appended configuration - 1.....	129
Figure 5.18 Pressure contours for fully appended configuration - 1.....	130
Figure 5.19 Pathlines for forward motion - 1.....	131
Figure 5.20 Pathlines for forward motion – 2.....	132
Figure 5.21 The domain used for vertical motion simulations .....	133
Figure 5.22 Wall $y^+$ contours for fully appended configuration .....	134
Figure 5.23 Pressure contours for fully appended configuration.....	134
Figure 5.24 Pathlines for vertical motion - 1 .....	135
Figure 5.25 Pathlines for vertical motion – 2.....	136
Figure 6.1 Model of the vehicle .....	138
Figure 6.2 Mold of the vehicle (during production) .....	138
Figure 6.3 Mold of the vehicle (finished product) .....	139
Figure 6.4 Hand lay-up process for the vehicle body .....	140
Figure 6.5 Vehicle body .....	140
Figure 6.6 Leakage test at the pool .....	142

## NOMENCLATURE

$A$	Area, $m^2$
$C_d$	Drag force coefficient
$C_\mu$	Modeling constant
$C_P$	Surface pressure coefficient
$C_t$	Total resistance coefficient
$D$	Diameter, m
$f$	Body force per unit mass, $m/s^2$
$k$	Turbulence kinetic energy, $m^2/s^2$
$k_p$	Kinetic energy at point P, $m^2/s^2$
$L$	Length of the vehicle, m
$l$	Characteristic length, m
$p$	Pressure, $N/m^2$
$p_{ref}$	Reference pressure, $N/m^2$
$R$	Total resistance, N
$Re$	Reynolds Number
$Re_y$	Turbulent Reynolds number
$S$	Wetted surface area, $m^2$
$t$	Time, s
$U$	Free stream velocity, m/s
$u$	Velocity in $x$ direction, m/s
$u_\tau$	Friction velocity, m/s
$u^+$	Non-dimensional velocity
$u'$	Fluctuation of velocity in $x$ direction, m/s
$v$	Velocity in $y$ direction, m/s
$v'$	Fluctuation of velocity in $y$ direction, m/s
$w$	Velocity in $z$ direction, m/s
$w'$	Fluctuation of velocity in $z$ direction, m/s
$y_p$	Distance from the point P, m

$y^+$	Wall yplus
$y^*$	Wall ystar

### Greek Letters

$\varepsilon$	Turbulent kinetic energy dissipation rate, $m^2/s^3$
$\mu$	Dynamic viscosity, $kg/m.s$
$\mu_{lam}$	Laminar viscosity, $kg/m.s$
$\mu_t$	Turbulent (eddy) viscosity, $kg/m.s$
$\rho$	Density of the fluid, $kg/m^3$
$\tau$	Shear stress, $N/m^2$
$\omega$	Turbulent kinetic energy specific dissipation rate, $1/s$

# CHAPTER 1

## INTRODUCTION

Unmanned underwater vehicles (UUV's) can be defined simply as robots which operate underwater. These robots do not require an operator on the vehicles and depending on their level of intelligence they can operate with little or no intervention by the operator on the surface to follow a pre-defined mission. To achieve this mode of operation, these vehicles carry various equipments. They need navigation systems, sensors, energy systems, a main computer, communication devices and an appropriate mechanical design.

First of all, navigation systems are obligatory to determine the location with respect to a certain reference point. Early vehicles counted on dead reckoning which determines the position of the vehicle using the initial position, speed and time data [1]. In this method, a suitable time interval is selected and the displacement of the vehicle within this interval is found by the speed and time while the final position is derived by adding this displacement to the initial position. Inertial Navigation Systems (INS) are more commonly used in modern vehicles. These systems work similar to dead reckoning. They include motion sensors like accelerometers and gyroscopes to determine the final position and orientation by integration process. Some modern vehicles take advantage of a Global Positioning System (GPS) to determine the current position and update their INS.

Secondly, various sensors are required depending on operation requirements. In fact, sensors are very important for these vehicles because they can be critical for the functioning of the vehicle and even the purpose of the vehicle may be carrying those sensors as payloads for experimentation and research. Sensors used in these vehicles

can be classified into two categories: motion sensors and underwater research/experimentation sensors [2]. Motion sensors are primarily used by the navigation systems. Accelerometers and gyroscopes are well-known examples for this category. On the other hand, experimentation and research sensors are primarily used for collecting underwater data. Radiation sensors, Conductivity/Temperature/Depth (CTD) sensors, pressure-sensitive depth sensors, digital cameras, hydrology sensors and environmental sensors are common examples for this category [3]. A sensor which is used by almost all vehicles is the side scan sonar. These sonars scan the environment in order to detect obstacles on the route of the robot and map the seabed. Later on, depending on the level of the autonomy of the robot, obstacle avoidance algorithms can be generated. Acoustic transducers are also used for mapping regions and determining obstacles which are close to the vehicle while side scan sonars are used for larger areas.

Thirdly, rechargeable energy systems are vital for these equipments to function. Unmanned underwater vehicles which are not connected to the surface ship with a tether carry their energy sources on their body so that they do not have energy transmitting underwater cables. Throughout unmanned underwater vehicles history, various types of energy sources were used. Lead acid and silver-zinc batteries were frequently used in the early history. Later on, lithium-ion and lithium-polymer batteries began to be used for current vehicles being developed. NiMH batteries are another source which is being commonly used in modern vehicles. Besides these technologies, modern techniques like fuel-cells and solar energy are investigated for future vehicles. An example of a solar-powered robot, SAUV, is developed by the Falmouth Scientific, Inc. (FSI) in cooperation with the Autonomous Undersea Systems Institute (AUSI) and Technology Systems Inc. (TSI) [4]. The vehicle possesses solar cells on the body. When the vehicle needs to recharge batteries, it surfaces and resides on the surface until the batteries are charged. This distinctive feature allows on-vehicle recharging of energy systems; therefore, the vehicle can operate for weeks to months while untethered underwater vehicles can operate for a few hours. These vehicles are convenient for long term operations like station-

keeping. For long-term operations, fuel cell technology is also being used in modern vehicles. Urashima, which is considered to be the first vehicle to be powered by fuel cells, is developed in Japan in March 2003 [5]. The vehicle is powered via Polymer Electrolyte Fuel Cell (PEFC) system. In their sea trials in 2004, they achieved a continuous autonomous operation of 43 hours at a depth of 800 m and had a journey of 220 km. They also claim that the vehicle can operate up to 300 km autonomously. It is obvious that fuel cell technology is extremely useful for long-term operations.

Almost all unmanned underwater vehicles need a communication device for human intervention no matter what the level of autonomy is. In cases where this intervention is desired to be wireless, acoustic modems are used. These modems work in pairs; one modem is placed on the vehicle while the other is placed on the surface buoy. The limitation is that these modems have narrow bandwidths and it makes it almost impossible to transmit data continuously. Receiving data when the vehicle surfaces or when it is recovered are considered as remedies for the insufficient bandwidth [6]. Nonetheless, underwater communication still remains as a major issue to be solved for the improvement of vehicles. Certainly, these robots require control units to coordinate the operation of sensors, navigation systems, communication devices and energy systems. This is accomplished with a main CPU. The main CPU is usually stacks of standard PC-104 electronic cards which are housed in a water-tight section.

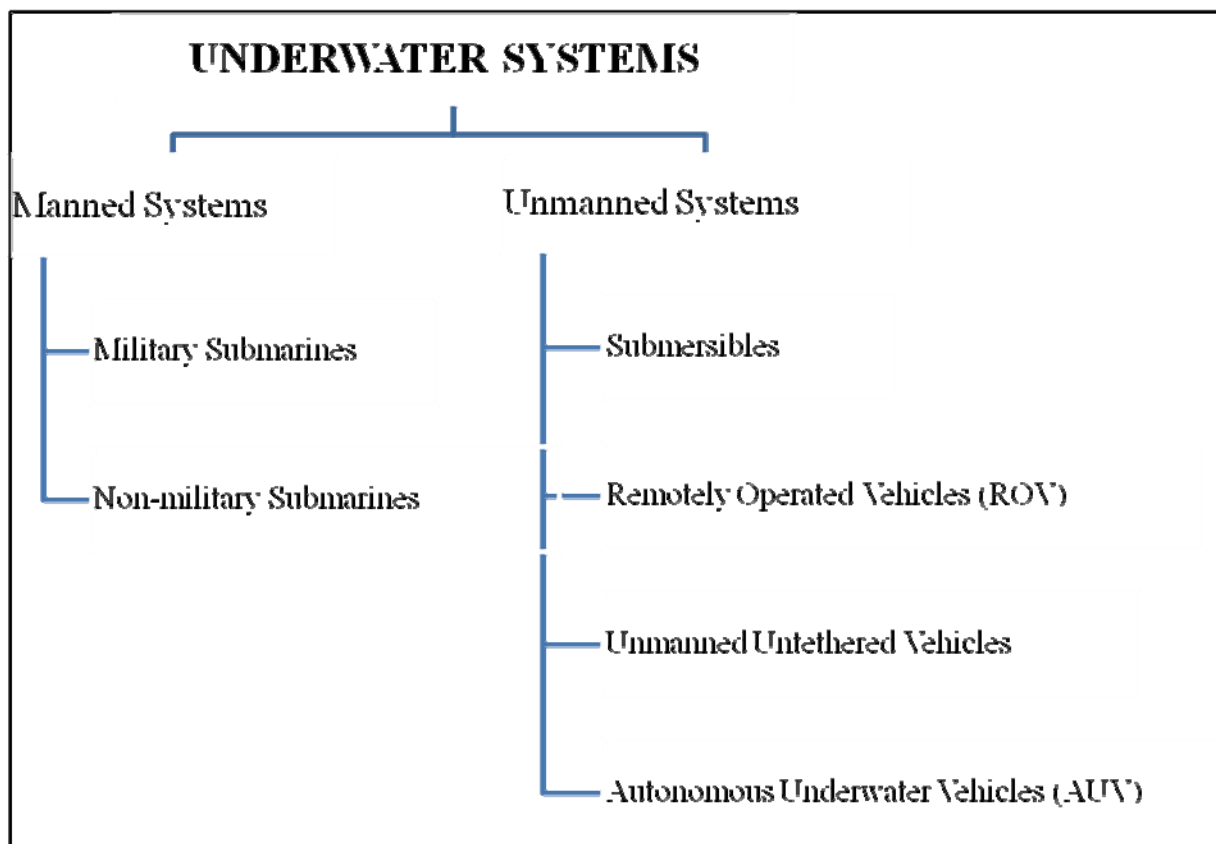
Finally, a body with an appropriate mechanical design is necessary for holding all of these systems together. Mechanical design has many steps like hull design, internal design, manufacturing and testing. In this thesis, hydrodynamic and thermal simulations performed during the conceptual design of an unmanned underwater vehicle are presented.

## 1.1 Classification of Underwater Systems

Underwater systems can be divided into two categories as manned systems and unmanned systems [1]. This classification is seen in Table 1.1. Submarines and non-military purpose underwater vehicles are well-known examples of manned vehicles. On the other hand, unmanned systems can be divided into four categories depending on their autonomy level. First category is simple submersibles. These are simple submersibles which float separately or towed behind a ship or submarine via cable. They are simple vehicles that may carry sensors or other devices. An acoustic jammer which is used for defense against torpedoes can be given as an example for this category. The second type is Remotely Operated Vehicles which is abbreviated as ROV's. ROV's are underwater robots that are connected to a ship or submarine with a tether. This tether is used for both transmitting power to the robot and communication with the robot. In some systems the power source may be on the vehicle so no tether for power transmission is necessary. However, they may not be completely autonomous and need some type of a communication system. Such systems are called Unmanned Untethered Vehicles. Finally, systems that carry their power on-board, that do not require any tether and that can operate autonomously without any operator intervention are called Autonomous Underwater Vehicles (AUV).



**Tablo 1.1** Classification of underwater systems [1]



## 1.2 Purposes of AUV's

Unmanned underwater vehicles are underwater robots which are capable of carrying out various underwater tasks without the need of an operator on the robot. They can be modified and equipped with tools like robot arms, cameras and research/experimentation sensors depending on the mission. ROV's require a tether to communicate with the operator on the surface; on the other hand, AUV's can complete a pre-defined mission autonomously without any human intervention. Nevertheless, all unmanned underwater vehicles are advantageous for underwater missions, no matter what the level of autonomy is. Today, unmanned underwater vehicles are mainly used for commercial, military and scientific purposes.

Commercial use of unmanned underwater vehicles has begun in the beginning of 2000s and has been intensifying since then. The most common commercial use is in oil and gas industry [7]. Inspection of the seabed for pipeline and underwater infrastructures along with the maintenance and repair of these structures are basic tasks that are executed. Utilization of AUV's or ROV's for these missions results in a significant decrease in expenditures. At present, ROV's and deep tow systems are default tools for underwater survey operations rather than AUV's. However, a comparison study performed in C&C Technologies by Thomas S. Chance, Art A. Kleiner, and Jay G. Northcutt exemplifies the advantage of AUV's [8]. It is mentioned that ROV's and especially deep tow Systems are default tools for underwater survey; however, they have serious deficiencies. Their operating depths are limited by tether length. A 10000 meter tether length not only brings up storage problems, but also limits the speed and maneuvering capabilities. On the other hand, AUV's overcome these deficiencies. A comparison in this study exemplifies that for a 45 mile pipeline survey in Gulf of Mexico, the use of AUV instead of a deep tow system reduces survey cost from \$707,200 to \$291,325 which means a 59% recovery on expenses. The emphasis on the use of unmanned vehicles in oil/gas industry is also expected to increase because companies are heading towards deeper parts of the ocean to discover new oil and gas resources. Besides the oil/gas industry, telecommunication industry makes use of this technology for the inspection and installation of fiber optic cables.

Unmanned underwater vehicles are frequently used for military applications as well. The majority of military applications are mine countermeasures. These vehicles can both search for underwater mines and destruct them if detected. It is obvious that the risk of losing staff for mine destruction is minimized. Other than mine countermeasures; reconnaissance and survey are military actions that are executed. At present, they are not used directly for battles and their roles are limited. However, there is a tendency to increase their military capabilities. For this purpose, in December 2003 U.S. Navy prepared their new AUV master plan [9]. In this plan, they state that their aim is to increase the capabilities of these vehicles for future

battle situations. Some of the potential military duties in the future are attacking, defense, delivering military payload, anti-submarine warfare, patrol, mine countermeasures, forming underwater communication links, surveillance and tactical oceanography.

Academic organizations have been using UUV's for scientific purposes for a long time. The aim is to gather underwater data which would be difficult to get with conventional methods. These can be oceanographic data like ocean current information, species concentration or seabed mapping. Moreover, behavior of underwater animals and plant life can be inspected. Underwater archeology is also another field where scientists make use of UUV's.

### **1.3 History of UUVs**

Although the history of UUVs commences from 1960s, the concept of developing underwater submersibles is much older. The first example of manned submersibles is the "Turtle", the first American Submarine which was built in 1775. As the submarine technology progressed, a need for building unmanned submersibles for the purpose of launching explosives to enemies grew. As a consequence, torpedoes were developed in the late nineteenth century. Torpedoes are, in fact, assumed to be the first examples of unmanned underwater vehicles. The first self-propelled torpedo was developed by Robert Whitehead in 1866 [10]. The first torpedoes were using compressed air supply for powering and by 1870 they could maintain a cruising speed of about 6 knots. Whitehead continued his torpedo experiments in Austrian Navy. In the meantime United States Navy also started a torpedo program. Until the World War 1, military rivalry forced navies of the world to put emphasis on the improvement of torpedo technology. Depth control, gyroscope and radio guidance were added to torpedoes. Moreover, propulsion systems were improved and a cruising speed of 30 knots was achieved in 1890. Prior to World War 1, war noses and Anti-Circular Run (ACR) systems were utilized. ACR system prevented torpedoes to explode if they are headed to the ship or submarine itself. In fact, these

attempts were for the sake of adding intelligence to torpedoes and they can be accepted as the first steps of “Autonomy” for torpedoes.

Another important step in adding intelligence to torpedoes was the development of the “Homing Torpedo” during the World War 2. The idea behind it is that the torpedo would try to sense any underwater noise like the noise of the propeller of a submarine before attacking. Later on when the enemy ship or submarine is detected, the torpedo would hit. For this purpose, Mine MK 24 was developed which showed good success in the latter years of the World War 2. After this point, there was a tendency to improve the homing characteristics of torpedoes in order to develop more intelligent torpedoes.

While torpedo technology was progressing, a new technology of unmanned submersibles originated in 1960s when ROV’s were developed. ROV’s are unmanned underwater vehicles where human intervention and power transmission is accomplished with a tether from a surface vessel where a human operator or control unit is present. Unlike torpedoes, ROV’s house more thrusters in different locations and orientations; hence, have more degrees of freedom and this ability in combination with the human intervention enables ROV’s to perform difficult underwater tasks. Therefore, ROV’s have been used for various tasks like underwater rescue operations, underwater archeology, mine destruction and etc. since 1960’s.

Even though ROV’s make it possible to reach underwater locations which would have been impossible to do with human divers and performed various missions, there are several drawbacks. Underwater tethers are necessary and the operation depth is limited to tether length. The necessity of a surface vessel and continuous human intervention is also undesired. The motivation to overcome these drawbacks led to the development of AUV’s.

The first example of AUV’s is believed to be the Special Purpose Underwater Research Vehicle, SPURV, which was developed by the University of Washington in

1957. It had an operating depth of more than 3000 meters and could maintain a cruising speed of 4 knots. Except SPURV, it is seen in the literature that most of the early AUV's were developed in 1970s [1]. SPURV2 was developed by the University of Washington, EAVE by the University of Hampshire, SKAT, L1 and L2 by the Russian Academy of Sciences in 1970s. Moreover, Massachusetts Institute of Technology (MIT) launched its AUV program in 1973 even though the Autonomous Underwater Vehicle Laboratory was truly established in 1988. In 1970s, several experiments with these AUV models were performed in order to learn more about AUV's and their potential. In this era, the progress of the AUV technology was slower compared to ROV technology. The main reason is that AUV's needed new technologies for their development more than ROV's. AUV's do not have tethers with which power transmission and communication can be achieved. Moreover, they need to have small, high-performance computers on the robot and complex algorithms to perform pre-defined missions. In 1980s, the advances in computer technology and energy systems paced the progress of AUV technology. The first "International Symposium on Unmanned Untethered Submersible Technology" was also held in 1980. In 1992, MIT developed the Odyssey AUV which showed great success. Until late 1990s, various new versions of Odyssey were developed and they were tested in various places like Bellinghousen Sea in Antarctica to Beaufort Sea in the Arctic. The success of Odyssey initiated a collaboration between MIT and NATO in 1998. Odyssey performed various oceanographic experiments as a result of this collaboration. This proves that in 1990s, AUVs evolved from proof of concept designs to operational vehicles. In 2000s, commercialization of AUV technology also began to take place. Many commercial AUVs like Hugin from Norway, Maridan 600 from Denmark, AQUA EXPLORER 2 from Japan, Sea Oracle and CETUS II from USA and Explorer from Canada were developed [7]. These vehicles were able to execute numerous commercial tasks. For instance Maridan 150 was used for underwater archeology and Maridan 600 was used by De Beers for diamond mining survey. Hugin vehicles were used for underwater pipeline survey. These vehicles prove that the commercial acceptance of AUVs has occurred in the beginning of 2000s.

## 1.4 Literature Survey – UUV Examples

In unmanned underwater vehicle technology, it is difficult to define a standard for the vehicles being designed. When the literature is searched from the view point of a mechanical engineer, it is seen that there are numerous vehicles with different subsystems, sizes, shapes, materials and capabilities. Therefore, before starting a design, one should carefully investigate previous unmanned systems and determine basic features of them. In this part of the thesis, some of the systems that were investigated throughout the literature survey will be presented.

The MAYA AUV is developed in the National Institute of Oceanography in India [11]. It is a small vehicle with a length of  $L = 1.742$  m and a diameter of  $D = 0.234$  m. As seen in Figure 1.1, the vehicle has a torpedo like hull shape.



**Figure 1.1** The AUV – MAYA [11]

There is a cylindrical middle hull section which is water tight. This section is sealed and equipments like control unit and batteries which should be kept in a dry environment are kept located in this section. This cylindrical section is enclosed by tapered rear and forward cones. The shape of the forward cone is ellipsoidal in order to form a low-drag hull. The rear cone, on the other hand, is made more slender so as to provide appropriate flow into the thruster which is at the rear end. The forward and rear end caps are floodable. It means that these sections are not sealed and water fills these sections during service. In this wet section, sensors that should reside in water are located. The connection between the free flooding section and the waterproof region is achieved by underwater connectors which are mounted on the rear and forward end faces of the middle section. The ability to detach the end caps from the middle section also enables the user to access the devices inside the middle section which is good in terms of modularity.

The middle section is waterproof; therefore, it is expected that it will withstand high pressure values especially in case of deep submergence. The middle section is manufactured from aluminum 6082 with 6 mm of thickness and it was tested in a at a depth of 178 meters. The end caps; however, are not loaded heavily since those sections are floodable. They are manufactured from glass fiber reinforced plastic.

The MAYA AUV is propelled with a single DC brushless motor that is attached to the rear end of the vehicle. Maneuvering is accomplished by control surfaces. Two horizontal planes for vertical movement and a single rudder for horizontal movement are utilized. Symmetric NACA section profiles are used for these planes. Thruster capacity and power requirement was determined from drag tests.

Their design led to a total weight of 54 kgf while the available buoyancy was only 43.4 kgf. In order to balance the vehicle, the hull was wrapped with PVC foams. A density of  $200 \text{ kg/m}^3$  was used for withstanding high pressures. After trimming, a net buoyancy of 0.5 kgf was attained. After these studies, the vertical distance between

the centers of buoyancy and gravity were 7 mm. This value is small when compared to other AUVs in service. Detailed information about stability will be given in the following chapters.

Woods Hole Oceanographic Institute (WHOI) is one of the most important institutions in ocean science. They have been in the leading role for a long time in this field. Nereus is one of their most recent underwater vehicles [12]. The operating depth of this vehicle is 11000 meters which is considered to be a very high value. Nereus is a hybrid remotely operated vehicle (HROV): it can operate both as an AUV or ROV. In case of deep-submergence, the vehicle switches to an untethered mode, in other words AUV mode; alternatively, in case of a shallow water investigation the tethered mode, or ROV mode, is on. The switch between operating modes can even be executed during the mission in the sea. As seen in Figure 1.2, the vehicle has two hulls which are attached to the chassis. These hulls are free-flooding and two lightweight pressure hulls are placed inside them to house dry equipments. Pressure hulls are made from ceramic and titanium to withstand high pressures in deeper parts of the ocean. Buoyancy is attained by hollow ceramic spheres which are located on the top of the vehicle as seen. Junction boxes and pressurized hoses are used for underwater cabling. The vehicle has movable foils (or wings) at the aft and mid parts of the body. Two thrusters are attached to the aft foils, while another one is attached to the middle plane; therefore, the thrusters are movable and this gives the ability of vectored thrust. The vehicle has no motion in lateral direction in the AUV mode.





**Figure 1.2** Nereus in AUV mode [12]

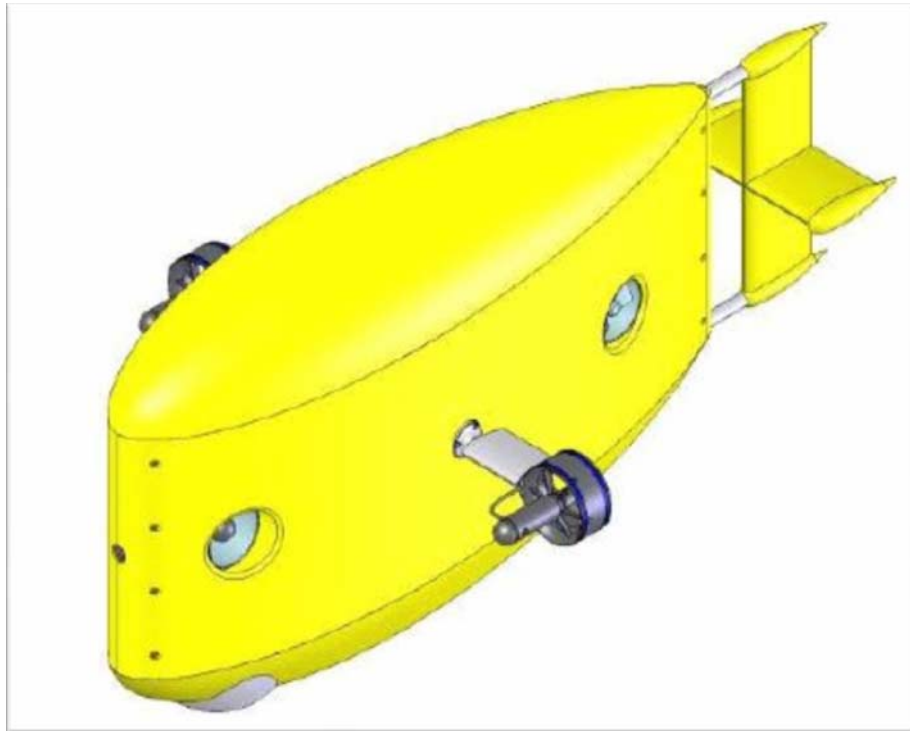
For AUV and ROV modes, different configurations are employed. As seen in Figure 1.3, an additional unit is attached to the bottom of the vehicle. It houses additional sensors, power source and a 6 Degrees of Freedom (DOF) robot arm. In ROV mode, lateral thrust is also available with an additional thruster.



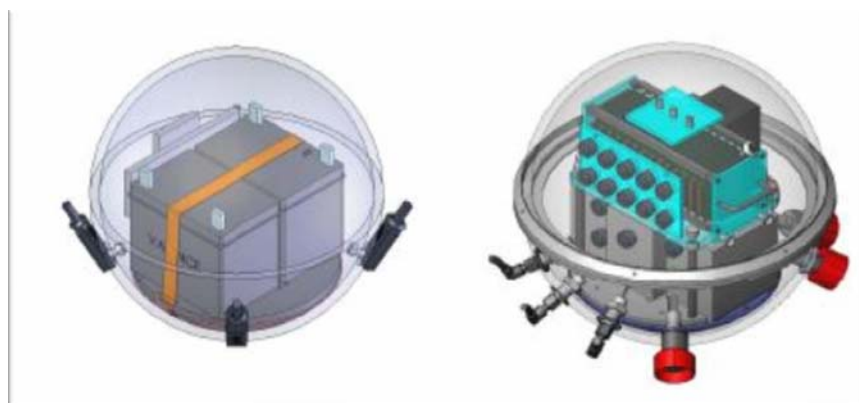
**Figure 1.3** Nereus in ROV model [12]

The Autonomous Underwater Vehicle Laboratory at MIT is one of the leading laboratories in AUV field. Hovering Autonomous Underwater Vehicle, HAUV, is an Odyssey 4 class AUV which is one of the AUV's that is being developed by the group [13]. It is designed as a relatively inexpensive vehicle which will be capable of performing deep submergence operations of about 6000 meters. Its weight is 350 kgf, length is 2.6 m, height and width are 1.3 m. As seen in Figure 1.4, the shape is different from previous examples and it does not have a torpedo shape. The vehicle has two forward looking thrusters which are attached to the body with movable foils. Their movement enables vectored thrust so three degrees of freedom are possible. An additional degree of freedom, lateral movement, is achieved by the lateral thrusters. These thrusters are placed inside a tunnel that passes through the body. Horizontal and vertical fins at the rear side of the body are fixed and they increase the roll stability of the vehicle. Like most of the designs of other MIT AUV's, HAUV has a floodable hull. Batteries and the control unit are housed in separate glass spheres as seen in Figure 1.5. These spheres are large; therefore, no buoyancy foams are needed. The vehicle is designed to be slightly buoyant. The spheres and other sensors

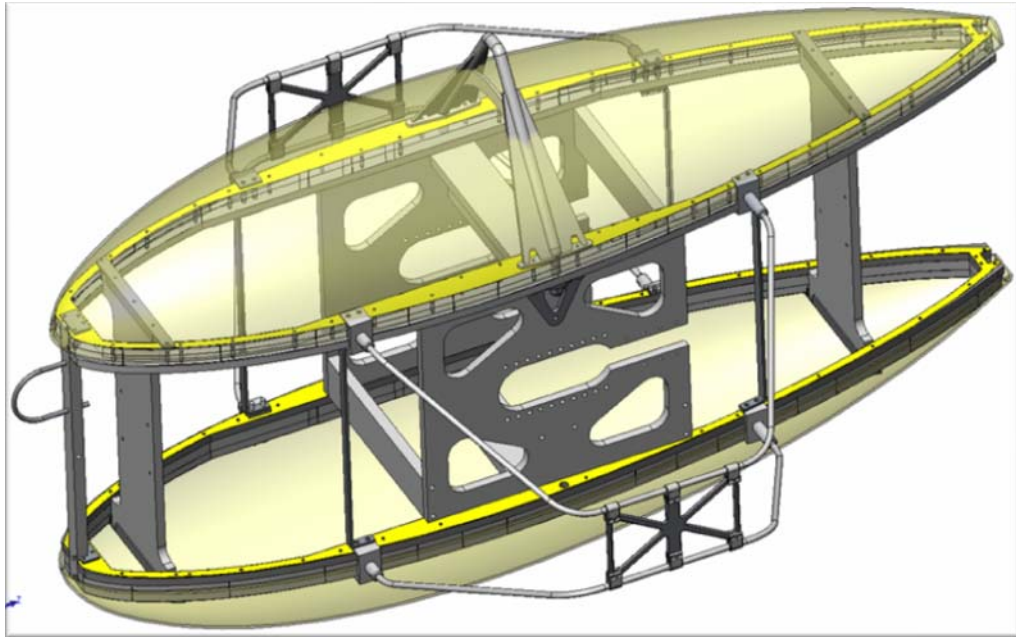
are attached to a plastic frame which is shown in Figure 1.6. Top, bottom and side faces of the frame are covered with 1/4" ABS fairings. The vehicle's distinctive shape results in decent pitch and roll stability characteristics. The horizontal and vertical fins that are attached from behind also improve stability.



**Figure 1.4** Odyssey 4 class HAUV [13]



**Figure 1.5** Pressure hulls of HAUV [13]



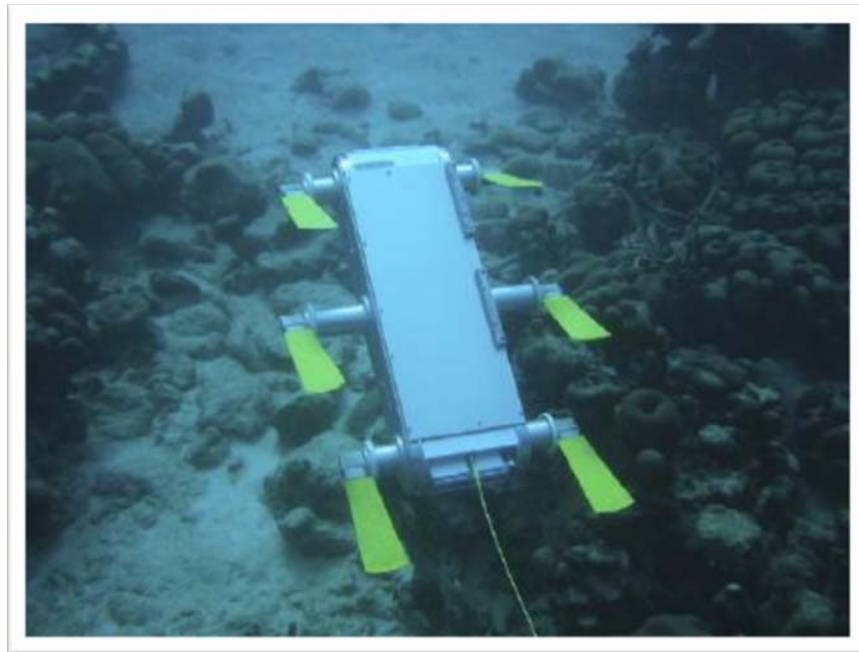
**Figure 1.6** Frame of HAUV [13]

Aquabox II is an AUV which is developed by the Kyushu Institute of Technology [14]. It is a small vehicle with a length of 1.2 meters and a weight of 60 kgf. The operating depth is 50 meters. As seen from Figure 1.7 the vehicle carries two forward thrusters at the sides and a vertical thruster which is mounted to a tunnel passing through the vehicle. Two additional thrusters are mounted horizontally for swaying motion. It should be noted that all thrusters are fixed so maneuvering is accomplished by the thrust differences between opposite thrusters. The vehicle has two pressure hulls inside the body both of which are manufactured from aluminum. Communication devices, batteries and motion sensors are placed at these hulls.



**Figure 1.7** Aquabox II

It should be noted that there are various unconventional unmanned underwater vehicles in literature as well. For instance, AQUA is an underwater robot based on the RHEX design as seen in Figure 1.8. It swims in water by paddling motion of six legs [15]. It also has the ability to crawl on the seabed.



**Figure 1.8** AQUA underwater robot

Besides that, unmanned vehicles that are built by inspiration from the nature, in other words biomimetics, are present. Vehicles that swim similar to turtles or sea snake are examples of such vehicles.

## **1.5 Basic Features of UUV's**

After the literature survey, it is seen that there are numerous unmanned underwater vehicles in use. All of these systems have different properties. They vary in shape, size and material selection. They have different thruster configurations and other subsystems. Therefore, it is difficult to define standards for these vehicles. Nonetheless; it is realized that all of these vehicles have common features even though their appearances are different. They all have an appropriate hull shape to house all subsystems and satisfy an efficient hydrodynamic design. Pressure hulls or vessels are present for keeping control units and batteries in a safe and dry environment. Propulsion and maneuvering systems are used depending on the required capabilities of the vehicle. A suitable internal design is done so that all

subsystems can function properly and the internal volume is used efficiently. Ballast systems are employed to maintain the balance and stability of the vehicle if necessary. Finally suitable underwater connectors and cables are used to connect separate subsystems.

## **1.6 Information about the Research Project ULISAR**

This thesis study is performed during the design of an unmanned underwater vehicle. The vehicle is designed and manufactured in a research project sponsored by the Scientific and Technological Research Council of Turkey and named as ULISAR Project (Ulusal İnsansız Sualtı Aracı Projesi). The primary goal of the project is to develop a prototype of an unmanned underwater vehicle that has the ability to swim steadily underwater, scan the environment, receive underwater image, store data and perform certain pre-defined missions. Later on, the level of the autonomy can be increased. New sensors or robot arms can be added for suiting different military, scientific or civilian purposes.

For achieving this, a multidisciplinary project group was formed. The project leader was Prof. Dr. M. Kemal Leblebicioğlu from the Electrical and Electronics Engineering (EEE) Department. Assistant Professor Dr. Afsar Saranlı from the same department and Professors M. Haluk Aksel, S. Kemal Ider and Levend Parnas from the Mechanical Engineering (ME) Department also worked actively for the project. Three graduate students from the EEE department and one graduate student from the ME department were present. Besides the academic part, BARIŞ Electrical Industry Inc. was responsible from the manufacturing and Yenel Ltd. Company was responsible from the manufacturing of certain electronic equipments.

The project is accepted to be one of the first efforts of designing an unmanned underwater vehicle in Turkey. Besides that, the vehicle has several distinctive features. It is able to operate autonomously to accomplish a pre-defined mission and when needed, wireless communication is possible via acoustic modem. Cameras and

lights are placed for getting underwater vision. Side scan sonar is used for scanning the environment and acoustic transducers are employed for detecting closer obstacles. Composite materials technology is utilized due to its novelty. The vehicle is capable of diving to 100 meters. The motion capabilities of the vehicle are surge, heave, pitch and yaw. No lateral motion is required and roll control is passive. The vehicle is expected to have a forward speed of 2 m/s.

## **1.7 Present Study**

In this thesis hydrodynamic and thermal analyses done during the design of an unmanned underwater vehicle hull are presented. In the first chapter, an introduction to unmanned underwater vehicles is made. Their classification, history and purposes are described. Moreover, some of the recognized unmanned underwater vehicles are examined and their common characteristics are defined.

In the second chapter, the conceptual design of the hull of the vehicle is carried out. Alternative hull shapes, propulsion and maneuvering systems are discussed and appropriate selections are made. Internal arrangement of components determines the dimensions of the hull. Furthermore, hull material and production method are decided at this step. In the third chapter the heating problem of the hull is investigated. Heating analyses and tests of the pressure hull are given in this part. In the fourth chapter drag calculations are performed in order to predict the performance of the proposed hull design and check whether speed requirements are met. In the fifth chapter brief information about the manufacturing and testing of the vehicle are provided. The thesis is concluded by mentioning experiences gained throughout the project and discussing alternative solutions that would make the design better and making suggestions for future studies.



## **CHAPTER 2**

### **CONCEPTUAL DESIGN OF THE HULL**

As mentioned before, all unmanned underwater vehicles have common features even if they may have different shapes, sizes, materials or capabilities. An appropriate hull design is one of the most important steps in the design. The hull, sometimes referred to as the fairing, is the structure that covers the vehicle. The hull has many important functions for the vehicle. First of all, it provides the vehicle an efficient hydrodynamic structure. This hydrodynamic structure decreases the drag force exerted on the vehicle during its motion and increases its stability which in turn causes a smooth and efficient motion. Moreover, it encloses subsystems in order to protect them from damages and environmental effects. Finally for vehicles that do not have a chassis, the hull serves as the chassis and holds equipments.

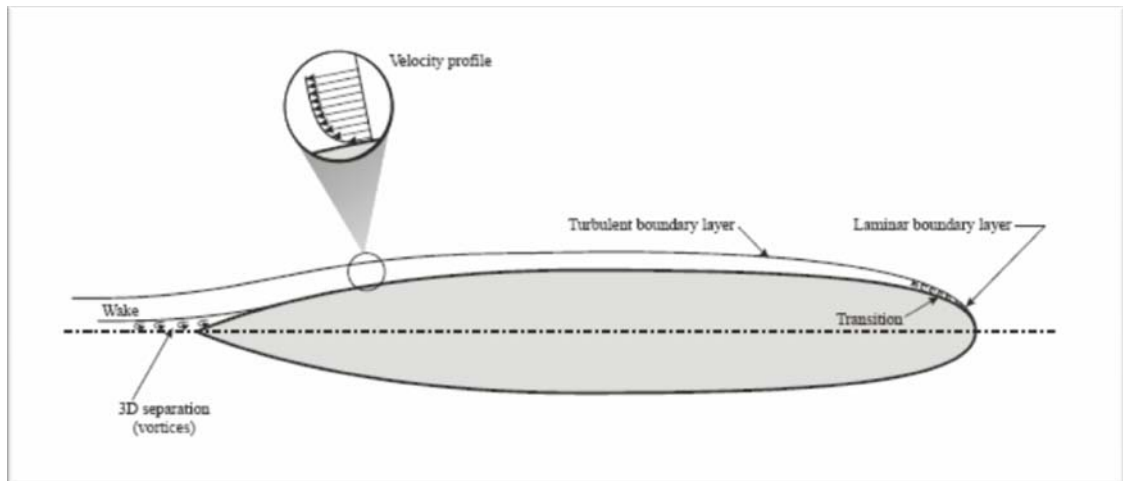
In this chapter, conceptual design of the hull of the unmanned underwater vehicle is presented. In this conceptual design effort, the first step is investigating hull shape examples in the literature and defining their advantages and disadvantages. Propulsion and maneuvering systems are also investigated. After these investigations, suitable selections are made depending on requirements. The dimensions of the hull are specified after the internal arrangement of components in the hull. Finally, hull material and production method selections are made.

## **2.1 Theoretical Background**

Before starting the conceptual design, it is beneficial to give information about parameters which are mentioned many times during the design process like skin friction drag, pressure drag, total drag force and prismatic coefficient. A body which travels underwater experiences water resistance. This resistance is called drag and it can be decomposed into two components: form (pressure) drag and skin friction drag. It should be noted that for bodies that travel close to the water surface, surface and wave effects are also significant and their effects should be considered in calculations as well.

### **2.1.1 Skin Friction Drag**

For the flow of viscous fluids over surfaces, a thin region named as the boundary layer forms on the surface. In this tiny region the flow velocity increases from zero at the wall to free stream velocity at the outer limit of the boundary layer; therefore, velocity gradients are very large in the boundary layer. This severe velocity gradient causes excessive shear stress in the region. Skin friction drag is the resultant force due these stresses and is a direct result of the viscosity of the fluid. The flow and the boundary layer over a submarine are seen in Figure 2.1.



**Figure 2.1** Flow around a submarine [16]

Skin friction drag value depends on the characteristics of the flow strongly. For laminar flow, thin layers of fluid slide over each other smoothly and this reduces the shear stress on the wall. On the other hand, in turbulent flows, the fluid is more chaotic with enhanced mixing, irregularities and 3D flows. As a result, velocity gradients and stresses are higher compared to laminar flow. Even though laminar flow is advantageous in terms of skin friction drag, it is hard to maintain laminar flow over the whole surface because laminar flow is very susceptible to disturbances. For almost all real flows, the flow remains laminar until a transition region and then turns into turbulent flow. This phenomenon is known as laminar to turbulent transition. Therefore, one should intend to delay transition as much as possible and increase the laminar portion of the flow on the body. At this point, it is beneficial to be able to guess the transition point and Reynolds number is used for this purpose. Reynolds number,  $Re$ , is a non-dimensional parameter which is the ratio of inertial forces to viscous forces.

$$Re = \frac{\rho U l}{\mu} \quad (2.1)$$

The characteristic length  $l$  is an appropriate length depending on the case. For a flow over a streamlined body, one can take  $l$  as the distance from the front end of the

body. It is obvious that Reynolds number increases with the distance and after a critical Reynolds number transition happens.

Skin friction is related to the surface of the body. Due to that decreasing the surface area, which may also be referred to as the “Wetted Surface Area”, reduces the skin friction.

### **2.1.2 Form Drag**

For a body moving in fluid, a pressure distribution forms on the body. The pressure is highest at the nose due to stagnation, decreases as the fluid accelerates on the contour of the body and is recovered at the stern. For ideal (inviscid) flows, the pressure is fully recovered and there is no pressure difference between the nose and the tail. However, for viscous flow there is a pressure difference which causes the pressure or form drag. Form drag depends strongly on the shape of the body and its pressure recovery characteristics.

An important phenomenon for form drag is the flow separation. When the fluid follows the contour of the body, it may experience pressure force against its motion and this pressure is known as adverse pressure. In such cases, the fluid is decelerated and can even be reversed at some point. At this point, the flow is not attached to the body anymore and it is separated.

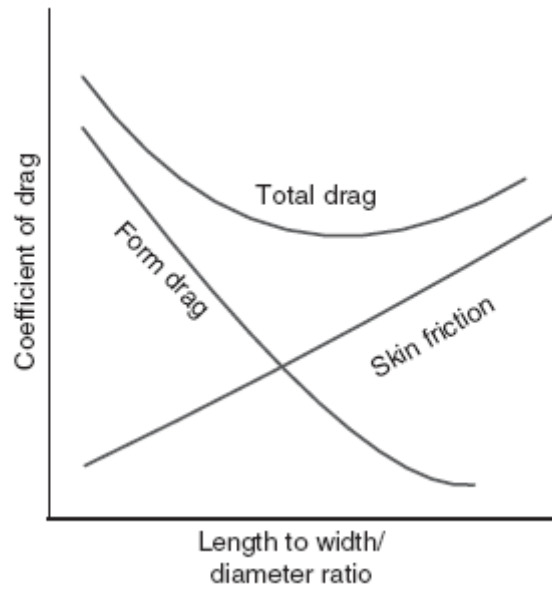
Separated flows damage pressure recovery. A wake region is formed behind the body which increases the pressure difference between the nose and tail sections. For that reason, one should pay attention to prevent separation. Turbulent flows are superior over laminar flows in this aspect because they attach to the surface better. Flow over a cylinder exemplifies this situation. For laminar flow, drag coefficient,  $C_D$  is equal to 1.2 due to early separation. For turbulent flow  $C_D$  drops to about 0.3-0.4 since the flow separation is delayed.

For reducing form drag, abrupt changes in the cross-section of the body should be avoided. A streamlined body with high length-to-diameter ( $L/D$ ) ratio is useful in this aspect. Furthermore, in critical cases designers tend to use methods like tripping the boundary layer to promote transition to turbulence and sucking the boundary layer in order to delay separation.

### **2.1.3 Total Drag Force**

Drag force is the combination of the form drag and the skin friction drag. In order to be able to decrease the total drag force (resistance), one should understand how these drag forces contribute to the total drag.

For streamlined bodies like an airfoil, the contributions of friction drag and form drag are comparable. In this case, it will be helpful to reduce the wetted surface area. On the other hand, for bluff bodies, resistance is mainly due to form drag. One has to make the body more slender in order to reduce the form drag. However, making the body slender increases wetted surface area which in turn increases the skin friction. It can be inferred that there is an optimum aspect ratio for minimum drag. A familiar graph is given in Figure 2.2. It is seen that there is an optimum  $L/D$  ratio. However, this ratio is not strict and using slightly lower or higher values do not change the resistance significantly. Numerous studies have been performed to determine the



**Figure 2.2:** Vehicle drag curves [3]

optimum aspect ratio. One of the earliest and most well-known studies was performed by Morton Gertler at David Taylor Model Basin in 1950 [17]. Gertler tested 24 underwater bodies at various velocities to collect drag data in deep submergence. Bodies were axisymmetric and formed from sixth degree polynomials. The purpose of the research was to understand the effects of various parameters like length-to-diameter ratio, prismatic coefficient and nose diameter on resistance. When he compared the results of bodies with the same volume, he reached the conclusion that optimum aspect ratio  $L/D$  is 6.5. Certainly, this value is not super critical. One can use higher or lower aspect ratios depending on limitations. For instance, for a vehicle like torpedo which probably has diameter limitation, the length should be increased to obtain the same amount of volume.

For comparing performance of different vehicles a non-dimensional coefficient called the drag force coefficient is defined as:

$$C_D = \frac{F_D}{0.5\rho V^2 A} \quad (2.2)$$

$F_D$  refers to the total drag force and  $A$  refers to the appropriate area. For bluff bodies, the area will be taken as the projected area normal to the flow direction or the frontal area, while it will be area as seen from top or the planform area for streamlined bodies.

#### **2.1.4 Prismatic Coefficient**

Prismatic coefficient is another important parameter for underwater vehicles. Prismatic coefficient is a non-dimensional value which represents the fullness of the body at the ends. It is the ratio of the volume of the body to a hypothetical volume of the prism which would be obtained by the mid-ship cross-sectional area and length. Gertler compared models with the same  $L/D$  ratio and varying prismatic coefficients. For both  $L/D$  ratio 5 and 7, he reached to the conclusion that a prismatic coefficient of 0.61 is optimum. This value is somewhat low and limits the internal space available. A solution for this problem is using a parallel mid-body [18]. In that solution, it is seen that when about half (0.495 times) of the total length is allocated as a parallel mid-body, one can achieve the same resistance as the prismatic coefficient increases from 0.682 to 0.84. The increase in prismatic coefficient means more internal space and it is obvious that it is extremely important for underwater vehicles.

## **2.2 Hull Requirements**

Before choosing the hull shape, one needs to define hull requirements clearly. The most important function of the hull is providing the vehicle an efficient structure in terms of hydrodynamics. Form drag and skin friction drag should be minimized in order to reduce power requirements. It should also be noted that drag losses also determine the range, maximum speed and operating time of the vehicle. The hull should also allow a smooth flow over it for stable motion. These items gain importance especially for cases where speed and maneuvering criteria are strict. The

hull should have a symmetric structure for rotational stability when it is stationary or moving. The size of the hull is also an important criterion. For most cases, it is not possible to try extreme values because there are restrictions in terms of size. The form drag usually limits the maximum diameter. On the other hand, minimum diameter is limited since the hull should be large enough to cover subsystems. Length of the hull should be large enough for all the subsystems to fit and small enough to minimize skin friction force and cabling requirements. The hull should be thick enough to have enough strength to withstand pressures and possible damages. If the whole hull is water tight, the thickness criterion becomes more important as the pressure force increases. Besides that, maximum length, thickness and diameter are limited to satisfy maximum weight criterion. One cannot design the hull without considering manufacturing. The design should be as simple as possible to reduce manufacturing costs and time. The hull should be ergonomic so that the users can easily carry and deploy the vehicle to water. Finally aesthetic issues become important especially for commercial uses.

These requirements are mostly determined from the project definition. It is clear that it is difficult to satisfy all of these requirements simultaneously. It should also be noted that some of these requirements may be coupled; therefore, care should be taken while changing parameters. The designer should begin with an initial design and modify that design depending on requirements. The modified design should be checked and modified again if necessary. At each step, the designer should make necessary calculations and simulations. This iterative procedure continues until the optimum design is achieved. It is important to note that feedbacks of other members of the design group should be taken into consideration for avoiding potential problems.

## **2.3 Hull Shape Selection**

During the literature survey, it was seen that hull shapes vary significantly for different vehicles and some examples of hull shapes were presented in the previous



chapter. It is difficult to make a classification for AUV shapes; however, one such classification is done by the Autonomous Undersea Vehicle Applications Center (AUVAC). According to AUVAC, autonomous underwater vehicles can have teardrop shape, torpedo shape, rectangular shape, oblate shape, open space frame and biomimetic shape [19]. Each of these shapes has advantages and disadvantages; therefore, the designer should pay attention to choose the shape that will fulfill project requirements best.

### **2.3.1 Teardrop Shape**

Some examples of teardrop shapes vehicles can be given as the Alistar of ECA and Odyssey II of MIT AUV Laboratory. These vehicles can be seen in Figures 2.3 and 2.4. Teardrop shape is an efficient shape in terms of resistance. In fact, it is assumed to be the ideal shape since the cross-section varies continuously and slowly preventing early separation of the flow. On the other hand, there are some drawbacks. The first one is the reduction in internal space (low prismatic coefficient). Moreover, the continuously changing contour can make it difficult to house equipments and use the space efficiently. Finally, depending on the production method, manufacturing the continuously changing contour can be a difficult task.



**Figure 2.3** Alistar of ECA [19]



**Figure 2.4** Odyssey IIc of MIT AUV Laboratory [19]

### 2.3.2 Torpedo Shape

Torpedo shape is a frequently used shape for underwater vehicles. These shapes have a cylindrical mid-body, ellipsoid fore body and a parabolic after body. Some examples of such vehicles are Autosub 6000 of Subsea7 and Hugin 1000 of Kongsberg Maritime, which can be seen in Figures 2.5 and 2.6, respectively.



**Figure 2.5:** Autosub 6000 [19]



**Figure 2.6:** Hugin 1000 [19]

Torpedo shape has several advantages. First of all it gives more free space to the designer due to its cylindrical middle body. These vehicles are usually manufactured in 3 pieces. The middle, fore and aft bodies are manufactured separately and then assembled. This property provides a modular structure. The vehicle can easily be

modified. Design and manufacturing is easier. Finally performance is comparable to teardrop shapes. Due to these advantages, they are commonly used for underwater vehicles, submarines and torpedoes.

### 2.3.3 Rectangular Shape

Rectangular shaped unmanned vehicles are very similar to torpedo shaped vehicles as it can be seen in Figure 2.7. Urashima of Japan Agency for Marine-Earth Science and can be given as an example for this category.



**Figure 2.7** Urashima AUV

These vehicles have a parallel mid body enclosed by tapered ends. They may house thrusters and control planes at the tail section like torpedo shaped vehicles. One advantage over the torpedo shapes is that rectangular cross-section can be used more efficiently for placing equipments. On the other hand, performance is lower.

### 2.3.4 Oblate Shape

Some AUVs are designed to have a more flat structure as seen in Figure 2.8. These vehicles allow more than one thruster to be located on the vehicle as seen in Double Eagle MKII of Saab Underwater Systems. This thruster configuration provides better maneuverability. However, performance is lower due to usage of more thrusters compared to the torpedo shape where one thruster is used.



**Figure 2.8:** Double Eagle MKII of SAAB Underwater Systems [19]

### 2.3.5 Open Space Frame

In certain cases, it may be helpful to use more than one hull. These hulls are usually attached to a main frame. One example can be seen in Figure 2.9. Different equipments like electronic cards, batteries and sensors can be placed at different tubes. One major advantage is that high stability can be achieved by placing heavy components at the lower tube and light components at the lower one. Separating certain equipments may also be beneficial for their proper functioning. Besides that, the chassis allows the user to place underwater equipments easily. On the other hand, performance will be lower compared torpedo shapes. Manufacturing may also be more difficult.



**Figure 2.9:** SeaQuest [19]

Other than these shapes, biomimetic shapes are gaining popularity in recent years. Vehicles that are inspired from animals like penguins, fishes, snakes are being developed. Even though these technologies are promising, it is hard to tell that they are reasonable alternatives.

Every hull shape has advantages and the selection depends on the designer. For the project study, a torpedo shaped body was chosen due to certain reasons. First of all it gives one of the best performances in terms of drag amongst hull shape examples. The cylindrical middle body allows an appropriate place for placing electronic equipments and the front and rear sections are allocated for the storage of underwater equipments. Torpedo shape is the most popular hull shape which provides a larger

database. For a project group who is designing such a vehicle for the first time, large database is helpful.

## **2.4 Propulsion and Maneuvering**

The selection of the appropriate hull shape is followed by the propulsion system selection. Choosing the systems depends on various considerations. Propulsion type, configuration, efficiency, power requirement, size, thrust value and cost are primary considerations in this selection.

### **2.4.1 Propulsion Type**

Electrical, hydraulic and jet propulsion are used by underwater vehicles [3]. Jet propulsion and hydraulic propulsion are very rarely used for special cases. During the literature survey it was seen that almost all systems rely on electrical propulsion. In this method, a shaft with propeller blades is rotated with an electric motor that can be located inside the vehicle or inside a separate casing. The rotation of propeller blades causes an increase in flow velocity across the propeller and this causes a forward thrust on the vehicle.

The efficiency of a thruster is affected by many parameters. One of the most important parameters is the quality of the flow leaving the thruster. Propeller blades create a rotational effect on the exiting flow and this effect reduces propulsion efficiency. Moreover, tip vortices are formed at the tips of propeller blades which also have negative effect on efficiency. To reduce the effects of these, underwater thrusters are enclosed by a casing named as the shroud, duct or a kort nozzle. This type of ducted propeller or kort nozzle give better performance compared to thrusters without a nozzle. One of the main reasons for this is that the tip vortices are suppressed by the duct and the flow exits axially with less swirl. On the other hand, the shroud causes an additional drag. For low speeds, the increase in propulsion

overcomes this drag; however, for higher speeds the shroud begins to have a negative effect on propulsion. Therefore; ducted propellers are used for low speed applications like unmanned underwater vehicles.

Podded propulsors are a special type of a propeller where the DC motor rotating the shaft is inside a hub and the propeller blades are attached to that hub from behind. The main advantage is that these propulsors house their own motors and do not require a DC motor on the vehicle. This gives more internal space for other equipments and eliminates the risk of water leakage through propeller shaft. Furthermore, the hub provides the vehicle an axial inlet flow which in turn increases propulsion efficiency. An example of a podded propulsor with kort nozzle is seen in Figure 2.10.



**Figure 2.10: Podded propulsor – BTD150 model of Seabotix**

Design and manufacturing of a thruster is a very complicated issue and it is in fact a separate project topic. Consequently, in this project thrusters were purchased. Due to the advantages stated above, a podded propulsor with a shroud around it used. The details of the propulsor will be presented in the chapter on performance chapter.

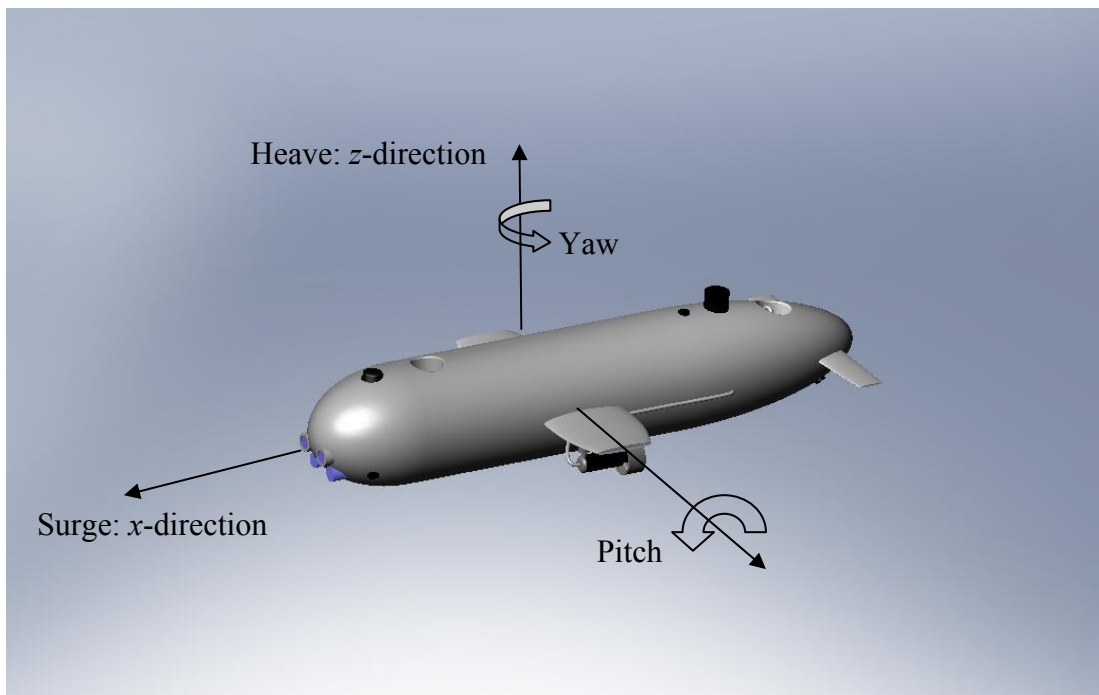
## **2.4.2 Maneuvering**

Besides the propulsion system selection, thruster configuration of the vehicle is important for maneuverability of the vehicle. Before considering alternatives, it is



necessary to define required degrees of freedom of the vehicle. ULISAR is expected to have surging motion, heave, pitch and yaw. Lateral (sway) motion and roll will not be available. In fact roll is intended to be eliminated passively. These degrees of freedom are presented in Figure 2.11.

During the literature survey, it was seen that there are various approaches for maneuvering. Generally, there are 3 means of maneuvering. The first one is using a single propeller at the aft and along with control surfaces. This is a common approach used for unmanned vehicles especially for torpedo shaped ones. Pitch and yaw motion is attained by the deflection of control planes. The advantage is that using a single propeller increases the efficiency of the system. Therefore; for long-range operations this solution is preferred. The disadvantage on the other hand is that these vehicles are maneuverable only when the vehicle is surging. They do not have the ability to do pitching, yawing and surging motion when stationary. Moreover, sealed shafts should be designed for controlling control surfaces.



**Figure 2.11:** Maneuvering capabilities of the vehicle

Another alternative is vectored thrust. In this method one or more thrusters are connected to a shaft which can change the angle of the thruster. As a result, the thrust vector changes direction and vehicle can maneuver. An example is Seawolf 2, seen in Figure 2.12, which is developed by the North Carolina State University Underwater Robotics Group. This thruster arrangement is the best in terms of maneuverability. It is preferable for missions where long range is not critical. The disadvantage is that these vehicles are inefficient for surging motion and require sealed shafts for thruster control. Unmanned underwater vehicles with vectored thrust are not so common. Systems with vectored thrust have only one rotational motion for each thruster. A conceptual design of a vectored thrust system which consists of a single thruster that has 3 degrees of freedom is seen in the study of Cavallo et al. [20]. In their study, they claim that such a system can satisfy propulsion and guidance tasks.

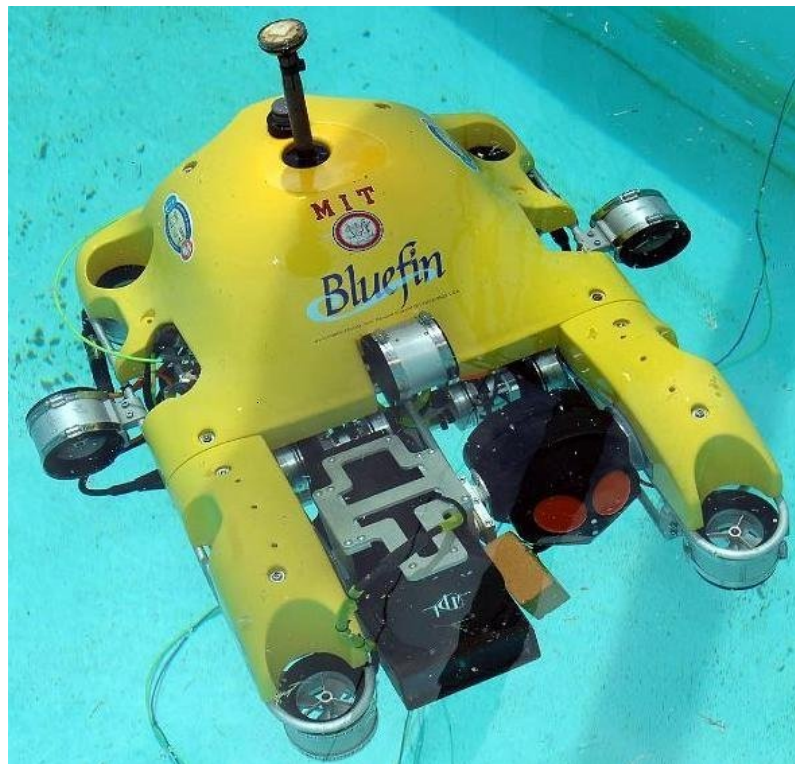


**Figure 2.12:** Seawolf II of North Carolina State University

Using multiple thrusters in different arrangements is a frequently used method for maneuvering. In this method thrusters are placed symmetrical with respect to the center of mass. Maneuvering is achieved by the thrust difference and the resulting moment. The required maneuvering ability determines the number of thrusters and their arrangement. In Figure 2.13 an example of such a vehicle is seen. This vehicle

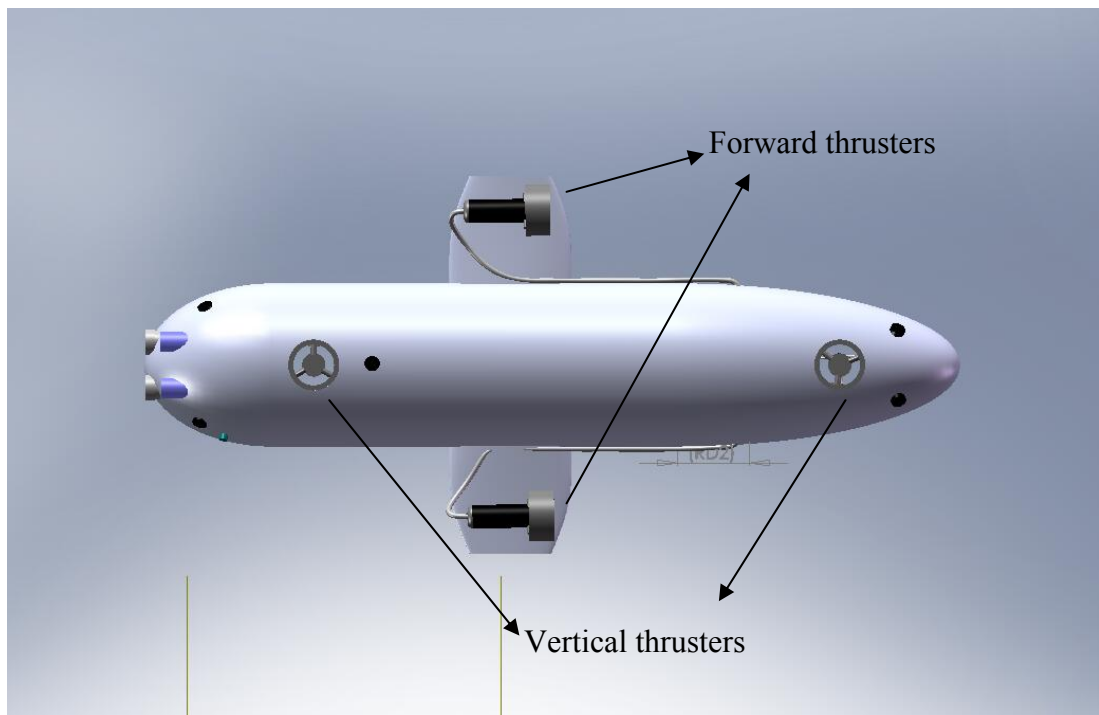
has 5 thrusters. Two of these thrusters are forward thrusters, two of them are vertical and one of them is lateral. Therefore, this vehicle is capable of surge, heave, sway, pitch and yaw motions.

Multi-thruster arrangements are useful in terms of maneuvering. However, they are not efficient for long range operations since increasing the number of thrusters decreases propulsion efficiency. One other drawback of multi-thruster arrangements is the torque steering [3]. As it was stated before, flow exits with swirl from the propeller. This swirling flow induces torque on the vehicle. When two forward thrusters are working simultaneously, their effects are combined which increases the torque on the vehicle as well. Shrouds around thrusters and flow straightening blades at the exit of propellers usually suppress this effect. However, the ideal solution is employing counter-rotating thrusters. These thrusters generate opposite torques which cancel each other.

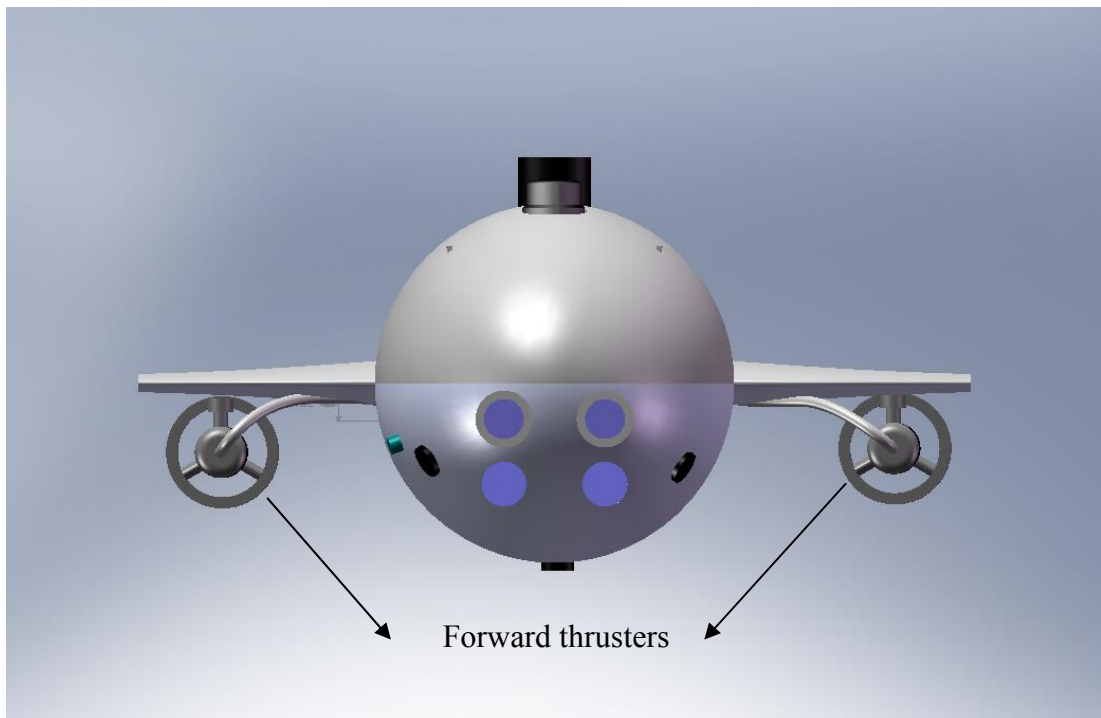


**Figure 2.13:** Bluefin unmanned underwater vehicle

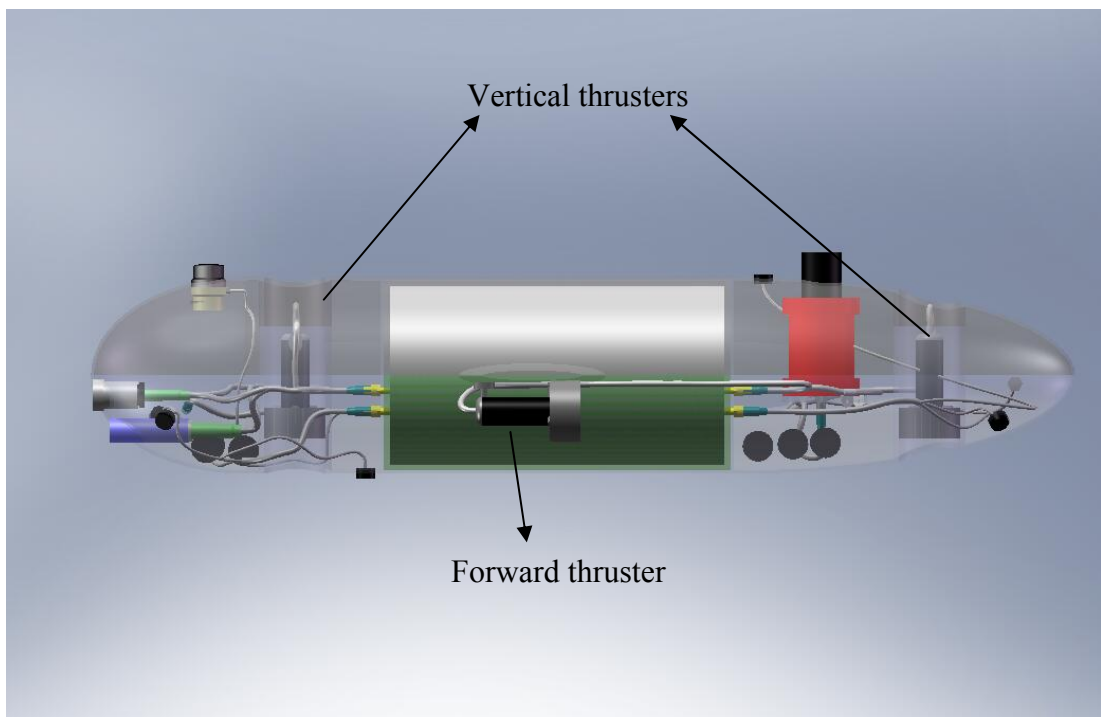
After investigating these thruster configurations, one can deduce that the designer has to choose the appropriate one depending on project requirements. Single thruster with control surfaces is efficient and used for long range operations; however, it will be less maneuverable. On the other hand multi-thruster and vectored thrust options give maneuverability with lower efficiency. ULISAR is the first prototype of an unmanned underwater vehicle and at this point it is not expected to have long range operations. Initial experiments will be at the pool. Considering that, maneuverability is more valuable than efficiency at this moment. Moreover, hovering motion will be beneficial for pool experiments as well. Due to these reasons, a multi-thruster configuration was chosen for propulsion and maneuvering. 4 thrusters will be used for surge, heave, pitch and yaw motions. The proposed configuration is seen in Figures 2.14 to 2.16.



**Figure 2.14:** Thruster configuration – Bottom view



**Figure 2.15:** Thruster configuration – Right view

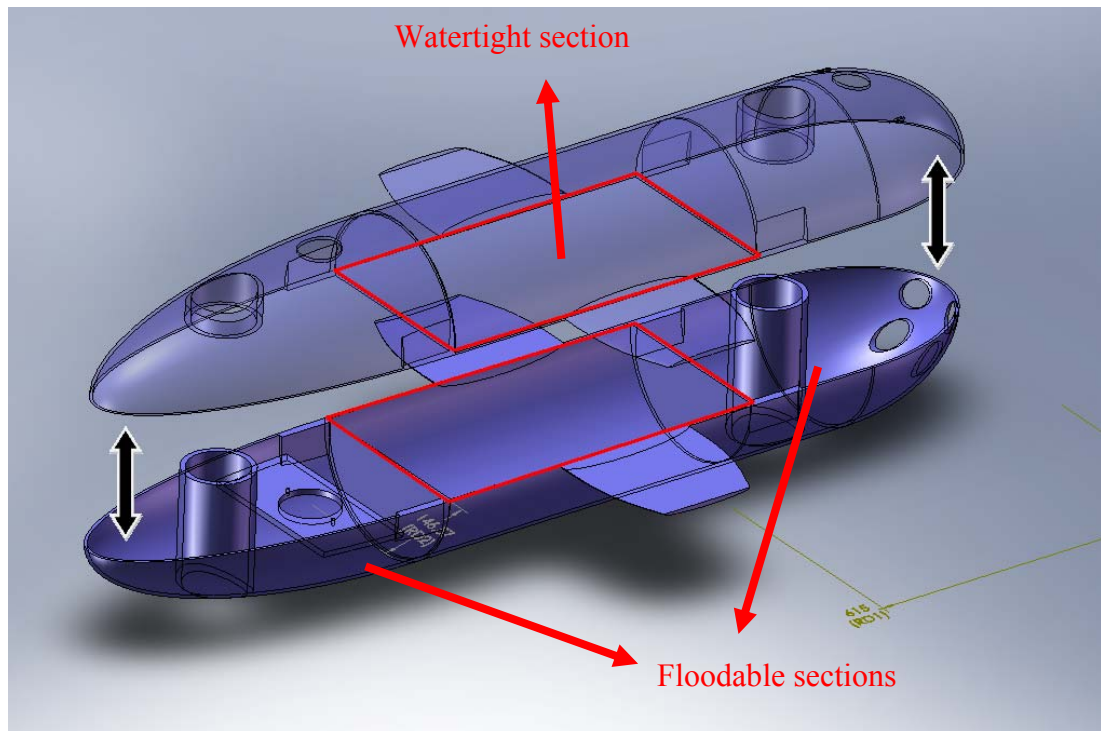


**Figure 2.16:** Thruster configuration – Front view

## 2.5 Internal Arrangement

After making a suitable selection for the hull shape and propulsion system, one has to specify the internal arrangement of the body and the dimensions of the hull. During the literature survey part of the thesis, several underwater configurations were investigated. From these investigations some conclusions were made. Firstly, almost all vehicles have floodable sections. It means that the vehicle houses electronic cards and batteries in a separate dry pressure hull while other equipments are kept in the floodable wetted sections of the vehicle. Such a design approach has benefits over a design which would have a hull that is completely sealed. The first advantage is that the sealing surfaces are reduced by this design because the waterproof region is smaller. Besides that, most of the equipments need to be in the wet section or at least must have protrusions out of the body to function properly. Acoustic modem, side-scan sonar, pressure sensor and temperature sensor are some examples of such devices. If the designer aims to seal of these protrusions, the risk of leakage will increase. The leakage of one of the systems in the design may even cause the failure of the vehicle. Finally, a floodable hull increases the wetted area of the pressure hull; hence, increases its heat removal capabilities. Due to these reasons a floodable hull is used in the design.

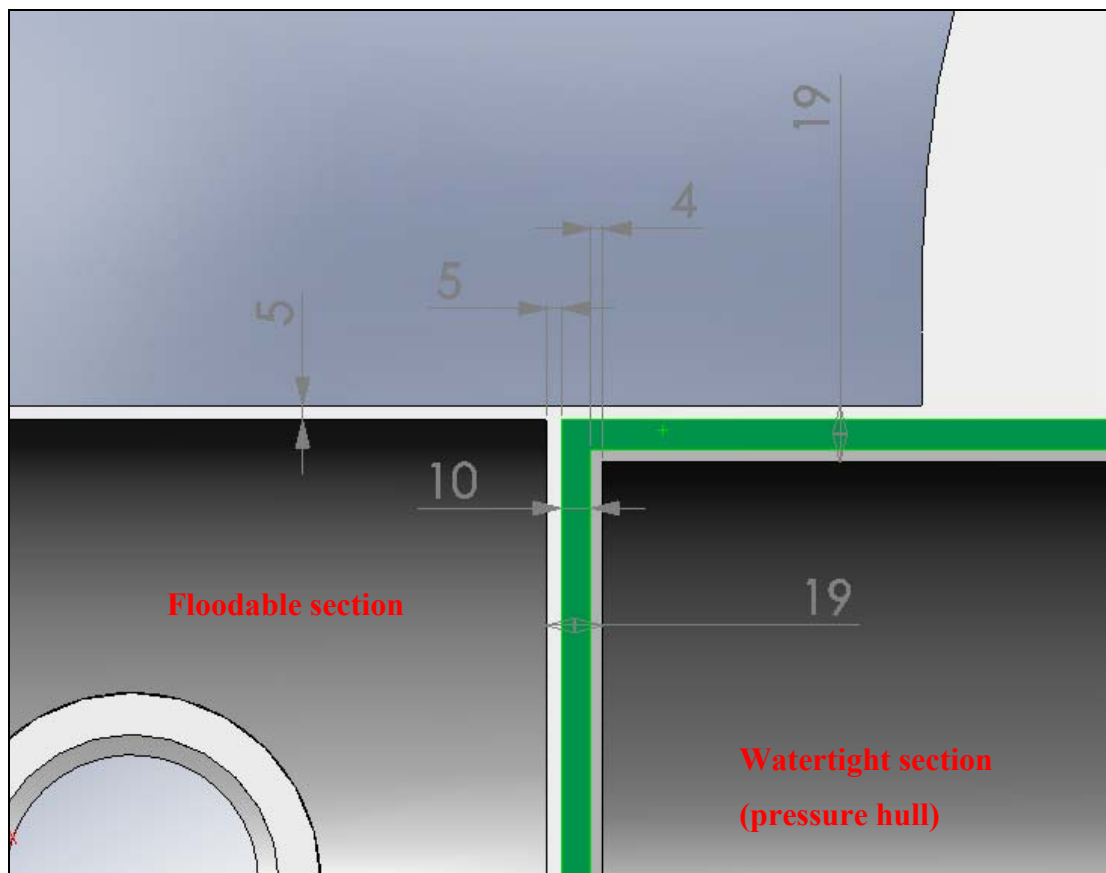
A floodable design was proposed as seen in Figure 2.17. It is seen that the hull consists of the upper and lower bodies. The red lines in the figure represent surfaces where sealing is applied. When these two bodies are attached properly, those sealing surfaces match and a watertight region, which may be referred to as the pressure hull, forms in the middle of the hull. Electronic equipments that should be kept inside a dry section are housed there. The sections in both sides of the pressure hull are floodable. Water enters into these sections as the vehicle is deployed to the water and drains the water when it is recovered.



**Figure 2.17** Lower and upper bodies

The idea behind designing such a two piece body was the ambition to use only one mold. As it is understood from Figure 2.17, the upper and lower bodies are exactly the same besides the internal parts. Therefore, these two bodies are produced from the same mold to reduce expenditures and manufacturing time. An alternative to this configuration may be designing three axisymmetric bodies that are attached to each other from sides. Compared to that configuration, this configuration is better in terms of strength because the upper and lower bodies are one piece solid components. Moreover, all of the connectors, electronic components and batteries are attached to the lower body. Therefore, the upper body can be detached easily and once it is detached, it makes it easy to access all of the equipments in the hull. On the other hand, this design hinders further modifications. For the three body configuration where three axisymmetric bodies are attached from the sides, it is easier to modify the vehicle by designing one of the bodies again. Conversely, for the present design, it is difficult to adjust the shape since one has to design and manufacture the whole body again.

Since the sections at the front and rear parts of the vehicle are floodable, there is no need to make the walls of those sections thick as they are not loaded under hydrostatic forces. On the other hand, the middle section resists excessive pressure forces because it is watertight. The walls of this section should be thicker to resist high pressures and create room for the seal to be attached. As a result, a variable thickness body was proposed. The details can be seen in Figure 2.18.



**Figure 2.18** Hull thickness differences

Wall thickness values and similar other critical values for manufacturing were determined by the manufacturing company. Strength analysis for composite structures is a very detailed topic and it is out of the scope of this thesis.



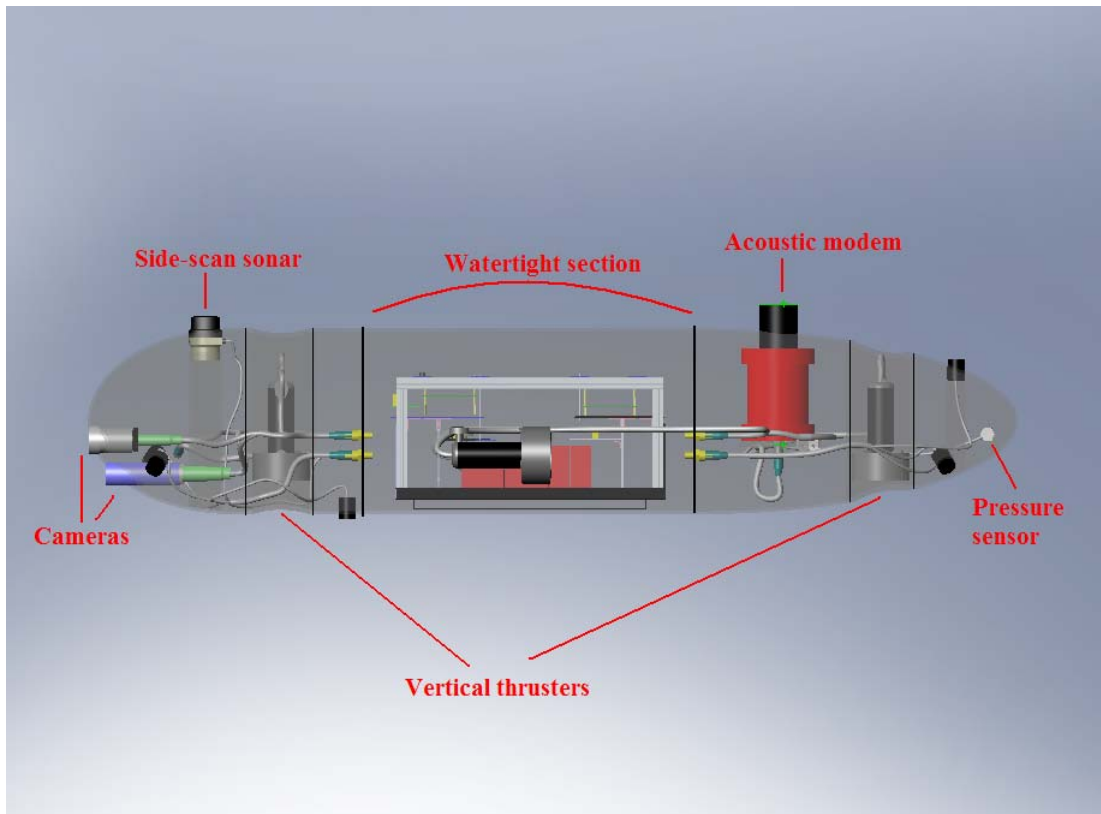
### **2.5.1 Internal Arrangement of Devices**

The dimensions of the hull are specified as the devices are located in it. The hull has two sections as stated before: floodable section and water tight section. Acoustic modem, sonar, cameras and lights, pressure sensor, thrusters and acoustic transducers are devices that are housed in the flooded section; on the other hand, electronic cards and batteries are located in the watertight section. These devices have conditions about their orientation in the hull:

- Acoustic modem which is seen in Figures 2.19 and 2.20 should remain vertical with its tip protruding out of the body.
- Sonar should remain vertical with its tip protruding out of the body. Since it is a side-scan sonar, it should be located near the front of the body to be able to scan forward.
- Cameras and lights should be located in front of the body to have forward vision. They should be separated with a distance of 4 cm.
- Vertical thrusters should be located in tunnels that pass through the body. They should be separated from each other as much as possible to increase the pitch motion capability.



the hull is set to be 32 cm. In order to meet location conditions of other devices, the internal arrangement seen in Figure 2.21 is proposed. Technical drawings of upper and lower bodies are also seen in Figures 2.22 and 2.23.



**Figure 2.21** Internal arrangement of devices

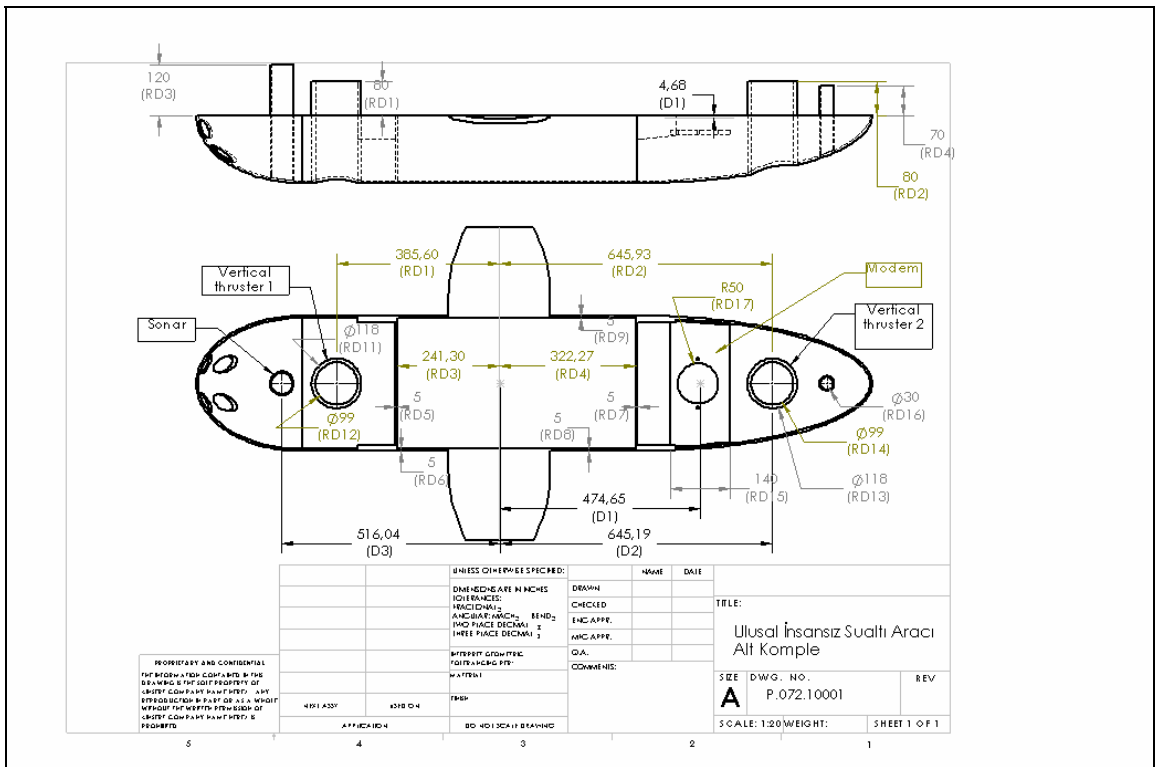


Figure 2.22 Technical drawing of the lower body part

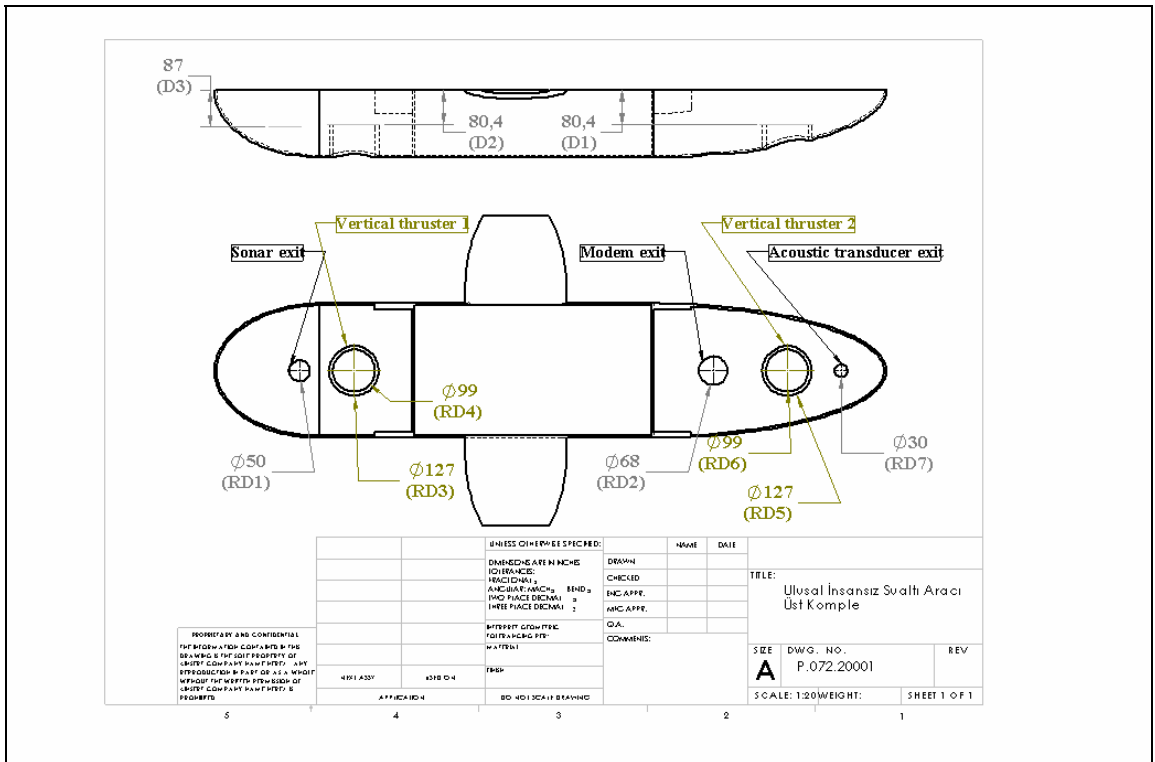


Figure 2.23 Technical drawing of the upper body part

While locating components inside the hull, a minimum separation of 5 cm is given between devices. The design proposal resulted in a body with a length of 1.6 m and a diameter of 0.32 m.  $L/D$  ratio becomes 5. In fact, this value is slightly lower than the optimum  $L/D$  ratio. Gertler states that  $L/D = 6.5$  gives the minimum total drag for streamlined bodies [17]. Increasing the length of the vehicle could be done; however, it would increase the weight of the vehicle further. Nevertheless, as it is seen in Figure 2.2, assigning slightly lower or higher values to  $L/D$  do not increase the total drag significantly.

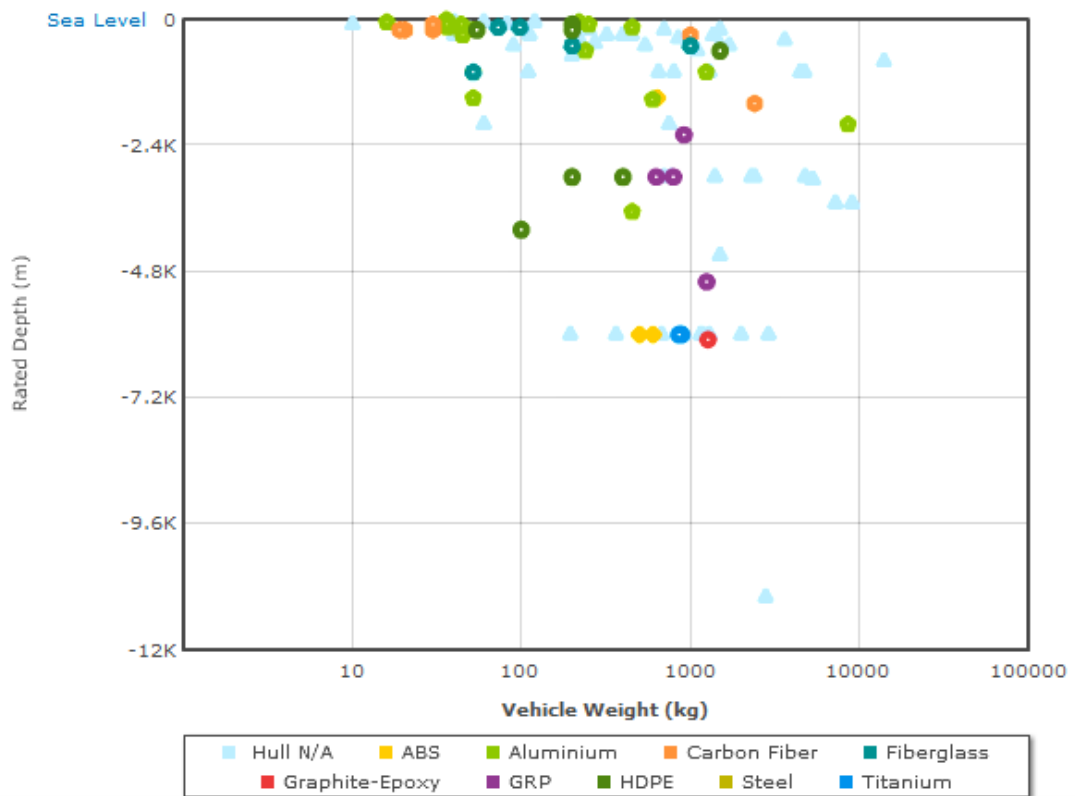
Gertler's studies show that prismatic coefficient value affects the resistance significantly [17]. He shows that a prismatic coefficient of about 0.61 is optimum for a range of  $L/D$  ratios between 5 and 7. A prismatic coefficient of 0.61 reduces the internal space available for devices significantly. A solution for this deficiency is employing a cylindrical middle section as it was given before [18]. Employing a cylindrical mid body with a length 0.495 times the length of the vehicle provides a prismatic coefficient of 0.84 with approximately the same resistance value which would be experienced with a prismatic coefficient of 0.682 with no parallel mid body. For this purpose a parallel mid body with a length of 0.79 meters is used which corresponds to a mid body ratio of 0.49 (close to 0.495). The dimensions of the front and rear ellipsoid sections are specified such that the prismatic coefficient of 0.84. The resulting volume of the body is  $0.107 \text{ m}^3$  which gives a prismatic coefficient of 0.832 which is close to the intended value of 0.84. Therefore, one can argue that the resistance is minimized for the  $L/D$  ratio of 5.

## 2.6 Material Selection

Determining the hull shape and its dimensions is an important step in the design. After choosing the shape, material of the hull should be determined. It should be noted that the material selection is significant because it affects various design parameters like weight, heat removal capabilities, manufacturing cost and time.

Moreover, design capabilities are affected. For instance an internal design approach which would be viable with a composite body may not be feasible with aluminum.

When the literature is searched, it is seen that various materials have been used for the manufacturing of underwater vehicles. The most common materials being used are aluminum, fiberglass composites and carbon fiber composites. A chart that is adopted from [auvac.org](http://auvac.org) exemplifies this situation; hull material alternatives and their suitability at different depths are demonstrated [19]. The chart can be seen in Figure 2.24.



**Figure 2.24** Hull materials chart [19]

Besides these common materials, there are other material alternatives as well. Titanium, steel, High-density polyethylene (HDPE) and graphite-epoxy are some of them. Each of these alternatives has distinctive properties. Titanium and graphite-epoxy offer high strength at relatively low densities; therefore, they are convenient

for operations at high ocean depths. However, their costs are relatively high and this inhibits them from being common at simpler vehicles.

Amongst the common materials (aluminum, fiberglass composites and carbon fiber composites), there is a higher preference of using aluminum over composites as the hull material. Nevertheless, before making conclusions about the material selection, it is beneficial to give some brief information about composite materials and compare them with traditional materials.

### **2.6.1 Composite Materials**

Composites are artificial materials that are formed by bringing two or more materials with different properties together. Materials form a rigid bond; however, they are separate at macroscopic level. The resulting material is usually anisotropic with different strength values at different directions. The strength is comparable to traditional metals alloys with significantly less weight. In other words, they have higher strength-to-weight ratios. This distinctive property makes composite materials attractive for weight critical applications like aerospace designs.

Composite materials are composed of two constituent materials. These are the reinforcing fibers and homogeneous matrix component. Fibers are strong and stiff elongated materials whose purpose is carrying the load on the material. On the other hand, they are not usable separately; therefore, they are impregnated in a matrix material. The matrix material, sometimes referred to as the resin, holds fibers together in a pre-determined orientation. They transmit and distribute the load to the fibers and also protect those fibers from environmental effects and damages. There are various types of fibers and matrix materials. The selection of the material determines the properties of the resulting composite.

#### **2.6.1.1 Fibers**

In composites technology, numerous materials have been used as reinforcing fibers. The most common fiber materials are carbon, graphite, glass, aramid, boron, aluminum, silicon carbide, quartz and polyethylene [21]. Carbon and graphite fibers provide great strength and stiffness to weight ratios. However, they are relatively expensive which limits their usage to cases where weight, not cost, is the priority. Moreover, carbon fibers are electrically conductive and may cause problems in that aspect. Aramid is also an advanced fiber material. Aramid fibers have high tensile strength and stiffness accompanied by high toughness. Especially, this high toughness and vibration damping characteristics make them valuable for armors and protection tools. Moreover, they have high thermal and chemical stabilities. Their high cost, however, make them inefficient for simpler applications. Glass fibers have been the most common ones being used in composites technology. Glass fiber composites have good strength properties with corrosion and moisture resistance. They are not electrically conductive which may be an advantage in some cases. They are not as advanced as carbon or aramid fibers in terms of tensile modulus and they are heavier; however, their lower cost makes them still attractive especially for simple applications.

#### **2.6.1.2 Resins**

Various resins with different chemical properties are used as matrix materials in composites. The most well-known examples are epoxy and polyester are the most common resin materials. Epoxies are used commonly due to their high strength, modulus, adhesion capability, corrosion resistance and ease of processing. They are, however, brittle. Polyester resins are inexpensive resins which are used for low-cost application. Moreover, they possess higher toughness compared to epoxies.

#### **2.6.1.3 Production Methods**

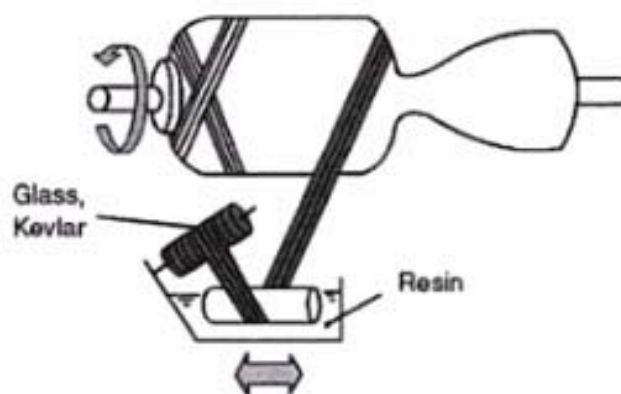
Method of combining the fiber and the matrix is a process which determines the quality of the final product. There are many ways of production and some of the



common ones are hand lay-up, spray-up, filament winding and resin transfer molding.

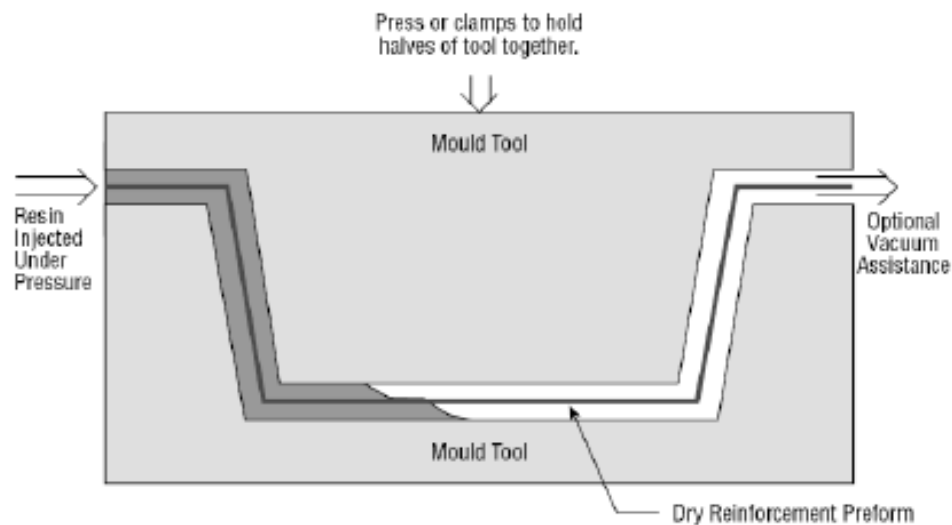
In hand lay-up method, first of all the mold is prepared like every other method. Fibers are laid on the mold manually and the resin is applied with a brush. As new layers are laid, rollers can be used to compress these layers and release the entrapped air. Hand lay-up is a manual process; therefore, it requires expertise and it cannot be automated. It is not suitable for mass production. In spray-up process resin is applied with a nozzle so it can be automated.

Filament winding is the production technique where continuous fiber which is wetted by resin is wound around a male mandrel. The winding head moves over the mandrel and the speed of this head determines the winding angle and the fiber orientation. Mandrel can be modified to leave grooves or holes on the body. Filament winding is an expansive but efficient method particularly for axisymmetric bodies like rockets. On the other hand, the product has continuous filaments. Due to that, operations such as drilling and cutting are not advised because continuous filaments are cut and the strength is reduced. Therefore, it is best to predict necessary modifications before the production. An illustration about filament winding is seen in Figure 2.25.



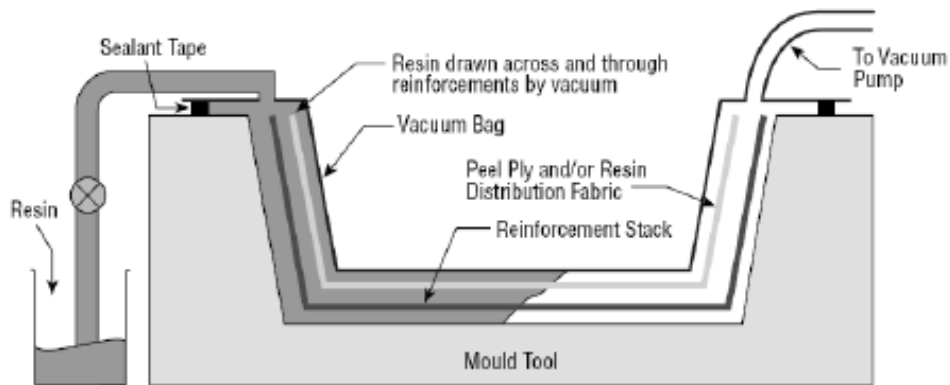
**Figure 2.25** Filament winding illustration [22]

Resin transfer molding (RTM) is a production method where fibers are laid between two molds and resin is applied to compressed fibers with pressure. Since the pressure can be controlled, the desired resin ratios can be obtained with low void content [23]. Since there are two molds, the surface quality is good. However, tooling is heavy and expensive. Illustration of this method is seen in Figure 2.26.



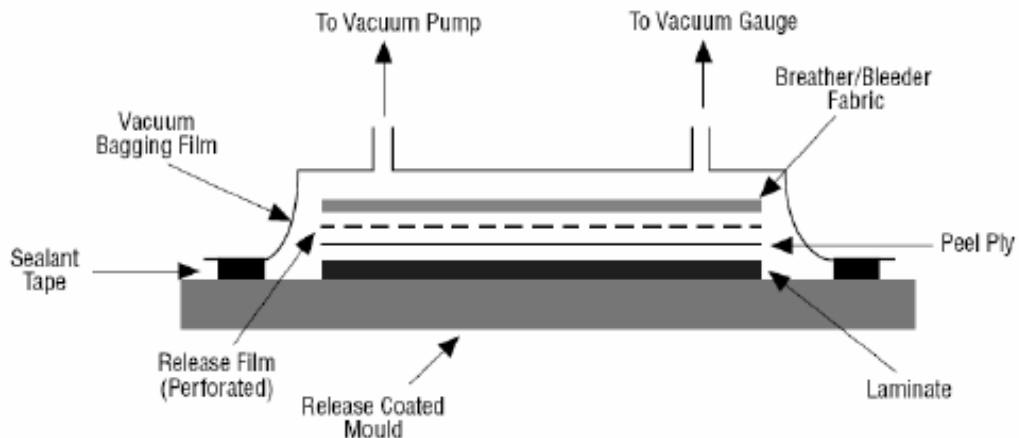
**Figure 2.26** Schematic view of Resin Transfer Molding process [23]

RTM method has several varieties. One variety is the vacuum assisted RTM (VARTM). Different than the traditional RTM process, one mold is used and the fibers are laid on this mold. Later on a plastic bag is covered on the mold and vacuum is applied. This vacuum compresses the fibers and decreases void content. Later on resin flows into these fibers due to the vacuum. Its advantage is that only one mold is used; therefore, it is easier to perform compared to RTM. However, it is still a complicated process and needs detailed study. Schematics can be seen in Figure 2.27 [23].



**Figure 2.27** Schematics of Vacuum Assisted RTM [23]

Vacuum bagging is a method which can be assumed to be a hybrid of the conventional hand lay-up process and the RTM. In this method, a mold is prepared first. Fiber reinforcement which is usually in the form of fabric is laid on the mold manually and resin is applied. The number of layers to be laid is important because the thickness of the final product is determined by that. After the laying-up process, a plastic sealing bag covers the laminate and vacuum is applied. This vacuum causes the plastic bag compress the laminate. This compression reduces the void content of the composite and enhances the wetting of the fibers. Moreover, excess resin is forced out into the bleeder which increases the resin ratio. Vacuum bagging produces better results compared to hand lay-up. Moreover, it is simpler and cheaper than the RTM method. On the other hand, highly-skilled labor is still required and the surface quality at the plastic bag side is not satisfactory. A schematic representation of this method is seen in Figure 2.28 [23].



**Figure 2.28** Schematic representation of Vacuum Bagging [23]

It should be noted that these processes are for the lay-up and application of the resin. After these, temperature should be applied to the laminate for the resin to cure.

#### **2.6.1.4 Composite Materials and Metal Alloys**

Composite materials have distinctive features that make them valuable for certain cases. Their most eminent advantage is the high mechanical properties at low weight. Due to this property, they are significant for aerospace applications where the weight is an important issue. Utilizing composite materials reduces the total weight of the system considerably. Moreover, they can be designed to be anisotropic. For situations where the loading is always in a certain direction, using composites can be very helpful because it can be designed such that it will possess higher strength in the critical direction. Composite materials, particularly aramid resin types, possess high values of damping and vibration reduction compared to metal alloys; hence, they are used for armors. Composite materials are produced by lying fiber reinforcement on a mold. As a result, it gives the designer the flexibility in design options and they are relatively easy to fabricate. Composites are excellent in terms corrosion resistance which means a lot for an underwater vehicle which will be subject to the sea water. On the other hand, composite materials have various disadvantages which inhibit them from being more commonly used. The most important issue is the cost. For

operations where financial issues are more critical than weight criteria, using composites materials is not feasible. It is hard to rely on universal mechanical properties data because properties of composites may change for different specimens. Their long-term durability is also questioned. Composites have temperature limitations; hence, they may not be applicable to high-temperature problems.

A comparison of hull materials is seen in [24] and given in Figure 2.29. It is seen that composites provide excellent strength with low weight. Moreover, their low magnetic susceptibility makes them attractive for vehicles that aim to avoid detection.

	Steel alloy	Aluminum alloy	Titanium alloy	C/peek composite	Ceramic
Ultimate stress (Kpsi)	60	73	125	300	100
Density (lb/in <sup>3</sup> )	0.283	0.1	0.16	0.056	0.13
Fabrication	Excellent	Very good	Good	Fair	Fair
Corrosion resistance	Poor	Fair	Very good	Excellent	Excellent
Magnetic susceptibility	Very high	Medium	High	Very low	Very low
Relative cost	Very low	Very low	Moderate	Moderate	Moderate

**Figure 2.29** Hull materials comparison [24]

In this project, the vehicle was produced from composite materials due to many reasons. First of all their ease of fabrication provides more freedom to the designer. Underwater vehicles have to be neutrally buoyant; therefore, weight is an important variable in the design. Composite materials provide the best strength-to-weight ratios. Corrosion resistance is advantageous for a vehicle which will operate underwater. Finally, the technical support given by the manufacturing firm, Barış Elektrik Endüstrisi Inc. has been important for the material selection. Glass-epoxy composite is utilized. Carbon was found unnecessary because diving depths are not so high (100 m). Moreover, carbons electrical conductance may cause problems with the electronic equipments. For the production method vacuum bagging is chosen. It

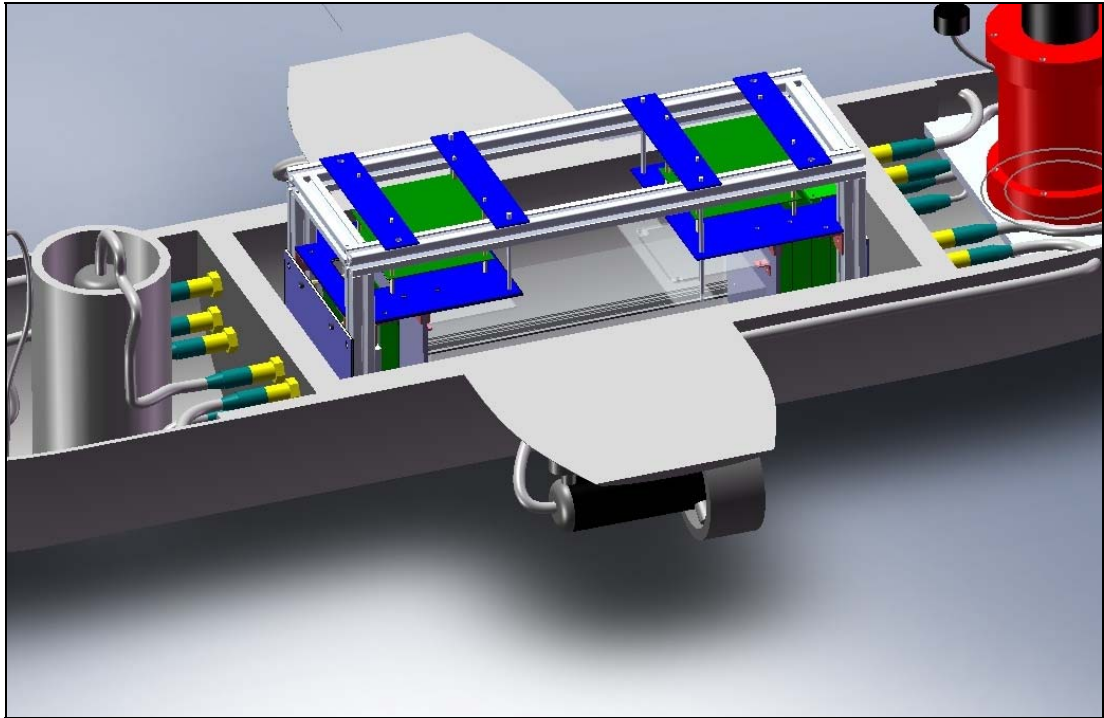
is a relatively simple and low cost method; nonetheless, it provides high resin ratio with low void content.

## **CHAPTER 3**

### **HEAT TRANSFER ANALYSES**

As it was mentioned before, the hull has a watertight section which houses electronic card and batteries. These devices dissipate heat during their performance. For ULISAR this heat dissipation rate is 55 W (50 W from cards and 5 W from batteries). Even though the vehicle will operate underwater, over-heating may still occur inside the pressure hull. The reason is that glass-epoxy composites have low thermal conductance. When it is recalled that the pressure hull has thick walls, it can be deduced that the resistance to heat rejection from the body will increase. As the heat rejection rate decreases, the temperature in the pressure hull increases and it may affect the operation electronic cards negatively. The maximum allowable

temperature in the pressure hull is 55 °C for ULISAR. The pressure hull can be seen in Figure 3.1.



**Figure 3.1** Pressure hull

### **3.1 The Software**

For heat transfer analyses FLUENT 6.3.26 and Gambit 2.3.16 software are used. Gambit is used in the pre-processing step. Gambit is a program that can import geometry data from computer aided design (CAD) softwares. Edge meshes, surface meshes, boundary layer meshes and volume meshes can be generated on the flow domain. Both unstructured and structured meshes are available in Gambit.

FLUENT 6.3.26 is used for solution and post-processing. It can import structured and unstructured meshes from different programs including Gambit. It has several physical models to model flow, turbulence, heat transfer and reactions for a wide range of applications. Both steady and unsteady analyses can be performed.

Boundary conditions for the problem are defined in Fluent and results are also seen after the solution. Contours of various flow parameters can be plotted. In this case, temperature distribution in the pressure hull is the most critical result.

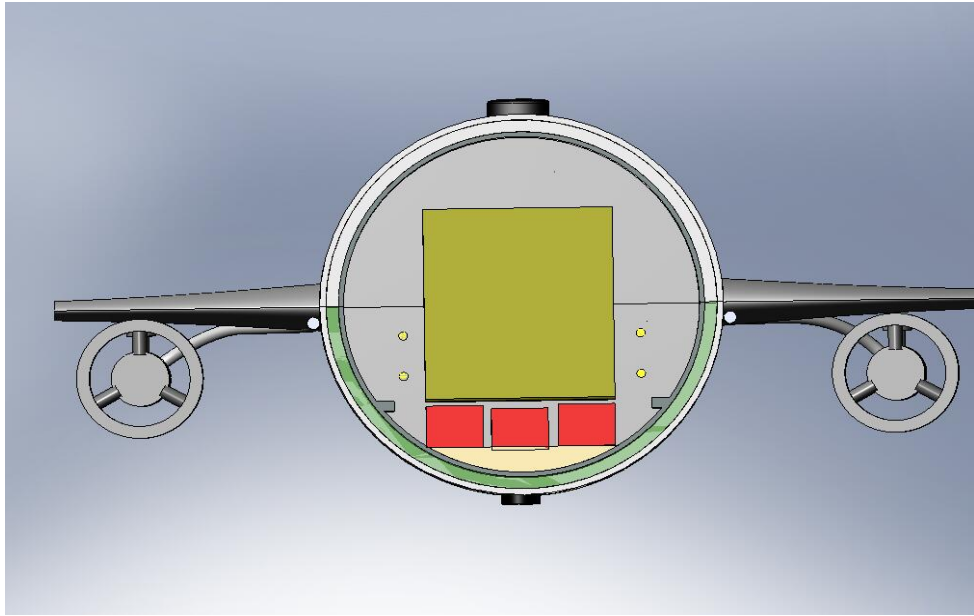
## **3.2 Analyses**

Unsteady solver of FLUENT is used in this case. Steady solver gives results for the steady-state solution and these results can be misleading. The maximum temperature obtained may be too high; however, reaching that temperature may take more than the operation time of the vehicle. Consequently, analyses were performed for 2 hours which is the expected operation time of the vehicle. 3D unsteady heat transfer analyses are very difficult to perform with the computational resources available. Therefore, 2D analyses were done. Two cases were solved.

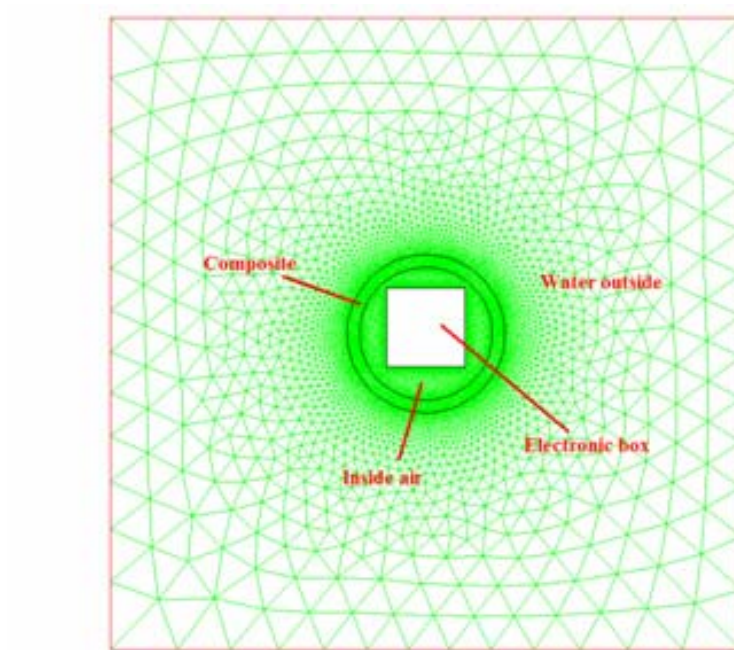
### **3.2.1 Analysis for the First Case**

For the first case, the front view cross-section of the pressure hull is modeled as seen in Figures 3.2, 3.3 and 3.4.

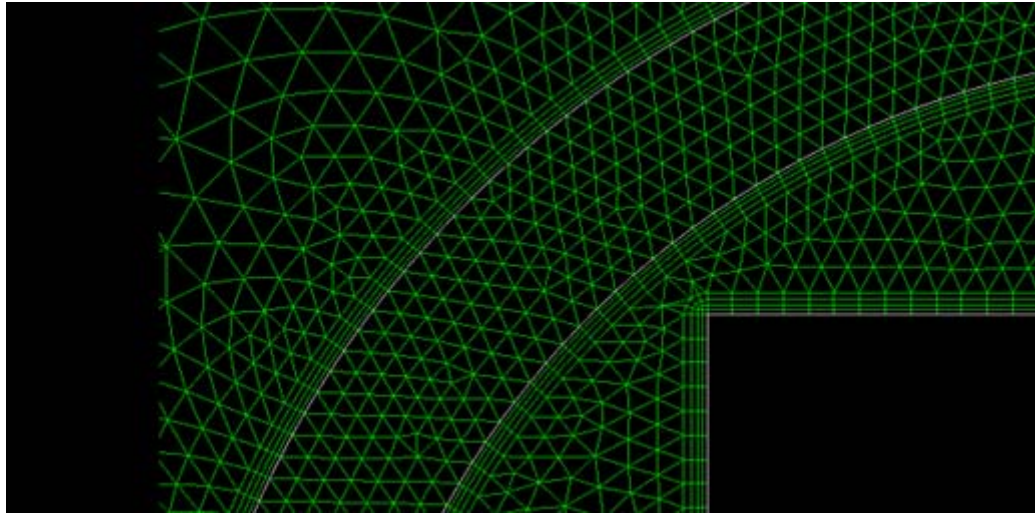




**Figure 3.2** Front view cross-section



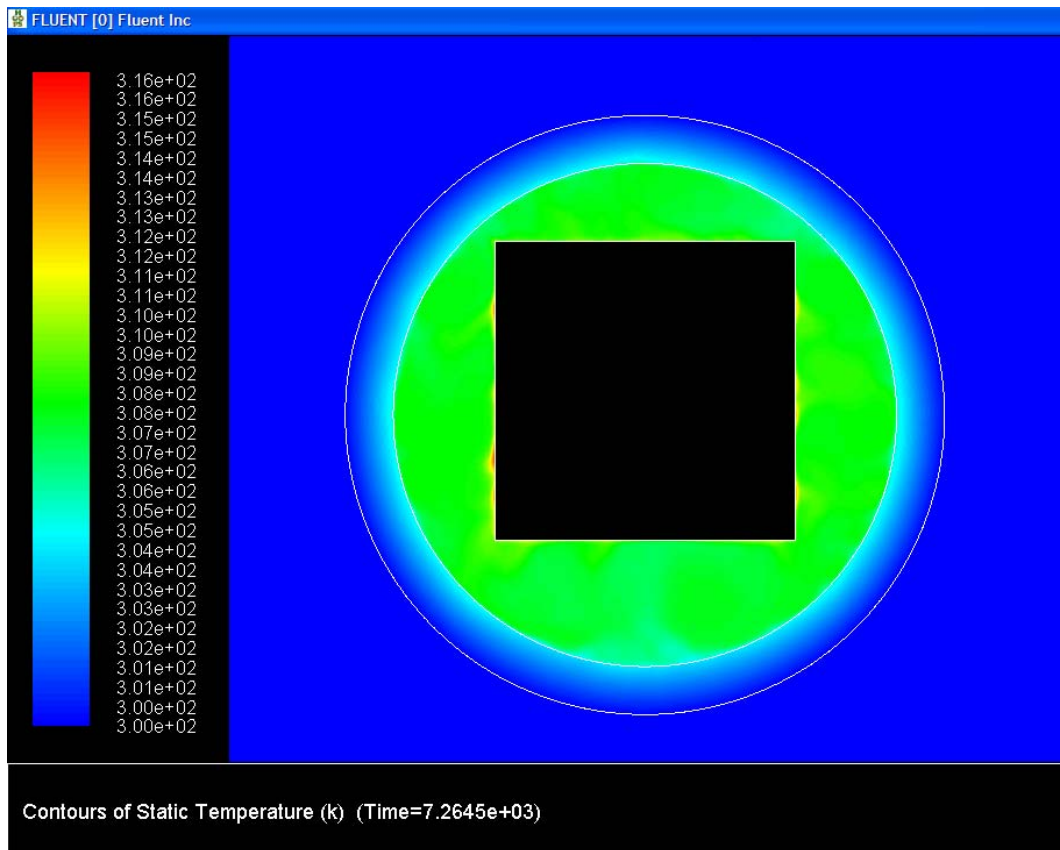
**Figure 3.3** Computational model - 1



**Figure 3.4** Details of the computational model - 1

2D, pressure based, unsteady solver was employed. Green-Gauss node based gradient option was enabled and laminar flow was solved. Air is defined to be compressible with the ideal gas formulation to be able to solve for natural convection inside the pressure hull. Water was defined to be incompressible. In order to be on the safe side, the vehicle was assumed to be stationary. It is obvious that the heat transfer rate will increase if the vehicle moves due to forced convection. Only the electronic box is modeled and a heat dissipation of 55 W was defined at the walls of the box. Initial temperature of the whole system is given 27 °C.

The temperature distribution in the pressure hull by the end of 2 hours (7200 s) can be seen in Figure 3.5. It is seen that the maximum temperature occurs on the box and has a value of 43 °C which is lower than the allowable temperature (55 °C).



**Figure 3.5** Temperature distribution inside the pressure hull

### 3.2.2 Analysis for the Second Case

An alternative analysis was performed later on. This time the cross section of the right side view is taken as seen in Figure 3.6. This time batteries are also added to the model. Moreover, the water that is entrapped in the flooded section is present; hence, heat rejection is expected to be predicted more accurately.

Solution parameters are the same as the previous case. This time the initial temperature of the water is 20 °C and the body is 30 °C. The temperature at an arbitrary time is seen in Figure 3.7. It is seen that the most critical points are on the box as expected.

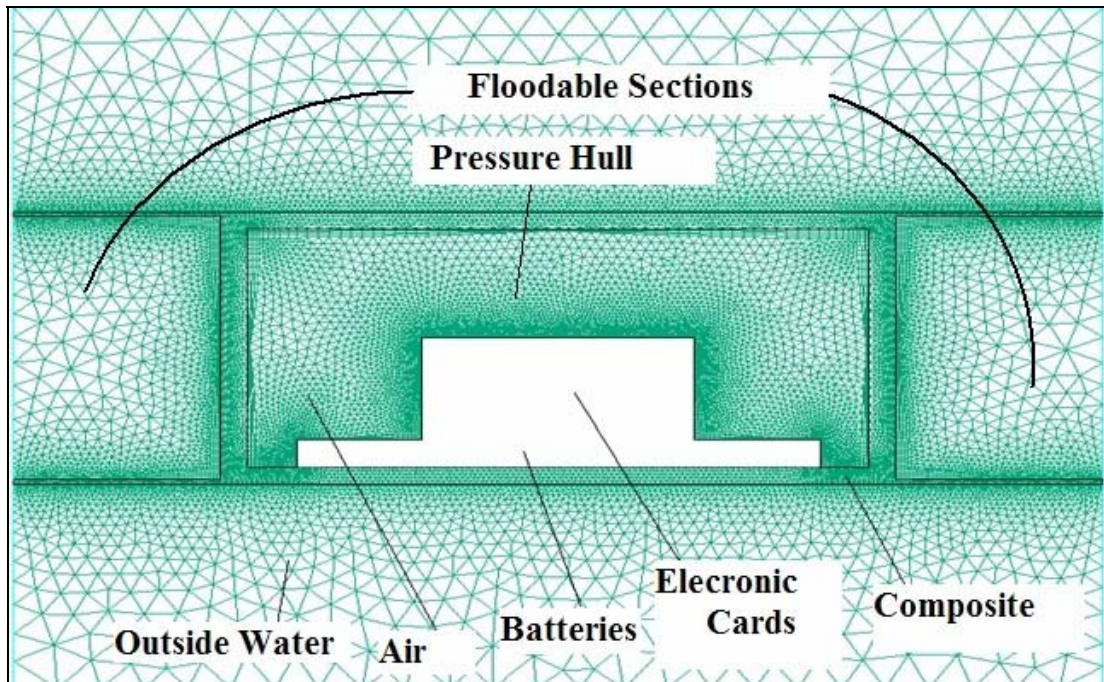


Figure 3.6 Details of the computational model – 2

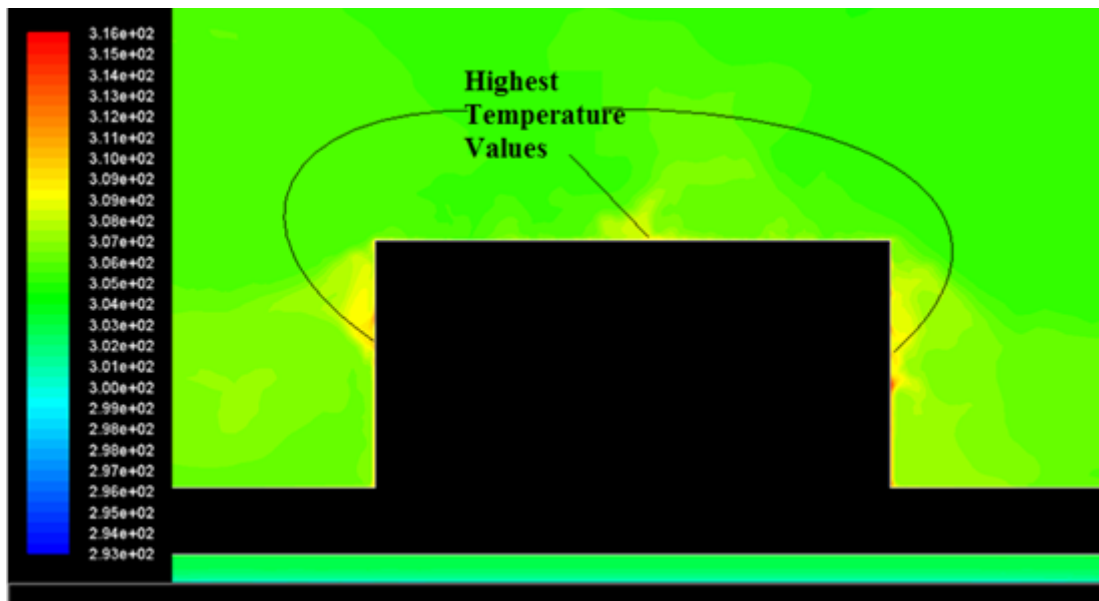
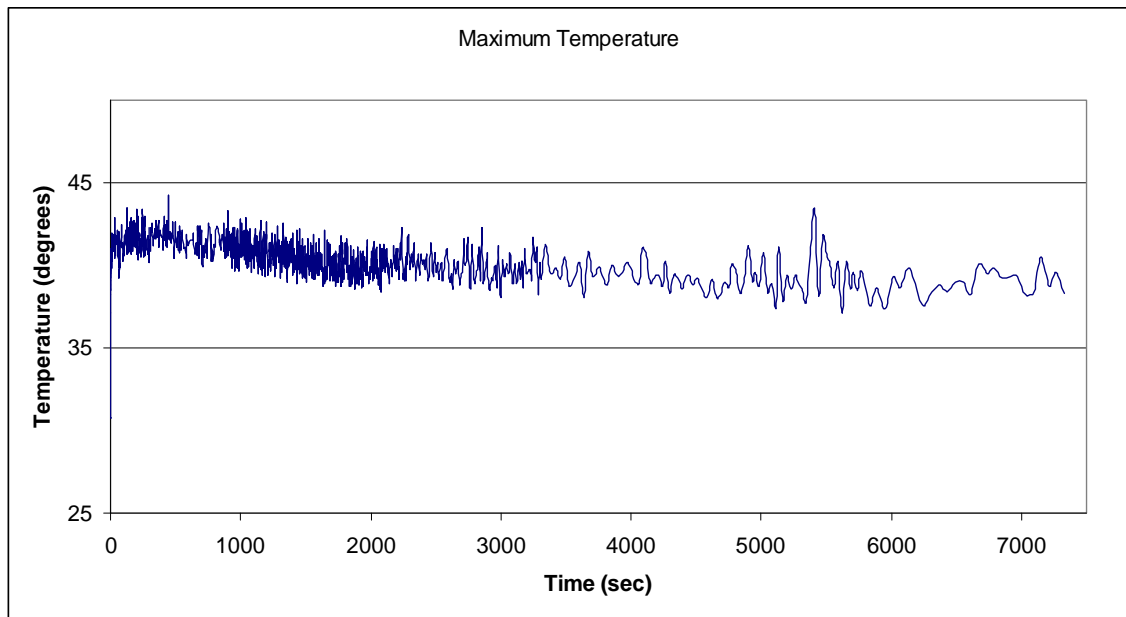


Figure 3.7 Critical temperature locations

The change of the maximum temperature in the pressure hull vs time is given in Figure 3.8. The graph has some oscillations; however, it is seen from the trend that the maximum temperature drops and converges to a steady value of about 37 °C.

This value is again below the critical limit. It should be noted that the maximum temperature was higher (43 °C) in the previous case. This can be attributed to the cooling effect of floodable section. A free-flooding hull is advantageous in terms of heat rejection.



**Figure 3.8** Maximum temperature vs time

These heat transfer analyses should be considered as preliminary analyses. Their results are not reliable because the analyses are two dimensional. Moreover, the electronic cards are modeled as a single box to simplify the model. The sandwich structure and resulting contact resistances are not included. Consequently, these analyses only give an idea on the temperature distribution.

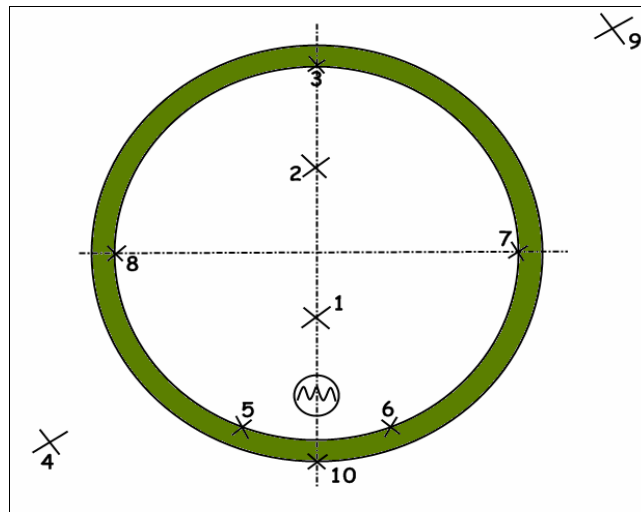
### **3.2.3 Heat Transfer Test**

To have a better understanding of the temperature distribution, a heating test is performed by Barış Elektrik Endüstrisi A.Ş. A composite hull which resembles the pressure hull of the vehicle is used. A 50 W lamp is placed and the body was immersed in water for 2 hours. The initial temperature of the body is 25 °C and water

is at 18 °C. Temperature sensors are placed at different locations inside the hull and temperature is recorded. Figure 3.9 shows the experimental setup and Figure 3.10 illustrates the positions of the sensors.

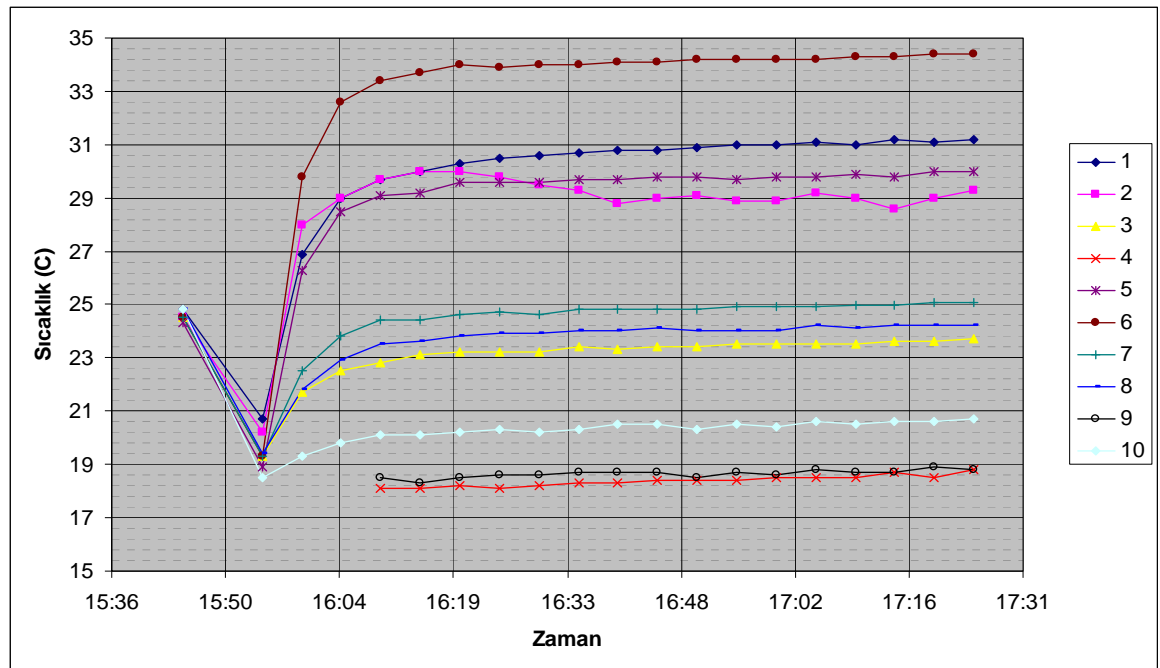


**Figure 3.9** Experimental setup



**Figure 3.10** Temperature sensor positions

The temperature vs time graph is seen in Figure 3.11. Point 6 is the most critical one and its temperature converges to an approximate value of 35 °C.



**Figure 3.11** Temperature vs time inside the hull

Experimental results converge to a value of 35 °C and second analysis converges to 37 °C. When considering the fact that initial temperature of the hull was 30 °C for the

analysis and 25 °C for the experiment, it is possible to deduce that experimental and simulation results match well. Therefore, we can conclude that the results of the second simulation are satisfactory even though they are very simple and preliminary calculations. Moreover, it is not expected to have over-heating at this point. Certainly, the actual results will be received when the vehicle is tested in real conditions.

## **CHAPTER 4**

### **VALIDATION STUDIES**

Determining the hull shape and its dimensions along with the thruster configuration is an important issue in vehicle design. However, before accepting the design proposal certain analyses are necessary. In our case the most critical requirement is about surging (forward) speed. Therefore; drag calculations should be performed to make sure that the offered design satisfies the speed criterion (2 m/s). For these analyses, Computational Fluid Dynamics (CFD) method is used and the commercial CFD solver software FLUENT 6.3.26 is employed. Before presenting the results of the analyses, it is helpful to give brief information on CFD and its applications in marine science.



## 4.1 Introduction to CFD

Computational fluid dynamics is a method used for determining fluid and flow properties like pressure, density, temperature and velocity vectors for different flow cases. These properties can be used for calculating important data like drag force, heat transfer rate, pressure distribution, acoustics, etc. In other words CFD enables the designer to predict the performance of a virtual system during the design. It is obvious that this is advantageous for the designer since the virtual system can be tested and modified if necessary until finding the optimum one. As a result, the chances of obtaining a successful design at the end increases, number of experiments required to find the optimum design decreases which in turn reduces the cost and duration of the design process.

In marine applications like an unmanned underwater vehicle design project, CFD can be used for many purposes. The most common usage is determining the drag force on the body in a specified scenario. This drag data is critical in choosing the propulsion system for achieving the speed requirement. For systems where speed criterion is strict or energy source is limited, the importance of performance prediction increases. As solver capabilities and computational resources have improved in recent years, CFD found new areas of application in marine science. One example is the propeller-hull interaction. The performance of a propeller strongly depends on the inlet flow quality of the propeller. Therefore, it is beneficial to make CFD analyses to predict the propeller performance. Maneuverability of the vehicle is another topic to investigate using CFD. It should be noted that unlike aerospace applications, experimental data for marine applications is scarce. Furthermore, it is harder to conduct experiments since experimental facilities like water tunnels are not common. These intensify the need for CFD simulations even more.

Speaking in general terms, a CFD simulation is the simultaneous solution of a set of governing equations of fluid flow in a discretized flow domain to obtain the

distribution of various flow parameters. A CFD simulation process consists of three steps: namely, pre-processing, solution and post-processing. In pre-processing the geometry and the flow domain around it are generated. Later on the domain is divided into numerous elements and the computational mesh is formed. Boundary conditions and material properties are defined at this step. In the solution step governing equations of fluid flow are solved for each cell in the mesh to obtain the distribution of fluid and flow parameters. Post-processing is the last step where relevant flow data can be seen and results can be assessed.

In the following parts of the chapter, detailed information about the governing equations and turbulence models which are used in the software FLUENT are presented.

#### 4.1.1 Governing Equations of Fluid Flow

Governing equations of fluid flow originate from conservation laws of physics; namely, conservation of mass, conservation of momentum in three dimensions and conservation of energy. Mathematically, one can derive governing equations for these laws. For a three dimensional, unsteady, viscous, compressible fluid flow following equations are given as [25]:

Continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0 \quad (4.1)$$

Momentum equation in the  $x$ -direction:

$$\frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u \mathbf{U}) = -\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + \rho f_x \quad (4.2)$$

Momentum equation in the  $y$ -direction:

$$\frac{\partial(\rho v)}{\partial t} + \nabla \cdot (\rho v \mathbf{U}) = -\frac{\partial p}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + \rho f_y \quad (4.3)$$

Momentum equation in the z-direction:

$$\frac{\partial(\rho w)}{\partial t} + \nabla \cdot (\rho w \mathbf{U}) = -\frac{\partial p}{\partial z} + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} + \rho f_z \quad (4.4)$$

It should be noted that  $\mathbf{U}$  refers to the velocity vector which is composed of velocities in x, y and z directions.

In marine CFD applications, energy equation is usually neglected since system is isothermal and flow is incompressible. For Newtonian flows where density and viscosity are constant and body forces are not present, equations become:

Continuity equation:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (4.5)$$

Momentum equation in the x-direction:

$$\rho \left( \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) = -\frac{\partial p}{\partial x} + \mu \nabla^2 u \quad (4.6)$$

Momentum equation in the y-direction:

$$\rho \left( \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) = -\frac{\partial p}{\partial y} + \mu \nabla^2 v \quad (4.7)$$

Momentum equation in the z-direction:

$$\rho \left( \frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) = - \frac{\partial p}{\partial z} + \mu \nabla^2 w \quad (4.8)$$

Equations (4.5) to (4.8) include four flow variables,  $p$ ,  $u$ ,  $v$  and  $w$  and there are four equations. Therefore, one can create a mesh and solve these equations simultaneously for all cells and find the value of  $p$ ,  $u$ ,  $v$  and  $w$  for each cell. This is true for the laminar, incompressible, three dimensional flow. However, in reality these variables are subject to instabilities and laminar flow usually leads to turbulent flow where flow turbulence modeling is necessary for solution.

#### 4.1.2 Introduction to Turbulence Modeling

Turbulence is one of the most important and complex topics in CFD. In this part of the thesis only brief information about it will be given. Basically, turbulent flow is a three dimensional, highly irregular and time dependent mode of flow.

Turbulent flows have a swirling character with local vortices. These vortices are called turbulent eddies and their time and length scales vary significantly. These eddies are responsible for the enhanced mixing of transportation quantities in the flow domain. It is no longer possible to define steady quantities for pressure and velocities. Instead, these quantities have unsteady characters.

It should be noted that these energy carrying eddies have varying length and time scales. For turbulent flow simulations, one should consider the effect of these eddies, regardless of their scale, to calculate flow variables accurately. The effort to calculate the effect of turbulence on flow variables is called turbulence modeling. The complexity of the turbulence model depends on the scale of eddies that are taken into account while modeling. Turbulence models can be divided into three categories

depending on their complexity: Direct Numerical Simulation (DNS), Large Eddy Simulation (LES) and Reynolds Averaged Navier-Stokes (RANS) method.

Direct Numerical Simulation (DNS) is the most complex simulation approach. In this method all of the scales of turbulence in the flow are solved no matter how small those scales are. In fact no modeling is made; instead, all the scales of turbulence are solved directly. DNS enables the researcher to get data of the fluid even at very small scales. On the other hand, solving the Navier-Stokes equations at such small scales requires extremely fine meshing and the total number of cells in the mesh increases to huge numbers. It is even worse for high Reynolds number flows where the length scale of the smallest eddies are even smaller. Even for low-Reynolds number flows it is not possible to solve complex problems with the computational resources available today.

Even though DNS makes extremely accurate predictions, most of the practical cases in CFD do not require such fine detail at small scales. Therefore; engineers tend to use simpler models which can still provide accurate results with reasonable computational time. One such approach is the Large Eddy Simulation (LES). As its name implies, LES considers only large scales of eddies. Large eddies are solved directly while small eddies are filtered and their effects are modeled. This requires the solution of Navier-Stokes equations with an additional stress term for small scales. Filtering small scales of turbulence makes LES require less computational resources compared to DNS and still give accurate results. The instantaneous flow characteristics are still captured. On the other hand, computational requirements are still high for LES which makes it unsuitable for most of the practical applications.

As it was mentioned, LES and DNS offer excellent solutions for flow simulation; however, the computational resources of today make them unsuitable for majority of practical cases. One approach to overcome this difficulty is Reynolds Averaging. In this approach, it is assumed that flow variables have mean and fluctuating components. When mean and fluctuating variables are used in the three dimensional

unsteady Navier-Stokes equations, a new set of equations called Reynolds-Averaged Navier-Stokes (RANS) equations are obtained. RANS equations solve for only mean values of the flow variables. Due to fluctuating quantities, additional terms appear in RANS equations. These terms represent the unsteady effect of turbulence and they act as an additional viscosity term in the equation. Therefore these terms are named as “turbulent stresses” or “Reynolds stresses”.

Modifying Navier-Stokes equations for obtaining RANS equations has two effects. First of all there is a great loss of information. Only mean variables are solved and the effect of eddies are modeled. This enables a steady solution of equations with less number of elements. Hence, computational requirements and solution time are much lower compared to LES and DNS. On the other hand, RANS methods provide only time-averaged results while LES and DNS give instantaneous flow variables. The second effect is that additional terms are added to Navier-Stokes equations. As a result, the set of equations are not closed anymore and additional equations are necessary for solution. Turbulence modeling is, in fact, performed for defining Reynolds stress components in terms of known variables of flow and additional equations are obtained.

RANS models give acceptable results with reasonable computational time and resources; therefore, they are widely used for practical applications. In this thesis, emphasis will be on RANS models and these models will be used for underwater vehicle simulations.

### **4.1.3 Reynolds Averaging**

Reynolds averaging is decomposing flow variables into its mean and fluctuating components. For instance  $x$ -velocity can be decomposed as

$$u = \bar{u}(x) + u'(x,t) \quad (4.9)$$

where  $\mathbf{u}(\mathbf{x})$  is the mean and  $\mathbf{u}'(\mathbf{x}, \mathbf{t})$  is the fluctuating component of velocity. Similar forms of equations can be written for the velocity components in  $y$  and  $z$  directions. When these pressure and velocity definitions are used in Equations (4.6) to (4.8), momentum equations become:

Momentum equation in the  $x$ -direction:

$$\begin{aligned} \rho \left[ \frac{\partial(\overline{u^2})}{\partial x} + \frac{\partial(\overline{uv})}{\partial y} + \frac{\partial(\overline{uw})}{\partial z} \right] = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial x} \left[ \mu \frac{\partial u}{\partial x} - \rho \overline{u'^2} \right] \\ + \frac{\partial}{\partial y} \left[ \mu \frac{\partial u}{\partial y} - \rho \overline{u'v'} \right] + \frac{\partial}{\partial z} \left[ \mu \frac{\partial u}{\partial z} - \rho \overline{u'w'} \right] \end{aligned} \quad (4.10)$$

Similar equations can be written for  $y$  and  $z$  momentum equations. It is seen that additional variables like  $\overline{\rho u'v'}$ ,  $\overline{\rho u'w'}$  are created. These are known as Reynolds stresses and they should be modeled.

For modeling these Reynolds stresses, two common approaches are held. The first one is the Boussinesq approach and the second one is directly calculating these stresses through transport equations.

#### 4.1.4 Boussinesq Approach

According to Boussinesq approach:

$$-\overline{\rho u'v'} = \mu_t \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \left( \rho k + \mu_t \frac{\partial u_k}{\partial x_k} \right) \delta_{ij} \quad (4.11)$$

where  $k$  is the turbulence kinetic energy per unit mass and  $\mu_t$  is the turbulent (eddy) viscosity. As it is seen in Equation (4.11), Reynolds stresses can be calculated for a

cell if  $\mu_t$  and  $k$  are known for those cells. Furthermore,  $\mu_t$  is related to  $k$  with a simple equation by the turbulence model being used. Hence, for computing Reynolds stresses, one has to know the turbulence kinetic energy per unit mass. On the other hand,  $k$  is calculated by defining a dissipation term for it and solving those equations simultaneously over the whole domain. The relationship between turbulent viscosity and turbulence kinetic energy along with transport equations for  $k$  and its dissipation depends on the turbulence model being used. The success of the turbulence model is determined by these relations.

Spalart-Allmaras,  $k - \varepsilon$  and  $k - \omega$  turbulence models use Boussinesq approach. Spalart-Allmaras is a one-equation model where only a single transport equation is solved for turbulent viscosity.  $k - \varepsilon$  and  $k - \omega$  turbulence models are two equation models. In  $k - \varepsilon$  model, two additional equations are solved for  $k$  and turbulence kinetic energy dissipation rate,  $\varepsilon$ . In  $k - \omega$  turbulence model, two additional equations are solved for  $k$  and turbulence kinetic energy specific dissipation rate,  $\omega$ . These models are commonly used in CFD and also by the users of FLUENT. Their advantage is that they assume that  $\mu_t$  is isotropic and this reduces computation costs significantly. On the other hand they may give inaccurate results when turbulence in the domain of interest has anisotropic character.

As an alternative to Boussinesq approach, one can solve six transport equations along with an equation for dissipation (usually  $\varepsilon$ ) for Reynolds stress components. This means that five additional equations for 2D case and 7 additional equations for 3D case will be solved. It is obvious that the computation time will increase; however, the accuracy is much better compared to two equation models especially for complex flows where anisotropy is present in Reynolds stresses. Some examples of complex flows are given in FLUENT 6.3 User's Guide [26] as cyclone flows, highly swirling flows in combustors, rotating flow passages, and the stress-induced secondary flows in ducts. Models that use this approach are called Reynolds Stress Models (RSM). Conversely, it will not be feasible to use this model for simple flows since the improvement in the accuracy will not overcome the increased computation cost.



Nevertheless, as computational resources are progressing, RSM models are gaining popularity.

#### 4.1.5 Turbulence Models

CFD analyses in this thesis are performed by the FLUENT 6.3.26 software. In FLUENT various turbulence models with different capabilities and complexities are offered. These models are Spalart-Allmaras,  $k - \epsilon$ ,  $k - \omega$ , RSM, Detached Eddy Simulation (DES) and LES.

It should be noted that none of these turbulence models are proven to be applicable for all cases. Each of them has advantages and disadvantages. The analyzer has to make a decision based on needs, computational resources and time.

Spalart-Allmaras model is a simple RANS model that adopts Boussinesq approach. This model solves only one transport equation for turbulent viscosity,  $\mu_t$ . Computation time is lower compared to two-equation models. For wall-bounded flows in aerospace applications, the popularity of the model is increasing. On the other hand, it is a relatively new model and its efficiency for complex flows is not verified yet [26].

RSM provides accurate results for complex flows; however, its feasibility is not justified for simple problems due to high computational cost. In this project, CFD simulations are performed for calculating the drag force on the streamlined hull shape. It is obvious that this problem is a simple external flow problem and the use of RSM is not critical. Similarly, LES and DES models have high computational costs which make them unsuitable for the analyses of the vehicle.

LES model is a much more complex model where instantaneous flow characteristics can be captured due to solving equations for large eddies in the flow. DES is a hybrid model between LES and RANS models.

Two-equation models solve a transport equation for  $k$  along with an additional equation for its dissipation. These models are the most popular ones for industrial applications because they provide accurate results with reasonable computational time. FLUENT offers  $k - \varepsilon$  and  $k - \omega$  models with their variations as two-equation models. In this thesis these models are chosen for review since they meet project requirements best.

#### **4.1.5.1 $k - \varepsilon$ Turbulence Models**

$k - \varepsilon$  turbulence model in FLUENT has three variations: standard, RNG and realizable  $k - \varepsilon$  models. Standard model is the earliest and simplest  $k - \varepsilon$  model. It is a semi-empirical model since the dissipation term  $\varepsilon$  is obtained through empirical relations [26]. As an early model, it has several weaknesses. It has been noted to give inaccurate predictions for flow cases involving separation, swirling, re-circulation, re-attachment and mixing [25]. Due to these deficiencies this model is improved to obtain the variations of the  $k - \varepsilon$  model.

RNG model is another variation of the  $k - \varepsilon$  model. The  $\varepsilon$  equation is modified and effect of swirl on turbulence is added. It is claimed that the model works for a wider class of flows and performs well for flows for rapidly strained flows and swirling flows.

Realizable  $k - \varepsilon$  is the most common model being used. It is a relatively recent improvement on the standard model. Turbulent viscosity and turbulent dissipation rate equations are modified over the standard model. It has been claimed that the model gives superior results for flows involving rotation, boundary layers under strong adverse pressure gradients, separation, and recirculation.

#### **4.1.5.2 $k - \omega$ Turbulence Models**

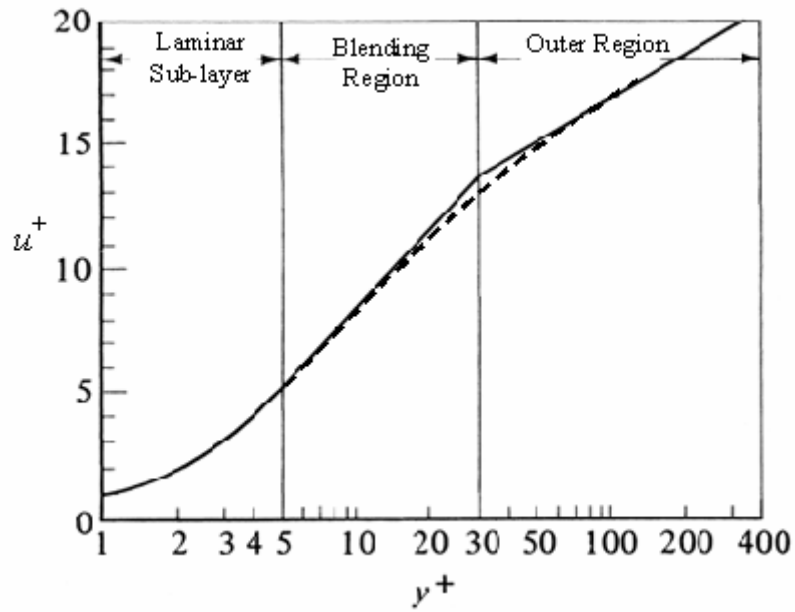
FLUENT has two  $k - \omega$  turbulence model variations. These are standard  $k - \omega$  model and the Shear-Stress Transport (SST) model. These models solve transport equations for turbulence kinetic energy,  $k$ , and its specific dissipation rate,  $\omega$ . The standard model is known to predict accurately for wall-bounded flows and free shear flows. One drawback of the model is its sensitivity to free stream effects.

To eliminate the inaccuracies due to free stream effects, a variation was developed. This model is called the SST  $k - \omega$  model. In this model the standard  $k - \varepsilon$  and standard  $k - \omega$  models are combined. For the near-wall region SST  $k - \omega$  is employed and for the free stream the less sensitive  $k - \varepsilon$  is used. This modification makes SST a suitable model for a wide range of flows.

#### **4.1.6 Near Wall Treatment**

One of the key points in turbulence modeling is the near wall treatment. In the region close to wall there are different layers with varying characteristics due to the strong effect of the wall. A turbulence model may not be valid for all of these regions; therefore, while creating the mesh and choosing the turbulence model, one has to be careful about this region. The accuracy of the simulation depends strongly on the accurate modeling near the wall.

In the region closest to the wall, viscous effects are dominant. This effect causes a region with laminar characteristics. It is called the viscous sublayer or laminar sublayer. In this region molecular viscosity is dominant over the turbulent viscosity. Due to this fact many turbulence models are not valid in this region. In the outermost region of the boundary layer there is the fully-turbulent or log-law region where viscous effects are suppressed and the flow is entirely turbulent. Turbulence models perform well in this region. Between these two regions, there is a transition region named as the buffer layer or blending region where the effects of the wall and free stream are comparable. These layers can be seen in detail in Figure 4.1.



**Figure 4.1** Near wall region

In Figure 4.1,  $u^+$  is the non-dimensional velocity and  $y^+$  is the non-dimensional distance of the cell to wall where

$$y^+ = \frac{\rho u_T y}{\mu} \quad (4.12)$$

$$u^+ = \frac{u}{u_T} \quad (4.13)$$

$$u_T = \sqrt{\frac{\tau_w}{\rho}} \quad (4.14)$$

In Equations (4.12) to (4.13)  $u_T$  is the friction velocity,  $y_p$  is the distance from the cell to the wall and  $\tau_w$  is the wall shear stress.

The most important issue in near wall treatment is deciding the level up to which the boundary layer needs to be resolved. As it was shown in Figure 4.1, it is seen that the

boundary layer consists of three regions. If it is desired to resolve the boundary layer all the way up to the laminar sublayer, first cell adjacent to the wall should be placed in the that region. Alternatively, one can place the first cell in the fully turbulent outer region if the viscous sublayer is not aimed to be captured exactly. These two alternatives lead to two approaches: near wall modeling and utilization of wall functions. In the wall function approach, viscous sublayer and buffer layer regions which require fine mesh are not modeled. Instead of this semi-empirical functions are called wall functions are used to represent these regions. In the near wall modeling approach, the boundary layer all the way up to the viscous sublayer is solved.

Wall functions are useful especially for high-Reynolds number flows. Since there is no need to generate extremely fine meshes, this approach reduces mesh size and computation time. However, for low-Reynolds number flows the mesh size may increase too much in order to locate the first cell in the fully turbulent region and this may cause bad mesh quality. Near wall treatment requires a fine mesh near the wall; hence, computation time increases; however, better accuracy is achieved. This method is beneficial for low-Reynolds number flows where required mesh size is not very small.

In FLUENT the turbulence model SST, when transitional option is activated, is a low-Reynolds number model; in other words, it solves all the way up to the viscous sublayer. Therefore; one has to create a very fine mesh near the wall when using SST. On the other hand,  $k - \varepsilon$  is a high-Reynolds number model so one has to begin with a coarse mesh so that the first element in the mesh falls into the fully-turbulent region.  $k - \varepsilon$  model use two types of wall functions; namely, Standard Wall Functions (SWF) and Non-equilibrium Wall Functions (NWF). Moreover,  $k - \varepsilon$  has an Enhanced Wall Treatment (EWT) option which can be activated if it is desired to solve the complete boundary layer with this model.

At this point it is necessary to define criteria to check whether the near wall mesh is fine enough or not. Wall  $y^*$ ,  $y^+$  and  $Re_y$  can be used for this purpose.  $y^+$  and  $y^*$  represent dimensionless distance from the wall to the cell and can be given as

$$y^* = \frac{\rho C_\mu^{1/4} k_p^{3/4} y_p}{\mu} \quad (4.15)$$

$$Re_y = \frac{\rho y_p u_T}{\mu_{lam}} \quad (4.16)$$

where  $C_\mu$  (=0.09) is the modeling constant,  $k_p$  is the kinetic energy of that point at point P,  $y_p$  is the distance from the point to the wall,  $u_T$  is the friction velocity and  $\mu_{lam}$  is the laminar viscosity.

When using the wall function approach, it is important to locate the first cell off the wall in an interval of  $30 < y^* < 300$  because wall functions are valid in this region (log-law region). It is advised not to exceed values of 100 [26].

For enhanced wall treatment, fine mesh is required. A mesh with the first cell in the viscous sublayer at  $y^+$  is 1 is the ideal. However, higher  $y^+$  values are also acceptable until  $y^+ < 4$  to 5 [26].

The aim while creating the boundary layer mesh should be making the mesh either fine or coarse so that the first element height does not fall in the interval  $5 < y^+ < 30$ . Besides the first cell height, number of cells inside the boundary layer should also be checked. FLUENT User's Guide [26] suggests that there should be at least 10 cells until  $Re_y < 200$ . After the analyses it is useful to make plots of  $y^+$  and  $y^*$  on the wall and contours of  $Re_y$  and check them to see whether the mesh is suitable for the given case.

## 4.2 Validation Studies

The brief introduction on turbulence models and near wall modeling approaches gives an idea on suitable modeling approach of the vehicle. The realizable  $k - \varepsilon$  model with wall functions, realizable  $k - \varepsilon$  model with enhanced wall treatment and SST  $k - \omega$  models seem to be appropriate for the simulations of the vehicle. Usually, it is not a good idea to begin simulations with this superficial knowledge. Beforehand some validation simulations should be performed on bodies that have experimental results in the literature. These bodies should be chosen such that the geometry is as similar as possible to the intended vehicle design. Validation studies are useful in many aspects. It helps the user check his/her capabilities in flow modeling. The user can check the pros and cons of various models and validate their applicability to the problem. The effect of solution parameters on the results can be seen by changing them in a controlled manner. Due to these reasons, some validation studies were performed before beginning the actual analyses.

### 4.2.1 The Software

For validation studies and actual simulations of the vehicle FLUENT 6.3.26, Gambit 2.3.16 and Tgrid 4.0.16 software were used. Gambit and Tgrid were used in the pre-processing step. Gambit is a program that can import vertex data or geometry from computer aided design (CAD) software. Edge meshes, surface meshes, boundary layer meshes and volume meshes can be generated on the flow domain. Both unstructured and structured meshes are available in Gambit. Tgrid is also capable of these duties; however, the main advantage of Tgrid is its high performance on creating boundary layer meshes. In this thesis surface meshes are generated by Gambit and boundary layer meshes are generated by Tgrid. Later on volume meshes are generated with either Gambit or Tgrid depending on the problem.

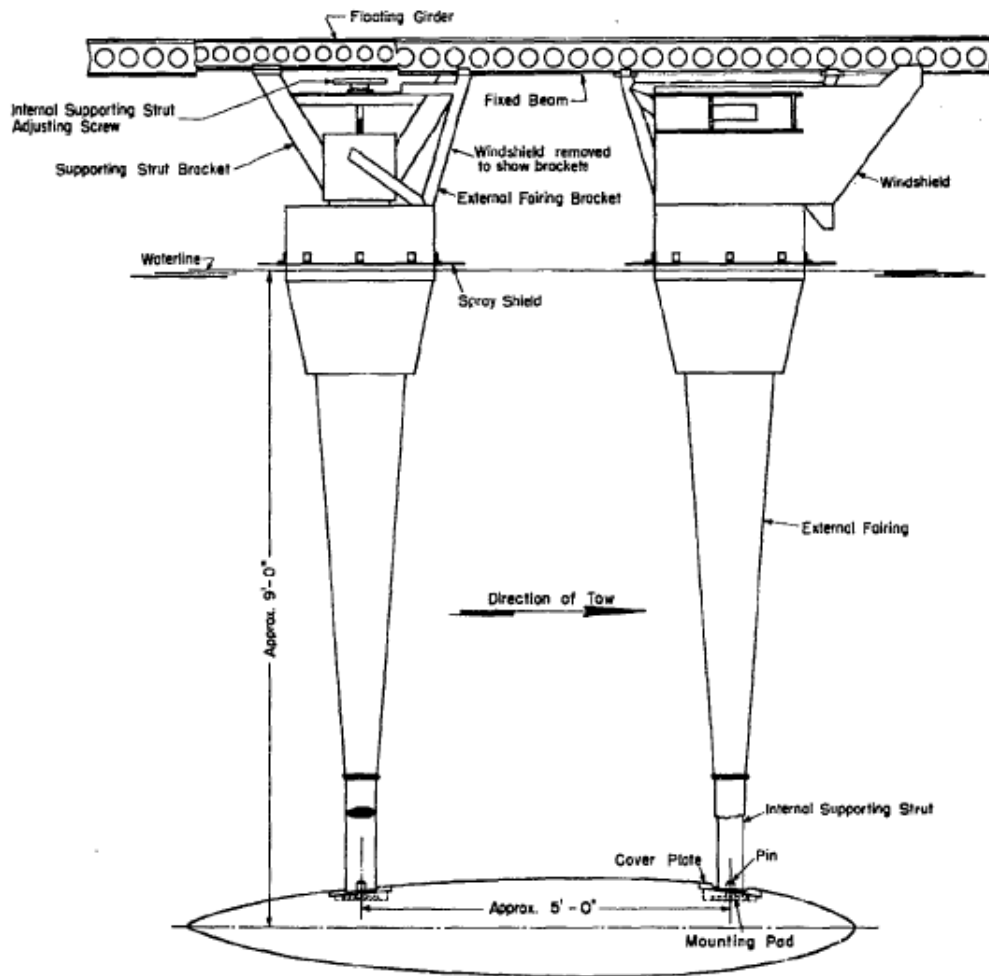
FLUENT is used for solution and post-processing. It can import structured and unstructured meshes from different programs. It has several physical models to model flow, turbulence, heat transfer and reactions for a wide range of applications. In this thesis an external flow problem is investigated by FLUENT using the above mentioned turbulence models. Results were post-processed and required data like pressure distribution, velocity vectors and drag force are obtained.

#### **4.2.2 Series 58 Studies**

The first validation studies were performed on Series 58 models. Series 58 experiments were held by Morton Gertler in 1950 [17]. The purpose of these experiments was determining the resistance of several axisymmetric underwater bodies. Geometrical properties like prismatic coefficient, nose radius, tail radius,  $L/D$  ratio and location of maximum diameter on resistance was observed by systematic experiments. The ultimate aim was presenting a guideline and database for submarine design. In fact, this is one of the earliest attempts to create a database on resistance of streamlined bodies in deep submergence case. Therefore, these results gained popularity and have been cited many times.

Investigated bodies are derived from a sixth degree polynomial. The coefficients of the polynomial were changed to vary the shape of the body and the geometric coefficients. 24 models were developed and tested. All 24 models have a length of 9 feet (2.74 m). Models were held inside the towing tank with a pair of struts. The additional drag due to these struts was later on removed from the results. The test apparatus can be seen in Figure 4.2.





**Figure 4.2** Test Apparatus for Series 58 Experiments [17]

Sand strips are placed near the leading edge of the model to stimulate the transition to turbulence. Tests were performed for both bare hull case and sand stripped case.

Experiments are performed for all of the bodies for a range of Reynolds numbers. Reynolds number usually changes from  $2 \times 10^6$  to a maximum of  $2 \times 10^7$  to  $3 \times 10^7$  based on body length. Total resistance coefficient,  $C_t$  is plotted against Reynolds number for all of the models. Gertler defines resistance coefficient as

$$C_t = \frac{R}{\frac{\rho}{2} S U^2} \quad (4.17)$$

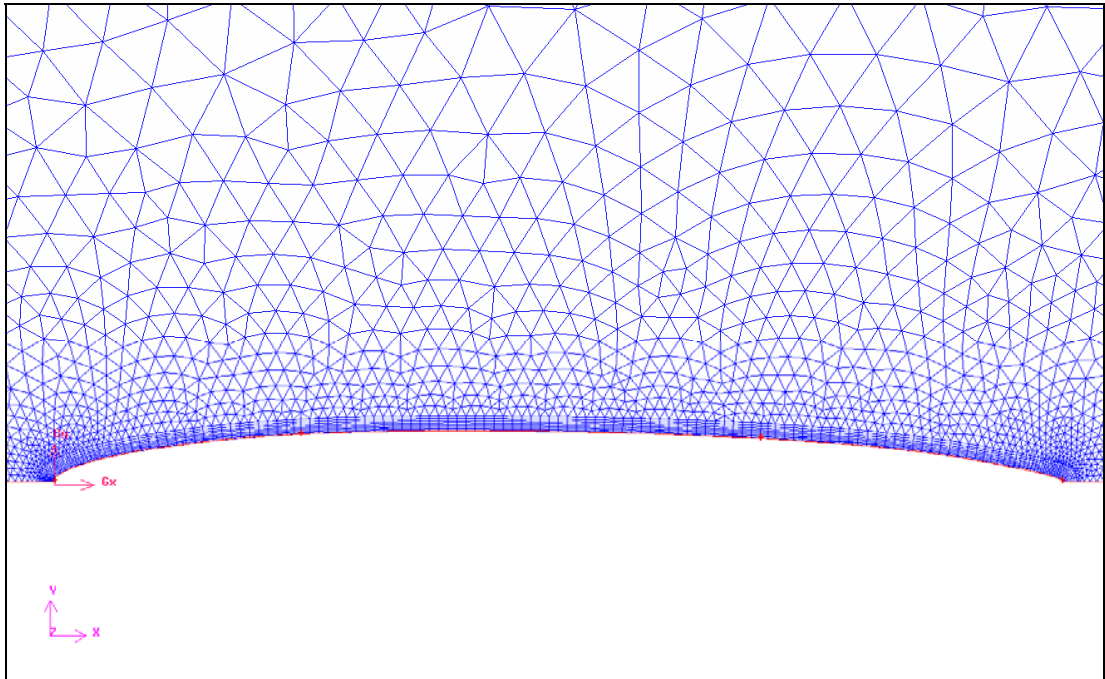
where  $R$  is the resistance (total drag) on the body and  $S$  is the wetted surface area for the corresponding body.

In this validation study, three of these models are chosen. These models are Model 4159, Model 4158 and Model 4154.

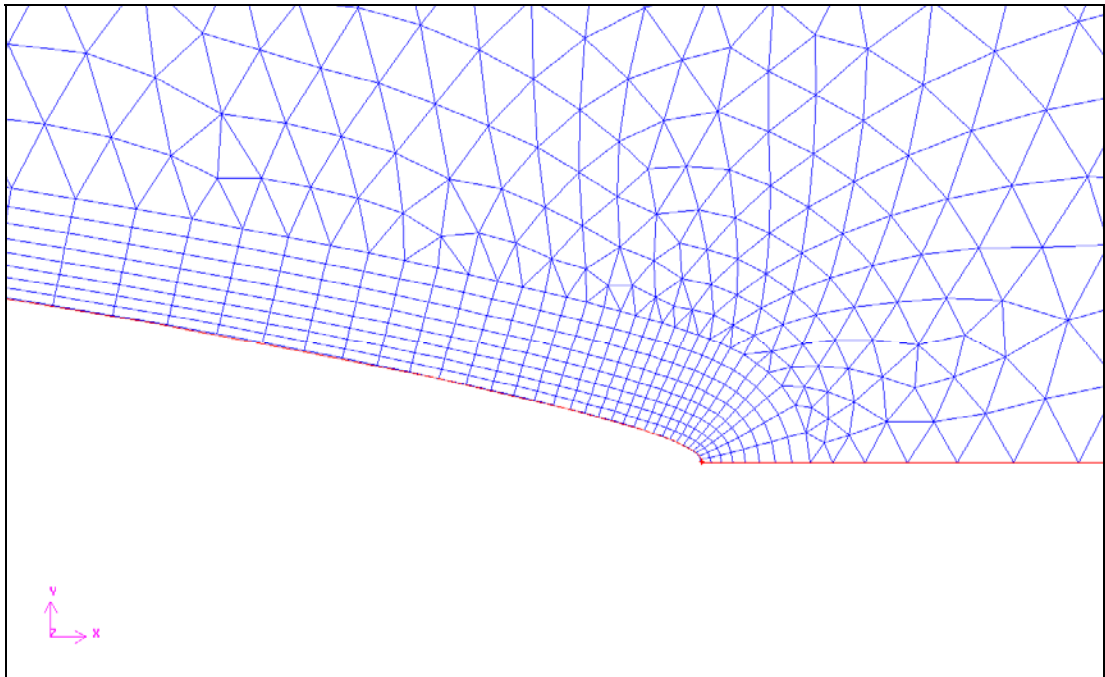
#### **4.2.2.1 Model 4159**

The first analyzed body is Model 4159. The forward (surging) motion of the body is tested. It is assumed that the body is stationary and the fluid is flowing over the body. 2D axisymmetric solver of FLUENT is employed for simulations. Axisymmetric solver was preferred over the 3D solver because for such simple geometries with no secondary flows or angle of attack, axisymmetric solver is as accurate as the 3D solver with considerably less solution time. Solution time is important for this case because many solution parameters are changed in a controlled manner over a range of Reynolds numbers. Axisymmetric simulation is commonly used for these cases and it can be seen in the literature.

Initially 3 types of mesh are created for Model 4159. Mesh 1 is coarse along the body and coarse in the boundary layer. It is used for turbulence models that do not require a fine mesh in the boundary layer ( $y^+ > 30$ ). Mesh 2 has the same number of grid points along the body; however, the mesh resolution is fine in the boundary layer to make it suitable for enhanced wall treatment. Finally, for some cases a third configuration, Mesh 3, is generated which is fine both along the body and in the boundary layer. Details of Mesh 1 are seen in Figures 4.3 and 4.4. Mesh 1 has 1264 quadrilateral boundary layer cells and 2740 triangular cells which results in a mesh with a total of 4004 cells.



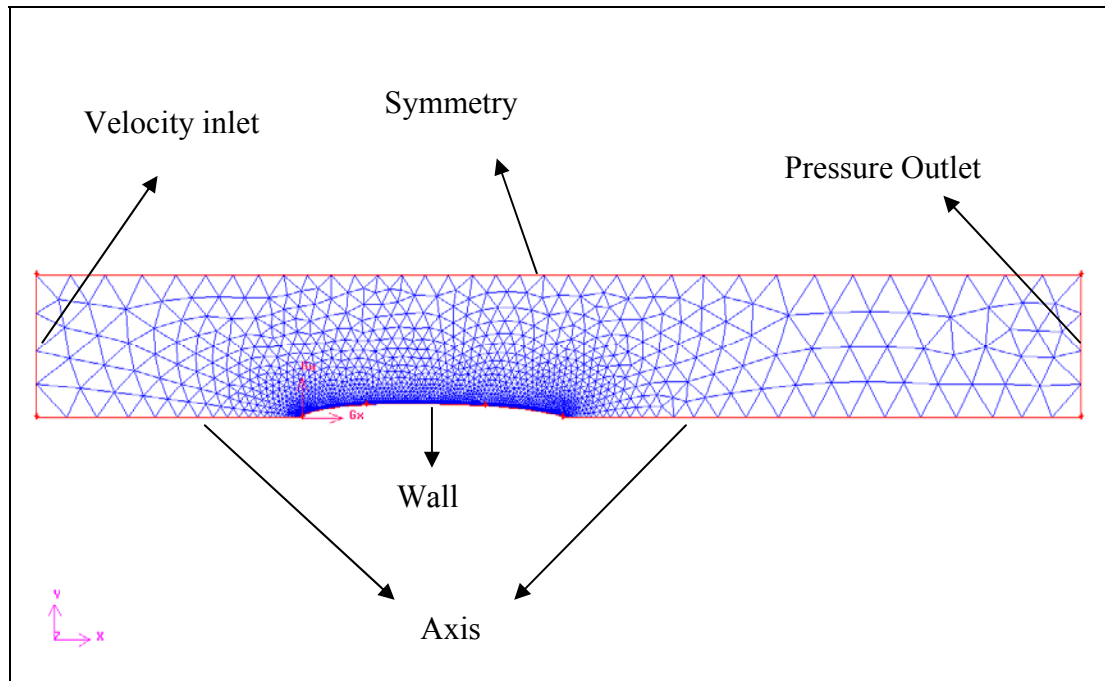
**Figure 4.3** Mesh 1 for Model 4159 – boundary layer



**Figure 4.4** Mesh 1 for Model 4159 - trailing edge detail

The domain around the body is seen in Figure 4.5. If the model length is assumed to be  $L$  and the maximum diameter is  $D$ ; the domain has a length of  $4L$  with a length of

$L$  ahead of the body and a length of  $2L$  behind the body and the height of the domain is  $11D$ . The domain has a total length of 11 m and a diameter of 1.5 m.



**Figure 4.5** Mesh 1 for Model 4159 – flow domain and boundary conditions

Some of the solution parameters are the same for all of the analyses. For instance, Green-Gauss node based gradient option is turned on. Green-Gauss node based is preferred over Green-Gauss cell based because it performs better for unstructured meshes like the one that is used in these analyses (triangular elements). Cell based option is more accurate for structured meshes [26]. Besides that, second-order discretization schemes are used for pressure, momentum and turbulence quantities. Second-order solutions are known to be much more accurate than first-order schemes. In fact, first order schemes are usually used for the initial iterations of the analyses for faster convergence. They are later on switched to second-order.

Boundary conditions are the same for all analyses: Velocity inlet at the inlet, pressure outlet at the outlet, symmetry condition at the upper edge of the domain and axis at the lower edge. Finally, wall boundary condition is applied for the body. Velocity

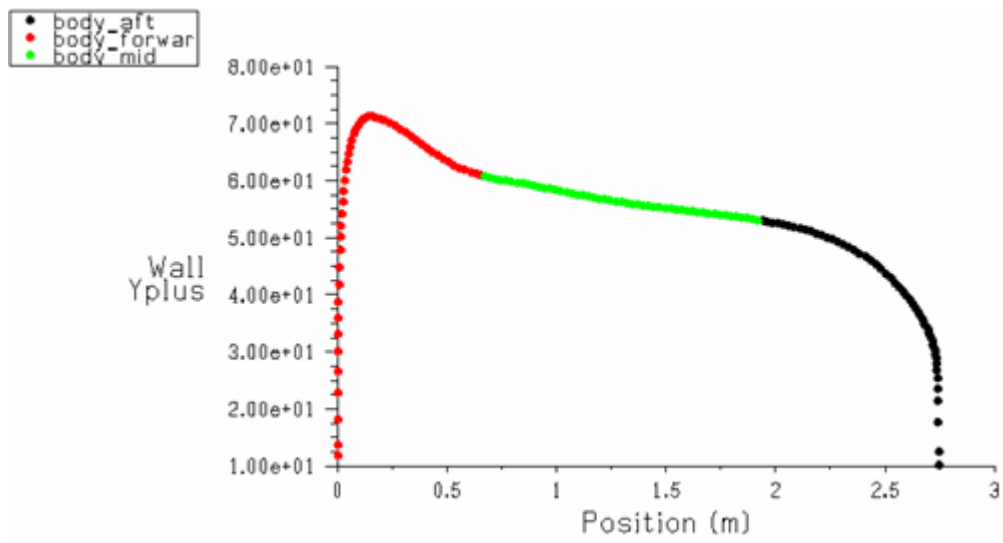
value at the inlet depends on the Reynolds number of the flow. On the other hand, turbulence quantity selection at the inlet is also critical. FLUENT offers four options for defining these quantities. Intensity and hydraulic diameter option is the most appropriate option for the present case. The hydraulic diameter is taken as the height of the inlet. Turbulence intensity, on the other hand, usually comes from the experimental data in these cases. Most of the modern wind tunnels provide the intensity value at the inlet of the tunnel. Unfortunately, for this towing tank experiment no such data is provided. Nevertheless, an approximate intensity value can be taken. Most of the simulations that are similar to this case use a turbulence intensity value of less than 1 % is used. For modern wind tunnels, this value may even drop below 0.1 %. Turbulence levels of some water tunnels in the literature can be given to demonstrate it. For instance, the water tunnel of Rolling Hills Research Corporation has a turbulence intensity level of less than 1 % [27]. In these analyses, turbulence intensity at the inlet is usually given a value less than or equal to 0.5 %. In some cases this value is modified.

Various cases are simulated for exploring the effect of turbulence mode and near wall modeling on solutions. These cases can be seen in Table 4.1.

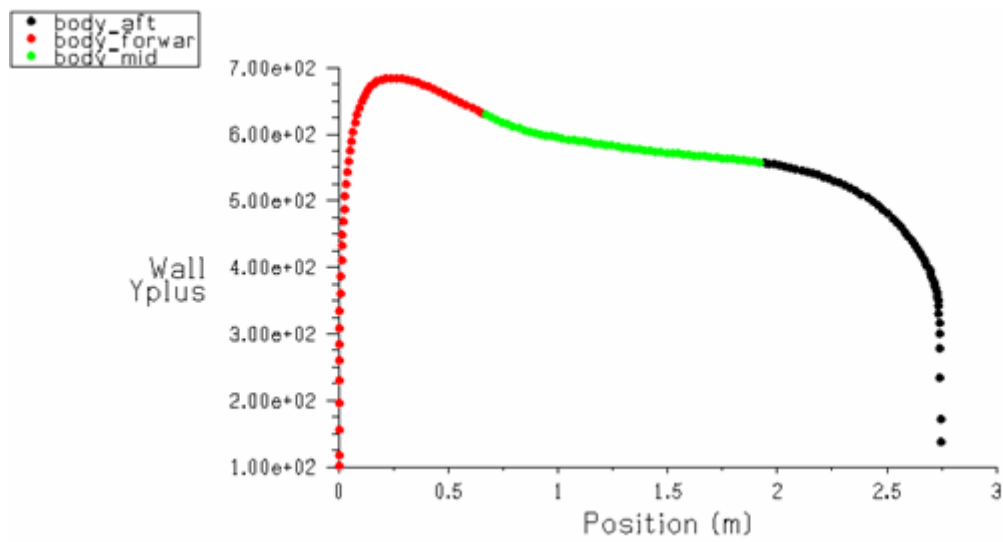
**Table 4.1** Simulation cases

<b>MESH</b>	<b>CASE</b>	<b>Viscous</b>	<b>Viscous Model</b>	<b>Near-Wall Treatment</b>
1,2,3	Case 1	laminar	-	-
1	Case 2	$k - \varepsilon$	Realizable	SWF
1	Case 3	$k - \varepsilon$	Realizable	NWF
2,3	Case 4	$k - \varepsilon$	Realizable	EWT
1	Case 5	$k - \varepsilon$	RNG	SWF
1	Case 6	$k - \varepsilon$	RNG	NWF
2,3	Case 7	$k - \varepsilon$	RNG	EWT
1	Case 8	$k - \omega$	SST	Transitional flows option (disabled)
2,3	Case 9	$k - \omega$	SST	Transitional flows option (enabled)

For Mesh 1, Case 2, Case 3, Case 5, Case 6 and Case 8 are solved; on the other hand for Mesh 2 and Mesh 3, Case 4, Case 7 and Case 9 are solved. Moreover, Mesh 1 and Mesh 2 are solved for laminar case. Details of this laminar solution is presented after turbulent analyses. The reason for using different cases for different meshes is that these meshes have different mesh densities and this affect the suitability of turbulence models for each. Mesh 1 is a coarse mesh and Cases 2, 3, 5, 6 and 8 use turbulence models which operate at coarse meshes. The intended wall  $y^+$  value is 30 as the guideline suggests. This aim was satisfied for  $Re=2 \times 10^6$  as it can be seen in Figure 4.6. Similar results are achieved for other cases. However, as the Reynolds number is increased, the wall  $y^+$  value gets very high values as seen in Figure 4.7 and exceeds 100 which is advised by the User's Guide as the upper limit. It should be noted that the transitional flow option is a low-Reynolds number correction to the turbulent viscosity. If this option is disabled like in Case 8, a coarse mesh should be used; however, if this option is enabled like in Case 9, a fine mesh should be used [26].

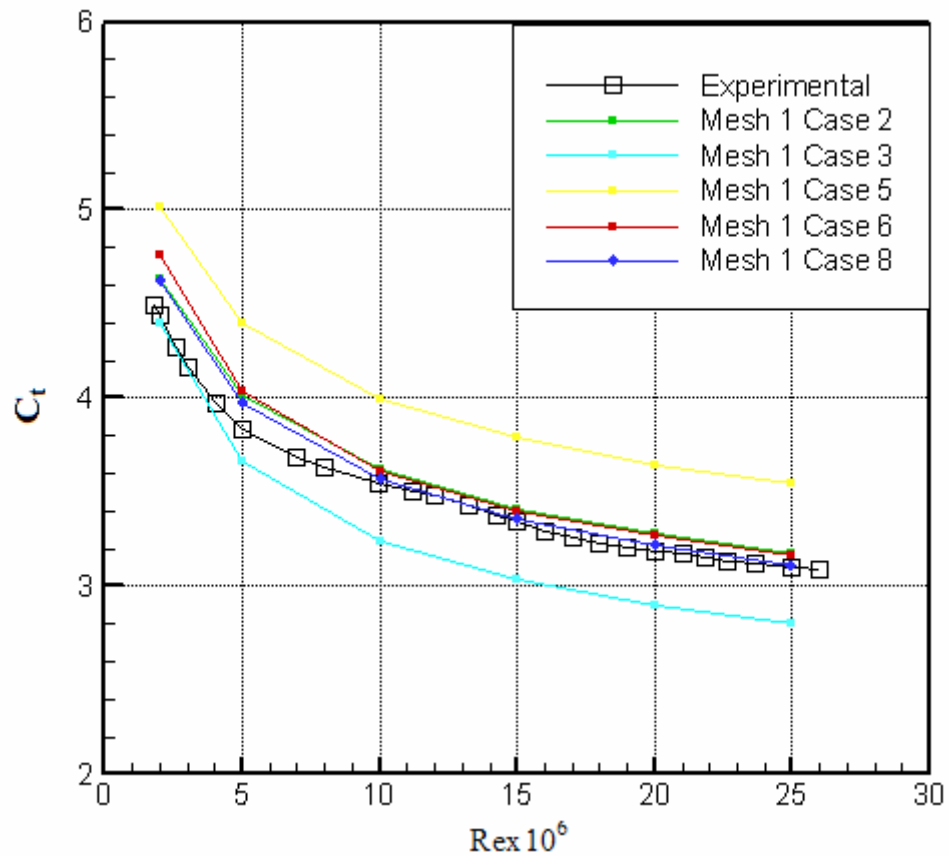


**Figure 4.6** Wall  $y^+$  for Mesh 1 Case 2 at  $Re=2 \times 10^6$



**Figure 4.7** Wall  $y^+$  for Mesh 1 Case 2 at  $Re=2.5 \times 10^7$

In Figure 4.8 a comparison of these cases with the experimental data for resistance coefficient can be seen.

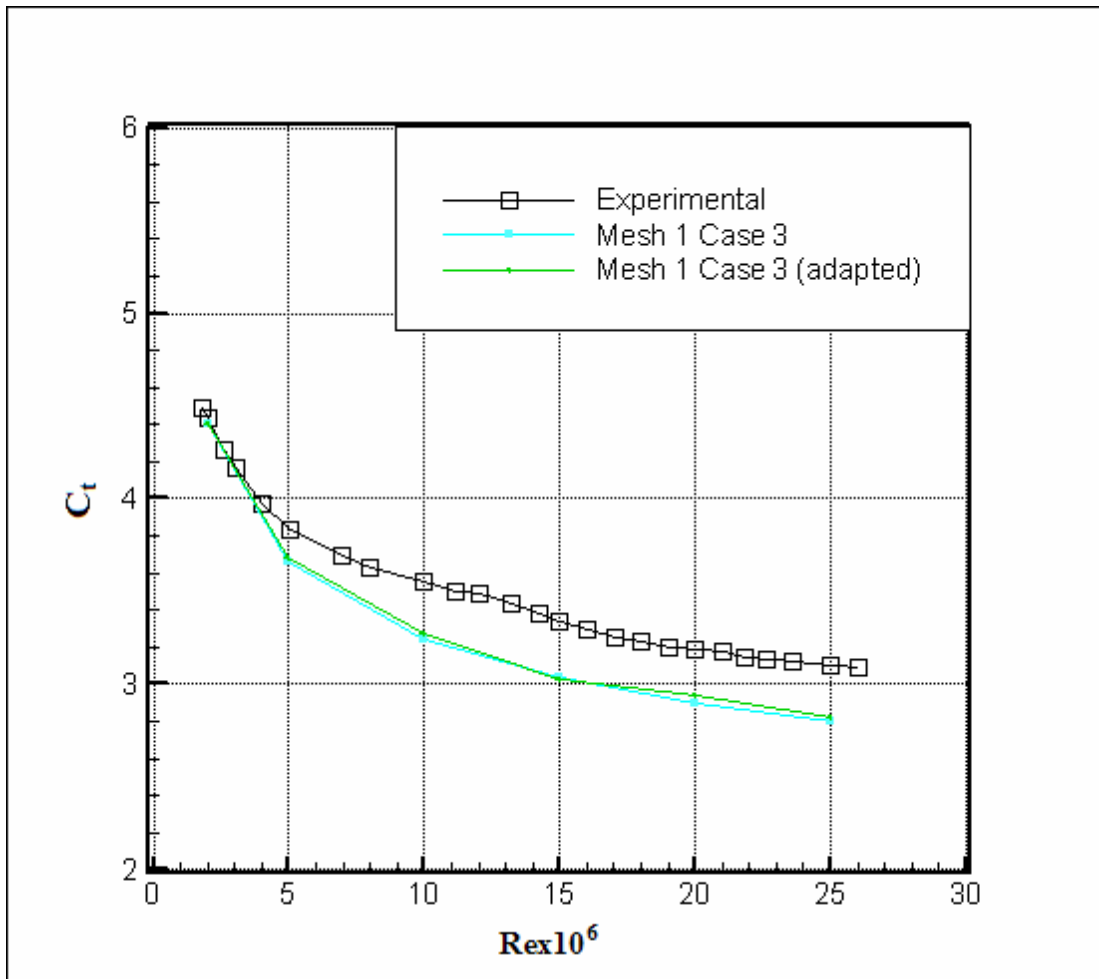


**Figure 4.8** Comparison of Mesh 1 results – total resistance coefficient

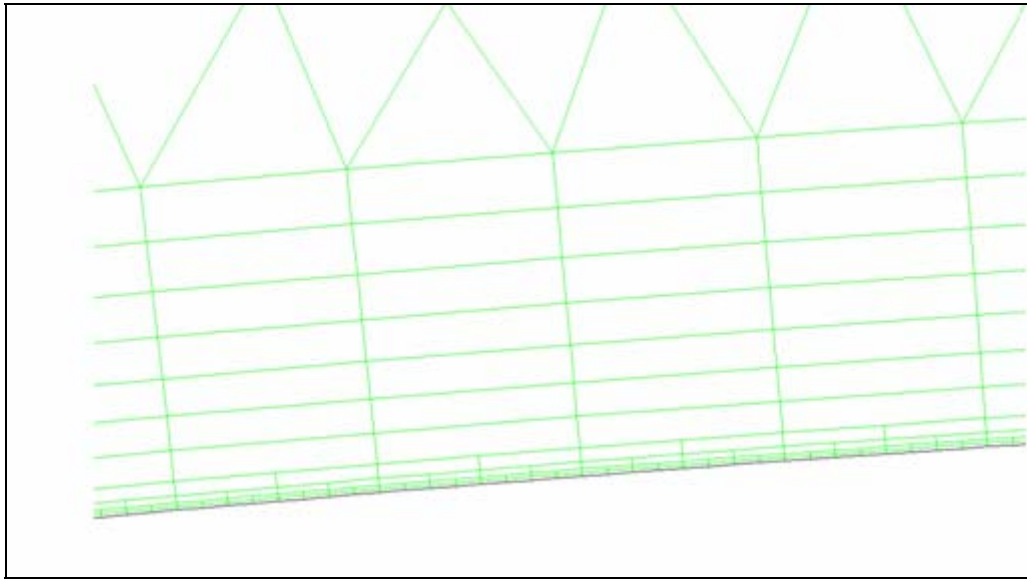
As it is seen, RNG model with SWF's fails to match the resistance coefficient at all Reynolds numbers. The  $k - \varepsilon$  model with NWF's is accurate for low-Reynolds numbers and loses accuracy at higher speeds. The  $k - \varepsilon$  model with SWF's, RNG with NWF's and SST gives inaccurate results for low-Reynolds numbers but matches the experimental data at higher  $Re$  numbers.

As it was mentioned before, at high  $Re$  numbers wall  $y^+$  values get values above 100. To check whether it is affecting the results or not, mesh is adapted so that wall  $y^+$  value is between  $30 < y^+ < 100$ . For Case 2, Case 3 and Case 6 results are almost identical. Comparison of case 3 and its adapted version is seen in Figure 4.9. Moreover, in Figure 4.10 mesh generated for  $Re=2 \times 10^7$  after adapting the grid is shown.



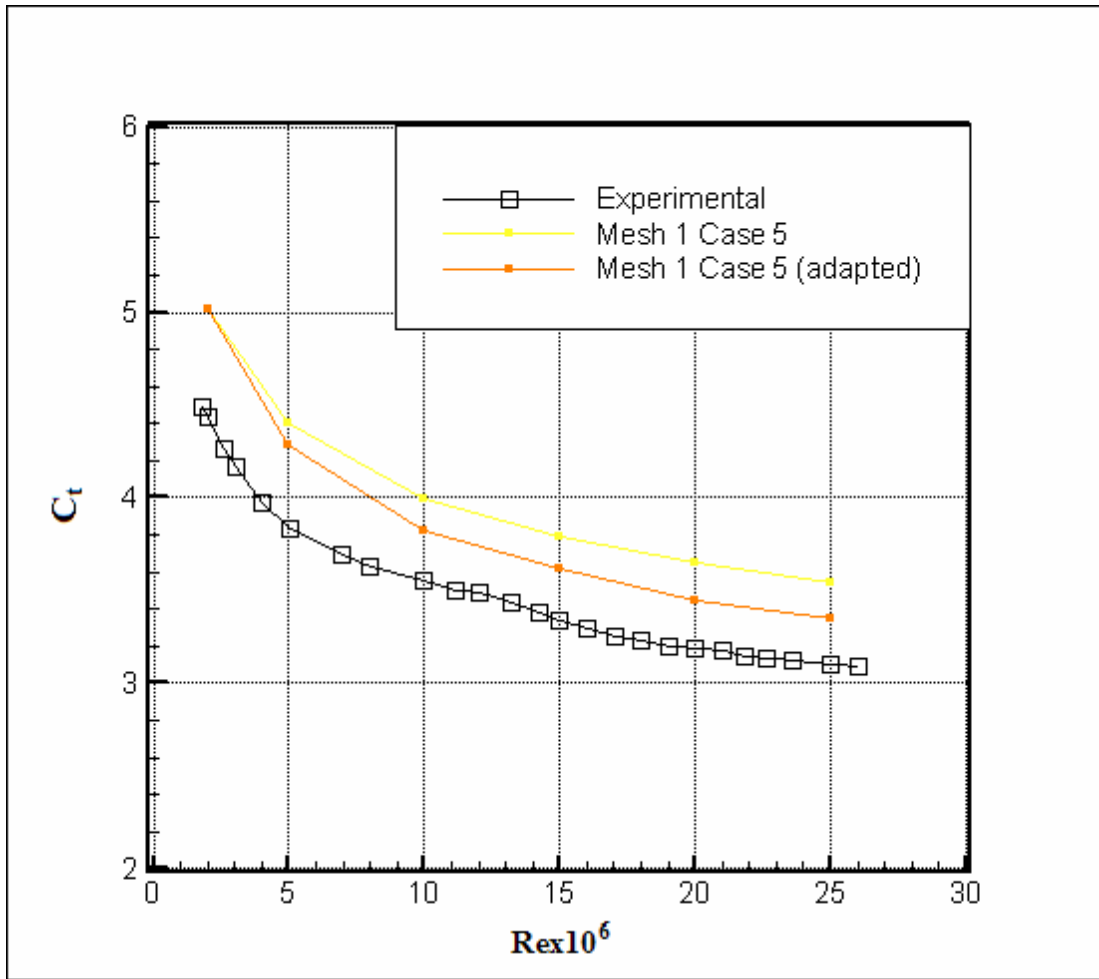


**Figure 4.9** Adaptation results for Mesh 1 Case 3



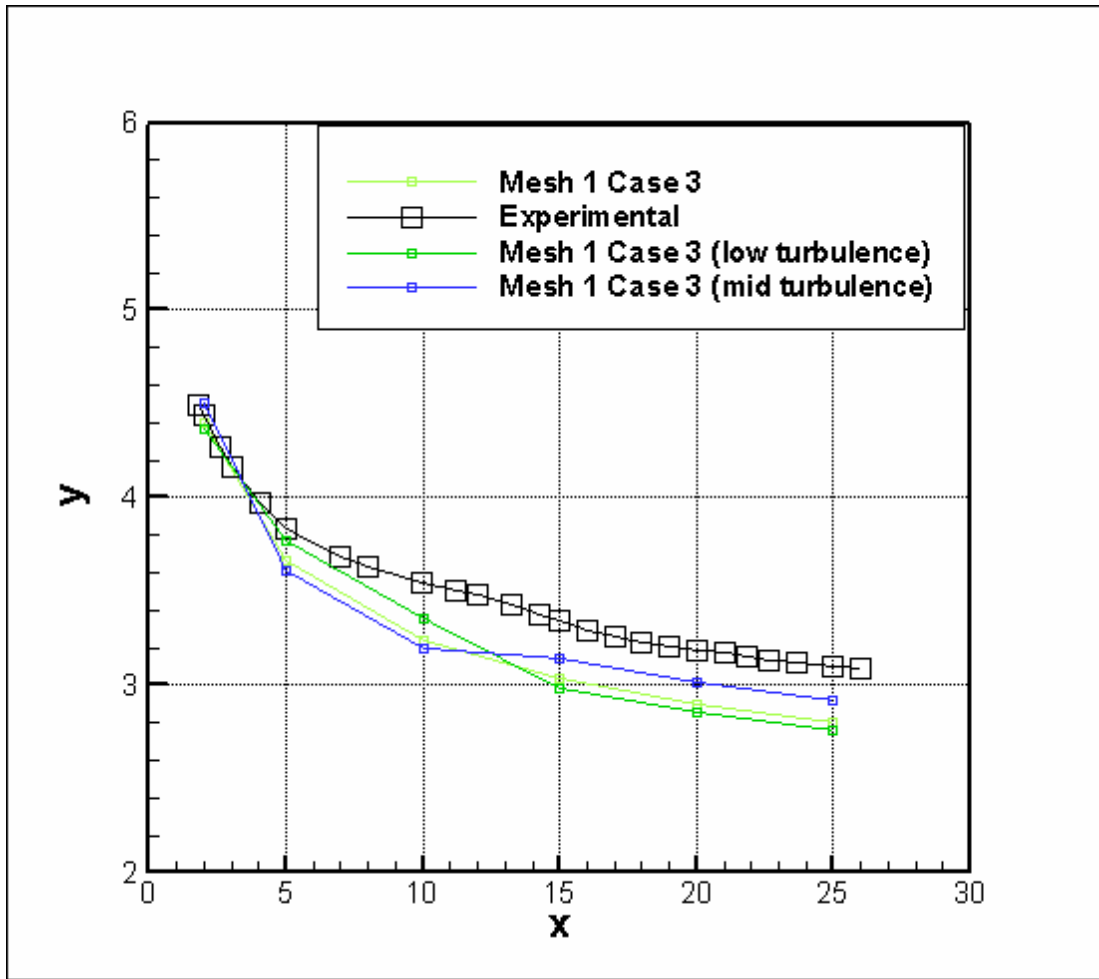
**Figure 4.10** Modified Mesh 1 after  $y^+$  adaption

For Case 5 results are improved but still not satisfactory as seen in Figure 4.11.



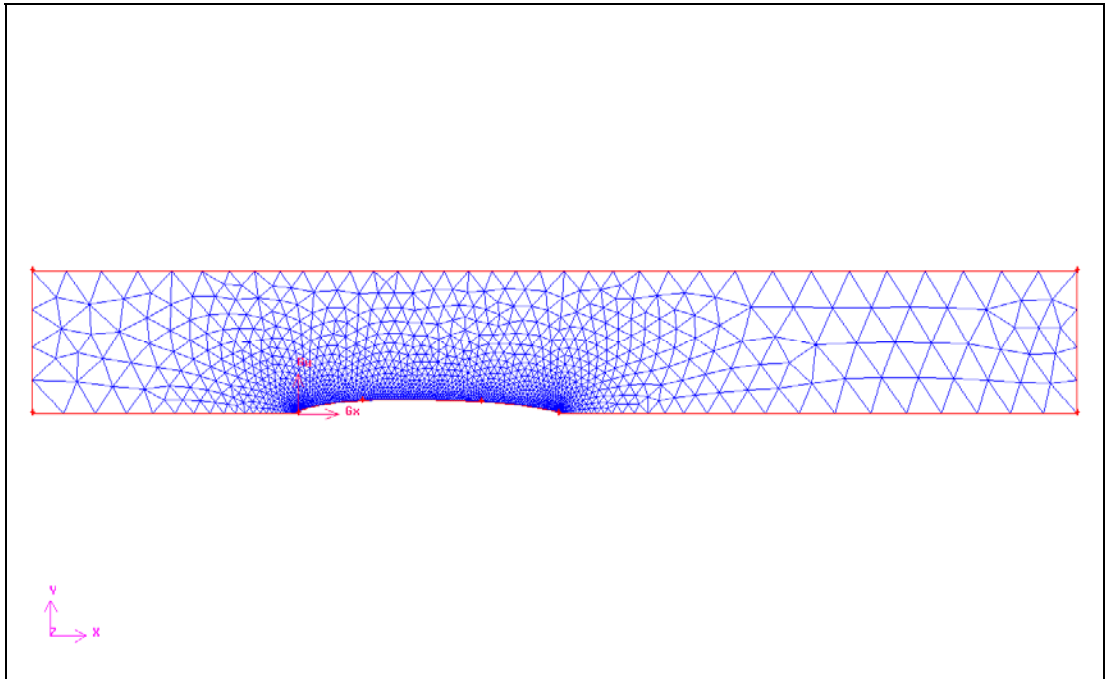
**Figure 4.11** Adaptation results for Mesh 1 Case 5

An attempt is taken to check whether the turbulence intensity level has negative effects on the solution. For Case 3, higher and lower turbulence intensity levels are assigned at the inlet. For the high turbulence case, intensity is taken as 1% and for the low turbulence case; it is taken as 0.1 %. Results are seen in Figure 4.12. Low turbulence case is accurate for Re numbers up to  $1 \times 10^7$  and high turbulence case is accurate for higher Re numbers. These results show the importance of the inlet turbulence intensity level on the accuracy of solutions. To sum up, for the coarse mesh Case 2, Case 6 and Case 8 are successful.

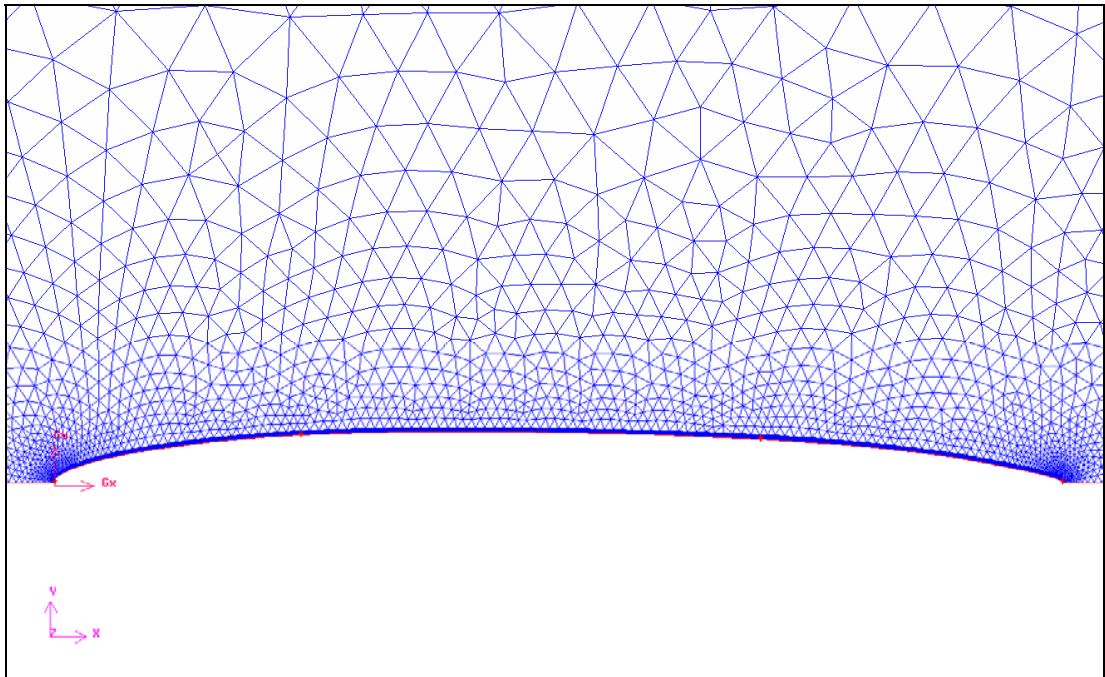


**Figure 4.12** Effect of turbulence intensity on total resistance coefficient

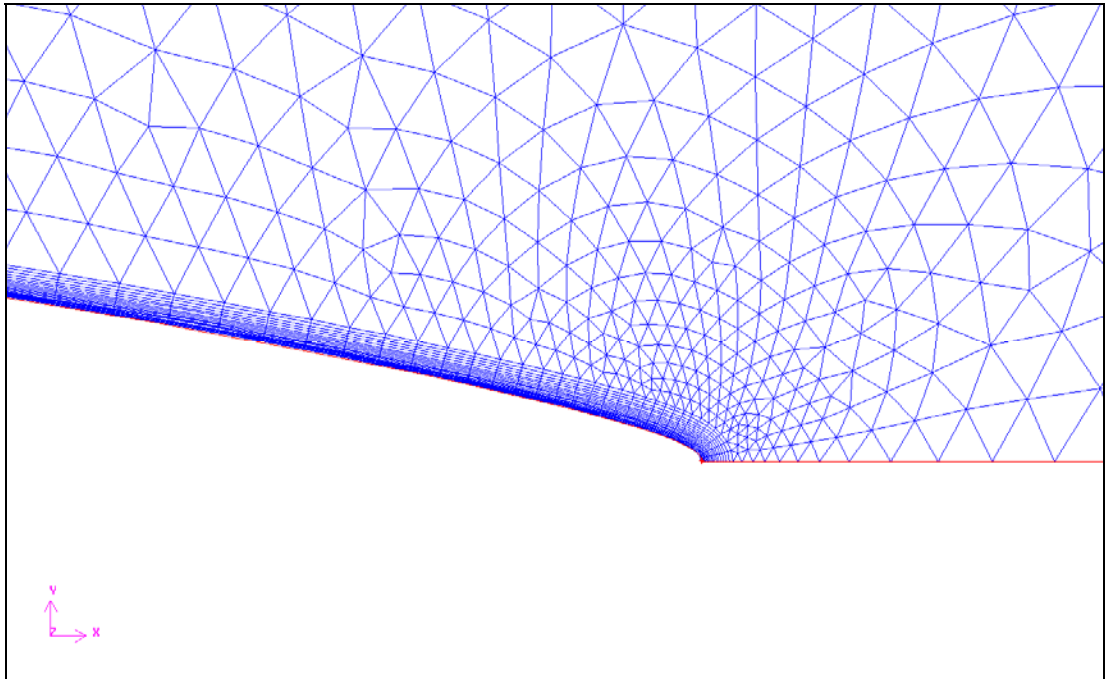
After Mesh 1 simulations, a finer mesh, Mesh 2, is used. This mesh has the same number of grid points along the body; however, the mesh is more tightly clustered in the boundary layer. The purpose is achieving a fine mesh with wall  $y^+ < 5$ . Details of Mesh 2 are seen in Figures 4.13 to 4.15. Mesh 2 has 3160 quadrilateral boundary layer cells and 3434 triangular cells which results in a mesh with a total of 6594 cells. The same domain with Mesh 1 is used. For  $Re=2 \times 10^6$   $y^+$  is around acceptable limits; however, at higher  $Re$  numbers  $y^+$  exceeds the limiting value of 5. Due to this fact, adaptation is applied to get the required mesh quality.



**Figure 4.13** Mesh 2 for Model 4159 – flow domain

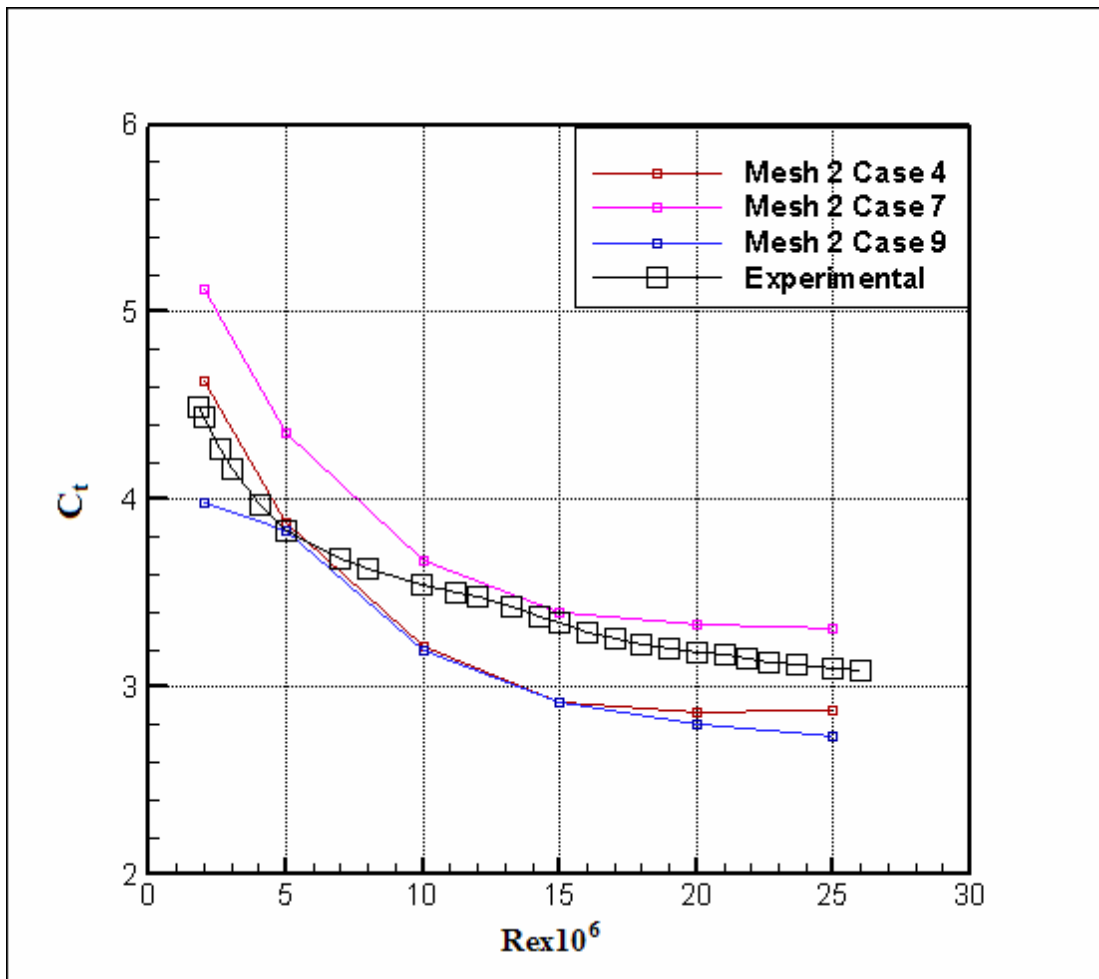


**Figure 4.14** Mesh 2 for Model 4159 – boundary layer



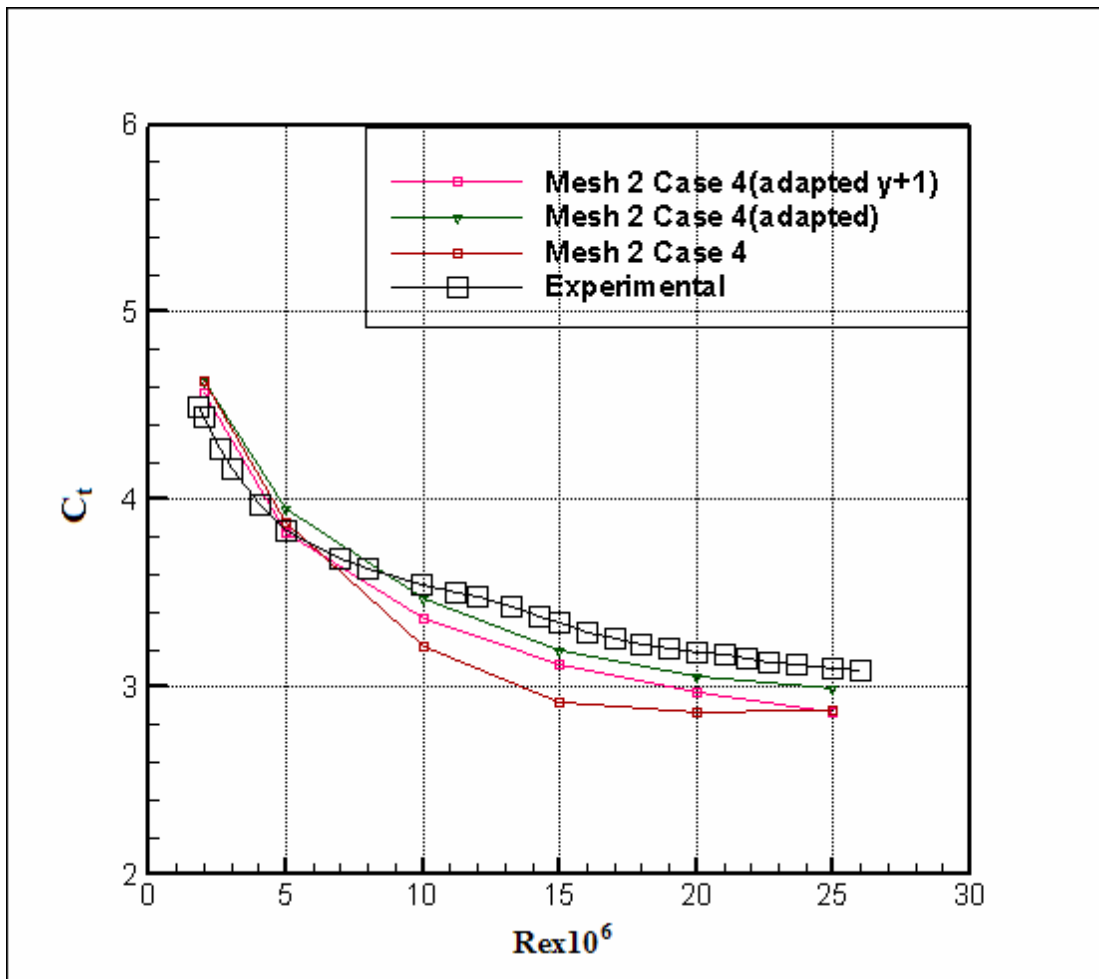
**Figure 4.15** Mesh 2 for Model 4159 – trailing edge

Mesh 2 is aimed to be a fine mesh to solve for Cases 4, 7 and 9 which require a mesh with wall  $y^+$  lower than 5. This dependency is especially critical for Cases 4 and 7 because these cases use EWT approach. Results of these cases without adaptation are seen in Figure 4.16. It is seen that all cases give inaccurate results. The most probable reason for this is the unsatisfactory mesh resolution especially at high  $Re$  numbers.



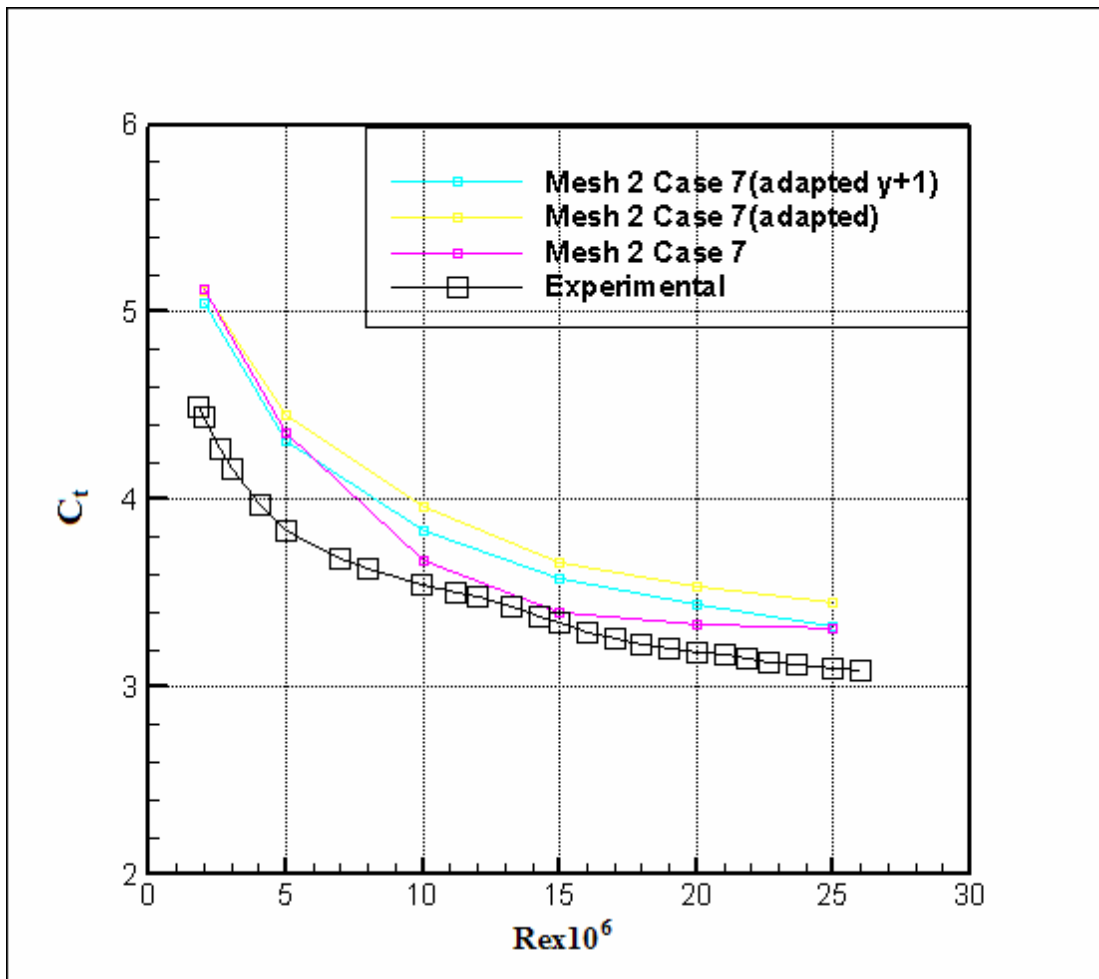
**Figure 4.16** Comparison of Mesh 2 results – total resistance coefficient

To overcome the poor mesh quality at the wall, adaptation is done. Cases 4, 7 and 9 are adapted for  $y^+ = 5$  and  $y^+ = 1$ . The results are presented in Figures 4.17 to 4.19.

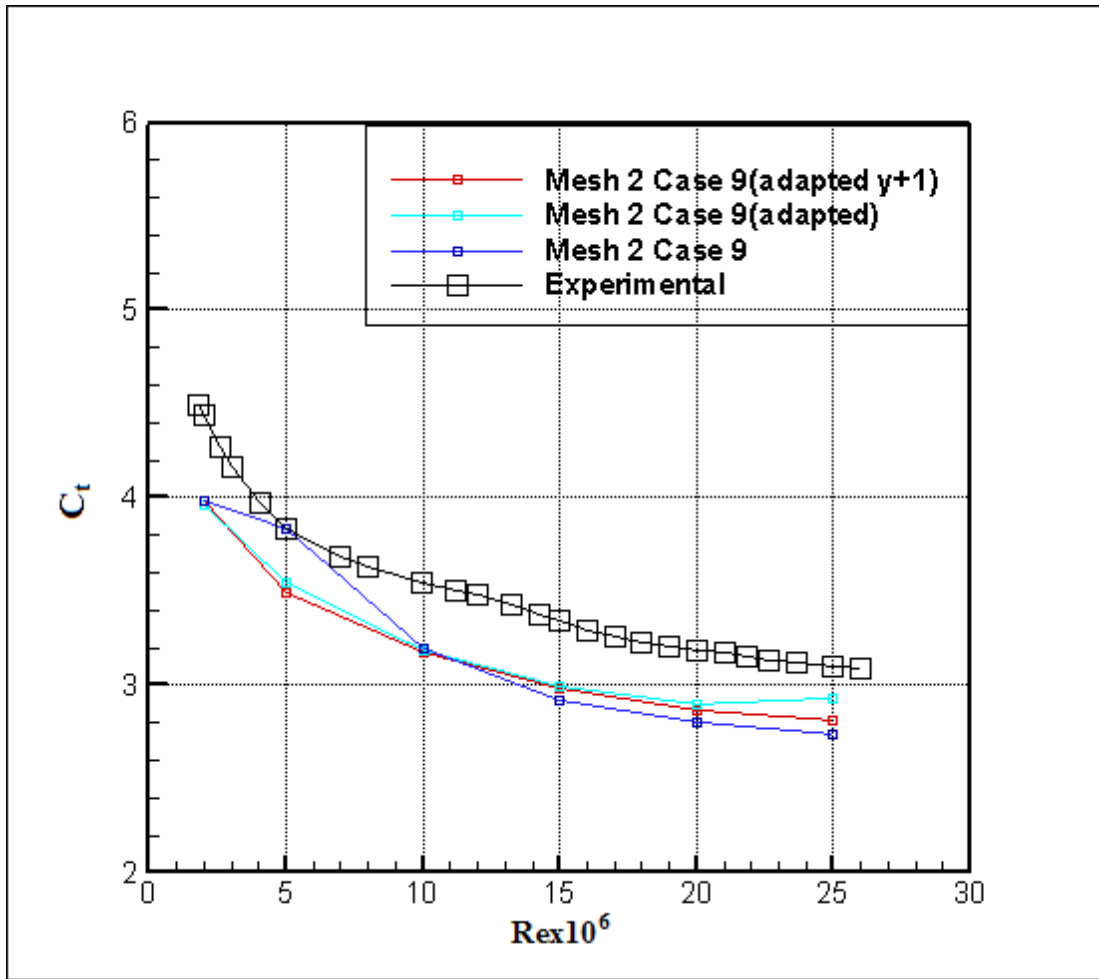


**Figure 4.17** Adaptation for Case 4 with Mesh 2 – total resistance coefficient





**Figure 4.18** Adaptation for Case 7 with Mesh 2 – total resistance coefficient

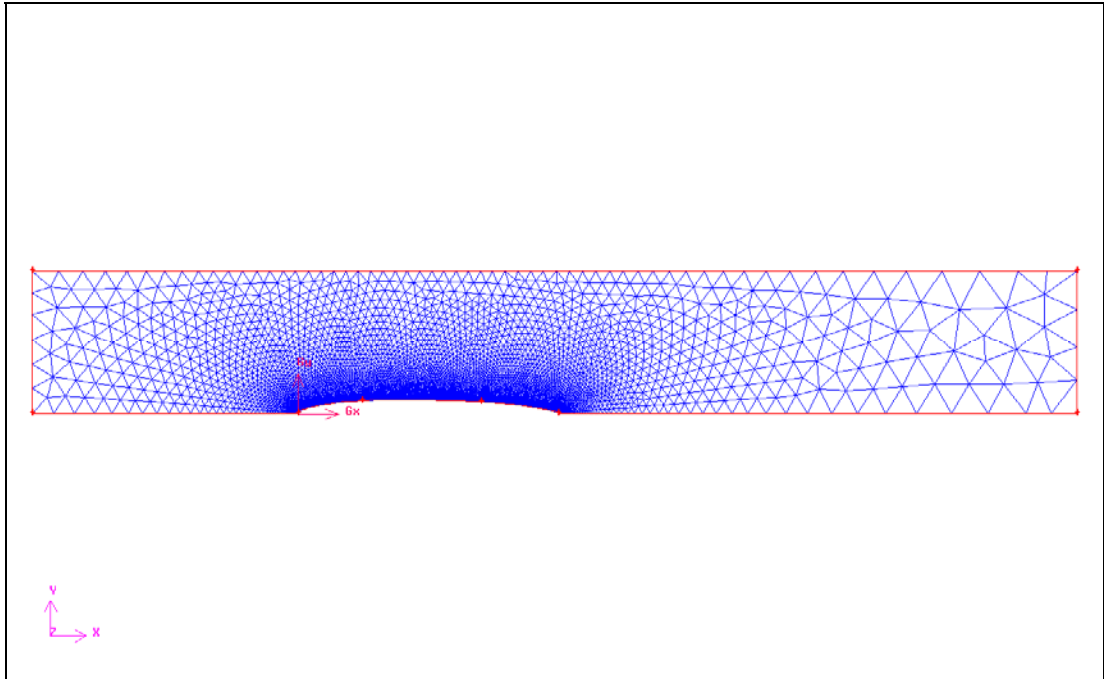


**Figure 4.19** Adaptation for Case 9 Mesh 2 – total resistance coefficient

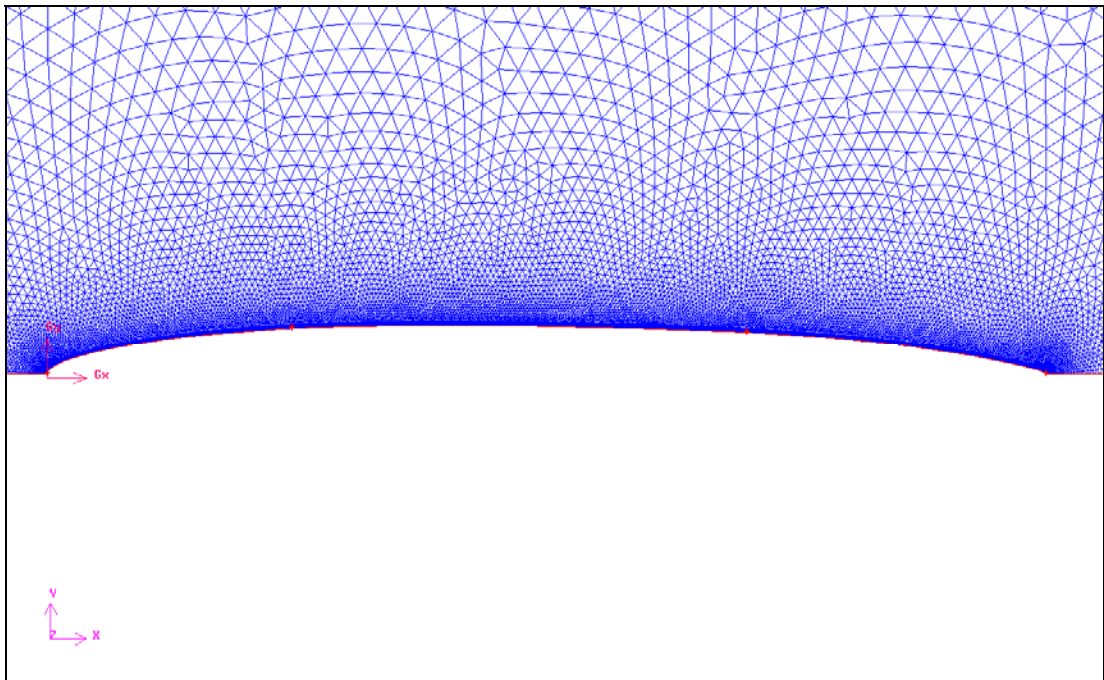
For case 4, adaptation is helpful. In Figure 4.17, Case 4 with Mesh 2 (adapted) refers to the case where adaptation is made for  $y^+ = 5$ . It is seen that  $y^+ = 5$  adaptation improved the results of case 4 significantly. Interestingly,  $y^+ = 1$  adaptation predicted worst. Conversely, for Cases 7 and 9, adaptation results are poor as seen in Figures 4.18 and 4.19, respectively. To sum up, for the fine mesh (Mesh 2), Case 4 which is adapted to  $y^+ = 5$  provides the best prediction.

After the analyses on Mesh 1 and Mesh 2, a final attempt is done with Mesh 3. This mesh has the same density in the boundary layer as Mesh 2; however, the mesh density is increased along the body. Mesh details are seen in Figures 4.20 to 4.22. Mesh 3 has 12760 quadrilateral boundary layer cells and 18002 triangular cells

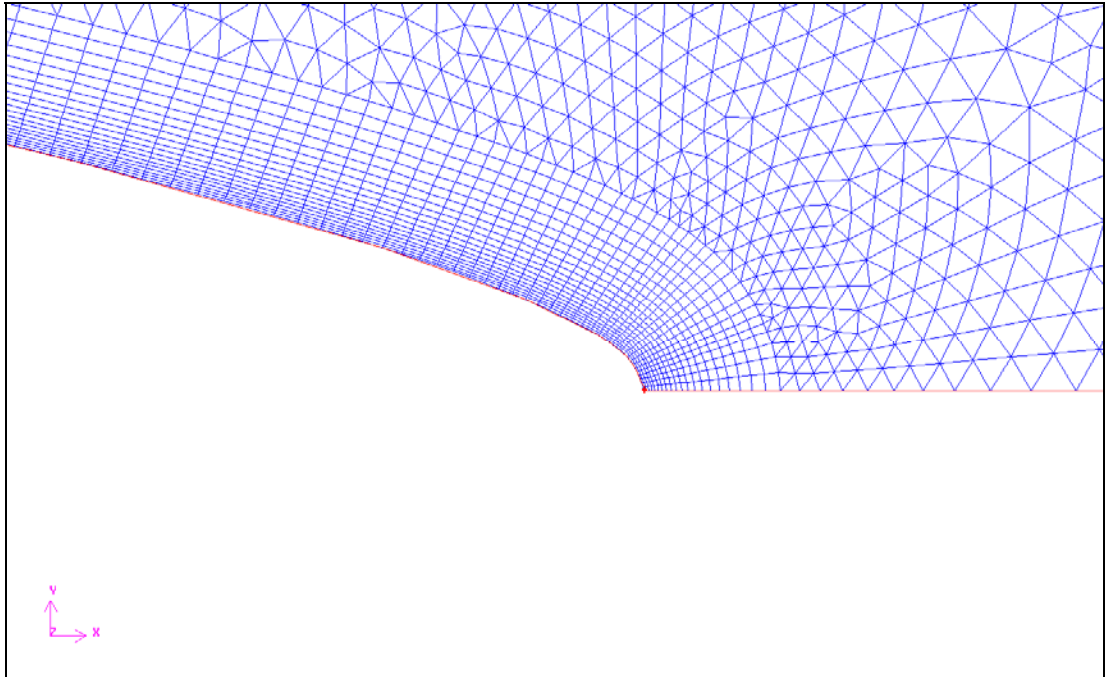
which results in a mesh with a total of 30762 cells. The same domain with Mesh 1 and Mesh 2 is used.



**Figure 4.20** Mesh 3 for Model 4159 – flow domain

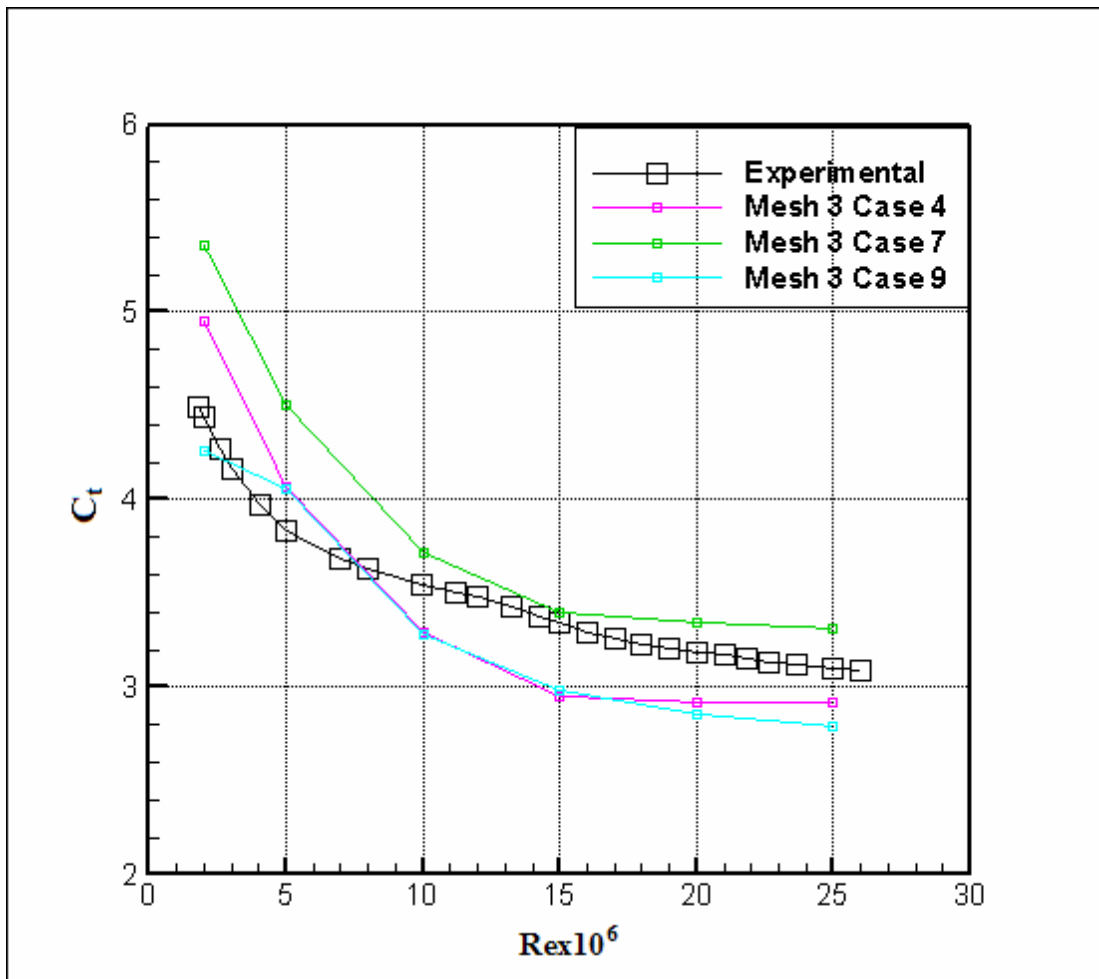


**Figure 4.21** Mesh 3 for Model 4159 – boundary layer

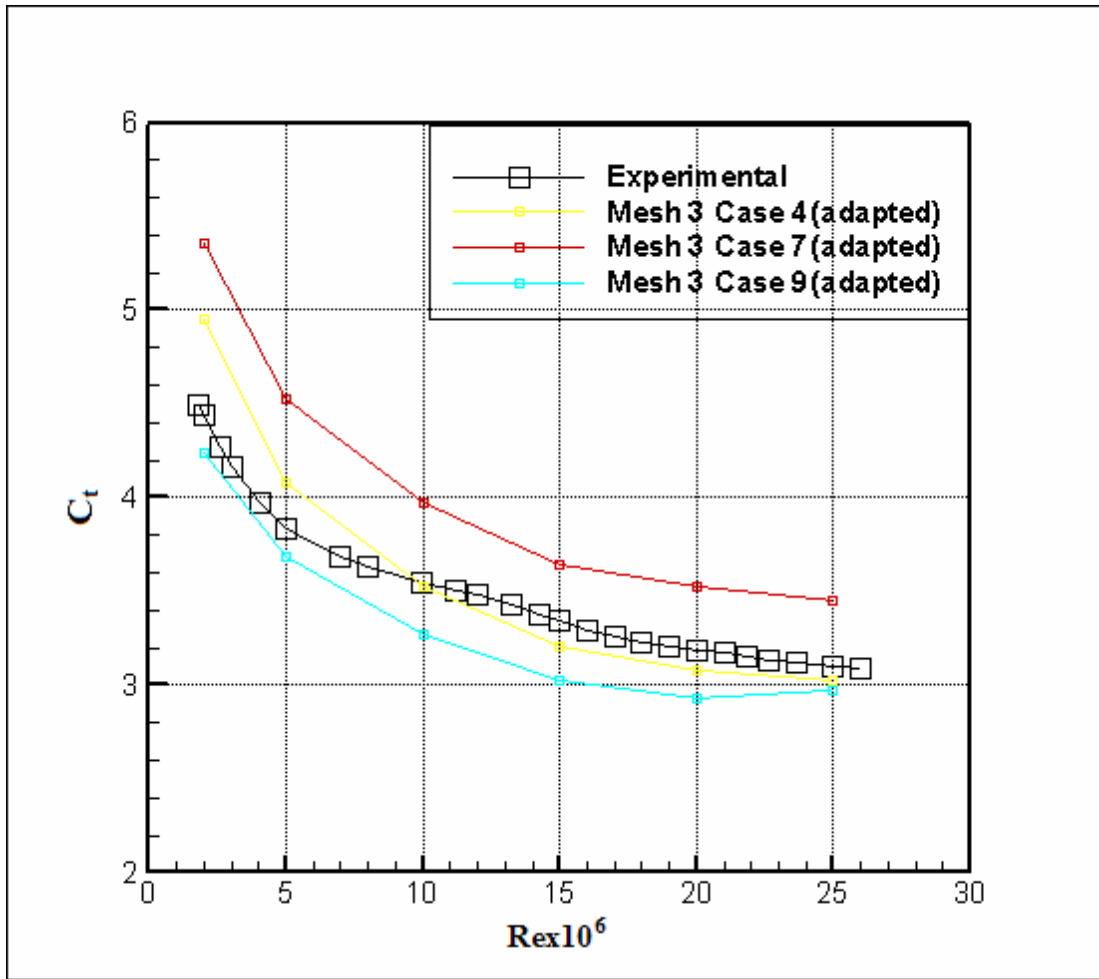


**Figure 4.22** Mesh 3 for Model 4159 – trailing edge

Since Mesh 3 is a low-Reynolds number mesh Case 4, Case 7 and Case 9 are simulated. As seen in Figure 4.23, none of the cases is accurate. When adaptation is applied, results improved, especially for Case 4, as seen in Figure 4.24.

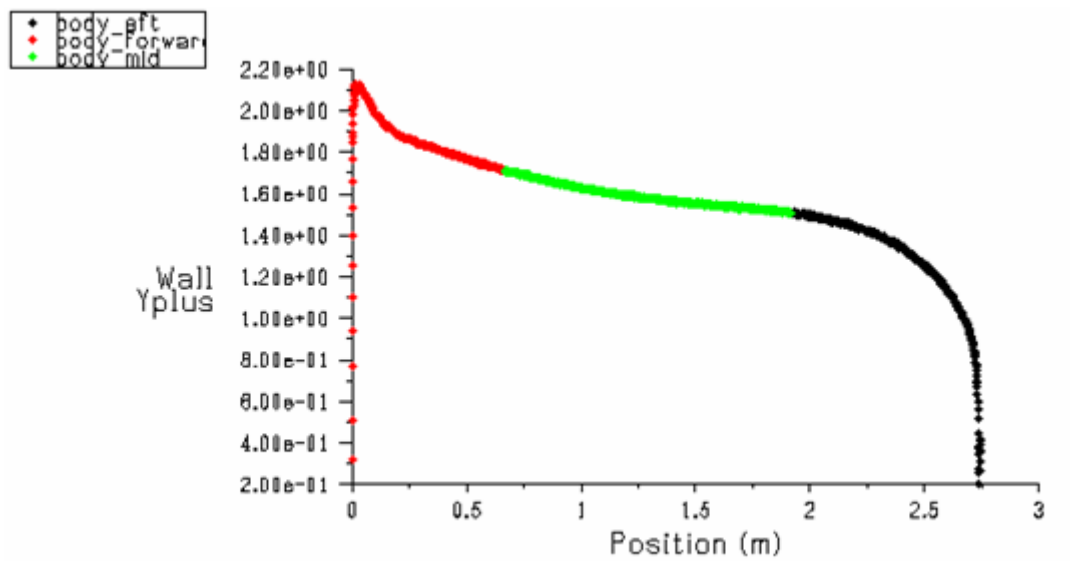


**Figure 4.23** Comparison of Mesh 3 results – total resistance coefficient



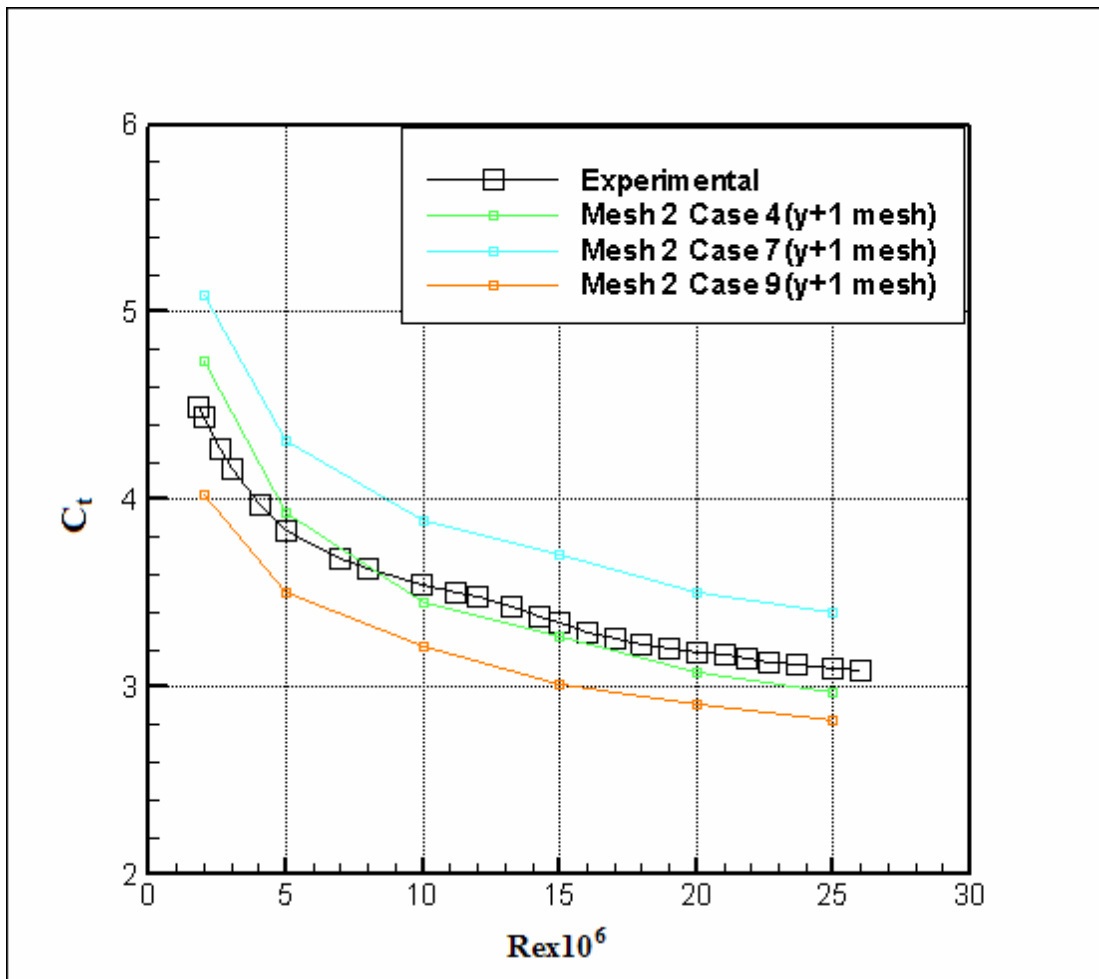
**Figure 4.24** Adaptation results for Mesh 3 – total resistance coefficient

After these simulations, several conclusions are made. First of all, adaptation is not always successful. Even though results are improved for some situations, such as Mesh 2 - Case 4 and Mesh 3 - Case 4, poor results are received as well. Due to erroneous results of adaptation, a new approach is tested for Model 4159. This time the mesh is generated for each Reynolds number separately so that wall  $y^+$  is about 1-2 for each case without any need for adaptation. For instance, for Case 4 at  $Re=2 \times 10^7$ , the  $y^+$  plot over the body is seen in Figure 4.25.



**Figure 4.25** Wall  $y^+$  plot for Case 4 at  $Re=2 \times 10^7$

This approach was applied for Case 4, Case 7 and Case 9. Results are seen in Figure 4.26. It is seen that Case 4 gives accurate results at all Reynolds numbers while other cases fail to match the experimental data.



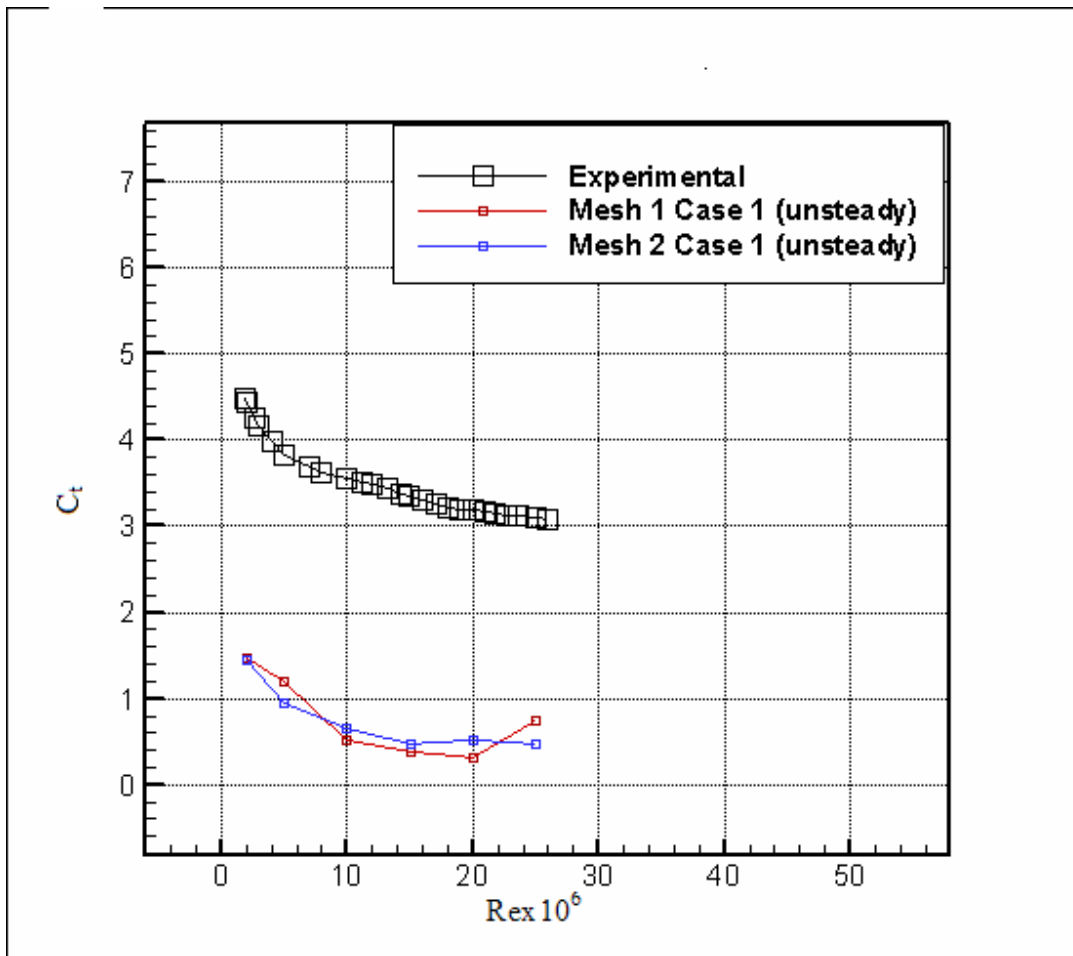
**Figure 4.26** Comparison of total resistance coefficient for  $y^+$  meshes

Up to now it is seen that Case 4 (realizable  $k - \varepsilon$ , EWT) gives consistently good results provided that the mesh resolution is fine enough at the wall. Other than Case 4; Case 6 (RNG  $k - \varepsilon$ , NWF), Case 2 (realizable  $k - \varepsilon$ , SWF) and Case 8 (SST, transitional option disabled) give good results for Mesh 1 (coarse mesh). Conversely, for fine meshes (Mesh 2 and Mesh 3), RNG and SST predict inaccurately. At this point, it seems that realizable  $k - \varepsilon$  model with enhanced wall treatment (Case 4) is the most appropriate choice for the simulation of the unmanned underwater vehicle. At first glance the fine mesh requirement of Case 4 may seem to be a drawback due to high computation time. However, it should be noted that marine applications usually have relatively lower Reynolds number values; hence, the mesh requirement does not increase to very high values. Using a fine mesh near the wall has another



advantage. As it will be given in the following chapter, there are several small appendages protruding out of the body. When these protrusions are included in the model, it is inevitable to use fine mesh. If a coarse mesh is used, the flow around the appendage will most probably be ignored. It is important to note that in some cases drag due to appendages may have a considerable contribution on the total drag of the vehicle.

As it was mentioned before, Case 1 which is the laminar case is solved for Mesh 1 and Mesh 2 of Model 4159. Steady laminar analyses usually have problems with convergence; hence, analyses are solved unsteady with a time step of 0.01 s. Results are seen in Figure 4.27. It is seen that laminar solutions predicted the flow case inaccurately.

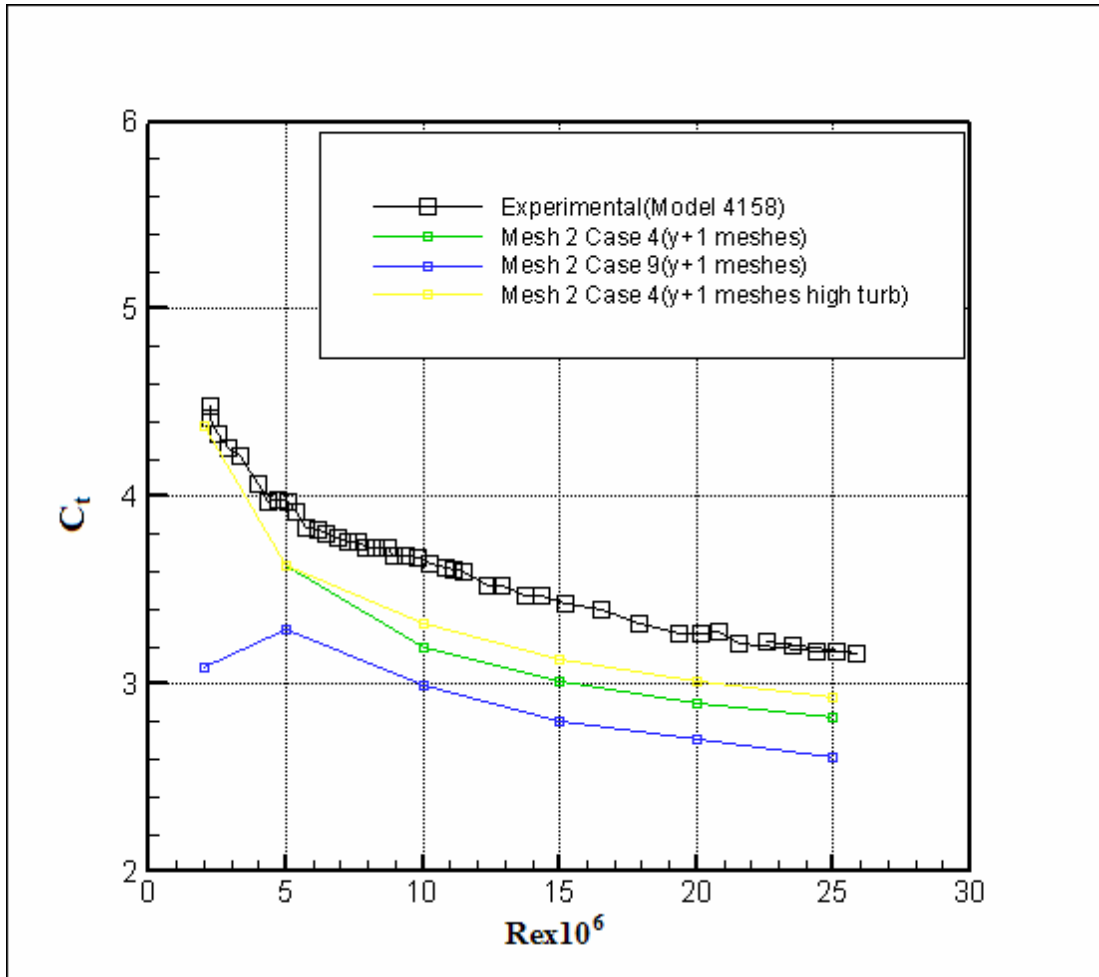


**Figure 4.27** Laminar solution of Model 4159 Meshes 1 and 2

To demonstrate the accuracy of Case 4, additional comparisons were made. This time Models 4158 and 4154 were analyzed.

#### 4.2.2.2 Model 4158

Model 4158 is one of the 24 models tested by Gertler [17]. Solution parameters similar to Model 4159 are used. Analyses are performed based on the concept explained in the previous part. A new mesh is generated for each Reynolds number so that the  $y^+$  is around 1 without any adaptation. Cases 4, 7 and 9 are used. Results are seen in Figure 4.28. None of the models predicted the total resistance correctly. Nevertheless, amongst these cases, Case 4 still provides the most accurate answer.



**Figure 4.28** Total resistance comparison for Model 4158

Gertler's experiments were performed for resistance only. No pressure distribution or other data were provided. There are, however, some computational studies done with the Series-58. One such study was performed by Gorski [28]. Gorski used Models 4159, 4158 and 4155 in his studies. A finite-volume based code which solves incompressible Navier-Stokes equations for axisymmetric cases was used for simulations. The algebraic eddy viscosity model of Baldwin-Lomax is used for turbulence modeling. Resistance coefficient results of Gorski match well with experimental data. Gorski also provided surface pressure coefficient,  $C_p$ , distribution of Models 4159 and 4158. Surface pressure coefficient is defined as

$$C_p = \frac{P - P_{ref}}{\frac{1}{2} \rho U^2} \quad (4.18)$$

where  $P_{ref}$  is the reference pressure which is usually taken as the operating pressure FLUENT.

$C_p$  data of Model 4158 are used for comparison with results that are obtained in this study. In Figures 4.29 and 4.30, these comparisons can be seen for Model 4158 with  $Re=2 \times 10^6$  and  $Re=2.5 \times 10^7$ , respectively. It is seen that computed surface pressure coefficient distribution of Gorski is very similar to those obtained here especially at the leading and trailing edges.

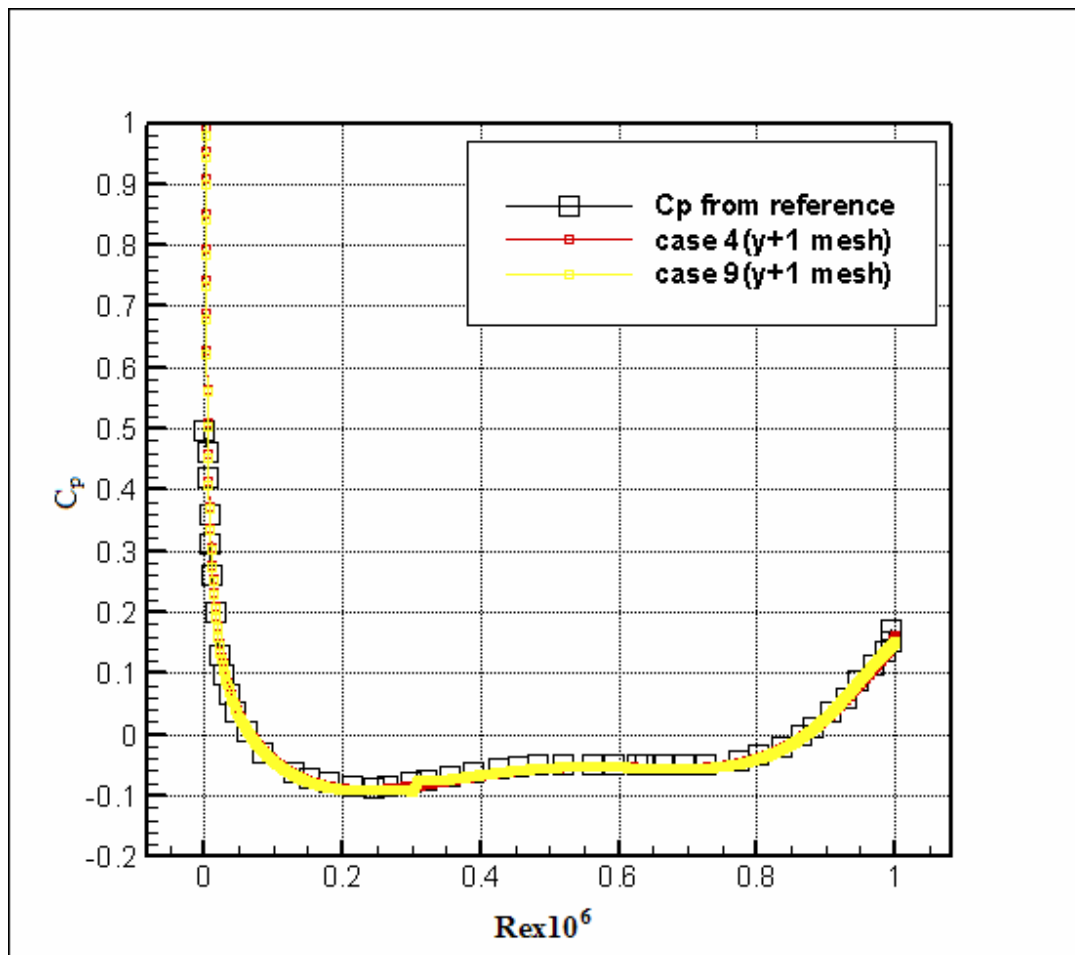
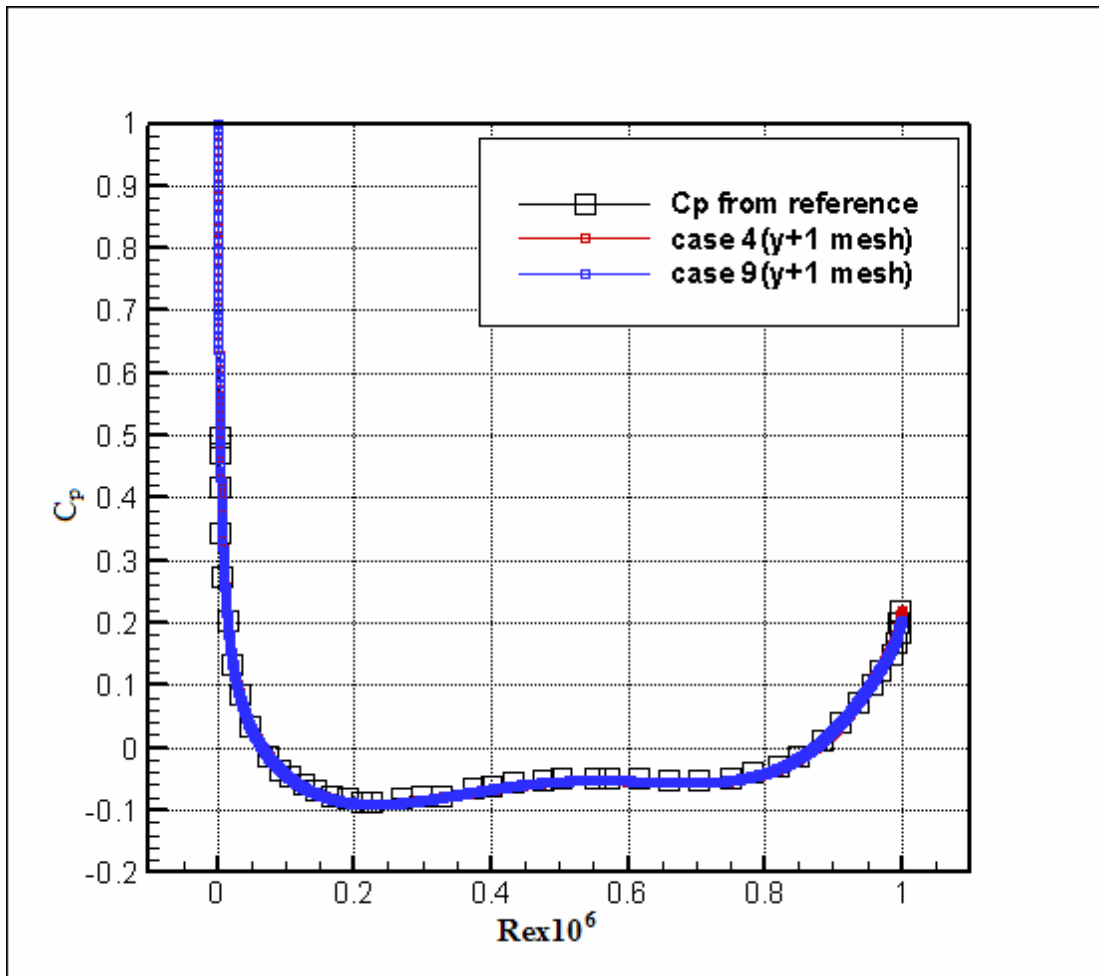


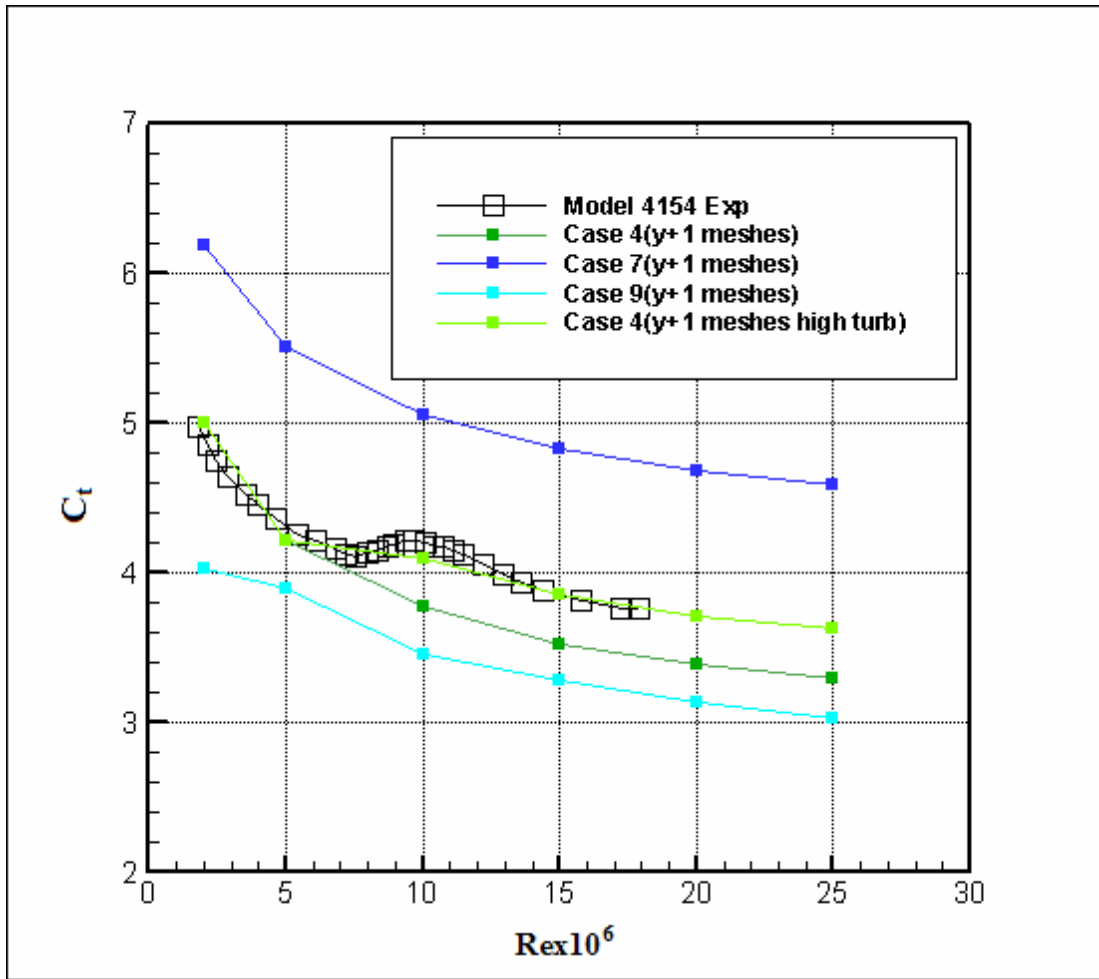
Figure 4.29 Pressure coefficient comparison at  $Re=2 \times 10^6$



**Figure 4.30** Pressure coefficient comparison at  $Re=2.5 \times 10^7$

#### 4.2.2.3 Model 4154

Model 4154 is the last model which is simulated for validation purposes. It is similar to Model 4159 and Model 4158; however, this model is relatively thicker with an  $L/D$  ratio of 4. Similar studies to Model 4158 were performed. Meshes with wall  $y^+ = 1$  are generated for each Reynolds number. Cases 4, 7 and 9 were tested again. Results are seen in Figure 4.31.



**Figure 4.31** Total resistance comparison for Model 4154

It is seen from Figure 4.31 that only Case 4 predicts resistance correctly at low Reynolds numbers. However, Case 4 fails to predict correctly for higher  $Re$  numbers while maintaining a parallel trend with the experimental data. Due to that, for Reynolds numbers higher than or equal to  $1 \times 10^7$ , turbulence intensity was increased to 1 % from 0.5 %. After this modification, Case 4 predicted almost the same resistance for high Reynolds numbers.

These additional studies also demonstrate the suitability of realizable  $k - \varepsilon$  model with EWT for the unmanned underwater vehicle simulations. It is obvious that turbulence intensity changes resistance results significantly. Studies performed here

serve as guidelines for determining turbulence quantities while simulating the vehicle.

## CHAPTER 5

### DRAG CALCULATIONS

CFD simulations given in this study are performed to determine the drag forces exerted on the vehicle. The most critical criterion is the surging (forward) speed. The vehicle is required to have a surging speed of 2 m/s. Therefore, emphasis is put on drag calculations in forward motion. Later on, heaving (vertical) motion of the vehicle is simulated as well.

In the previous chapter, it was shown that realizable  $k - \varepsilon$  model with EWT gives accurate results provided that the mesh is fine enough near the wall. This approach is adopted in this part of the study.

#### 5.1 Hull Geometry

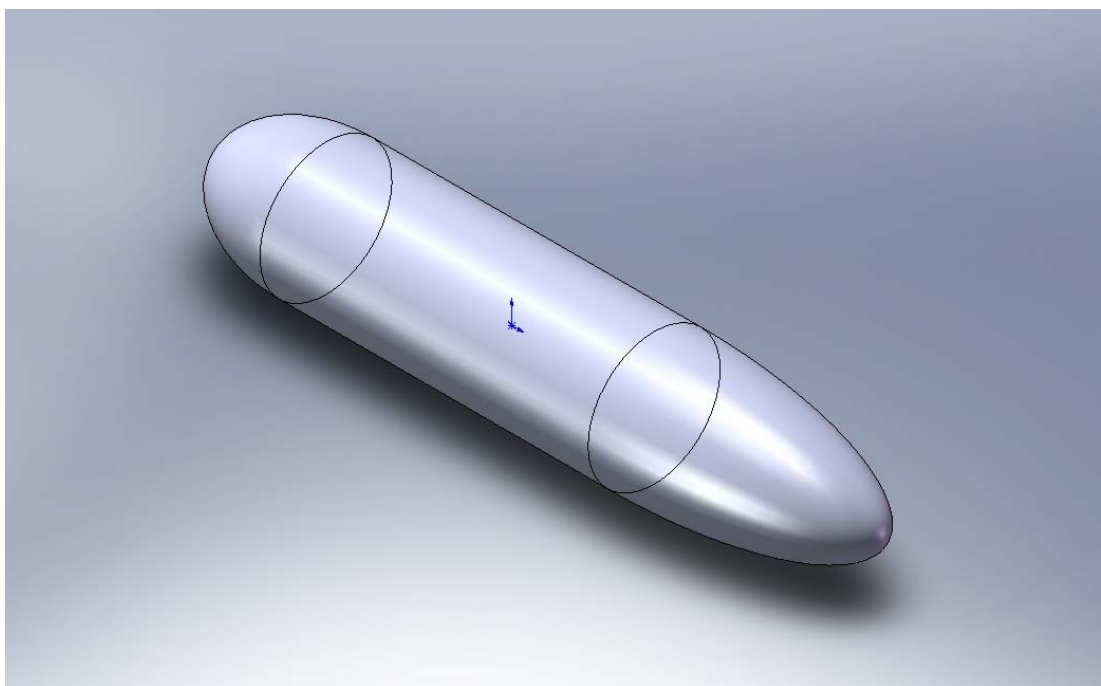
As it was stated in Chapter 2, a cylindrical hull shape with wing-like appendages is proposed. Two wing-like appendages on the sides of the hull are used for attaching the forward thrusters. There are two vertical tunnels throughout the hull inside which vertical thrusters are placed.

Apart from determining the hull shape in principle, one has to specify the dimensions of it. Dimensions are also determined in Chapter 2 during the internal design process. The hull is a streamlined body with a cylindrical mid-body and ellipsoid ends at the fore and aft. The aft ellipsoid is elongated to ensure a smooth flow over the body. During simulations, three configurations were used. These configurations are bare hull, hull with wings and full-appended hull.

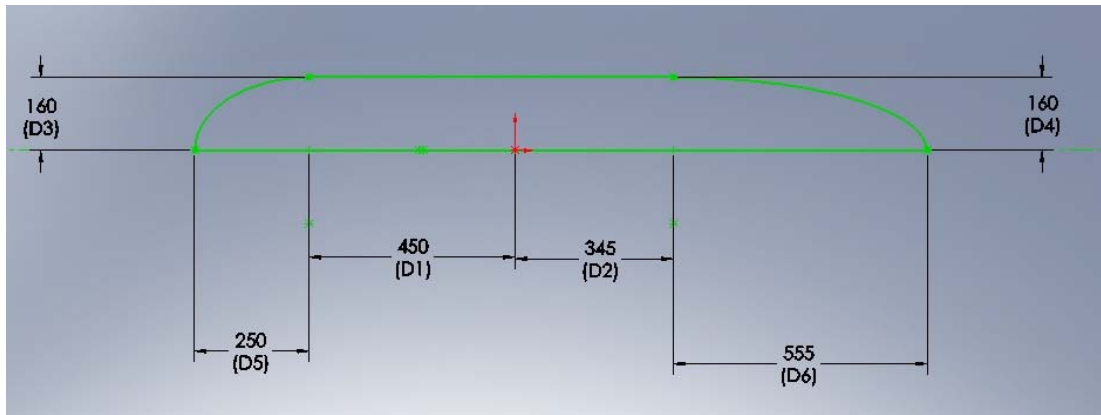


### 5.1.1 Bare Hull

Bare hull configuration consists of only the axisymmetric hull. No appendages are included in the model. Details of this model are seen in Figures 5.1 and 5.2. This configuration is used for preliminary analyses. Moreover, it is beneficial for stressing the effects of appendages.



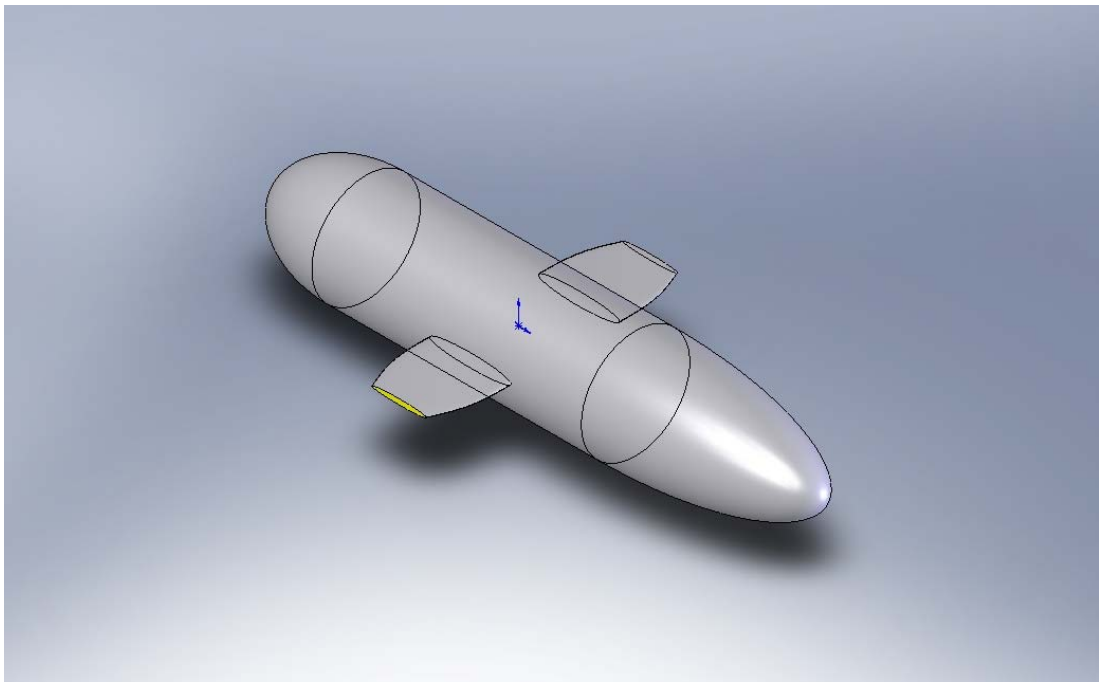
**Figure 5.1** Bare hull of the vehicle



**Figure 5.2** Dimensions of the bare hull

### 5.1.2 Hull with Wings

In this configuration side wings are added to the model as seen in Figure 5.3. The contribution of these wings on drag is seen by simulating this configuration.



**Figure 5.3** Hull with wings

### 5.1.3 Fully Appended Hull

This is the most complex model. Apart from the wings; cameras, lights, sonar, acoustic modem and vertical tunnels are added to the model. This configuration was beneficial for understanding the extra drag due to appendages thoroughly. Model can be seen in Figure 5.4.

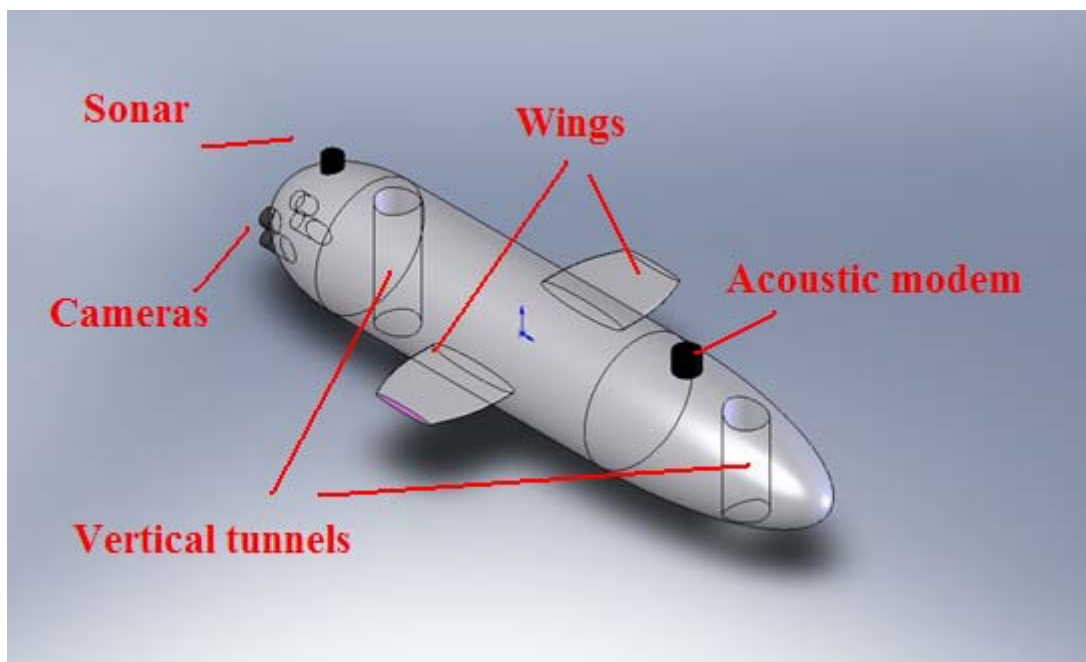


Figure 5.4 Fully Appended Hull

## 5.2 Solution

Solution of these configurations is similar to the validation studies performed before. Surface meshes are created at the surface in Gambit and 3D boundary layer was created in Tgrid. For volume meshes, tetrahedral elements were used. Length of the domain is chosen to be five times the length of the body with 3 times the length of the body left behind the trailing edge.

Green-Gauss node based gradient option was turned on and realizable  $k - \varepsilon$  turbulence model with enhanced wall treatments was used. Turbulence intensity was specified as 0.5 % at the inlet. The most important difference between the validation studies is that this time 3D solver is utilized whereas 2D axisymmetric solver was used for validations.

At this point one may question whether the flow over the vehicle is turbulent or not. It is known that laminar flow is very unstable and it is likely to be turbulent due to disturbances. For the case of an underwater vehicle, there are many disturbances. Appendages on the body, propeller-hull interaction and surface roughness are some examples. Therefore, one can infer that flow will be mostly turbulent in real conditions. Nevertheless, it is wise to check critical Reynolds numbers.

Two modes of motion are simulated in this thesis study: forward motion and vertical motion. For forward motion, the vehicle may be treated as a streamlined body and characteristic length is taken as body length (1.6 m). Reynolds number in this case becomes:

$$Re = \frac{998.2 \left(\frac{kg}{m^3}\right) \times 2 \left(\frac{m}{s}\right) \times 1.6 (m)}{0.001003 \left(\frac{kg}{m \cdot s}\right)} \cong 3.18 \times 10^6 \quad (5.1)$$

which is higher than the critical Reynolds number  $5 \times 10^5$  for a flat plate. Even though the flat plate is not a true representative of the streamlined body, still it is logical to expect turbulent flow over the vehicle. It is obvious that the Reynolds number of the vehicle is higher; hence, turbulent flow assumption is reasonable.

For the vertical motion, the vehicle may be treated as a cylinder in cross-flow; hence, bluff body motion is present. For this case characteristic length can be taken as the diameter of the vehicle (0.32 m). There is no criterion for vertical speed. A vertical speed of 0.5 m/s can be assumed as a reasonable value. In this case, Reynolds number becomes:

$$Re = \frac{998.2 \left(\frac{kg}{m^3}\right) \times 0.3 \left(\frac{m}{s}\right) \times 0.32 (m)}{0.001003 \left(\frac{kg}{ms}\right)} \cong 159234 \quad (5.2)$$

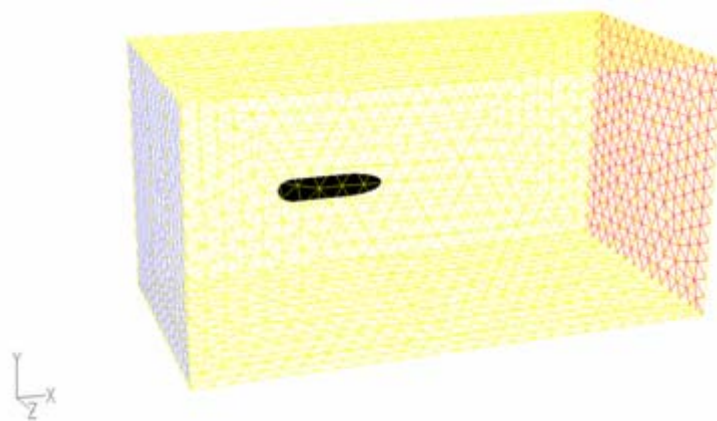
For  $Re > 2 \times 10^5$ , transition to turbulence occurs in the front part of the body. This value is close to the  $Re$  number of the body. When the effects of irregularities on the body are considered, it can be inferred that assuming turbulent flow is a reasonable choice.

## 5.2.1 Forward Motion

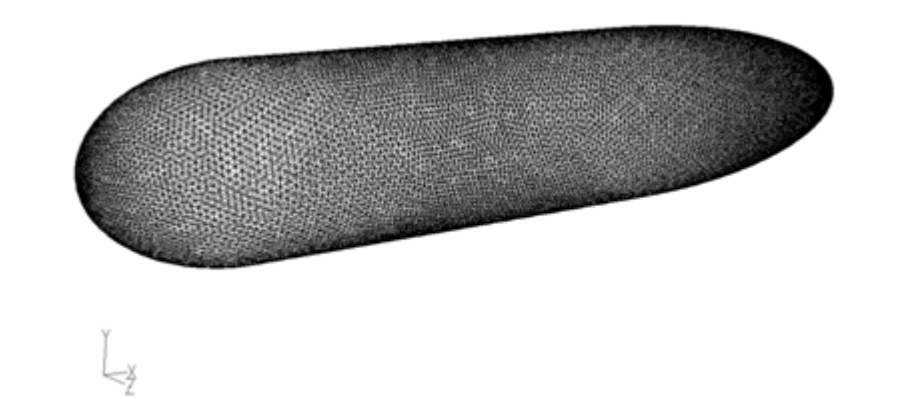
Forward motion is the most critical case because it determines the thrust and power requirements of the vehicle. Forward motion is analyzed for all three configurations.

### 5.2.1.1 Forward Motion with Bare Hull Configuration

The purpose of this simulation is to determine the drag for the single bare hull. The flow domain and the mesh on the model are seen in Figures 5.5 and 5.6, respectively.

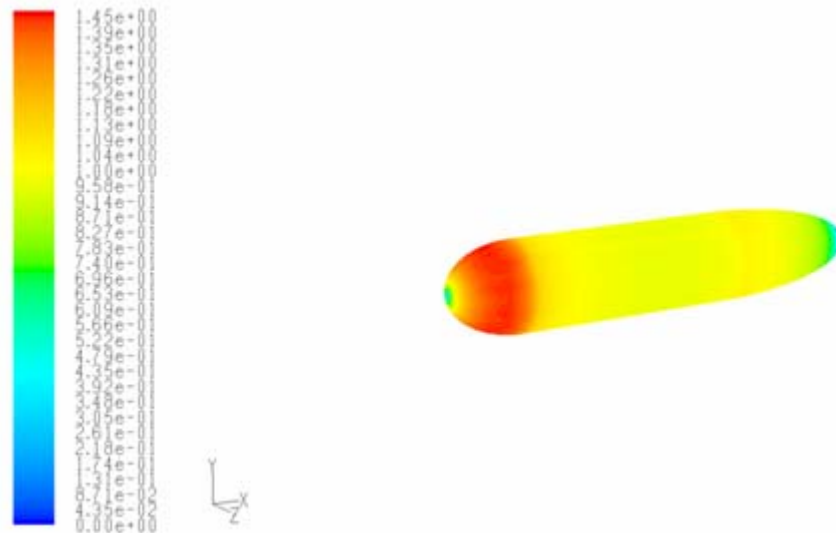


**Figure 5.5** Flow domain for bare hull configuration in forward motion

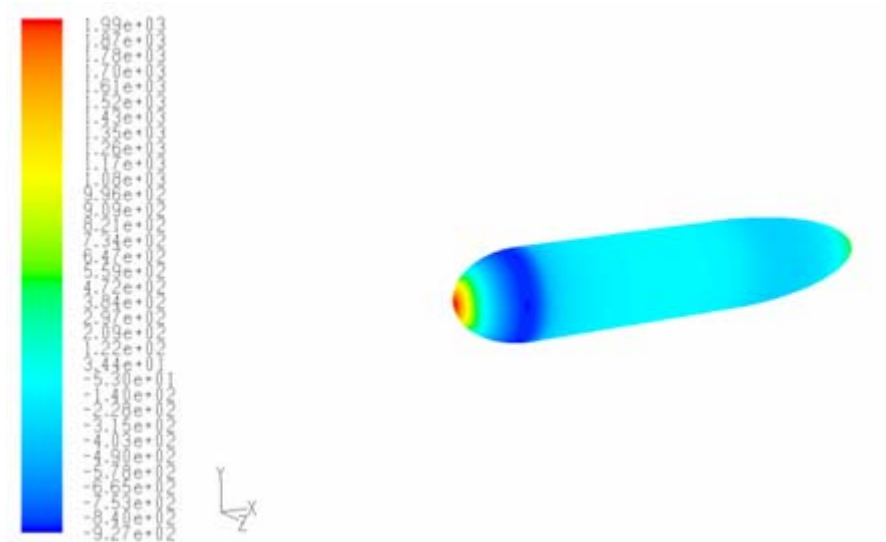


**Figure 5.6** Surface mesh for bare hull configuration in forward motion

It should be noted that the aim is to obtain a fine mesh on the surface so that the wall  $y^+$  value is around 1 throughout the surface. While making the initial guess for the first cell height, experience from the validations studies and preliminary 2D studies are used. The wall  $y^+$  contours on the body, as seen in Figure 5.7 show that the mesh is fine enough for enhanced wall treatment. Moreover, for EWT it is suggested to put at least 10 elements in the boundary layer until  $Re_y > 200$ . This condition is also satisfied. Pressure contours on the body can be seen in Figure 5.8.



**Figure 5.7** Wall  $y^+$  contours for bare hull configuration in forward motion



**Figure 5.8** Pressure contours for bare hull configuration in forward motion

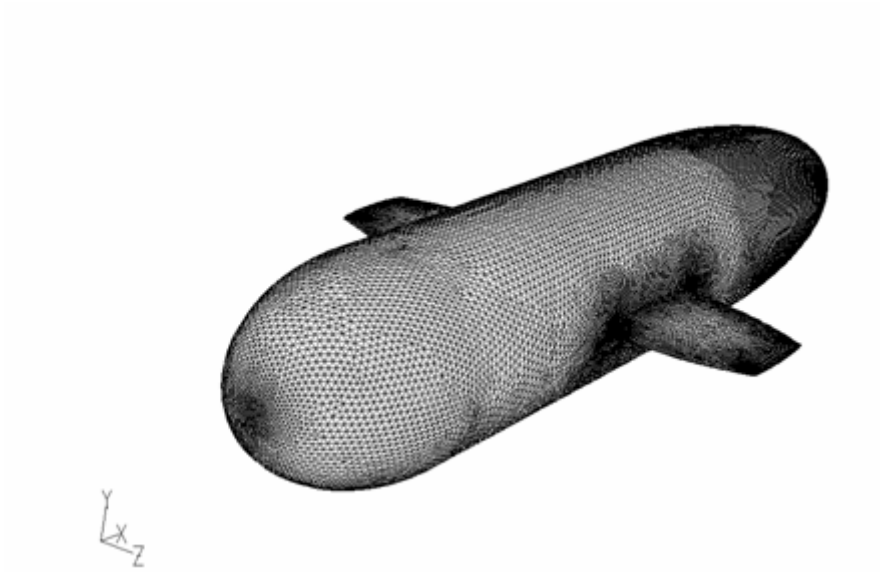
The total resistance force on the body is seen in Table 5.1.

**Table 5.1** Drag components on bare hull for forward motion

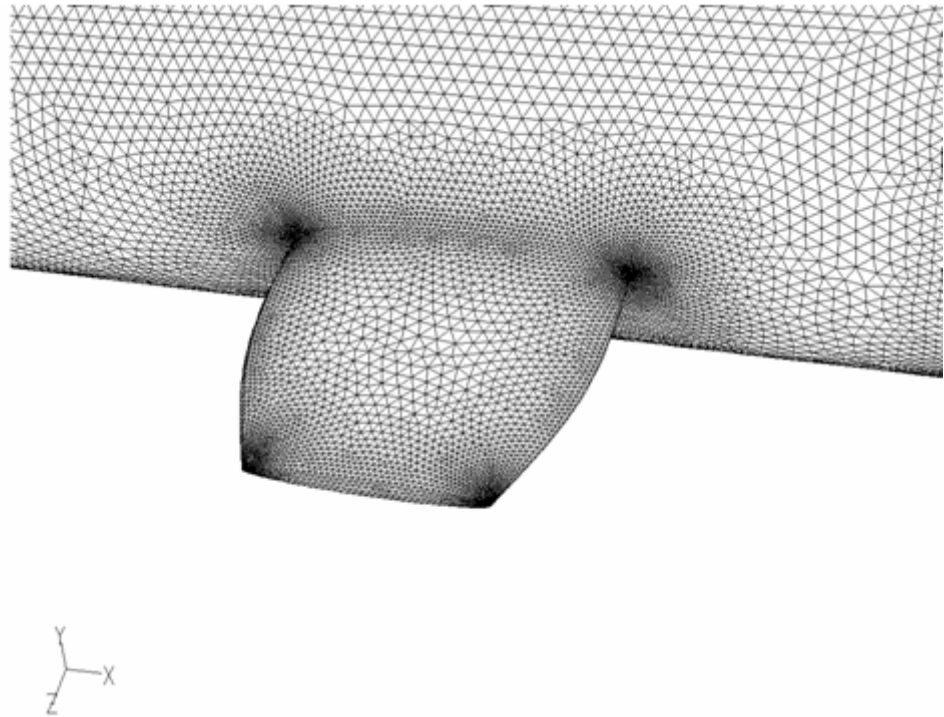
	Friction drag (N)	Pressure drag (N)	Total drag (N)
Bare hull (total)	3.19	10.83	14.02

### 5.2.1.2 Forward Motion for Hull with Wings Configuration

This time the effect of the wing appendages is also seen. The same domain with the bare hull configuration is used. Mesh details can be seen in Figures 5.9 and 5.10.  $y^+$  values are again satisfactory and can be seen in Figure 5.11. Pressure distribution is seen in Figure 5.12.

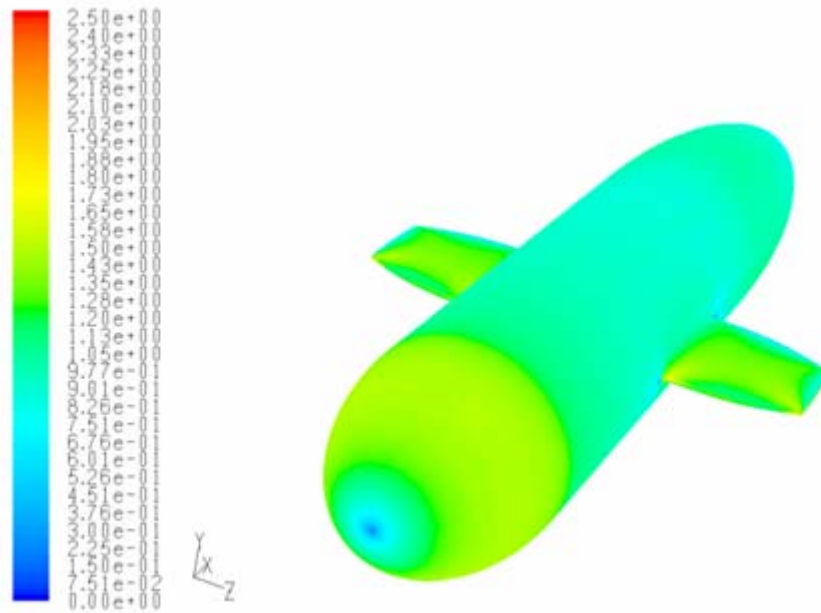


**Figure 5.9** Surface mesh for hull-wings configuration in forward motion

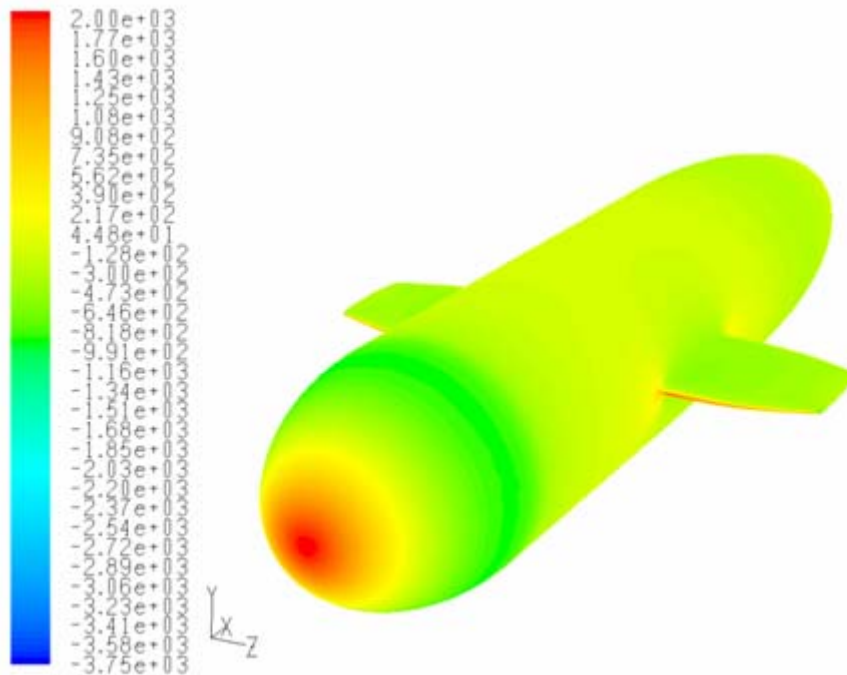


**Figure 5.10** Mesh details for hull-wings configuration in forward motion





**Figure 5.11** Wall  $y^+$  contours for hull-wings configuration in forward motion



**Figure 5.12** Pressure contours for bare hull configuration in forward motion

The total resistance force on the body is seen in Table 3.3.

**Table 5.2** Drag components for forward motion for hull-wings configuration

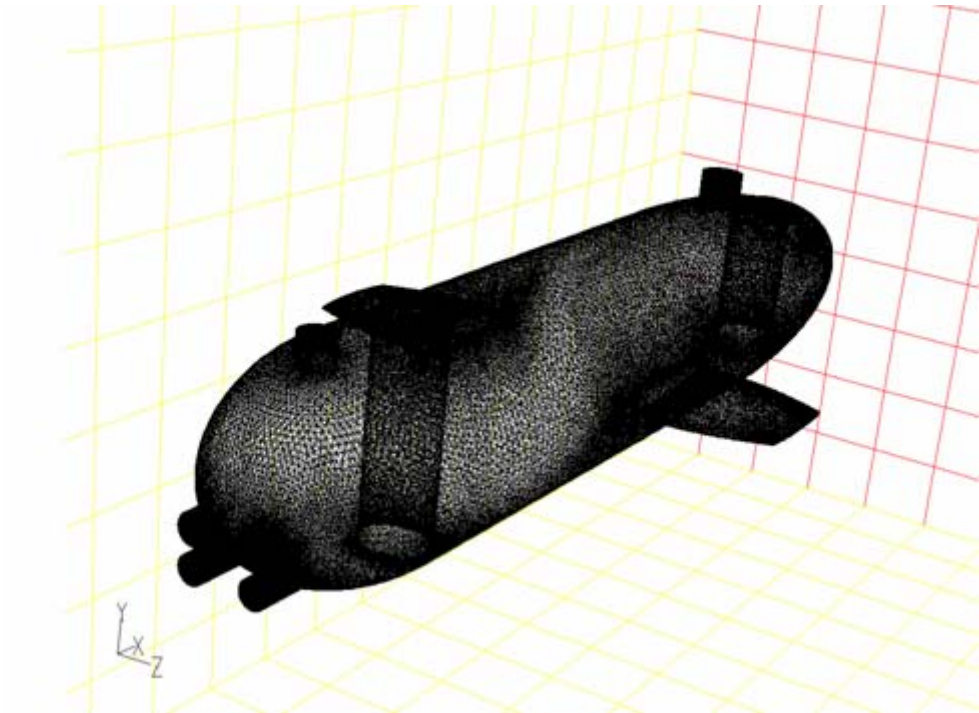
	Friction drag (N)	Pressure drag (N)	Total drag (N)
Hull	9.85	3.32	13.18
Wings	2.26	1.41	3.66
Total	12.11	4.73	16.84

Friction drag is a surface property; therefore, it is expected to have higher friction drag for hull which has a larger surface area. On the other hand, even though these wings are thin, they still produce comparable amount of pressure drag (about 30% percent).

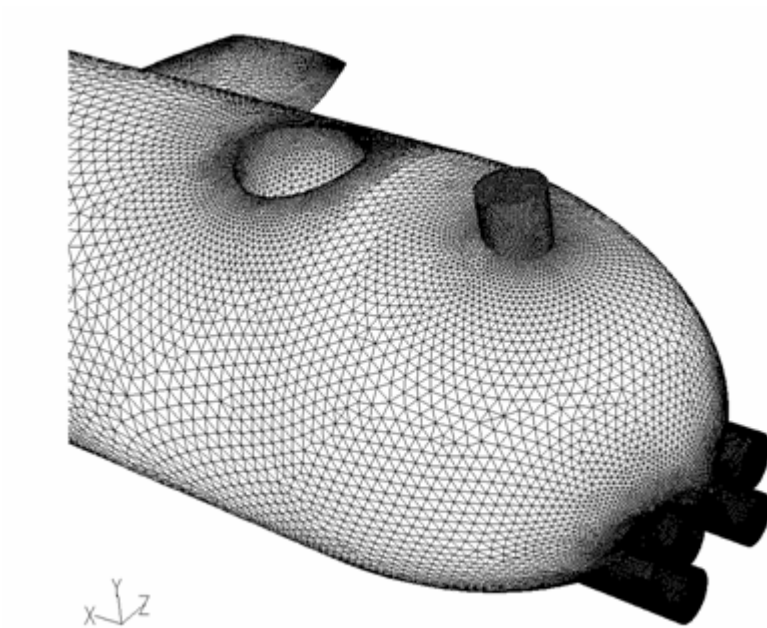
### **5.2.1.3 Forward Motion for Fully Appended Hull Configuration**

In this configuration, almost all of the appendages are included in the model: sonar, modem, tunnels, cameras and wings. The increase in the complexity of the model results in a large number of grid points and eventually longer computation time. Moreover, the complexity makes it difficult to generate boundary layer mesh at the surface. Fortunately, Tgrid provides easy and efficient means of boundary layer and volume mesh generation.

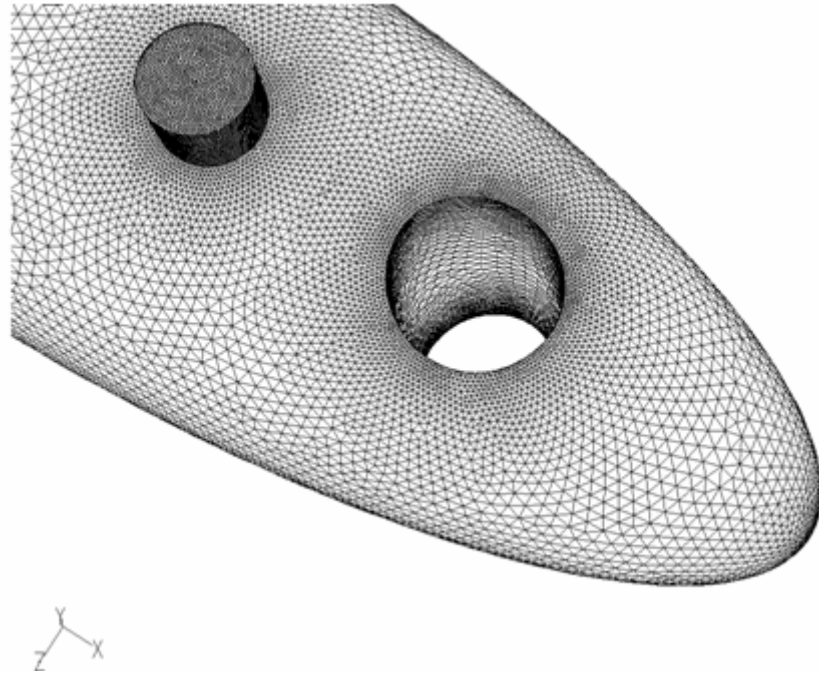
The domain, boundary conditions and solution parameters are chosen the same for the previous two models. The surface mesh can be seen in Figures 5.13 to 5.15.



**Figure 5.13** Mesh for fully appended configuration in forward motion – 1

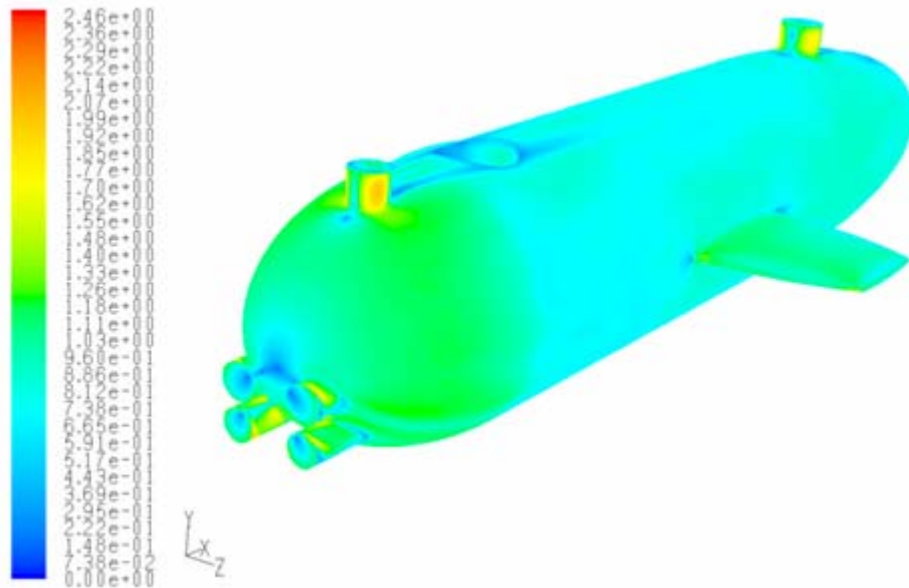


**Figure 5.14** Mesh for fully appended configuration in forward motion – 2



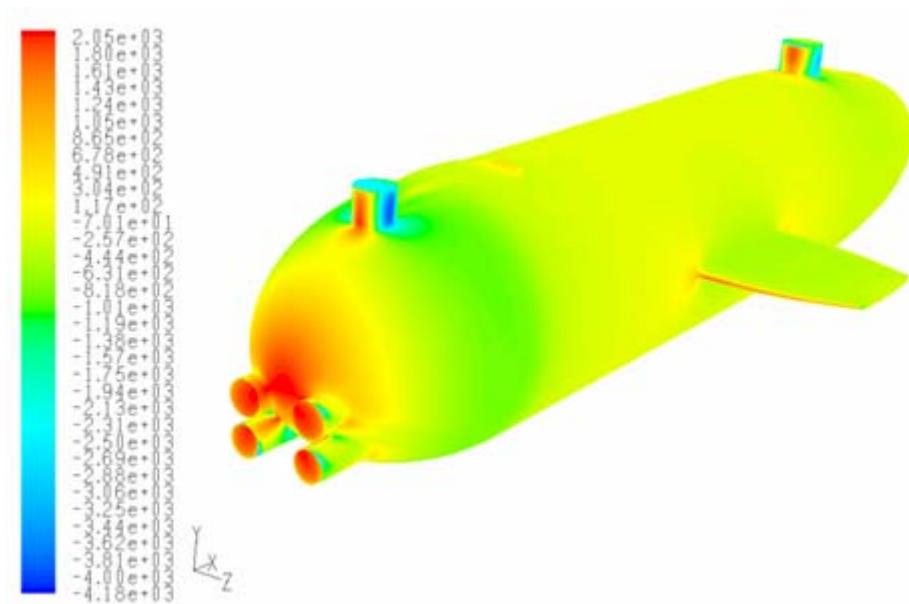
**Figure 5.15** Mesh for fully appended configuration in forward motion – 3

The contours of wall  $y^+$  on the body are seen in Figure 5.16. Only for a few accelerating regions around appendages, the  $y^+$  value exceeds 2. Other than that, wall  $y^+$  is mostly around a value of 1 which makes the mesh suitable for enhanced wall treatment.

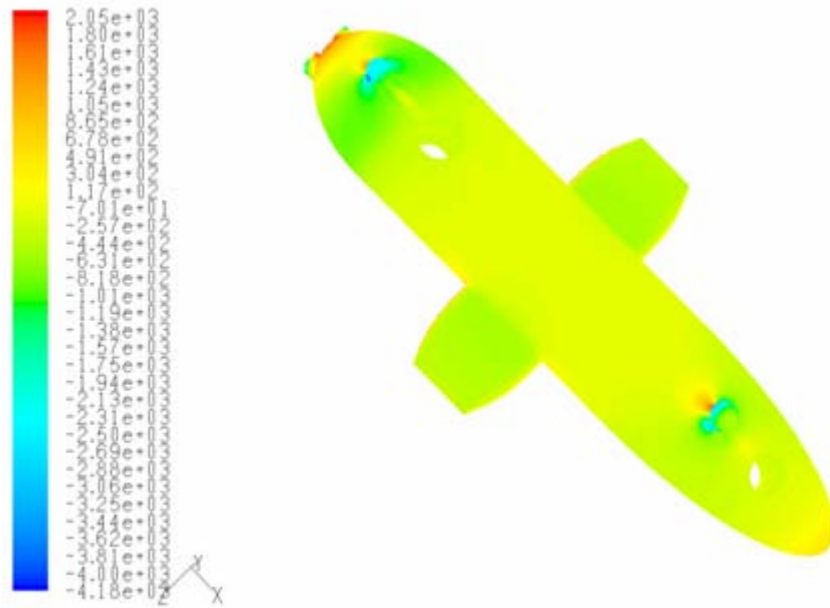


**Figure 5.16** Wall  $y^+$  contours for fully appended configuration

In Figures 5.17 and 5.18, static pressure contours are seen.



**Figure 5.17** Pressure contours for fully appended configuration - 1



**Figure 5.18** Pressure contours for fully appended configuration - 1

Total resistance on the body is seen in Table 5.3.

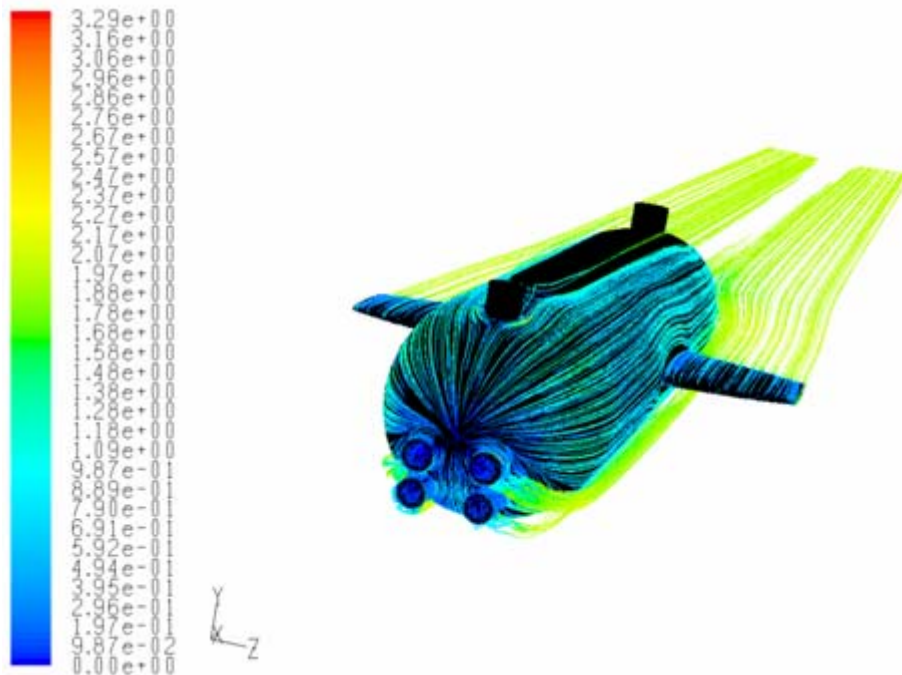
**Table 5.3** Drag components for forward motion

	Friction drag (N)	Pressure drag (N)	Total drag (N)	Percentage (%)
Modem	0.09	2.81	2.90	9
Sonar	0.06	1.46	1.52	4.7
<b>Camera</b>	<b>0.06</b>	<b>9.19</b>	<b>9.26</b>	<b>28.8</b>
Wings	2.18	1.64	3.82	11.9
Tunnel – aft	-	1.56	1.56	4.8
Tunnel – forward	-	1.00	0.99	3.1
Hull	9.56	2.47	12.03	37.5
Total	11.95	20.13	32.08	

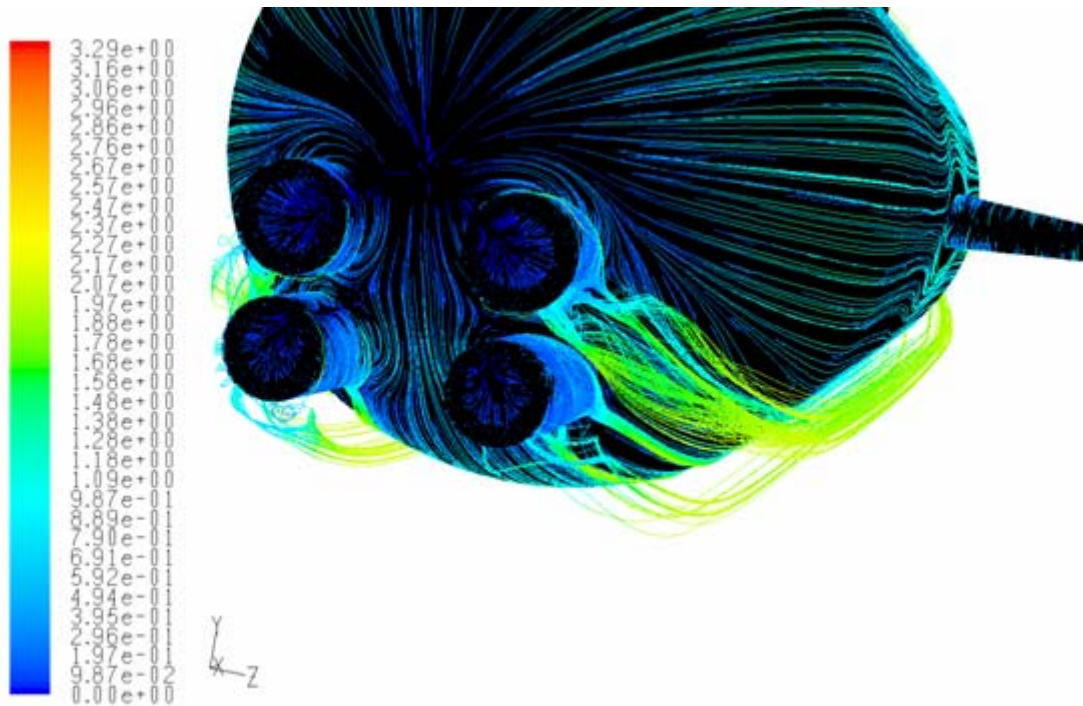
Even though extra drag due to appendages may seem insignificant, the cumulative effects of these appendages cause a significant increase in total drag force. Total drag is 14.02 N for bare hull, 16.84 for hull with wings and 32.07 N for the fully appended configuration which means that the drag value is twice as high as the bare hull case.

This increase is not only due to the extra drag of appendages, but also due to the distortion of the flow field. Therefore; one should pay extra attention to these protrusions and streamline them if possible.

Pathlines of the flow also provide interesting data for the designer. The appendages which distort the flow of the flow can be detected and modified if possible. For instance when Table 5.3 is examined, it is seen that the drag increase due to cameras is very high compared to other appendages. When pathlines which are given in Figures 5.19 and 5.20 are observed, it is seen that the protrusion of cameras are very significant and this results in separated flow and increased drag.



**Figure 5.19** Pathlines for forward motion - 1

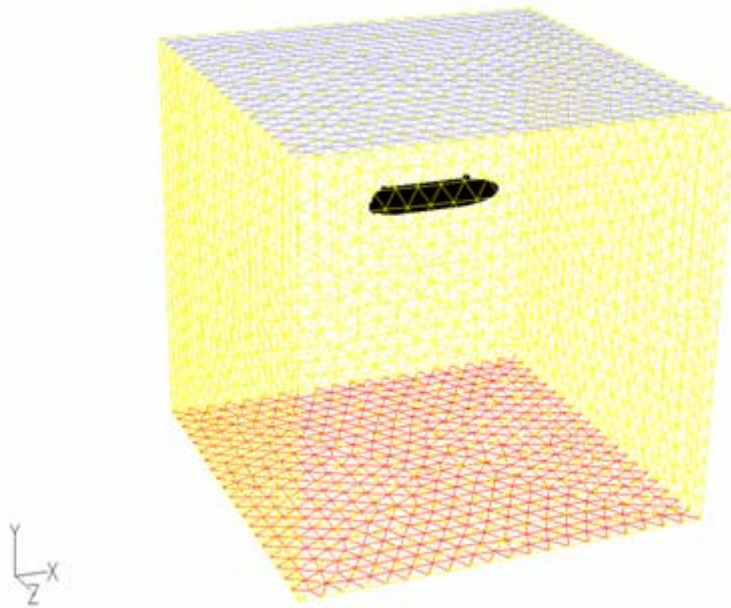


**Figure 5.20** Pathlines for forward motion – 2

## 5.2.2 Vertical Motion

Vertical motion is achieved with two vertical thrusters that are placed inside a tunnel that passes throughout the body. In the project definition, no criteria were defined for vertical motion. Nevertheless, it is important to verify that the vehicle will at least be able to surface at a steady speed. For this purpose a vertical speed of 0.5 m/s is selected. The mesh which gives good resolution for forward motion is used again. This time the domain changes, however, a cube with a length of 5 m is chosen. The domain can be seen in Figure 5.21.

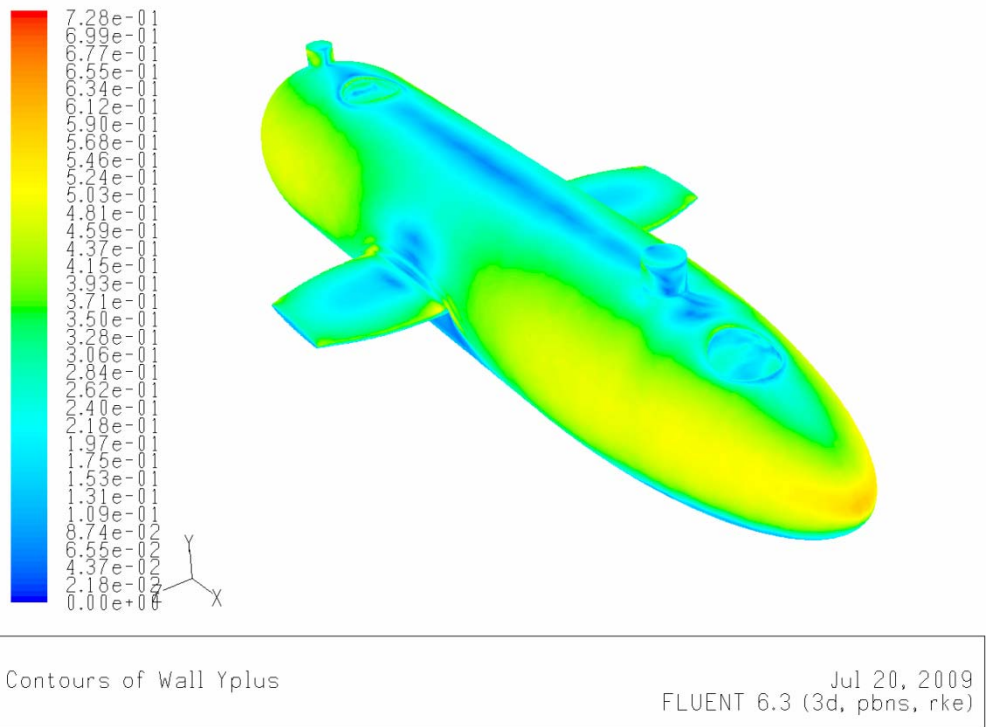




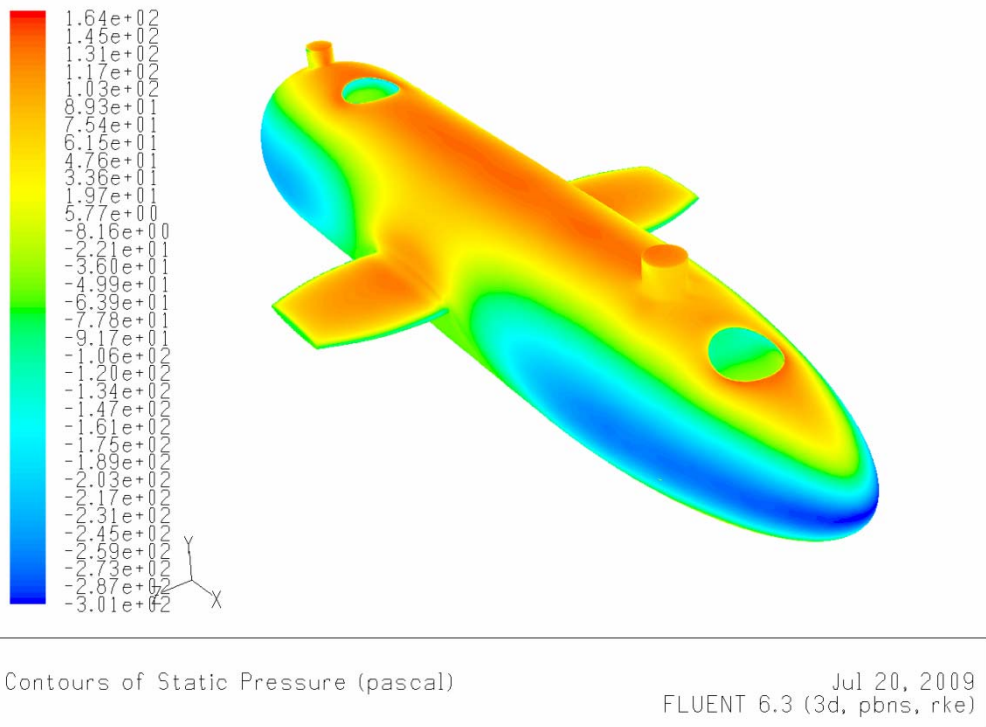
**Figure 5.21** The domain used for vertical motion simulations

Solution parameters are similar to the previous simulations. Simulations are done by using the hull-wing configuration and the fully-appended configuration. Pressure, wall  $y^+$  and pathline graphs of only fully-appended region will be provided here. Mesh details will not be given also since the same mesh for the forward motion is used.

Wall  $y^+$  contours on the body are given in Figure 5.22. As seen from the figure, wall  $y^+$  values are less than 1 all over the wall and mesh resolution is satisfactory. Pressure distribution on the body is seen in Figure 5.23.

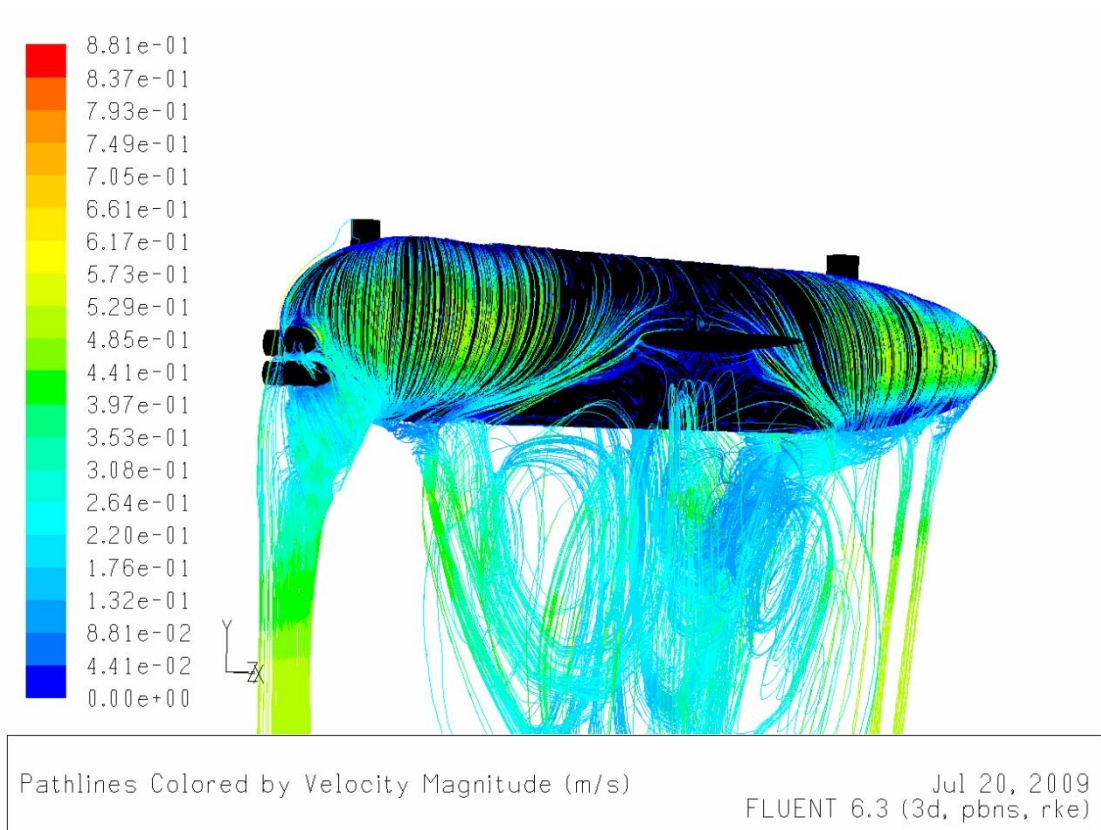


**Figure 5.22** Wall  $y^+$  contours for fully appended configuration

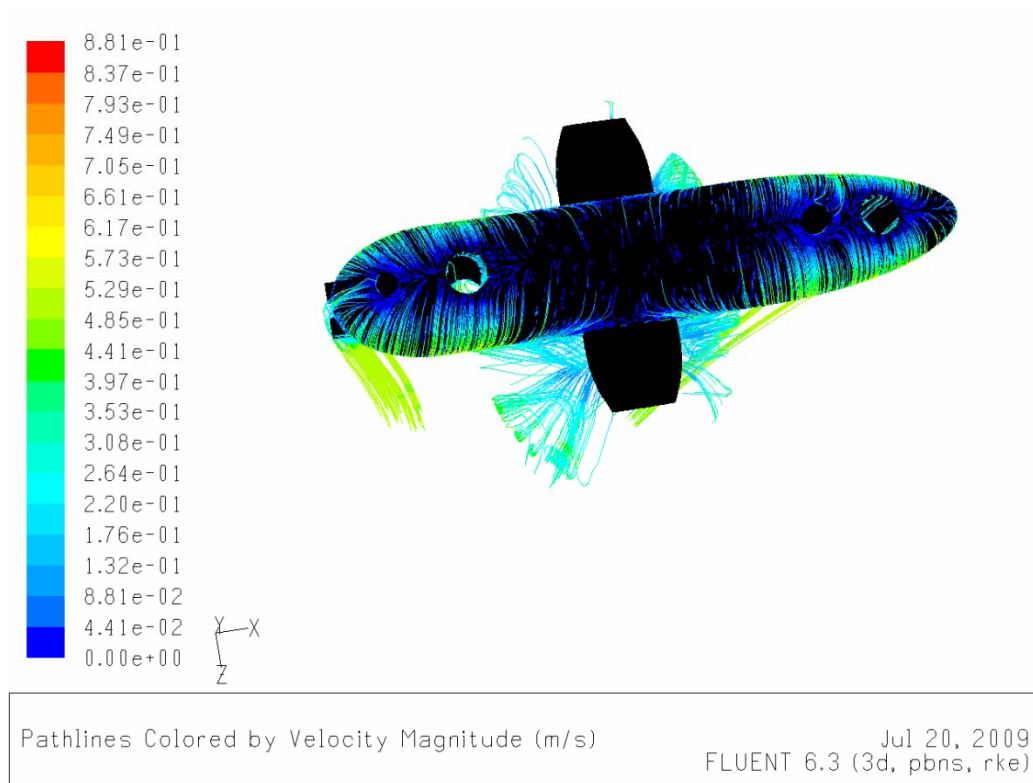


**Figure 5.23** Pressure contours for fully appended configuration

Finally, pathlines for vertical motion are seen in Figures 5.24 and 5.25. From these figures, it is seen that appendages like cameras and wings distort the flow and cause separation. On the other hand, wings are useful for supporting the thrusters, providing rolling stability and creating a moment arm for forward thrusters. Therefore, the choice depends on the needs and priorities of the project group.



**Figure 5.24** Pathlines for vertical motion - 1



**Figure 5.25** Pathlines for vertical motion – 2

Resistance predictions of hull-wings configuration and fully-appended configuration are almost equal. The resulting drag is about 53 N.

### 5.3 Thruster Selection

After determining the thrust requirements for forward and vertical motion, thruster selection can be done. For the 2 m/s forward motion of the fully-appended body, a resistance of 32 N is expected. BTD150 thrusters of Seabotix can give a continual bollard thrust of 2.2 kgf (21.6 N). Since two thrusters in parallel will be used for forward and vertical motion, a total of 43.2 N thrust is achieved which is probably sufficient for the speed criteria. BTD150 is a popular thruster for unmanned underwater vehicles. It is a podded propulsor so there is no need for sealed shafts to drive the motor. It should be noted that the demands of the electronics engineers in the group were also taken into account while selecting the thruster.

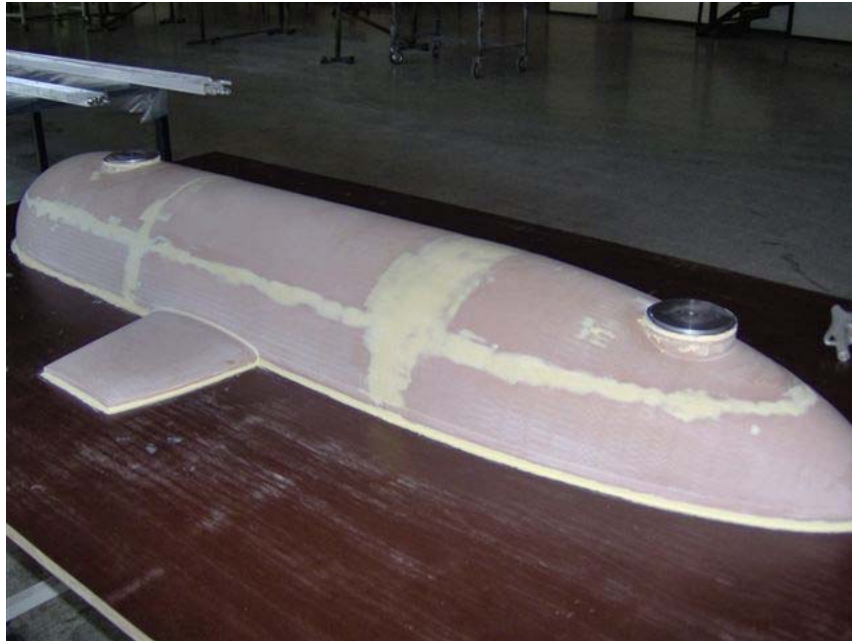
## **CHAPTER 6**

### **MANUFACTURING AND TESTING**

In this chapter of the thesis, information about the manufacturing and testing of the vehicle is provided. It should be noted that Barış Elektrik Endüstrisi Inc. is responsible from the manufacturing of the vehicle using composite materials; therefore, most of the efforts mentioned here are performed by them. Decisions such as determining the hull thickness, designing the mounting interface of devices and modifying the hull is done by them. Due to that reason, only an overview of the manufacturing process will be presented in this thesis. At this stage, only preliminary tests are performed; therefore, brief information is provided for tests as well.

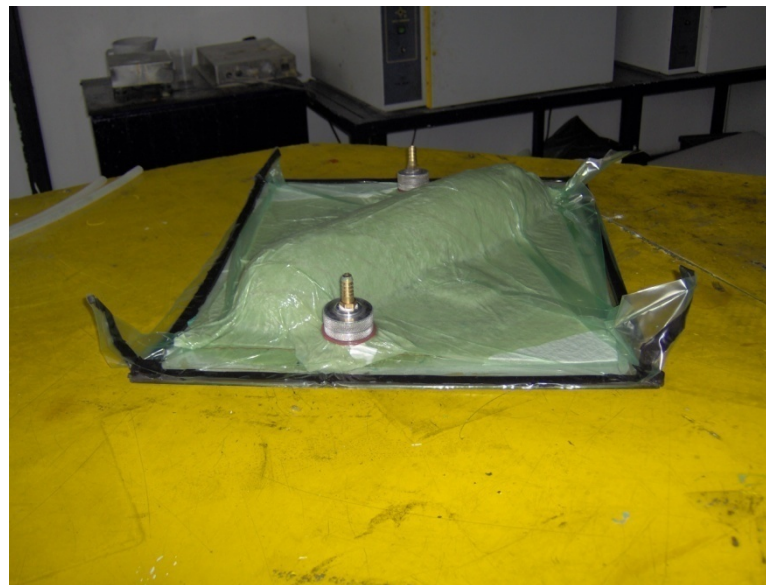
#### **6.1 Manufacturing**

The manufacturing process consists of three steps: namely, manufacturing the model, manufacturing the mold and manufacturing the body. First of all a model that has the same shape of the body is manufactured with CNC machine. The model that was produced is seen in Figure 6.1. The cylindrical protrusions are used for the vertical tunnels later on.



**Figure 6.1** Model of the vehicle

The actual mold is obtained by using the model as the mold. The mold will be produced from glass-epoxy with vacuum bagging process. Lamination is done on the model as seen in Figure 6.2. The mold is the female counter-part of the vehicle and is seen in Figure 6.3.



**Figure 6.2** Mold of the vehicle (during production)

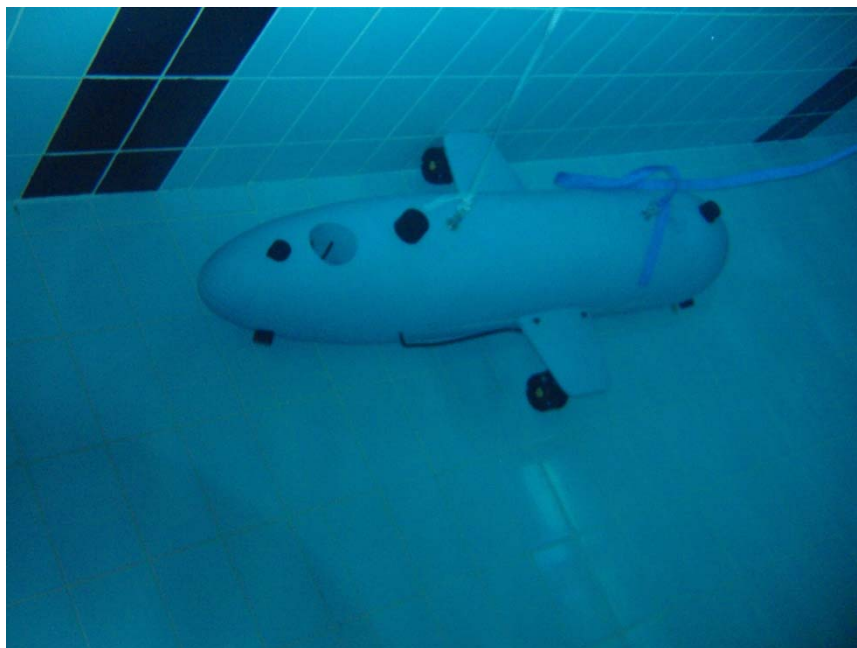


**Figure 6.3** Mold of the vehicle (finished product)

This mold is used for producing the vehicle body. Similar to the previous process, fibers are laid on the mold, vacuum is applied and curing is done. Laying up the fibers is seen in Figure 6.4. After these steps, the raw vehicle body is obtained and it is processed to achieve the final design. Some of these processes are obtaining the walls of the pressure hull, fitting underwater connectors to those walls, piercing the necessary holes, providing a sealing system at the interface, manufacturing the mounting hardware of devices, placing the draw latches on the body, surface operations and painting. The final product is seen in Figure 6.5.



**Figure 6.4** Hand lay-up process for the vehicle body



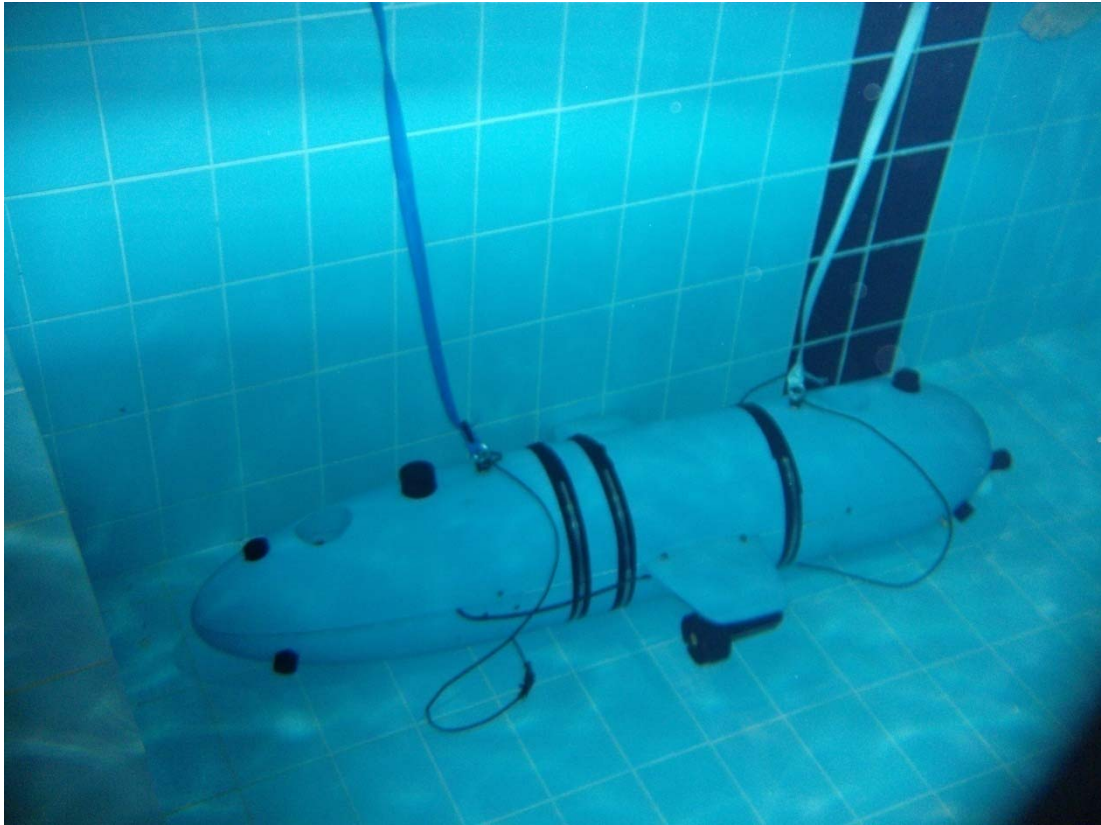
**Figure 6.5** Vehicle body



## 6.2 Tests

Testing the vehicle is the final step in the project. In the scope of this project, tests that are or will be applied are leakage tests, heating tests, deep submergence tests and performance tests. It is obvious that before beginning performance tests, one should conduct leakage, heating and deep submergence tests and receive successful results.

The leakage test is a difficult step. Two types of leakage tests were performed. In the first one, vacuum is applied to the pressure hull and the vehicle is submerged in water. Vacuum is applied in order to simulate the effect of increased depth. 0.2 and 0.8 bars of vacuum is applied which correspond to water depths of 2 and 8 meters respectively. Initial trials were not successful. However, it was seen that leakage was lower at higher values of vacuum. It was assumed that the problem was insufficient compressive force provided by draw latches. As a result, additional latches were assembled to the vehicle and tests were conducted again. Leakage was reduced but still present. A final test was performed. This time four clamps were attached to the vehicle from the outside as seen in Figure 4.32 and the vehicle was submerged into the pool. The aim was to increase the compression force. The clamps proved to be helpful and no leakage was detected. It should be noted that these tests are performed for shallow water and the system has to be tested at deep submergence up to 100 meters.



**Figure 6.6** Leakage test at the pool

Initial heating tests were performed during the design step and results are presented in the previous part. In these tests a representative hull was used along with lamp as the heat source. Those results claim that no over-heating will occur inside the hull. However, it is important conduct these tests with actual system.

At present, leakage tests are successfully completed at the pool. Heating tests are being performed and their results will be received soon. Deep submergence test at sea and performance tests to check the maximum speed of the vehicle should be the next steps in the project.

## CHAPTER 7

### CONCLUSIONS AND DISCUSSIONS

#### 7.1 Conclusions

The aim of this thesis study is to present the performance prediction simulations that were performed during the design of an unmanned underwater vehicle. First of all conceptual design of the hull is explained. Different hull shape examples and propulsion systems are investigated and appropriate selections are made. The dimensions of the hull are determined from the internal arrangement of devices. Besides the internal arrangement, it is decided to employ a floodable hull. Glass-epoxy is chosen as the hull material due to its high strength-to-weight ratio, ease of molding and high corrosion resistance. Vacuum bagging is chosen as the manufacturing method since it provides ease of manufacturing with relatively low cost compared to other RTM methods.

After the conceptual design of the hull, performance prediction analyses are done. The first set of analyses is about the heating problem of the watertight middle section (pressure hull). This section houses electronic equipments and has thicker walls to withstand the high hydrostatic pressure in deep submergence. Furthermore, glass-epoxy is known to be a poor thermal conductor. When the effects of thick walls and high thermal resistance of glass-epoxy are considered, it is inferred that the heat dissipation may not be rejected from the pressure hull easily and the temperature inside the hull may increase. This increase should be avoided for the proper operation of electronic equipments. FLUENT is used for analyzing this heating problem. Due to

computational resources, only two-dimensional unsteady analyses of the hull are performed for 2 hours (the expected operation time of the vehicle). Two different sections are analyzed with FLUENT. The results of these analyses revealed that the maximum temperature in the hull is below the maximum allowable value. Heating tests are also performed to check the validity of the analyses. A glass-epoxy pressure hull with a lamp inside as the heat source is immersed in water. A number of temperature sensors are placed in the pressure hull to record the temperature against time. The results of this test give close results to analyses. Results of the analyses and tests are presented in Chapter 3.

The second set of performance prediction analyses are about the forward and vertical motions of the proposed hull design. The most important speed requirement is the forward speed which is 2 m/s. Before analyzing whether the hull shapes satisfies this limitation, some validation studies are performed in Chapter 4. Series 58, which is a series of axisymmetric streamlined bodies, is chosen as the experimental reference. Three models of Series 58; namely, Model 4159, 4158 and 4154 are chosen for investigation in this thesis. Different mesh configurations with varying mesh densities in the boundary layer and along the body are generated for Model 4159. These mesh configurations are analyzed with different turbulence models and near wall modeling approaches. Moreover, the mesh is adapted to suitable wall  $y^+$  values depending on the mesh requirements of the turbulence model being used. The variation of total resistance coefficient in a Reynolds number range between  $2 \times 10^6$  and  $2.5 \times 10^7$  is obtained and compared with experimental data. Results revealed that realizable  $k - \epsilon$  with EWT gives accurate results over the range of Reynolds numbers provided that the mesh is fine enough with wall  $y^+$  value around 1. The same conclusions are drawn from the analyses of Models 4158 and 4154 as well. Therefore, realizable  $k - \epsilon$  turbulence model with EWT is used for the rest of the studies. Results of the validation studies are given in Chapter 4.

Forward motion of the vehicle at 2 m/s is analyzed using the realizable  $k - \epsilon$  turbulence model with EWT. Three different body configurations with different

appendages are used. The aim is to understand the effect of appendages on total drag force. Same body configurations are also analyzed for the vertical motion with a velocity of 0.5 m/s. Drag force values that are obtained from these analyses are used for the selection of an appropriate thruster for the vehicle. Results of forward and vertical motion analyses are given in Chapter 5.

In Chapter 6, manufacturing of the vehicle is given. It should be noted that manufacturing is done by a company; therefore, only an overview of the process is provided.

## **7.2 Future Work Recommendations**

Studies performed in this thesis provide many beneficial recommendations for future unmanned underwater vehicle design projects.

Employing a floodable hull is beneficial in many aspects. First of all the watertight section is limited to a smaller volume which reduces the amount of sealing surfaces and the risk of leakage. Secondly, floodable sections provide room for equipments like pressure transducer which need to operate in water. Finally, the floodable hull increases the wetted surface area of the pressure hull and increases its heat rejection capabilities. As a result, the maximum temperature in the pressure hull reduces. This effect is demonstrated in Chapter 3 where two different models are used for heat transfer analyses. Second model includes the effect of flood water; hence, the maximum temperature in the pressure hull is found to be lower.

Even though preliminary heat transfer analyses performed in this thesis provide beneficial information, it is hard to rely on these data due to certain reasons. First of all, they are two-dimensional. Secondly, electronic equipments are not modeled in detail; instead, only simplified models are used. It may be a better choice to make use of three-dimensional analyses where components are modeled with more detail. Commercial programs like ICEPAK may also be useful for such analyses.

Although simple two-equation models like realizable  $k - \varepsilon$  with EWT and  $k - \omega$  SST provide accurate results for simple flows like the flow over a bare hull, their accuracy is questionable for more complex flows like the flow over the vehicle with appendages. Therefore, utilization of more complex turbulence models like Reynolds Stress Model (RSM) or Large Eddy Simulation (LES) approach should be considered.

The protrusions out of the body contribute to the total drag of the vehicle considerably as shown in Chapter 4. The largest contribution is from cameras in that they have significant extensions out of the body in the front end of the vehicle. High pressure values are experienced at this region due to stagnation of flow. Moreover, cameras disturb the flow in this region and cause separation as seen from pathlines of the flow. The combined effect of high pressure and separation cause a significant contribution from cameras. Locating cameras inside the hull is a remedy to this problem. The nose of the hull can be made to be transparent so that the cameras can work in the hull.

For vehicle design projects with higher demands on movement capabilities, it will be necessary to perform additional analyses. Maneuvering of the vehicle should be investigated for such cases. Developing the dynamic model of the vehicle is usually performed to solve this problem. Adding thrusters into the computational model is also an important issue. Thrusters affect the flow field and surface pressure distribution on the vehicle. Consequently, drag force values change considerable. It should be noted that thruster performance is also affected by the flow field. Due to that modeling the vehicle with thrusters helps the designer to investigate the performance of thrusters and enable to adjust thruster configuration.

## REFERENCES

- [1] Blidberg, D. R., *The Development of Autonomous Underwater Vehicles (AUVs); A Brief Summary*, IEEE International Conference on Robotics and Automation, Seoul, Korea, 2004.
- [2] Langebrake, L. C. (2003), *AUV Sensors for Marine Research*, In: G. Griffiths (ed.), *Technology and Applications of Autonomous Underwater Vehicles*, CRC Press, p:245-246.
- [3] Wernli, R. L., Christ, R. D., *The ROV Manual: A User Guide for Observation Class Remotely Operated Vehicles*, Elsevier Butterworth-Heinemann, 2007.
- [4] Jalbert, J., Baker, J., Duchesney, J., Pietryka, P., Dalton, W., D.R. Blidberg, Chappell, S. G., Nitzel, R., Holappa, K., *Solar-Powered Autonomous Underwater Vehicle Development*, Proceedings of the 13<sup>th</sup> International Symposium on Unmanned Untethered Submersible Technology, Durham, NH, 2003.
- [5] Hyakudome, T., Aoki, T., Tsukioka, S., Yoshida, H., Ishibashi, S., Inada, T., Kabeno, T., Maeda, T., Hirokawa, K., Yokoyama, K., Tani, T., Sasamoto, R., Nasuno, Y., *Fuel Cell Underwater Vehicle URASHIMA*, 14th International Offshore and Polar Engineering Conference, Toulon, FRANCE, May 23-28, 2004.
- [6] Küçük, K., *Modeling and Motion Simulation of an Underwater Vehicle*, MS Thesis in the Middle East Technical University, 2007.

- [7] Wernli, R., *AUV Commercialization – Who’s Leading the Pack?*, Oceans 2000, Providence, RI, 2000.
- [8] Chance, T. S., Kleiner, A. A., Northcutt, J. G., *Reducing Deepwater Survey Costs: The Impact of Contrasting Technologies*, C & C Technologies, Inc.
- [9] Department of the Navy (USA), *The Navy Unmanned Undersea Vehicle (UUV) Master Plan*, November 2004.
- [10] Jolie, E. W., *A Brief History of U.S. Navy Torpedo Development*, NUSC Technical Document 5436, September 1978.
- [11] Madhan, R., Desa, E., Prabhudesai, S., Sebastião, L., Pascoal, A., Desa, E., Mascarenhas, A., Maurya, P., Navelkar, G., Afzulpurkar, S., Khalap, S., *Mechanical Design and Development Aspects of a Small AUV-MAYA*, IFAC MCMC2006, Conference on Manoeuvring and Control of Marine Craft, Lisbon, Portugal, 2006.
- [12] Bowen, A. D., Yoerger, D. R., Taylor, C., McCabe, R., Howland, J., Gomez-Ibanez, D., Kinsey, J. C., Heintz, M., McDonald, G., Peters, D. B., Fletcher, B., Young, C., Buescher, J., Whitcomb, L. L., Martin, S. C., Webster, S. E., Jakuba, M. V., *The Nereus Hybrid Underwater Robotic Vehicle for Global Ocean Science Operations to 11,000m Depth*, OCEANS 2008, Vols. 1-4, p:1281-1290, 2008.
- [13] Dasset, S., Damus, R., Hover, F., Morash, J., Polidoro, V., *Closer to Deep Underwater Science with ODYSSEY IV Class Hovering Autonomous Underwater Vehicle (HAUV)*, IEEE Oceans 2005 Europe, Brest, FRANCE, 2005.



- [14] Ohata, S., Ishii, K., Sakai, H., Tanaka, T., Ura, T., *Development of an Autonomous Underwater Vehicle for Observation of Underwater Structures*, Oceans 2005, Vol. 3, p:1928-1933, 2005.
- [15] Georgiadesa, C., Nahon, M., Buehlera, M., *Simulation of an Underwater Hexapod Robot*, Ocean Engineering, Vol. 36, Issue 1, p:39-47, January 2009.
- [16] Joubert, P., *Some aspects of submarine design Part 1. Hydrodynamics*, Defence Science and Technology Organisation, Technical Report, 2004.
- [17] Gertler, M., *Resistance experiments on a systematic series of streamlined bodies of revolution for application to the design of high-speed submarines*, DTMB Report C-297, Bethesda, 1950.
- [18] Bertram, V., Alvarez, A., *Hydrodynamic aspects of AUV design*, 5th Conf. Computer and IT Applications in the Maritime Industries (COMPIT), Leiden, pp.45-53, 2006.
- [19] <http://auvac.org/resources/browse/>
- [20] Cavallo, E., Michelini, R. C., Filaretov, V. F., *Conceptual Design of an AUV Equipped with a Three Degrees of Freedom Vectored Thruster*, Journal of Intelligent and Robotic Systems, Issue 39, p:365–391, 2004.
- [21] Department of Defense, *Composite Materials Handbook Volume 3*, Department of Defense Handbook, 2002.
- [22] Gay, D., Hoa, S. V., Tsai, S. W., *Composite Materials*, CRC Press, 2003.
- [23] Balya, B., Senel, F., Miskbay, O., Parnas, L., *Comparison of Resin Transfer Molding Methods in Production of Composite Shell Structural Aerospace*

*Components*, 4. Ankara International Aerospace Conference, September 2007, Ankara.

[24] Yuh J., *Design and Control of Autonomous Underwater Robots: A Survey*, *Autonomous Robots* 8, 7-24 (2000).

[25] WS Atkins Consultants, *Best Practice Guidelines for Marine Applications of Computational Fluid Dynamics*.

[26] FLUENT 6.3 User's Guide.

[27] *Research Water Tunnels Specifications*, Rolling Hills Research Corporation.

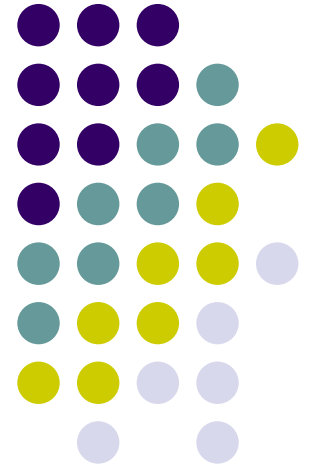
[28] Gorski, J. J., *Drag Calculations of Unappend Bodies of Revolution*, Hydromechanics Directorate Research and Development Report, Naval Surface Warfare Center Carderock Division, 1998.

# ULİSAR

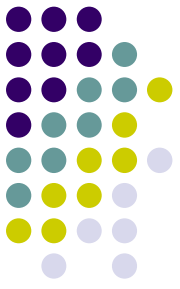
## Ulusal İnsansız Sualtı Aracı

**Emre EGE**

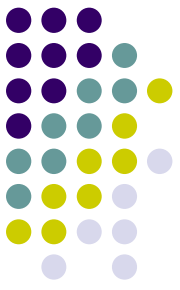
ODTÜ, Elektrik ve Elektronik Mühendisliği Bölümü



# İçerik



- Proje tanımı
- Mekanik tasarımı ve üretimi
- Donanım tasarımı ve üretimi
- Yazılım tasarımı ve geliştirilmesi
- Sualtında yapılan çalışmalardan örnekler
- Gelecek çalışmalar



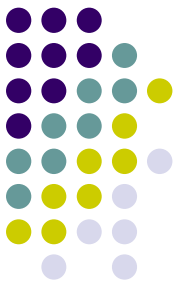
# Proje Grubu

- Proje Yöneticisi : Prof. Dr. M. Kemal LEBLEBİCİOĞLU  
(ODTÜ, Elektrik ve Elektronik Müh.)
- Projede görev alan kurum ve kuruluşlar:
  - ODTÜ Elektrik ve Elektronik Mühendisliği Bölümü
  - ODTÜ Makina Mühendisliği Bölümü
  - ODTÜ Bilgisayar Mühendisliği Bölümü
  - Yenel LTD. (Elektronik kart üretimi)
  - Barış Elektrik Endüstrisi A.Ş. (Gövde üretimi)

# İnsansız Sualtı Araçları



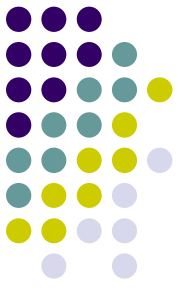
- Denizaltıları andıran sualtı robotları
- Otonom ve uzaktan yönetilen araçlar
- Bağlantı kablosuz/kablolu ve mürettebat gereksinimleri yok



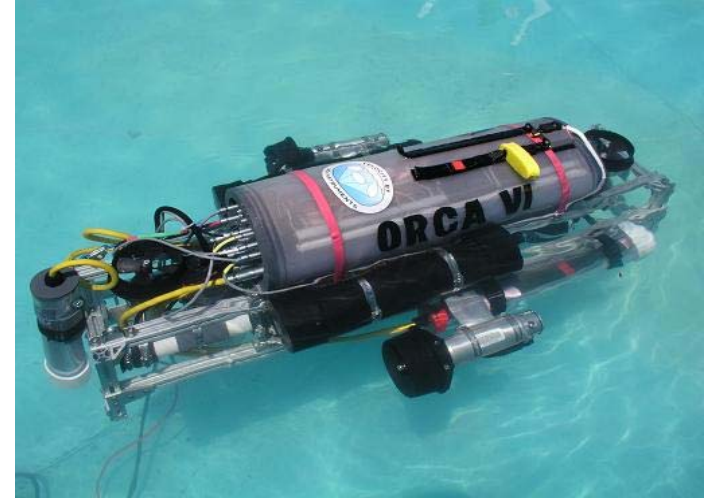
# Kullanım Alanları

- Mayın imha
- Keşif
- Sualtı kablolarının döşenmesi
- Sualtı hedeflerine ekipman taşınması
- Deniz ve okyanus suyu incelemeleri
- Petrol ve doğalgaz boruları gözlem ve bakımı
- Sualtı arkeoloji çalışmaları
- Deprem araştırmaları

# Dünyadaki Örnekleri



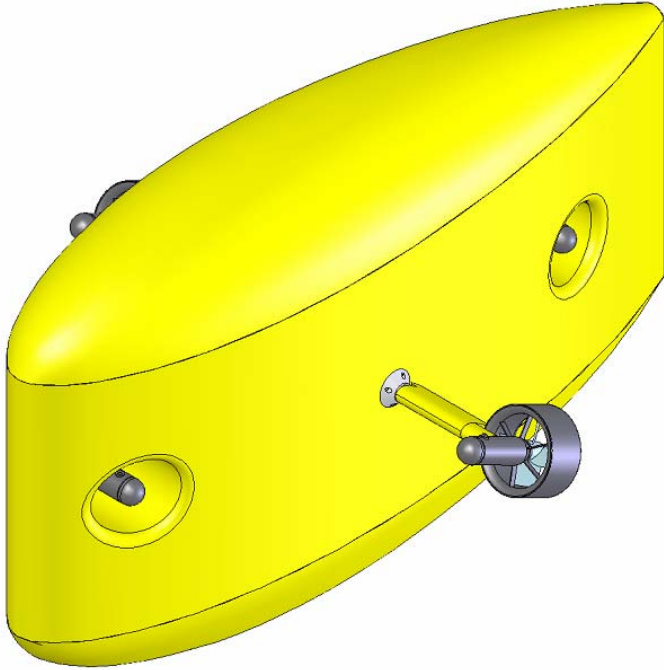
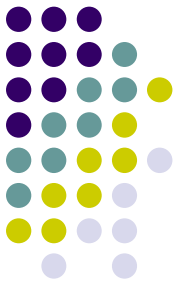
Cornell University (sea Monkey)



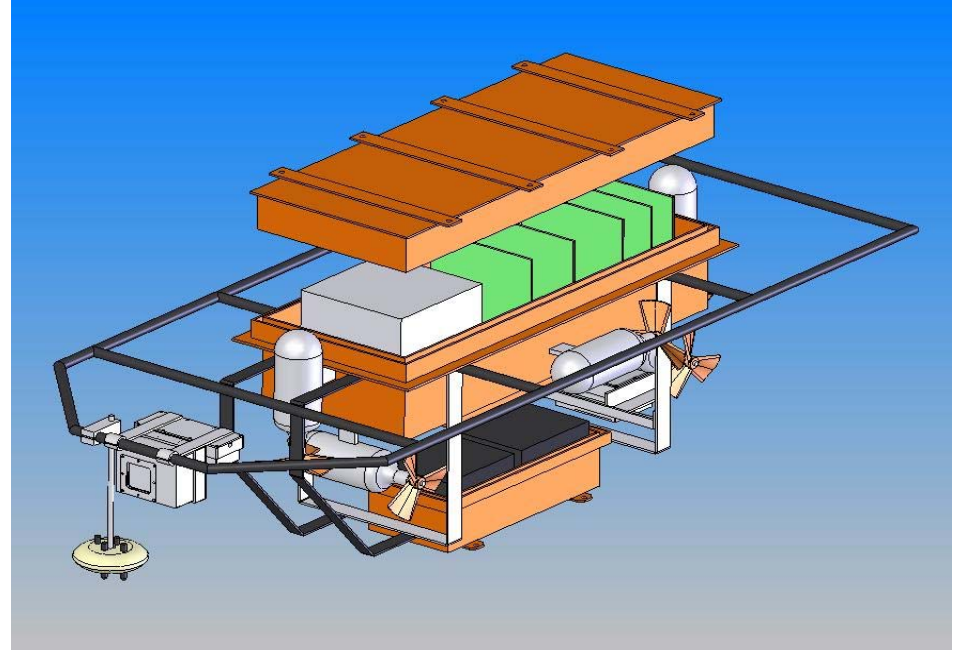
MIT AUV Tasarımı (ORCA VI)



# Dünyadaki Örnekleri

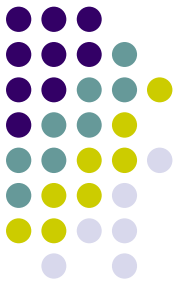


Odyssey IV (MIT)

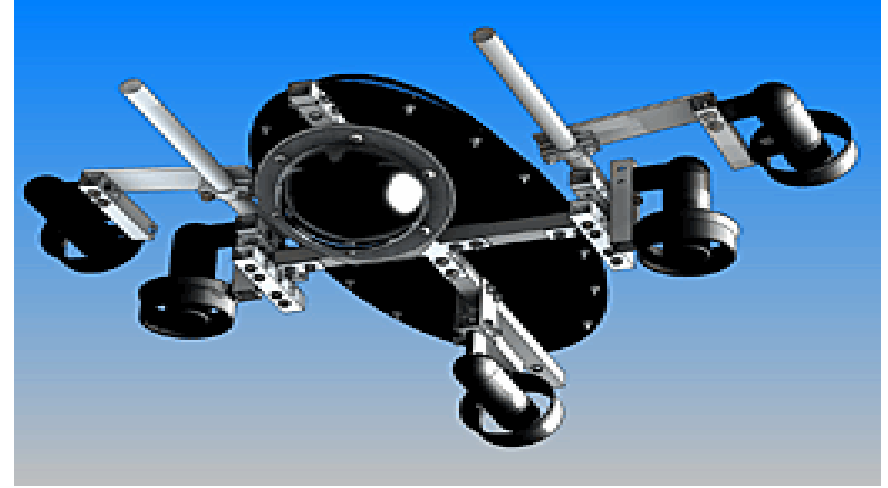


Quebec University Submarine

# Dünyadaki Örnekleri

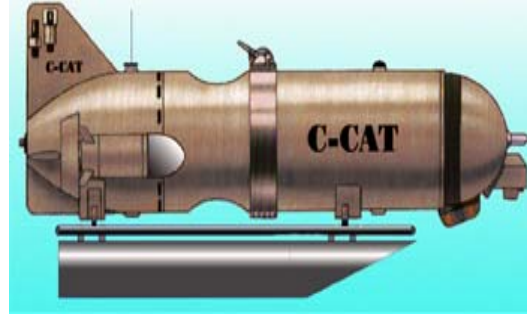
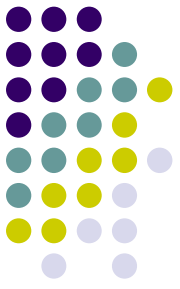


North Carolina State Uni. AUV Tasarımı  
(Seawolf II)

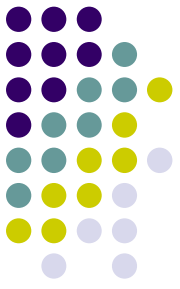


University of Toronto AUV Tasarımı

# Dünyadaki Örnekleri



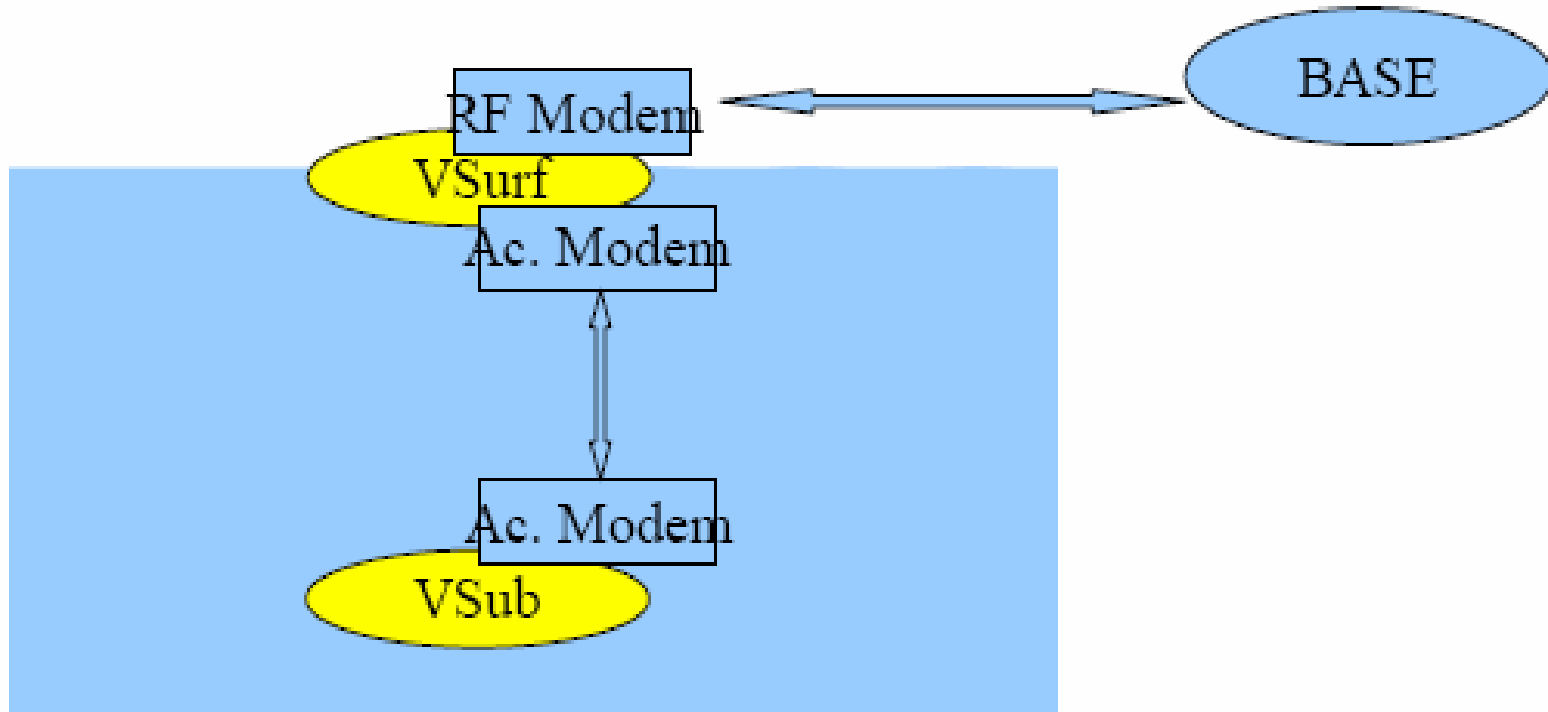
# ULİSAR



## Temel Fikir:

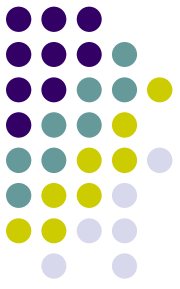
- Sualtı aracı ile beraber uyumlu olarak çalışan bir suüstü aracının varlığı,
- Aradaki haberleşme: Akustik modem kullanarak düşük rezolüsyonlu görüntü ve navigasyon bilgilerinin aktarımı,
- Suüstü haberleşmesi: RF kullanarak.

# ULİSAR

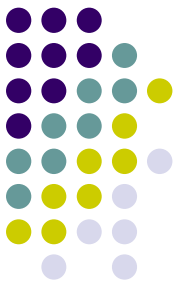


Önerilen insansız sualtı aracı sistemi

# Mekanik Tasarım ve Üretim

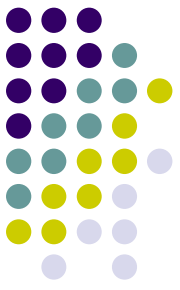


- Su sızdırmaz bölme tasarımı
- Dış gövde tasarımı
- İtme ve manevra sistemleri
- Denge ve yüzerlik
- Kablolama
- İmalat yöntemi



# Dış Gövde Tasarımı

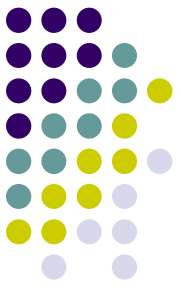
- Dış gövde: aracı en dışta saran yapı
- Ekipmanın korunması, hidrodinamik yapı ve aracın şekli
- Denizaltıların aksine çeşitli şekillerde olabilirler



# Dış Gövde Tasarımı

- Dönme kararlılığı açısından simetrik yapıda
- Ekipmanı kapsayacak ölçüde uzun ve büyük çapta
- Sürüklenme ve sürtünme kuvvetlerini azaltacak ölçüde kısa ve küçük çapta
- Basınca dayanacak ve ağırlığı çok arttırmayacak ölçüde bir et kalınlığında
- Elektronik ekipmanın fazla ısınmasını engelleyecek ölçüde ısı transferine izin veren yapıda
- Modüler yapıda



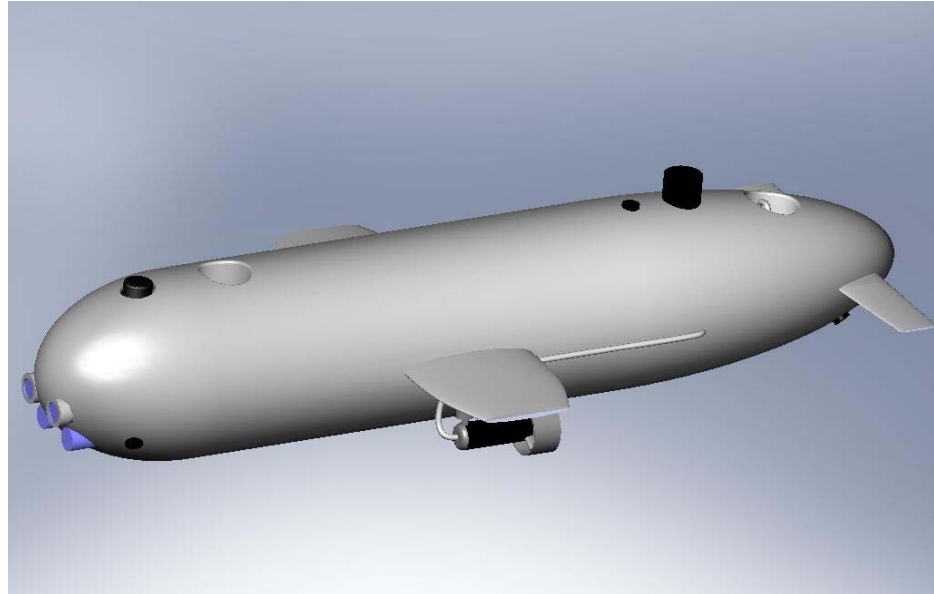


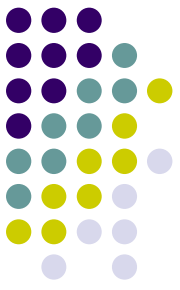
# Dış Gövde Tasarımı

- Düşük karşı kuvvet
- Üretim kolaylığı
- Geniş veritabanı



Silindirik yapı

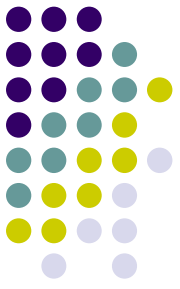




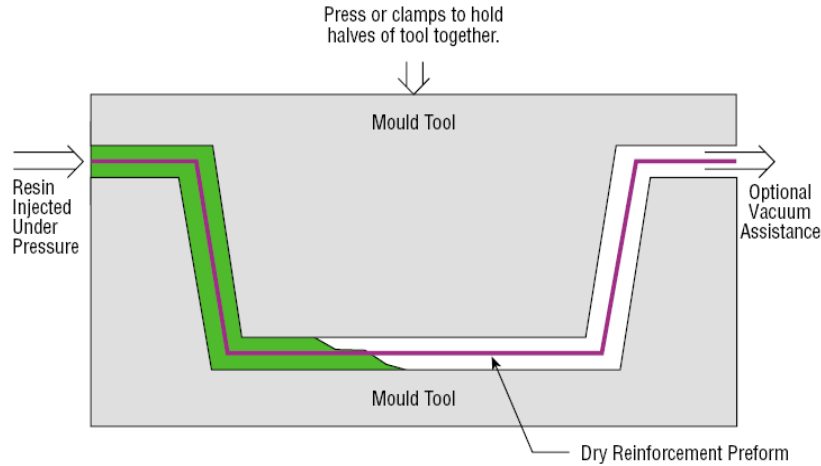
# Üretim Malzemesi

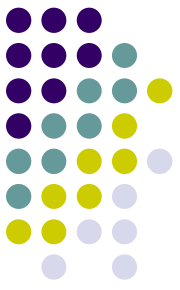
- Gövde kompozit malzemedan yapılmıştır:
  - Düşük yoğunluklarda yüksek mukavemet
  - Tasarım esnekliği
  - Kalıplama kolaylığı
  - Çeşitli yüzey uygulamalarına izin vermesi
  - Gövde içine metal parçalar yerleştirilebilmesi
  - Yapıştırma işlemleri uygulanabilmesi
- Malzeme: Cam dokuma kumaş – epoksi reçine

# Üretim Tekniği



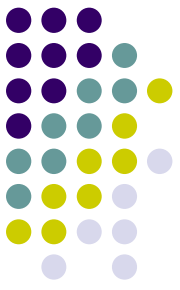
- Reçine Transfer Kalıplama (RTM)
  - Kalıp parçaları arasına elyaf yatırma
  - Basınç ve/veya vakum ile kontrollü reçine besleme
  - Isıl işlem ile polimerizasyon



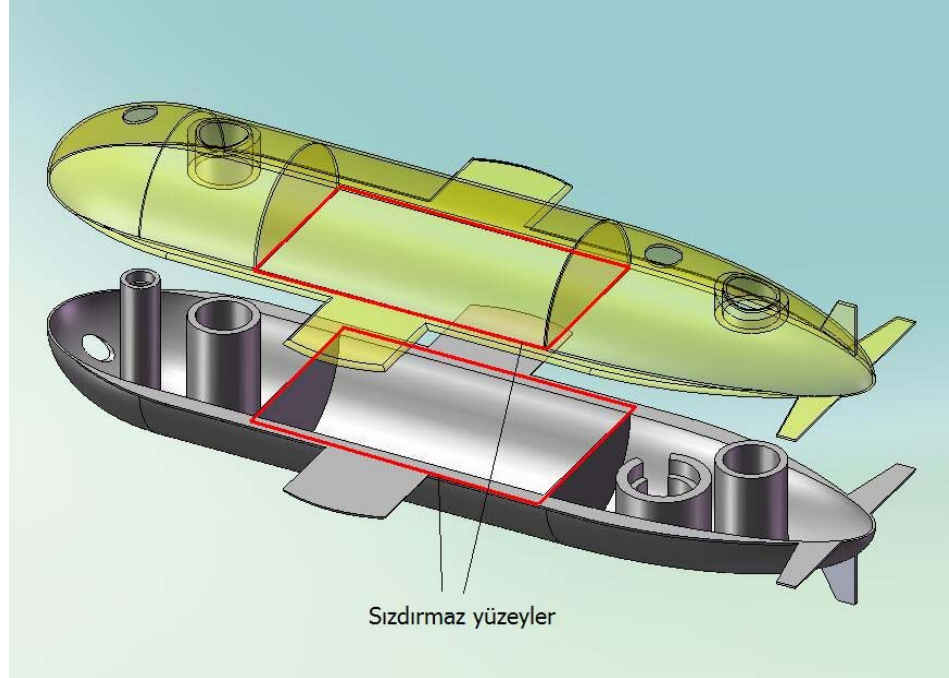


# Üretim Tekniđi - RTM

- Avantajlar:
  - - Yüksek elyaf-reçine oranı
  - - Düşük hava kabarcığı oranı
  - - Düzgün yüzey kalitesi
  - - Tasarım esnekliđi
- Dezavantajlar:
  - -Artan kalıplama masrafı
  - -Büyük parçalara uygulama zorluđu

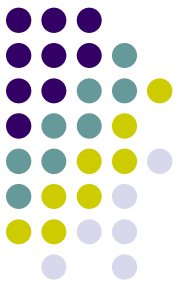


# Dış Gvde Tasarımı



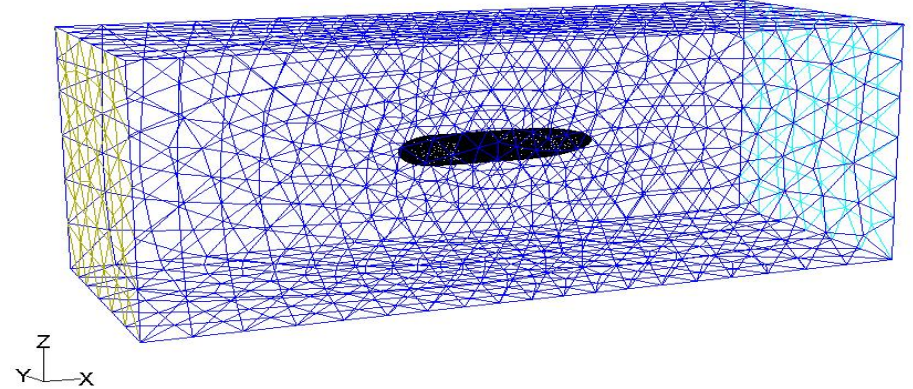
- Maliyeti dşrmek
- Yksek mukavim yapı elde etmek

# Dış Gövdenin Tasarımı

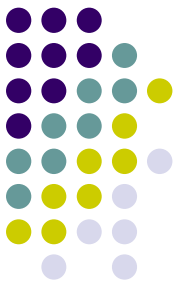


## Hidrodinamik analizler:

- Gambit-Fluent
- Basitleştirilmiş model
- 3B, zamandan bağımsız
- k- $\epsilon$  türbülans modeli
- Sıcaklık ve özellikler sabit

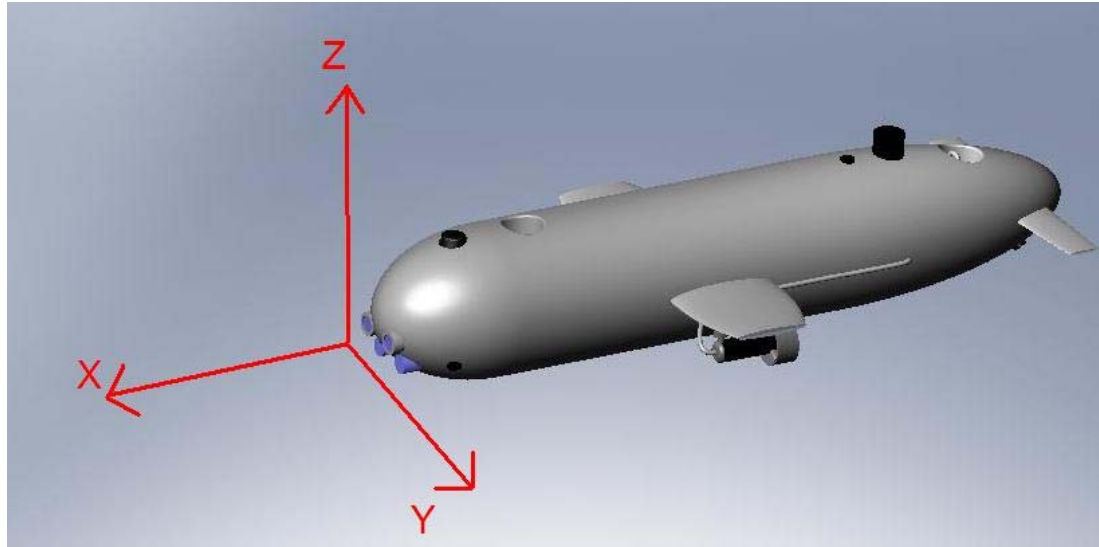


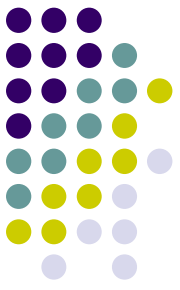
- Analizler sonucu araca etkiyen sürüklenme kuvvetleri elde edilmiş ve bu değerlerin seçilen motorların sağladığı itki değerlerini geçmediği görülmüştür.



# İtki ve Manevra

- İleri geri hareket (x eksenini boyunca)
- Yukarı aşağı hareket (z eksenini boyunca)
- Sağa sola dönüş (z eksenini etrafında)
- Yukarı aşağı dönüş (y eksenini etrafında)

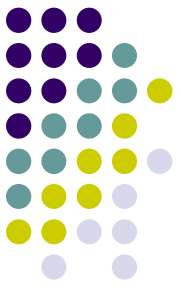




# Denge, Kararlılık ve Yüzerlik

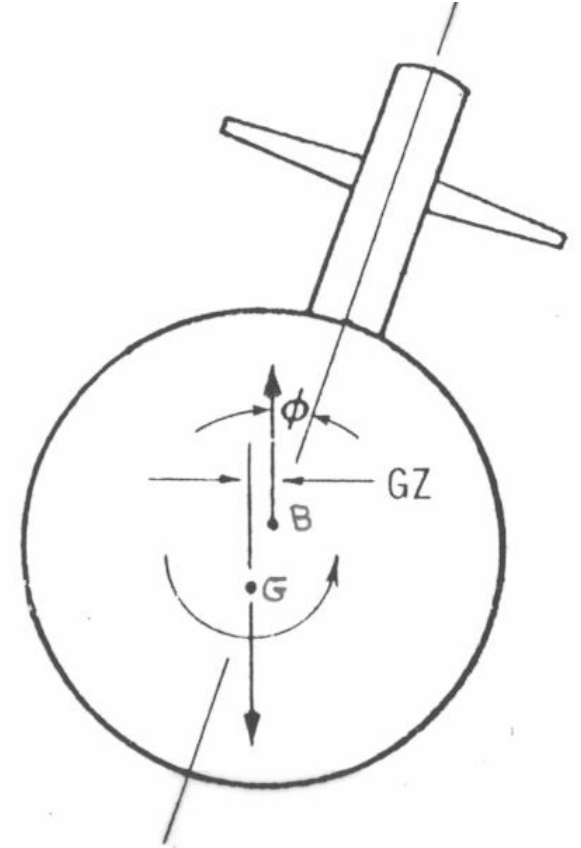
- Yüzerlik özelliğine sahip araçlar suda asılı kalabilmektedir. Bu sayede hareketsizken güç harcanması yoktur.
- Olası sistem problemlerinde aracı kaybetmemek için araç 0.5-1 kg arasında kaldırma kuvvetine sahip olacak şekilde tasarlanabilir.
- Denge ve kararlılık ise aracın istenilen şekilde durabilmesi ve hareket edebilmesi açısından önemlidir.
- Aracın denge ve kararlılığında ağırlık ve kaldırma kuvveti merkezlerinin konumları belirleyicidir.

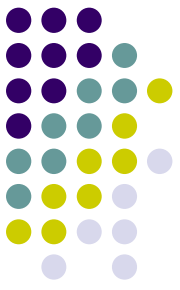




# Denge, Kararlılık ve Yüzerlik

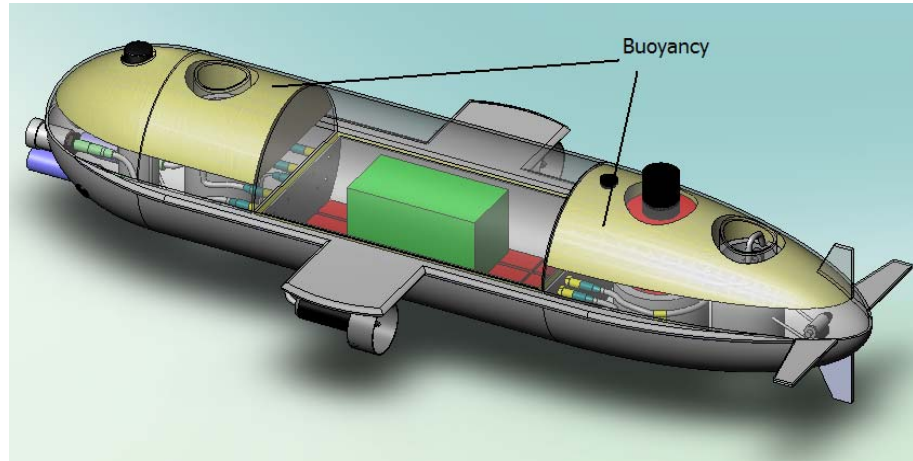
- Denge açısından iki merkezin x ve y konumları aynı olmalıdır.
- Kararlılık açısından ise kaldırma kuvveti merkezi, ağırlık merkezinin mümkün olduğunca üstünde olmalıdır
- İki merkez arasındaki bu mesafe aracın maruz kaldığı dönme etkilerine karşı düzeltici etkide bulunup kararlılık sağlamaktadır





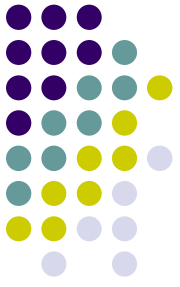
# Denge, Kararlılık ve Yüzerlik

- Araca belirtilen özellikleri sağlamak amacıyla alt bölümlere metal ağırlıklar, üst kısımlara ise yüksek yoğunluklu su emmeyen köpük eklenmektedir.



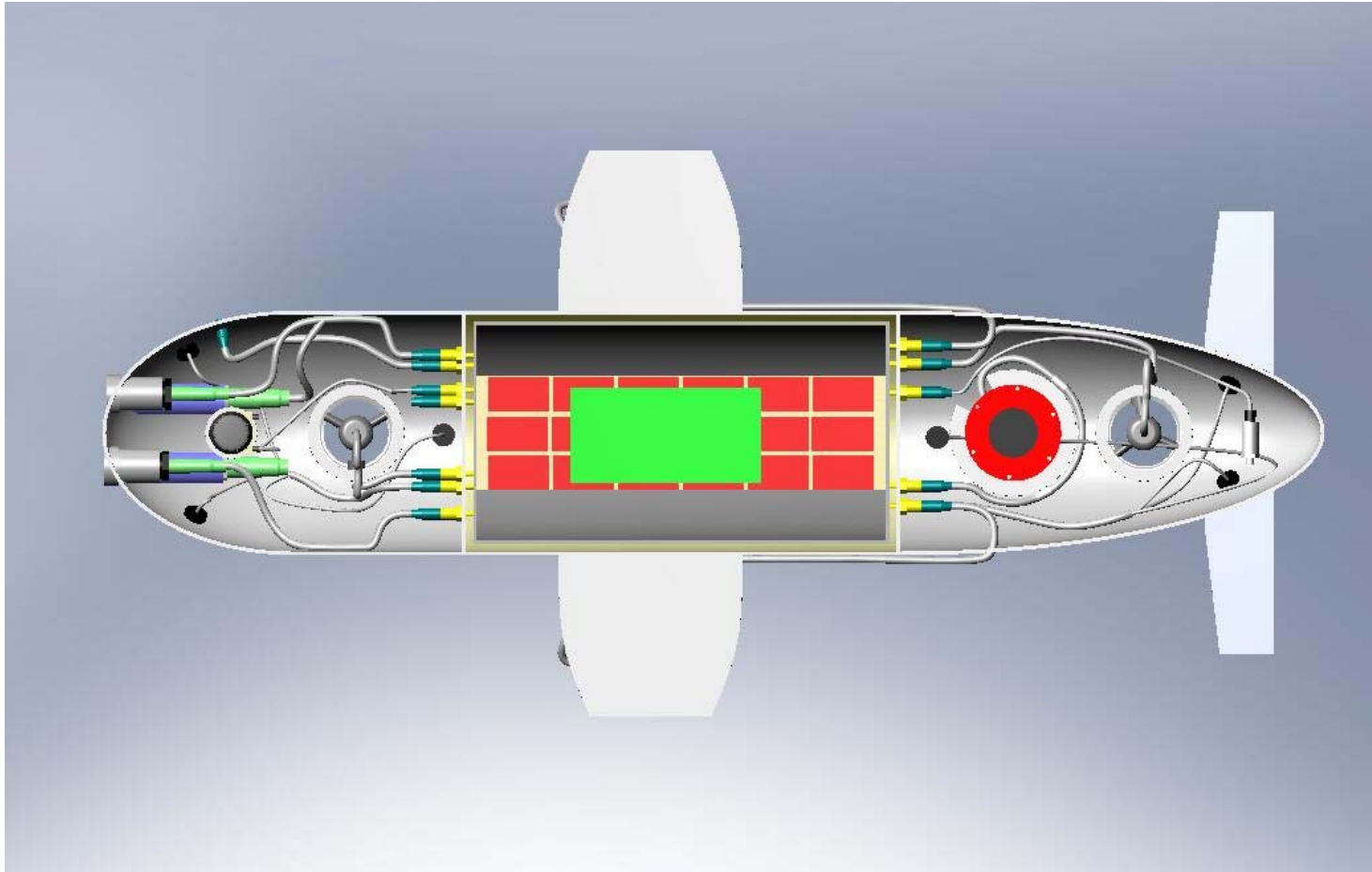
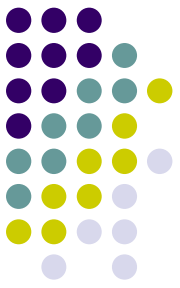
- Bu işlemler sonucu ağırlık ve kaldırma kuvveti merkezleri yaklaşık 3 cm kadar ayrılmışlardır.

# Kablolama

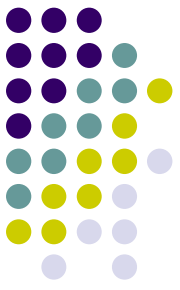


- Denetim elemanları ve bataryalar su geçirmez bölümde saklanırken diğer cihazlar su geçirmez olduklarından su alan bölümde bulunmaktadır.
- Su geçirmez kablolar, aracın su geçirmez orta bölümü ile su alan bölümündeki cihazlar arasındaki bağlantıyı sağlamaktadır.
- Su geçirmez konektörler ise su geçirmez bölüm ile su alan bölümü ayıran duvardan geçişi sağlamaktadır.

# Kablolama



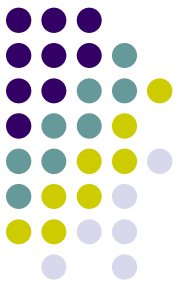
# Üretim Aşamalarından Görüntüler



## Model



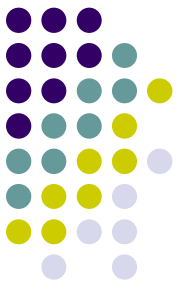
# Üretim Aşamalarından Görüntüler



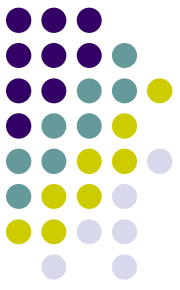
## Kalıp



# Gövde Üretimi

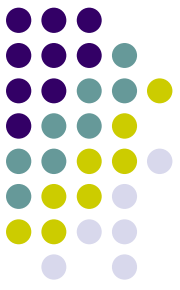


# Gövde Üretimi

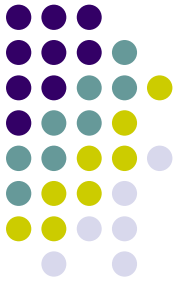




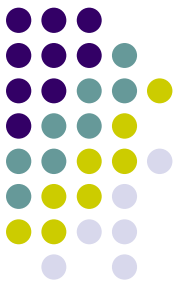
# Gövde Üretimi



# Donanım

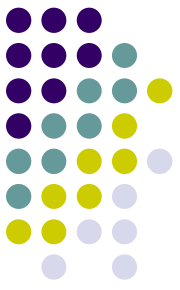


- Denizaltı aracında bulunan çevresel donanım birimleri:
  - 4 adet itici
  - 2 adet kamera ve 2 adet aydınlatma
  - 1 adet sonar
  - 1 adet akustik modem
  - 6 adet yaklaşma almacı (ultrasonik transducer)
  - 1 adet basınç almacı
  - 2 adet sıcaklık ölçer
  - 1 adet ataletsel ölçüm birimi (IMU)
  - 1 adet uzaktan kumandalı güç-açma kapama anahtarı



# Donanım

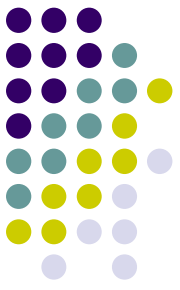
- Denizaltı aracını kontrol etmek amacı ile 2 adet PC/104+ form faktöründe bilgisayar ve bu bilgisayarlara bağlı değişik amaçlarda kartlar kullanılmıştır.
  - 2 adet bilgisayar(kontrol ve görüntü)
  - 1 adet MPEG-4 kodlayıcı kartı
  - 2 adet DC/DC çevirici
  - 1 adet PCMCIA-Ethernet kartı
  - 1 adet Compact Flash kartı



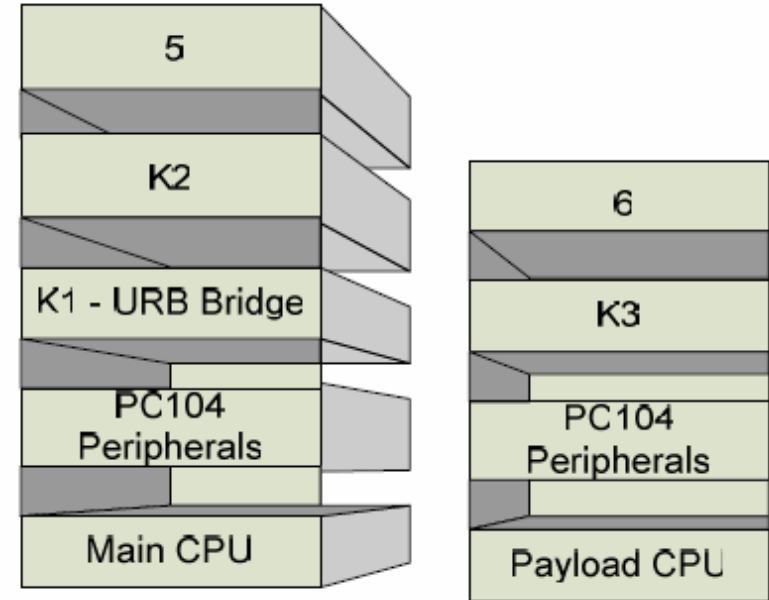
# Donanım

- Denizaltı aracındaki çevre birimlerle haberleşmek ve güç dağıtımını sağlamak amacıyla çeşitli kartlar tasarlanıp üretilmiştir.
  - URB Köprü Kartı
  - Güç – Kamera- Işık – Motor – Analog Almaç Kartı
  - Ultrasonik Almaç Düğümü
  - Denetim bilgisayarına bağlı yığın arayüz kartı
  - Faydalı yük bilgisayarına bağlı yığın arayüz kartı
  - Güç durumu ölçme ve güç dağıtım kartı

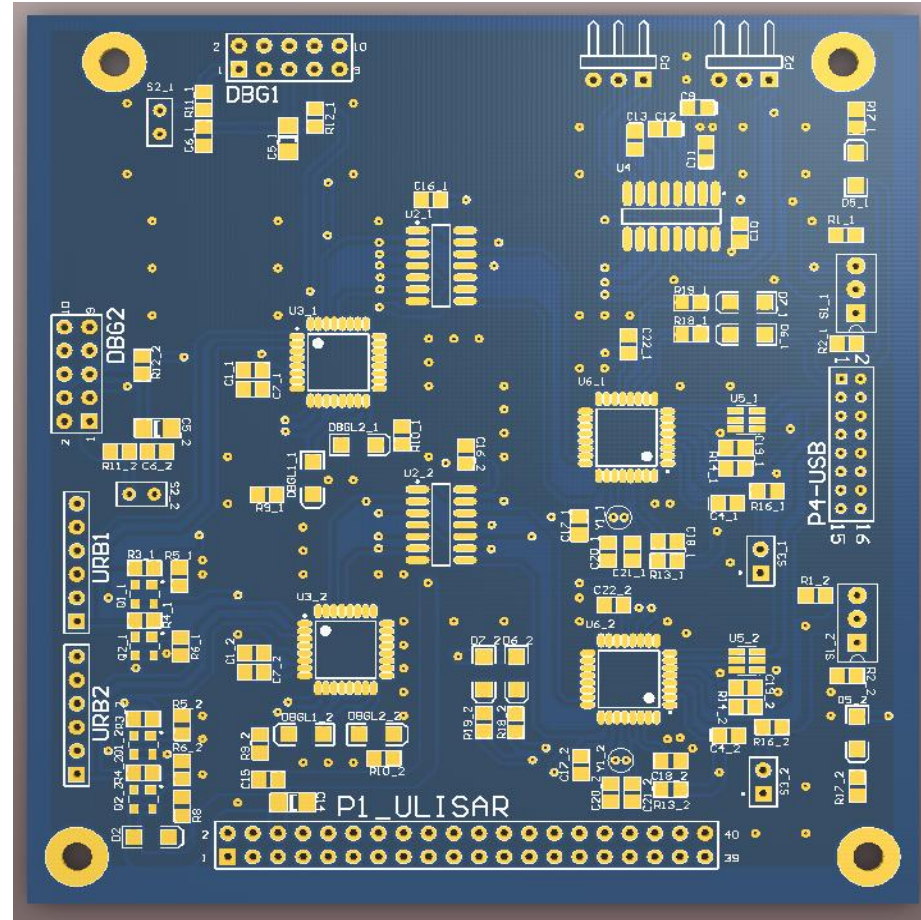
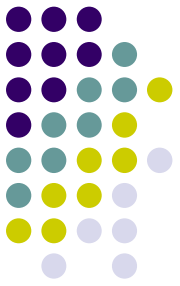
# Donanım



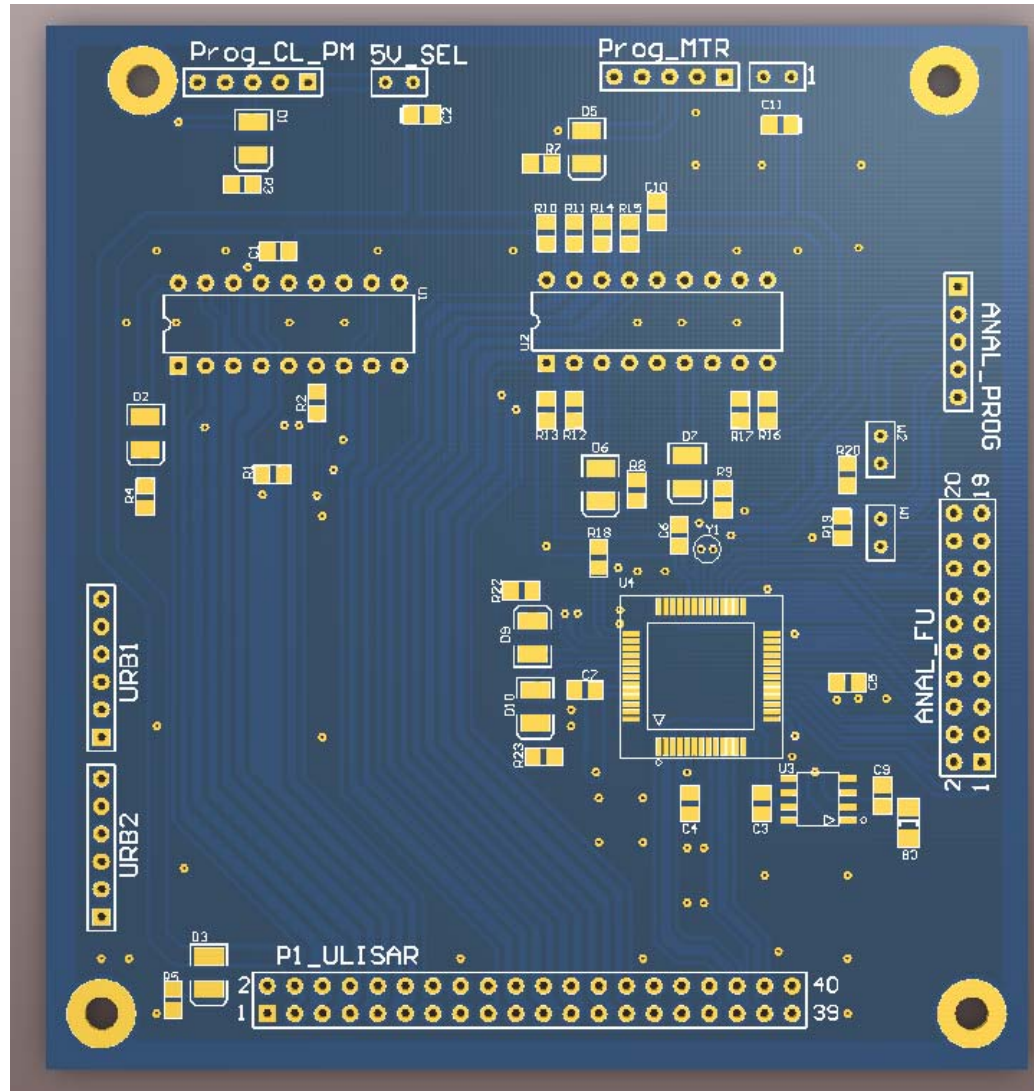
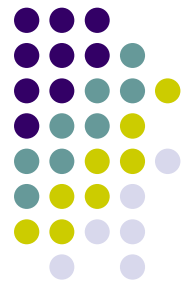
- Ön ultrasonik almaç seçme ve kamera-ışık denetleme kartı
- Arka ultrasonik almaç seçme kartı
- Yönelim denetleme kartı

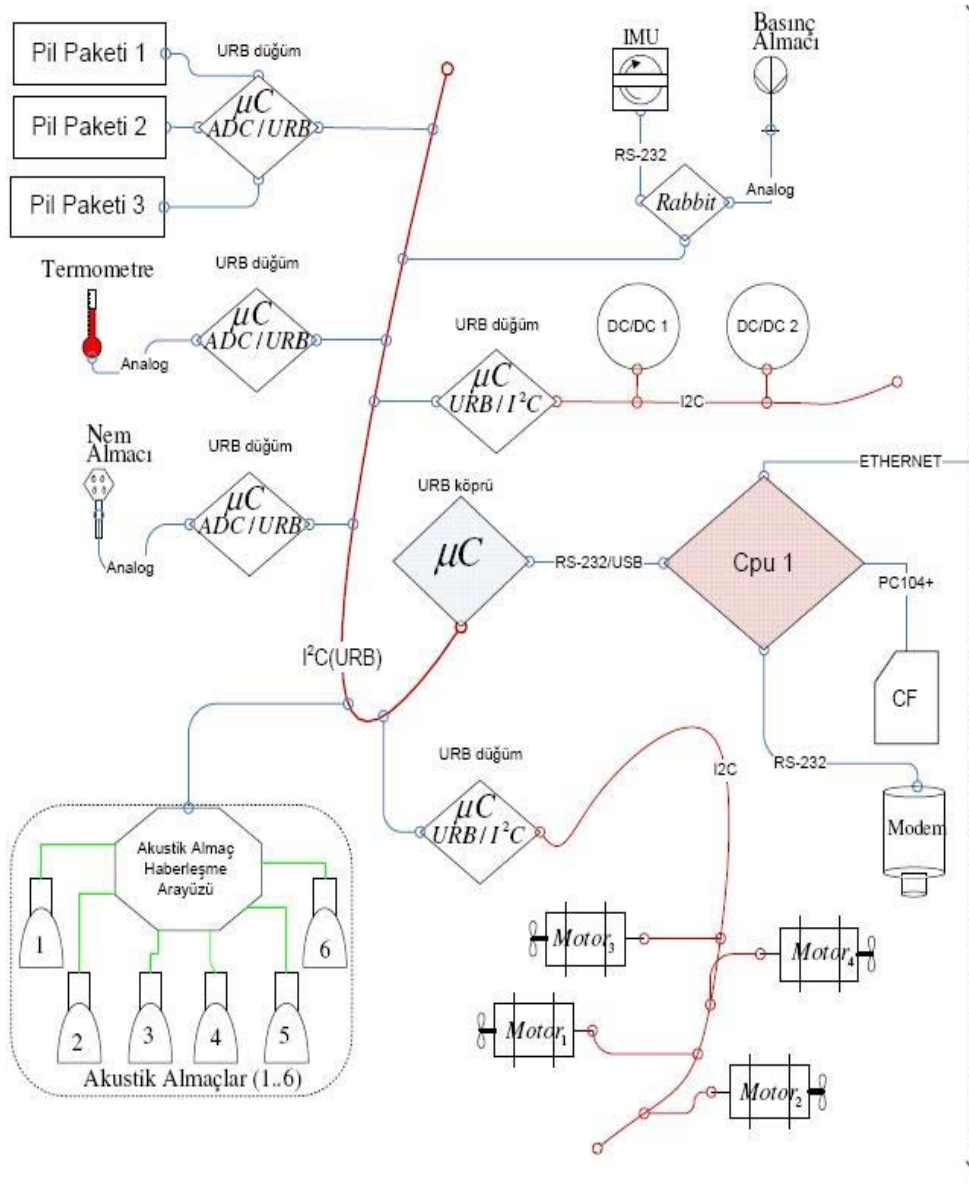


# URB Köprü Kartı

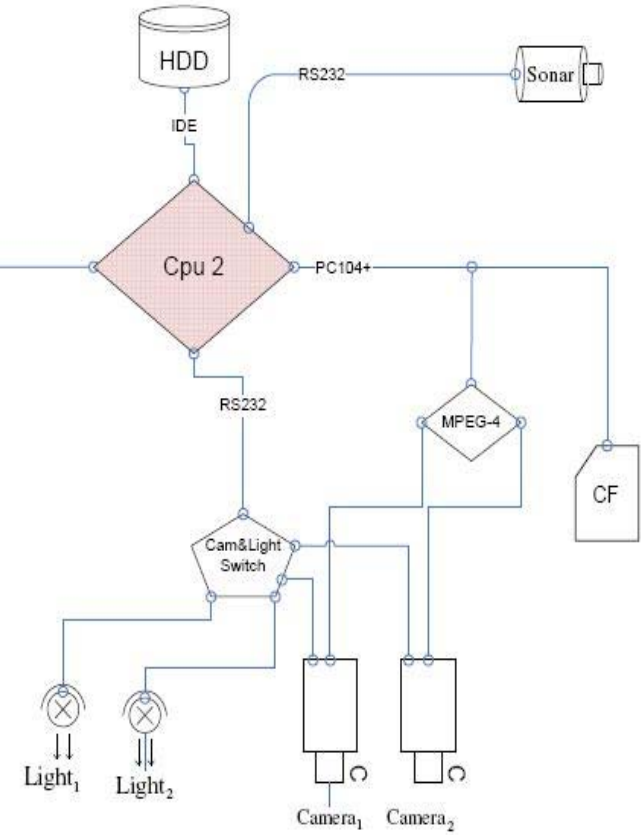


# Güç – Kamera- Işık – Motor – Analog Almaç Kartı

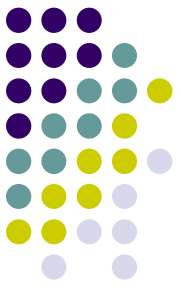




CPU 1: Ana denetim bilgisayarı  
 CPU 2: Faydalı yük denetim bilgisayarı  
 URB/I2C: URB'den I2C veri yolu çeviricisi  
 URB/ADC: Analogdan URB'ye çevirici  
 URB DÜĞÜM: URB Arayüz birimi (URB NODE)

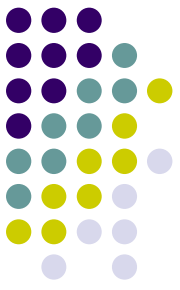






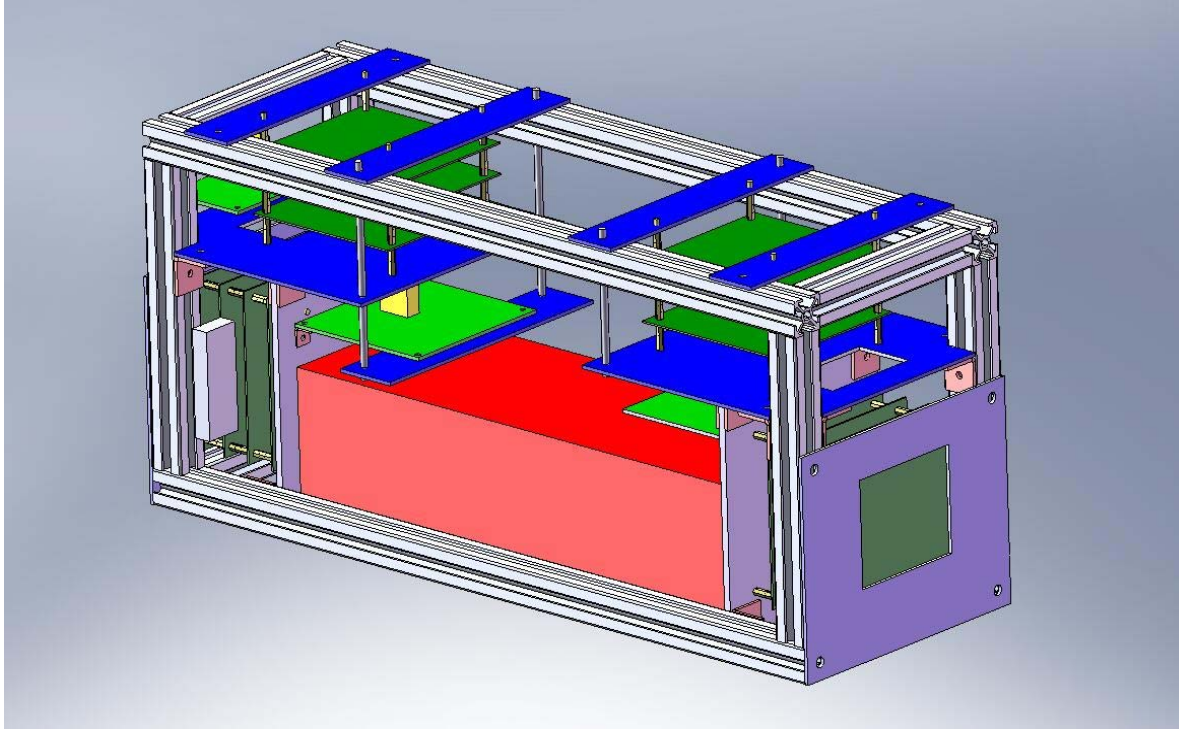
# Bataryalar

- Denizaltı aracı yaklaşık olarak sualtında 2 saat süreyle operasyon yapacak şekilde batarya yerleşimi yapılmıştır.
- Yüksek kapasitesi ve boyutlarının küçüklüğü bakımından LiPo bataryalar kullanılmıştır.
- 3 ayrı batarya grubu kullanılmıştır.
  - 4 hücreli 6 paralel batarya(14.4V, Elektronik)
  - 8 hücreli 4 paralel batarya(28.8V, 2 motor)
  - 8 hücreli 4 paralel batarya(28.8V, 2 motor)

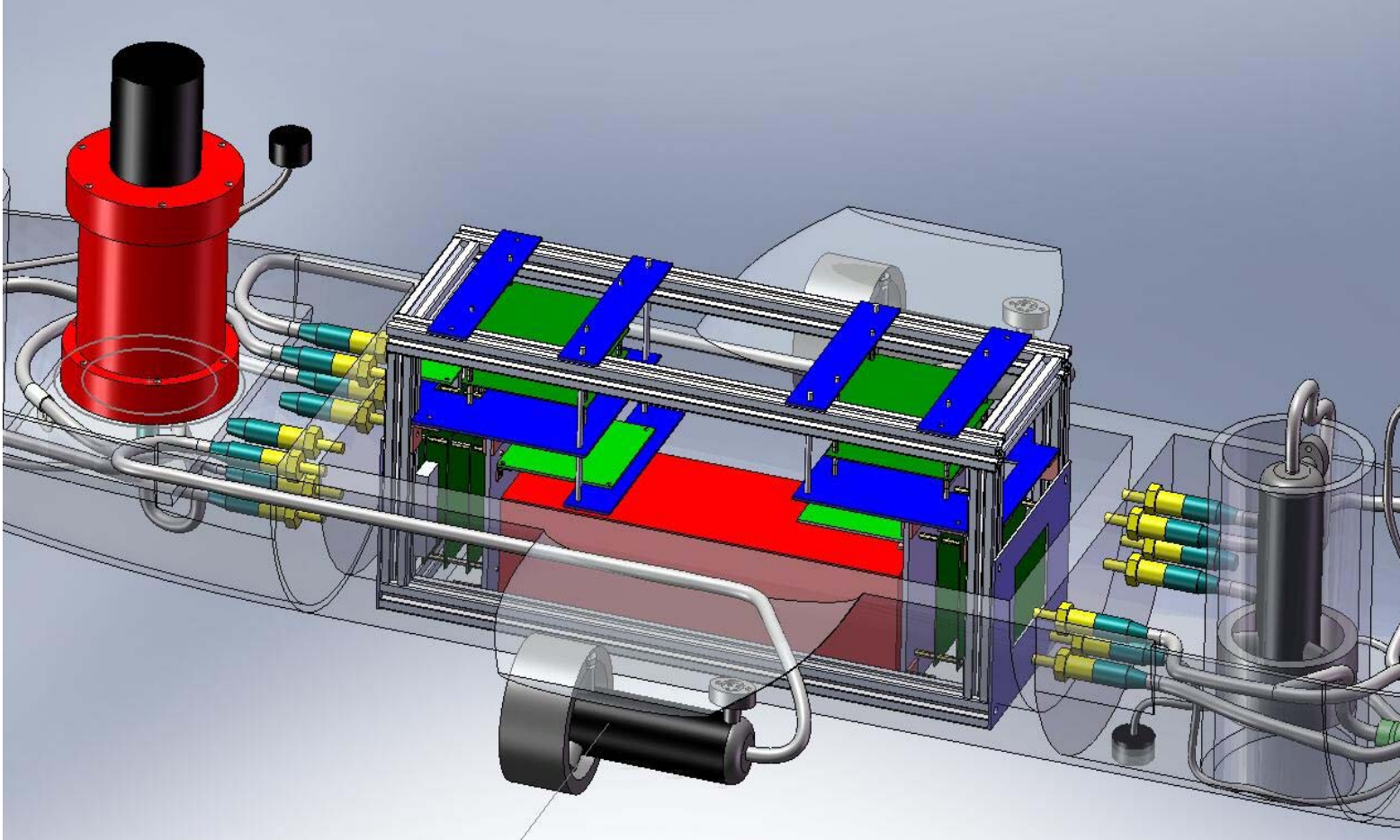


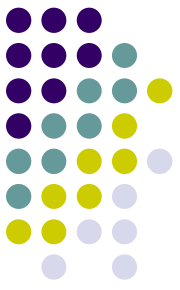
# Elektronik Kutu

- Gvde ierisinde bulunan kartlar ve bataryaları sabit tutmakta kullanılmaktadır.



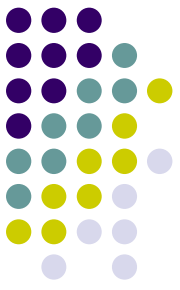
# İç Yerleşim





# Yazılım

- Denizaltı aracı için bir çok deęişik donanımda kořacak çeřitli yazılımlar geliřtirilmiřtir.
- Kontrol bilgisayarında QNX, grnt(faydalı yk) bilgisayarında ise Linux iřletim sistemi alıřmaktadır.
- Sistemde bulunan mikroiřlemcilerde farklı amalarda yazılımlar geliřtirilmiřtir.
- Kamera ve dięer almalardan alınan veriler operatr bilgisayarına gnderilmekte, ve operatr bilgisayarından gelen emirler motorlara ve dięer birimlere aktarılmaktadır.



# Operatör Arayüzü

- Denizaltı aracı bir dizüstü bilgisayar aracılığı ve bir adet kumanda ile kontrol edilmektedir.
- Denizaltı aracı ve operatör bilgisayarı arasındaki haberleşme fiberoptik kablo ile sağlanmaktadır.

# Operatör Arayüzü



UlisarGUI

**Sensor Degerleri**

Basinc Sensörü: 0 Mesafe Sensörü 2 0  
Nem Sensörü: 0 Mesafe Sensörü 4 0  
Mesafe Sensörü 1 0 Mesafe Sensörü 5 0  
Mesafe Sensörü 2 0 Mesafe Sensörü 6 0

**Denizalti Durumu**

ACIL DURUM

**Sonar Kontrol Ekranı**

Sonar Aktif

**Motorlar**

On Motor M.D: 0  
Sol Motor M.D: 0  
Sag Motor M.D: 0  
Arka Motor M.D: 0

**Batarya Voltaj Seviyeleri**

1. Batarya Grubu 0 3. Batarya Grubu 0  
2. Batarya Grubu 0 4. Batarya Grubu 0

**Ag Durumu**

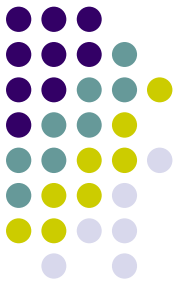
**IMU Bilgileri**

**Kamera ve Isik Kontrol**

Isiklar Acik  
 Kamera 1 Acik  
 Kamera 2 Acik

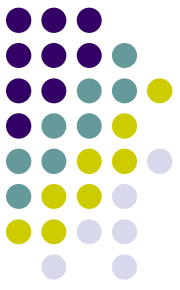
Kamera 1 bulunamadi  
Kamera 2 bulunamadi

# Kumanda kolu



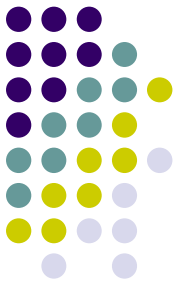
Numara	İşlev
1	Kamera 1 açma-kapama
2	Kamera 2 açma-kapama
3	Işıkları açma-kapama
4	Sonar açma-kapama
5	Denizaltı aracının yönlendirilemesi. İleri-geri,sağ,sol.
6	Denizaltı aracının yunuslama(pitch) kontrolü
7	Kullanılmıyor
8	Denizaltı aracının dik olarak yukarı çıkması
9	Denizaltı aracının dik olarak aşağı inmesi
10,11	10 ve 11 numaralı düğmelere aynı anda basıldığında Acil Durum konumuna geçirmek

# Testlerden Görüntüler

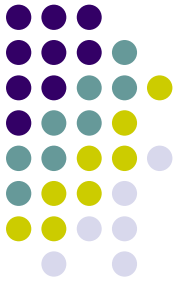




# Testlerden Görüntüler



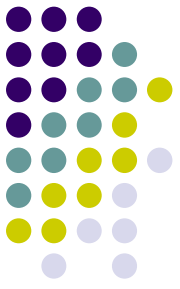
# Testlerden Görüntüler



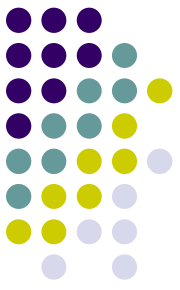
Yüzerlilik denge testi sualtı görüntüsü



# Sonuç

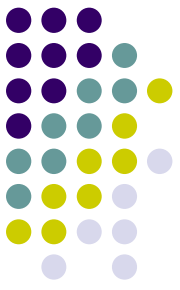


- Mekanik tasarım ve üretim evresi projenin başarısı açısından son derece önemlidir.
- Bu evrede çeşitli tasarım öğeleri göz önünde bulundurulmuş ve mühendislik çalışmaları gerçekleştirilmiştir.
- Proje çalışmalarında makina, elektronik ve bilgisayar mühendislerinden oluşan bir disiplinlerarası proje grubu görev almıştır.
- Gereken noktalarda analizler ve deneyler gerçekleştirilmiştir.



# Sonuç

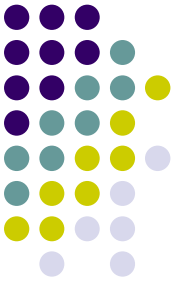
- Oldukça yoğun çalışmalar daha gerekmektedir
- Navigasyon çalışmaları konusunda bilinenlerin ötesini zorlamak gerekiyor,
- Çalışmaların hiçbir zaman sonlandırılmaması gerekli,
- Bu konuda ülkemizin çok eksikliği var,
- Aynı zamanda bu konulara ihtiyaç çok fazla,
- Sadece askeri amaçlar değil sivil ihtiyaçlar düşünüldüğünde pazar çok büyük.



# Sonuç

- Projede elde edilen tecrübe ve bilgi birikiminin önümüzdeki yıllarda gerçekleşebilecek benzer projelerde faydalı olması beklenmektedir.
- Böyle bir çalışmanın sonunda görüntü işleme, sonar/kamera görüntülerinin birleştirilmesi, yol planlama, vs., gibi takip eden çalışmalar yapılabilir.

# Teşekkür



*Dinlediğiniz için teşekkürler*

*Soru ve Cevap*

# BİR İNSANSIZ SUALTI ARACI: ULİSAR



**M. Kemal Leblebiciođlu**

ODTÜ, Elektrik ve Elektronik Müh. Bölümü

[kleb@metu.edu.tr](mailto:kleb@metu.edu.tr)

# Proje Ekibi



- Prof. Dr. M. Kemal Leblebiciođlu (ODTÜ, Elektrik ve Elektronik Müh. Bölümü) (Proje koordinatörü),
- Y. Doç. Dr. Afşar Saranlı (ODTÜ, Elektrik ve Elektronik Müh. Bölümü),
- Y. Doç. Dr. İlkay Ulusoy (ODTÜ, Elektrik ve Elektronik Müh. Bölümü),
- Hüseyin Yiğitler (ODTÜ, Elektrik ve Elektronik Müh. Bölümü),
- Emre Ege (ODTÜ, Elektrik ve Elektronik Müh. Bölümü),
- Örsan Aytekin (ODTÜ, Elektrik ve Elektronik Müh. Bölümü),
- Üst. Kadir Isıyel (ODTÜ, Elektrik ve Elektronik Müh. Bölümü),
- Binbaşı İsmail Çalışkan (ODTÜ, Elektrik ve Elektronik Müh. Bölümü),
- Doç. Dr. Halit Oğuztüzün (ODTÜ, Bilgisayar Müh. Bölümü),
- Alper Kılıç (ODTÜ, Bilgisayar Müh. Bölümü),
- Haldun Mıdođlu (Yenel Ltd.),
- Fikret Şenel, (Barış Elektrik Endüstrisi A.Ş.),
- Prof. Dr. S. Kemal İder, (ODTÜ, Makina Müh. Bölümü),
- Prof. Dr. M. Haluk Aksel (ODTÜ, Makina Müh. Bölümü),
- Prof. Dr. Levend Parnas (ODTÜ, Makina Müh. Bölümü),
- Prof. Dr. Kemal Özgören (ODTÜ, Makina Müh. Bölümü),
- Necmettin Cevheri (ODTÜ, Makina Müh. Bölümü),
- Mine Cüneyitođlu (ODTÜ, Makina Müh. Bölümü),

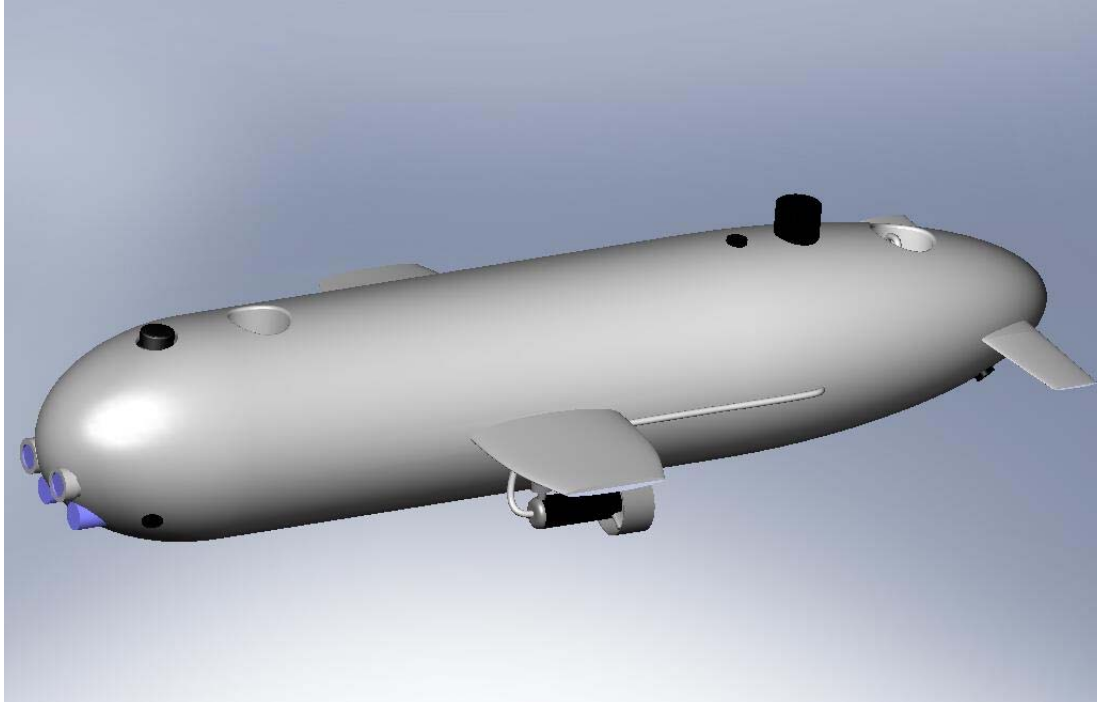


# İnsansız Sualtı Araçları



- Denizaltıları andıran sualtı robotları
- Otonom araçlar
- Bağlantı kablosu ve mürettebat gereksinimleri yok

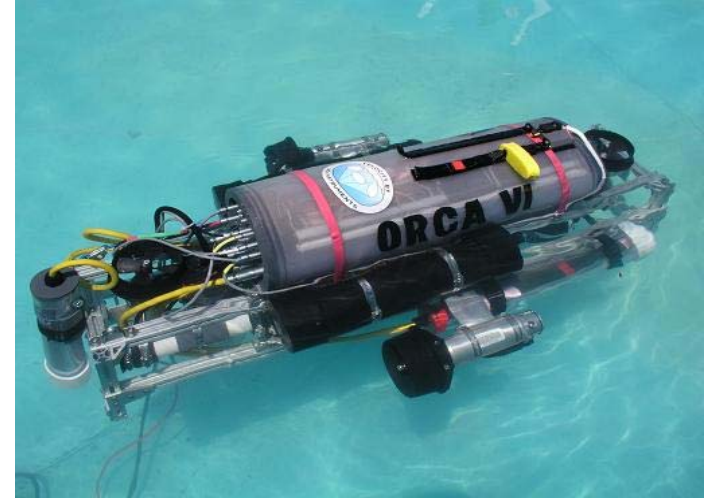
# İnsansız Sualtı Araçları



# İnsansız Sualtı Araçları

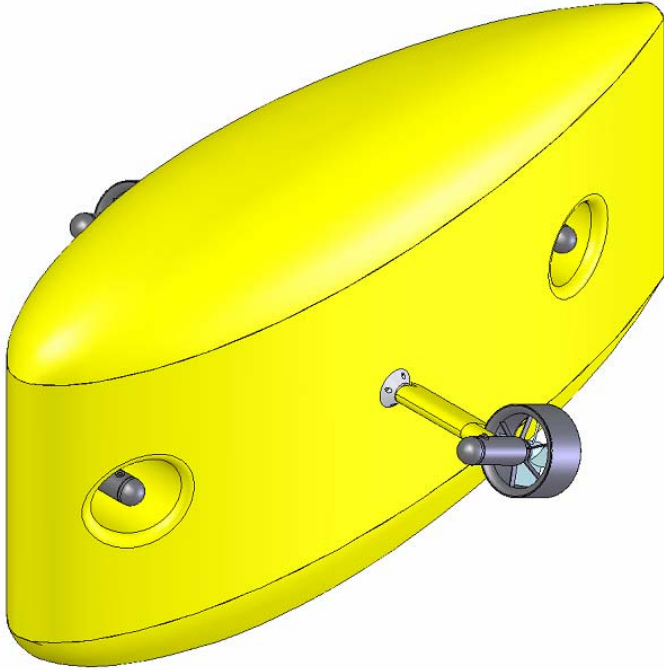


Cornell University (sea Monkey)

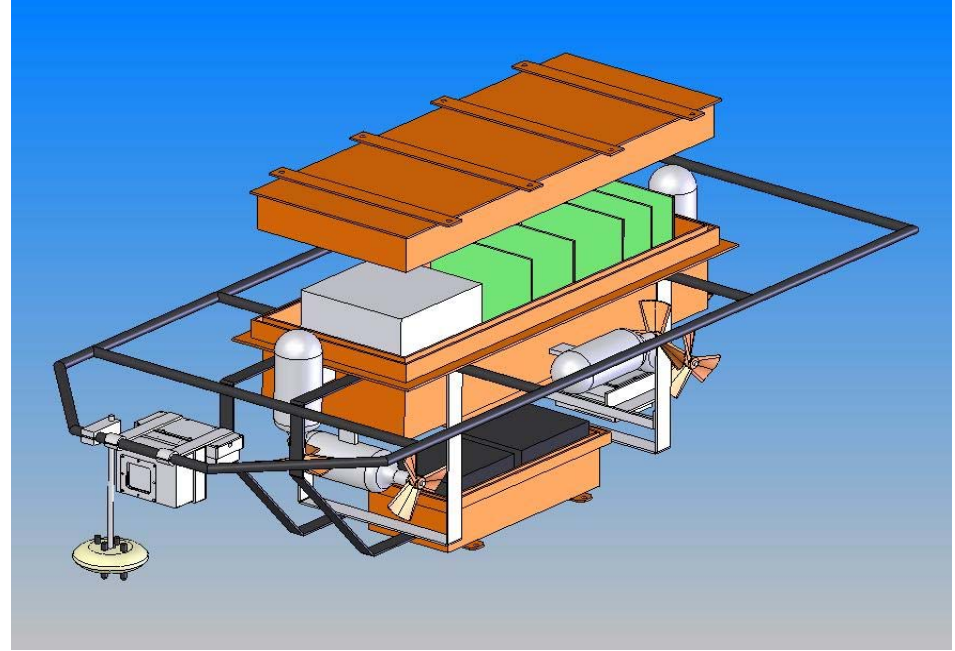


MIT AUV Tasarımı (ORCA VI)

# İnsansız Sualtı Araçları

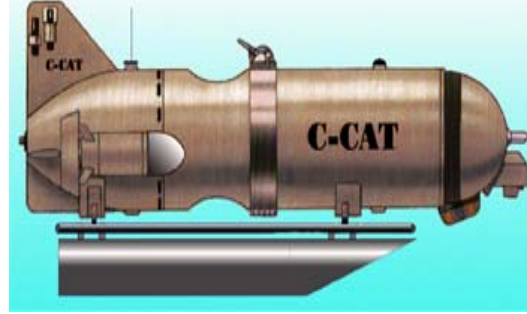


Odyssey IV (MIT)



Quebec University Submarine

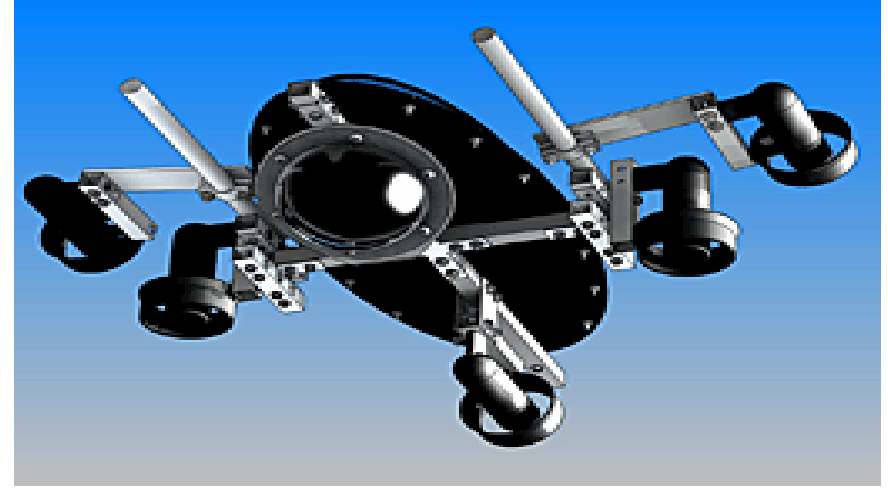
# İnsansız Sualtı Araçları



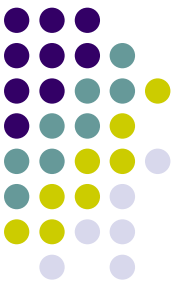
# İnsansız Sualtı Araçları



North Carolina State Uni. AUV Tasarımı  
(Seawolf II)



University of Toronto AUV Tasarımı



# Kullanım Alanları

- Mayın imha
- Keşif
- Sualtı kablolarının döşenmesi
- Sualtı hedeflerine ekipman taşınması
- Deniz ve okyanus suyu incelemeleri
- Petrol ve doğalgaz boruları gözlem ve bakımı
- Sualtı arkeoloji çalışmaları
- Deprem araştırmaları

# ULİSAR

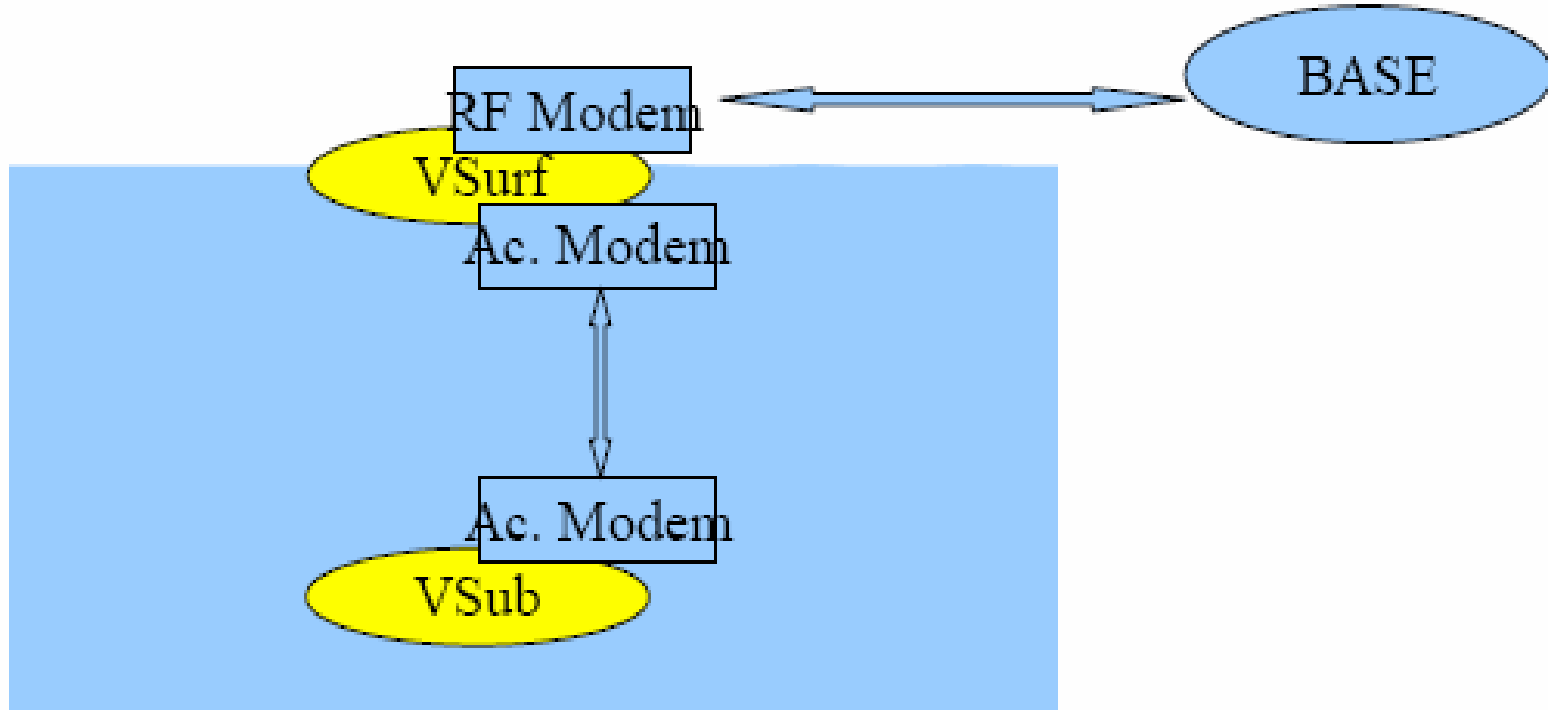
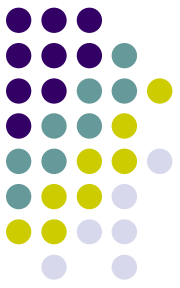


## Temel Fikir:

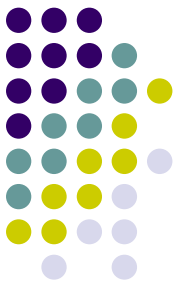
- Sualtı aracı ile beraber uyumlu olarak çalışan bir suüstü aracının varlığı,
- Aradaki haberleşme: Akustik modem kullanarak düşük rezolüsyonlu görüntü ve navigasyon bilgilerinin aktarımı,
- Suüstü haberleşmesi: RF kullanarak.



# ULİSAR



## Önerilen insansız sualtı aracı sistemi



# Tasarım Ögeleri (Gövde)

- Su sızdırmaz bölme tasarımı
- Dış gövde tasarımı
- İtki ve manevra sistemleri
- Denge ve yüzerlik
- Kablolama
- İmalat yöntemi



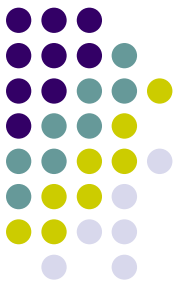
# Dış Gövde Tasarımı

- Dış gövde: aracı en dışta saran yapı
- Ekipmanın korunması, hidrodinamik yapı ve aracın şekli
- Denizaltıların aksine çeşitli şekillerde olabilirler



# Dış Gövde Tasarımı

- Dönme kararlılığı açısından simetrik yapıda
- Ekipmanı kapsayacak ölçüde uzun ve büyük çapta
- Sürüklenme ve sürtünme kuvvetlerini azaltacak ölçüde kısa ve küçük çapta
- Basınca dayanacak ve ağırlığı çok arttırmayacak ölçüde bir et kalınlığında
- Elektronik ekipmanın fazla ısınmasını engelleyecek ölçüde ısı transferine izin veren yapıda
- Modüler yapıda

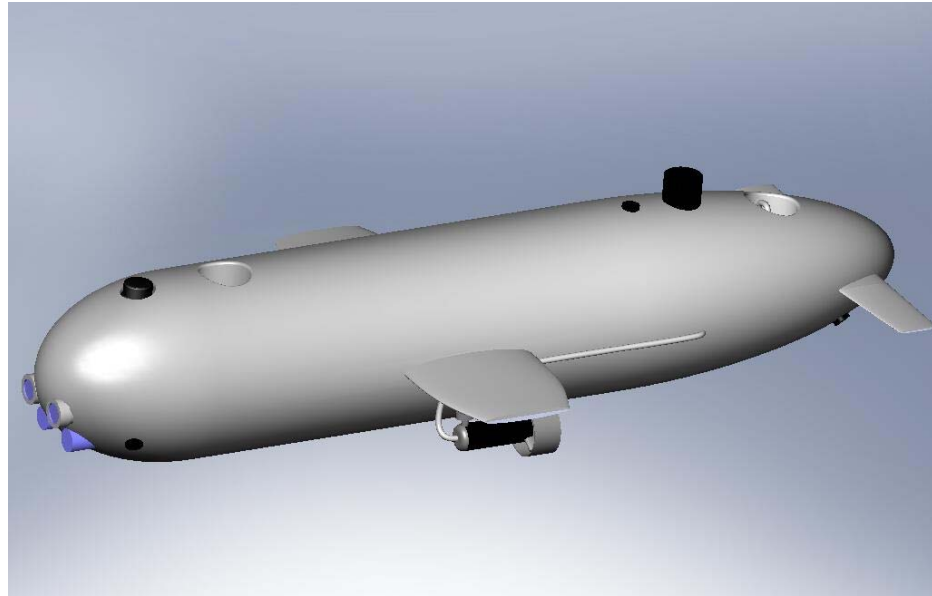


# Dış Gövde Tasarımı

- Düşük karşı kuvvet
- Üretim kolaylığı
- Geniş veritabanı



Silindirik yapı





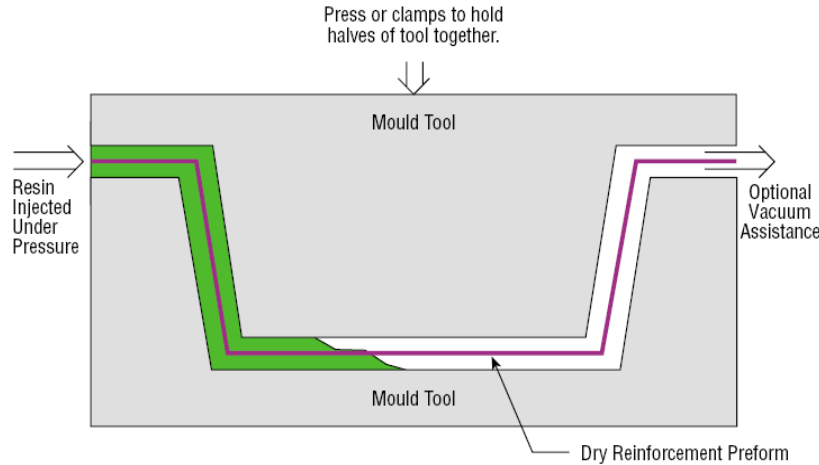
# Malzeme

- Gövde kompozit malzemedan yapılacaktır:
  - Düşük yoğunluklarda yüksek mukavemet
  - Tasarım esnekliği
  - Kalıplama kolaylığı
  - Çeşitli yüzey uygulamalarına izin vermesi
  - Gövde içine metal parçalar yerleştirilebilmesi
  - Yapıştırma işlemleri uygulanabilmesi
  - Barış Elektrik Endüstrisi A.Ş.' in verdiği destek
- Malzeme: Cam dokuma kumaş – epoksi reçine

# Üretim Tekniği



- Reçine Transfer Kalıplama (RTM)
  - Kalıp parçaları arasına elyaf yatırma
  - Basınç ve/veya vakum ile kontrollü reçine besleme
  - Isıl işlem ile polimerizasyon





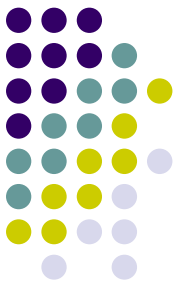
# Üretim Tekniđi - RTM

- Avantajlar:
  - - Yüksek elyaf-reçine oranı
  - - Düşük hava kabarcığı oranı
  - - Düzgün yüzey kalitesi
  - - Tasarım esnekliđi
- Dezavantajlar:
  - - Artan kalıplama masrafı
  - - Büyük parçalara uygulama zorluđu

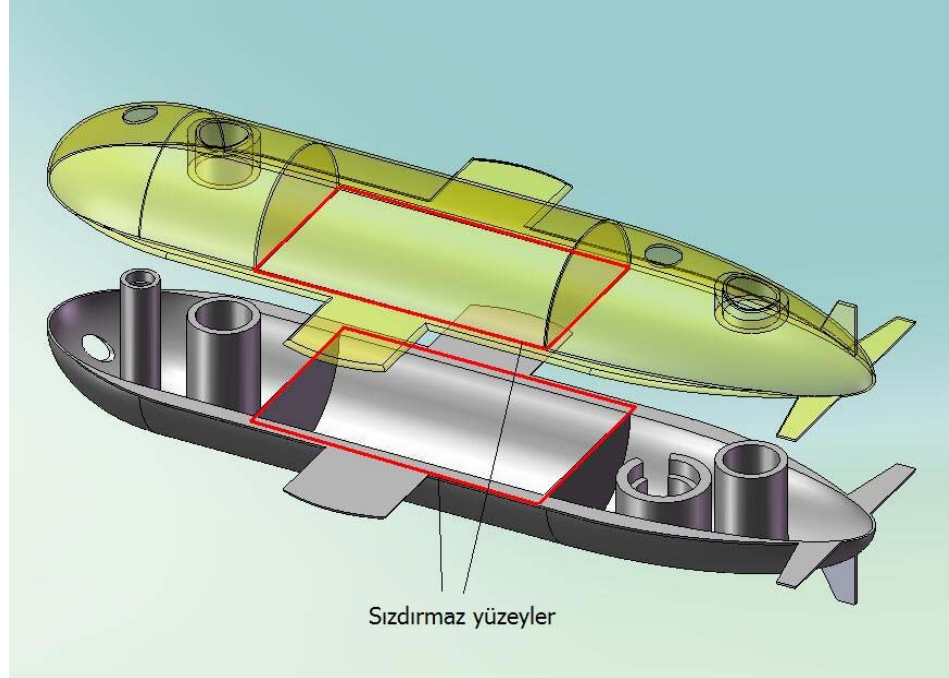


# Üretim Tekniđi - RTM



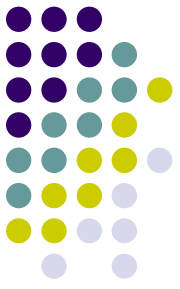


# Dış Gövde Tasarımı

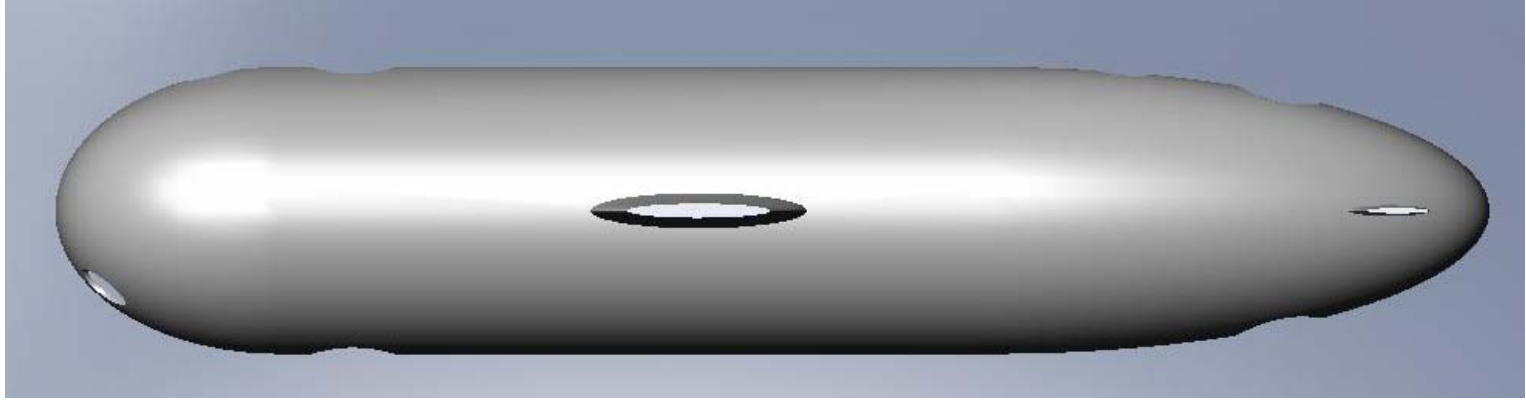


- Maliyeti düşürmek
- Yüksek mukavim yapı elde etmek

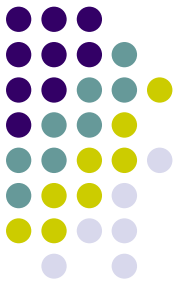
# Dış G?vde Tasarımı



Hidrodinamik yapı:

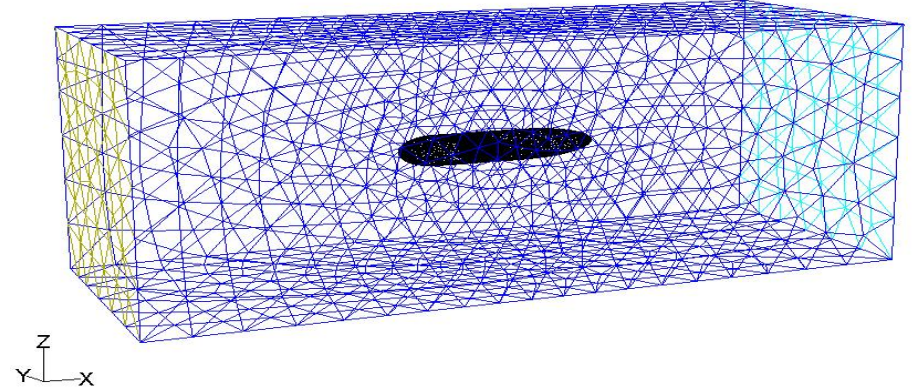


# Dış Gövdenin Tasarımı

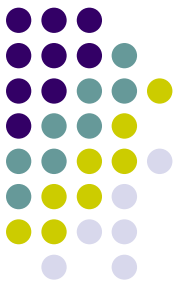


## Hidrodinamik analizler:

- Gambit-Fluent
- Basitleştirilmiş model
- 3B, zamandan bağımsız
- k- $\epsilon$  türbülans modeli
- Sıcaklık ve özellikler sabit

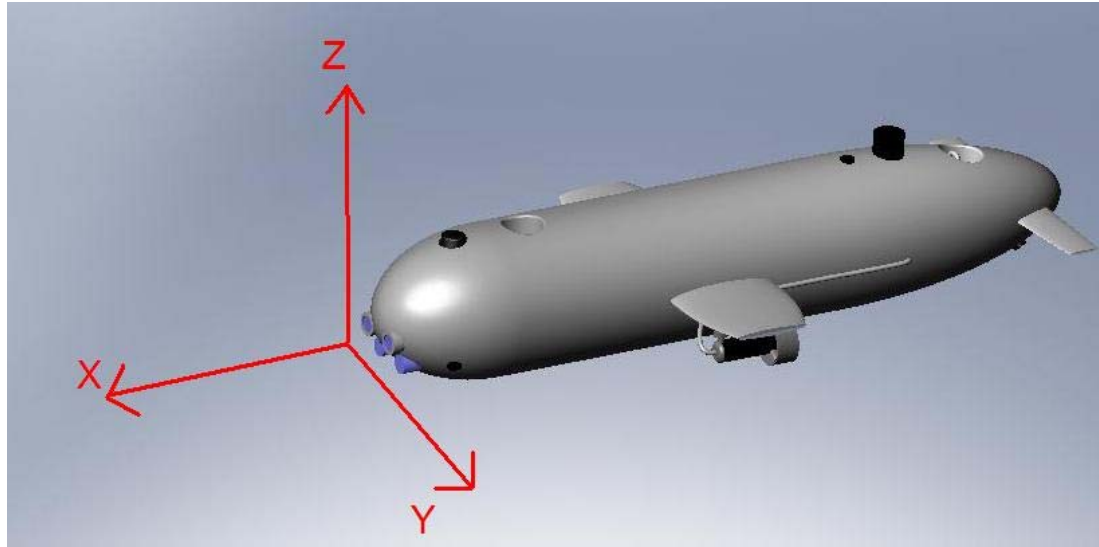


- Analizler sonucu araca etkiyen sürüklenme kuvvetleri elde edilmiş ve bu değerlerin seçilen motorların sağladığı itki değerlerini geçmediği görülmüştür.



# İtki ve Manevra

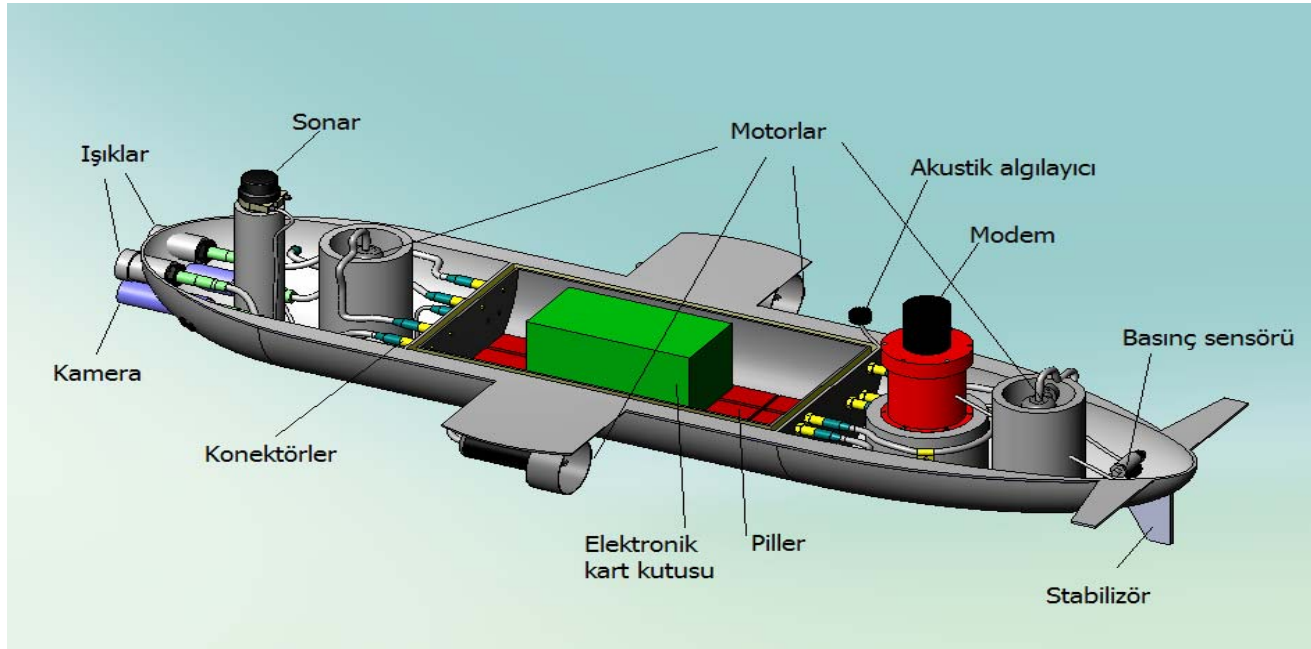
- İleri geri hareket (x eksenini boyunca)
- Yukarı aşağı hareket (z eksenini boyunca)
- Sağa sola dönüş (z eksenini etrafında)
- Yukarı aşağı dönüş (y eksenini etrafında)

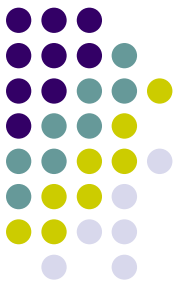


# İç Yerleşim ve Isınma Problemi



- İç yerleşimde önemli nokta: ağır parçaların alt bölümlere yerleştirilmesi





# Isınma Problemi

- Isınma probleminin öncelikli olarak sızdırmaz bölgede gerçekleşebileceği tahmin edilmektedir.
  - Denetim ekipmanlarının ve pillerin bulunması
  - Et kalınlığının fazla olması
  - Kompozit malzemenin yalıtkan özelliği
- Deneyler ve analizler gerçekleştirilmiştir.

# Isınma Problemi

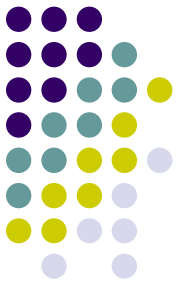


Su geçirmez bölümde pillerin haricindeki donanım elemanlarının güç tüketimleri

Donanım Eleman Adı:	Adet	Güç Tüketimi(W)	Toplam Güç(W)
Parvus CPU-1462	2	13.00	26.00
Parvus ACS-5150 DC-DC çevirici	2	4.00	8.00
Parvus CTR-1461 PCMCİ çevirici	2	0.35	0.70
Parvus CTR-1475 MPEG 4 sıkıştırıcı	1	10.00	10.00
Microstrain 3DM-GX1 IMU	1	0.78	0.78
Z-World RCM 4000 Mikrodenetleyici	1	0.10	0.10
Diğer	N/A	5.00	5.00
Kokam 2000mAh LiPo hücreler	-----	0.10	-----
Piller hariç toplam			50.58



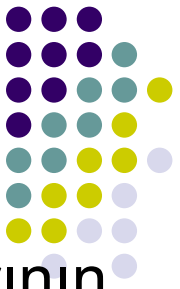
# Isınma Problemi



Pillerde harcanan güç değerleri

Adı	Eşdeğer direnci( $m\Omega$ )	Çekilen akım (mA)	Güç tüketimi(W)
Pil Paketi 1	4.20	12000.00	0.60
Pil Paketi 2	16.80	12000.00	2.42
Pil Paketi 3	16.80	12000.00	2.42
Toplam			5.44

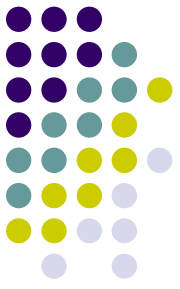
# Isınma Problemi



Su geçirmez bölümde bulunacak olan donanım elemanlarının çalışma sıcaklıkları

Donanım Eleman Adı:	Sıcaklık aralığı (°C)	En yüksek sıcaklık (°C)
Parvus CPU-1462	-40 - 85	80
Parvus ACS-5150 DC-DC çevirici	-40 - 85	80
Parvus CTR-1461 PCMCİ çevirici	-40 - 85	80
Parvus CTR-1475 MPEG 4 sıkıştırıcı	-40 - 85	80
Microstrain 3DM-GX1 IMU	-40 - 70	65
Z-World RCM 4000 Mikrodenetleyici	0 - 70	65
Diğer	-----	-----
Kokam 2000mAh LiPo hücreler	-20 - 60	55

# Isınma Problemi

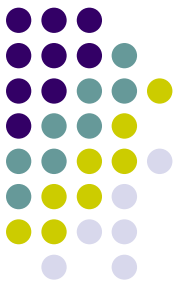


Bu durumda, eğer diđer donanım elemanlarının çalışma sıcaklıkları 60 °C 'ın üstünde olursa, sualtı aracının su geçirmez bölümünün sıcaklığı 55 °C'ın altında olmalıdır.



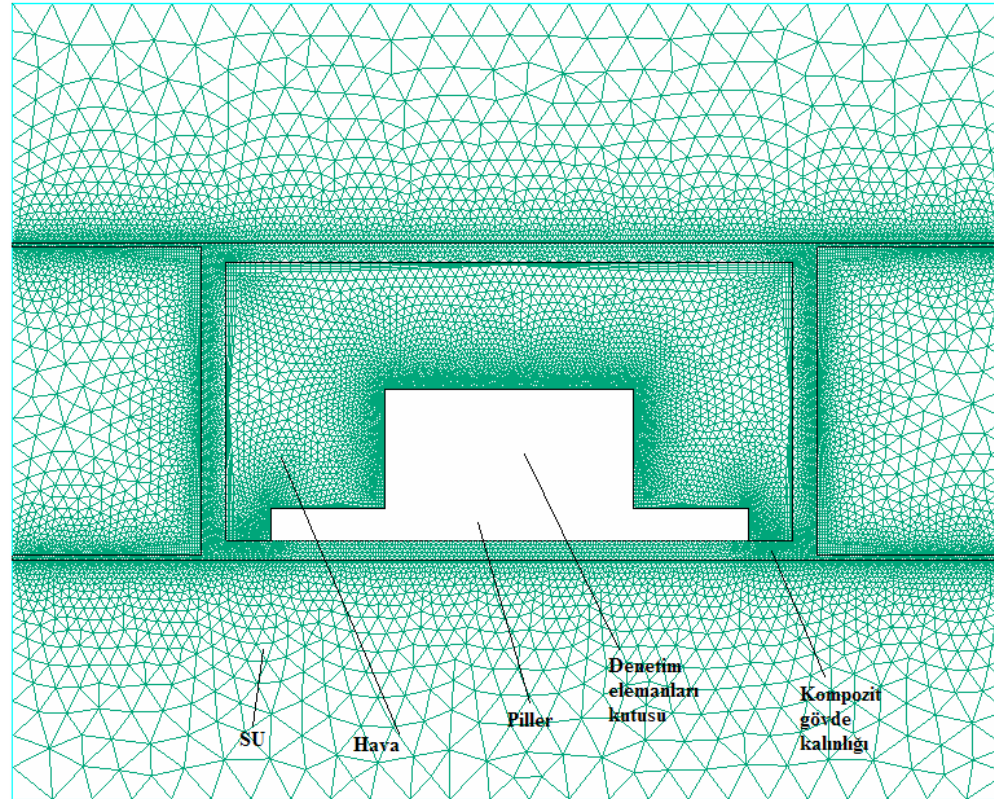
# Isınma Problemi - Analizler

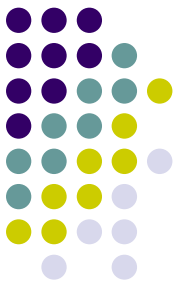
- Gambit ve FLUENT® yazılımları kullanılarak aracın çalışma süresi olarak düşünülen 2 saatlik sürelerde analiz edilmiştir.
- Zamana bağlı 3 boyutlu doğal konveksiyon ısı problemlerinin çözümü yüksek hızlarda bilgisayarlar gerektirdiğinden, problemler öncelikle 2 boyutlu çözülmüştür.
- 2 boyutlu analizlerden elde edilen sonuçlar tatmin edici olmaz ise 3 boyutlu analizlere başvurulacaktır.



# Isınma Problemi - Analizler

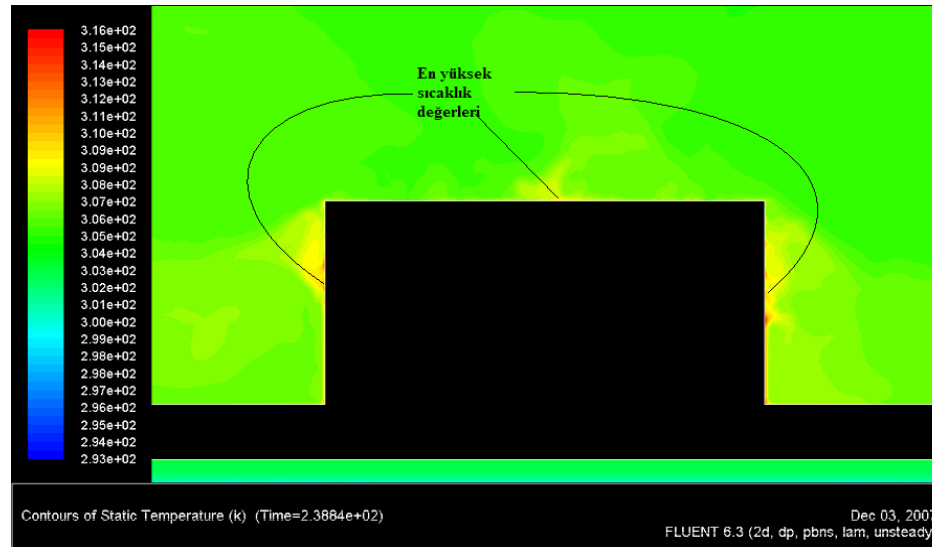
- Kritik olması beklenen sızdırmaz orta bölüm için analizler

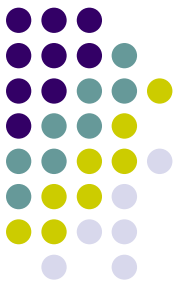




# Isınma Problemi - Analizler

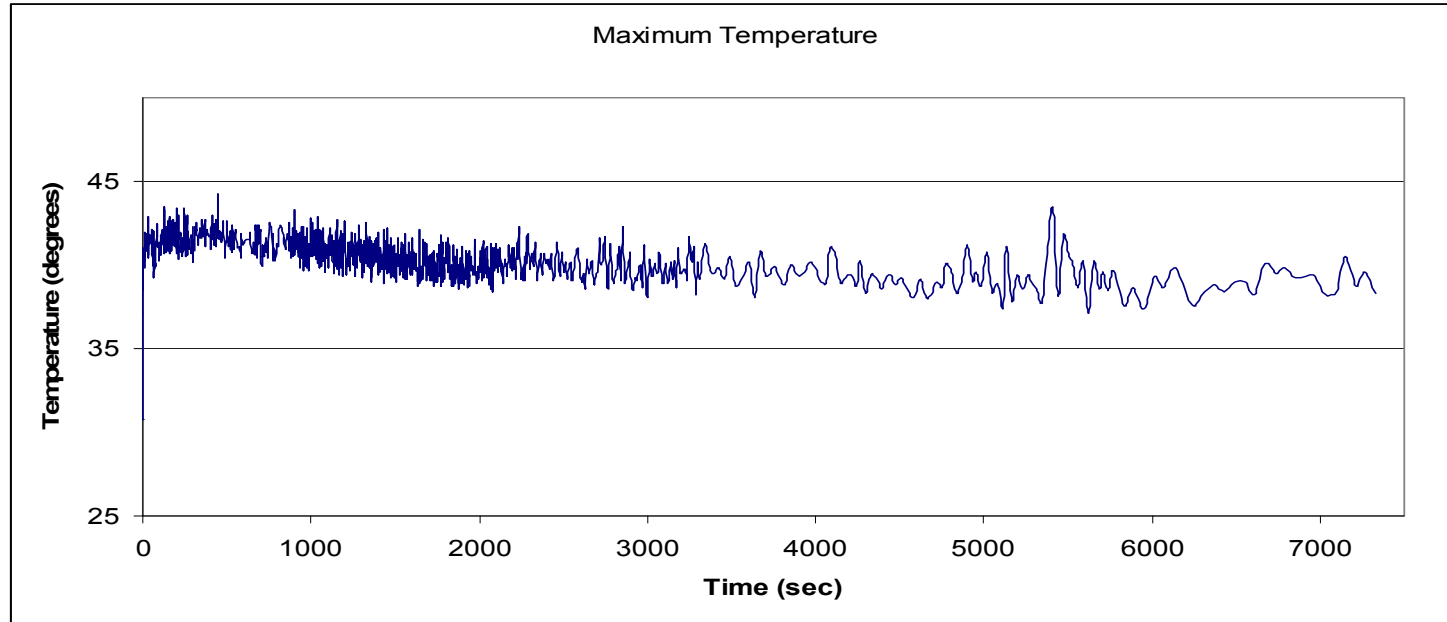
- Sınır ve başlangıç koşullar için su ve havanın alması beklenen sıcaklık değerleri kullanıldı.
- Analiz esnasında her hangi bir anda sıcaklığın, maksimum çalışma sıcaklığını geçmemesi kontrol edildi.





# Isınma Problemi - Analizler

- En kritik noktanın sıcaklık-zaman eğrisi:

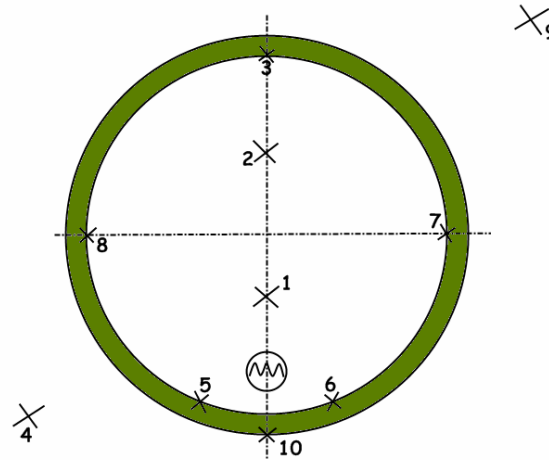


- En yüksek sıcaklık 36-37 derece arasında



# Isınma Problemi - Deneysel

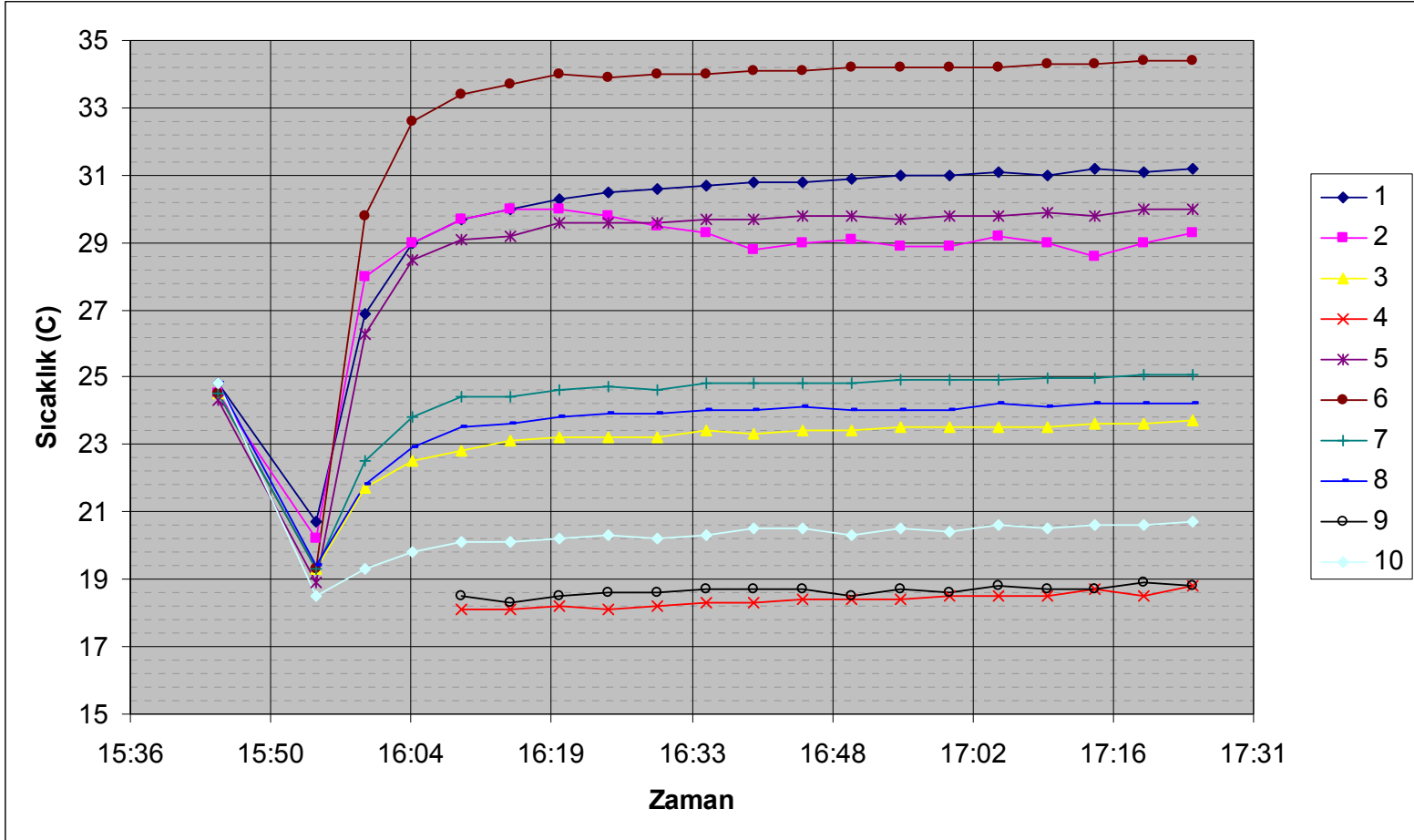
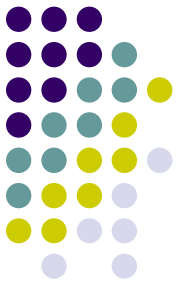
- Analizlerin doğrulanması amacıyla üretici firma tarafından yapılan ısı analiz testleri

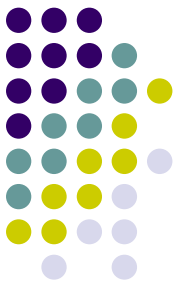


- Çeşitli noktaların sıcaklık-zaman eğrileri



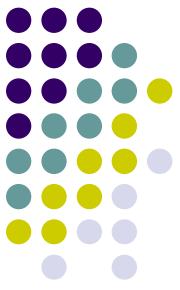
# Isınma Problemi - Deneysel





# Isınma Problemi - Sonuç

- Deney ve analiz sonuçlarının yakınlığı
- Bu sonuçlara göre araç içinde bir ısınma problemi beklenmemektedir.
- Buna rağmen araç içi ısı almaçları ile aracın çalışması sırasında ölçüm.
- Isınma problemleri yaşanması durumunda ısı borusu gibi ek çözümler gündeme gelebilir.



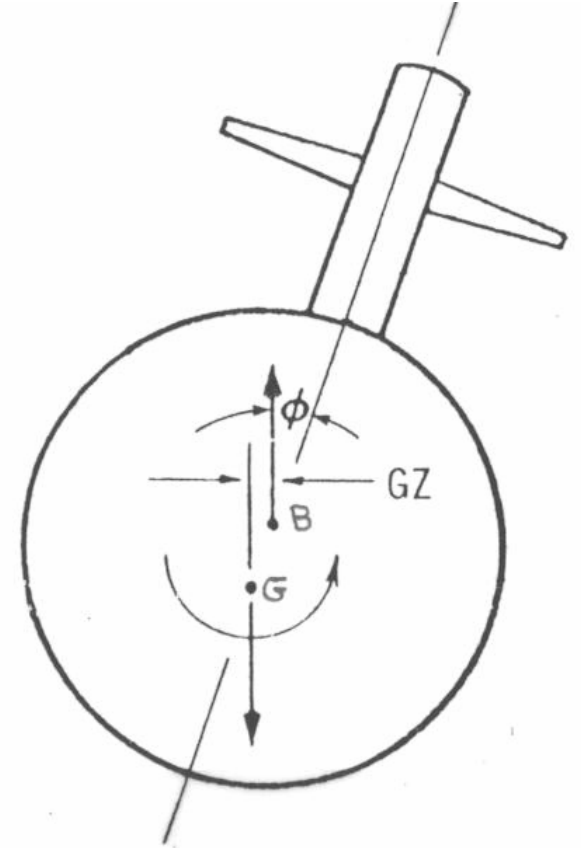
# Denge, Kararlılık ve Yüzerlik

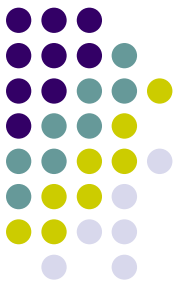
- Yüzerlik özelliğine sahip araçlar suda asılı kalabilmektedir. Bu sayede hareketsizken güç harcanması yoktur.
- Olası iletişim problemlerinde aracı kaybetmemek için araç 0.5-1 kg arasında kaldırma kuvvetine sahip olacak şekilde tasarlanabilir.
- Denge ve kararlılık ise aracın istenilen şekilde durabilmesi ve hareket edebilmesi açısından önemlidir.
- Aracın denge ve kararlılığında ağırlık ve kaldırma kuvveti merkezlerinin konumları belirleyicidir.



# Denge, Kararlılık ve Yüzerlik

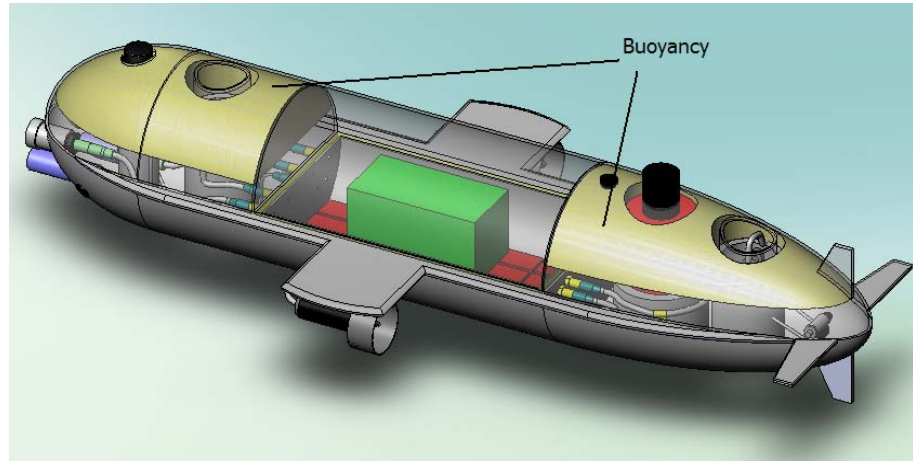
- Denge açısından iki merkezin x ve y konumları aynı olmalıdır.
- Kararlılık açısından ise kaldırma kuvveti merkezi, ağırlık merkezinin mümkün olduğunca üstünde olmalıdır
- İki merkez arasındaki bu mesafe aracın maruz kaldığı dönme etkilerine karşı düzeltici etkide bulunup kararlılık sağlamaktadır



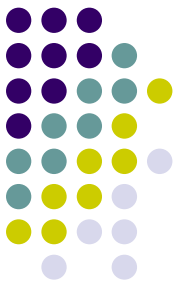


# Denge, Kararlılık ve Yüzerlik

- Araca belirtilen özellikleri sağlamak amacıyla alt bölümlere metal ağırlıklar, üst kısımlara ise yüksek yoğunluklu su emmeyen köpük eklenmektedir.



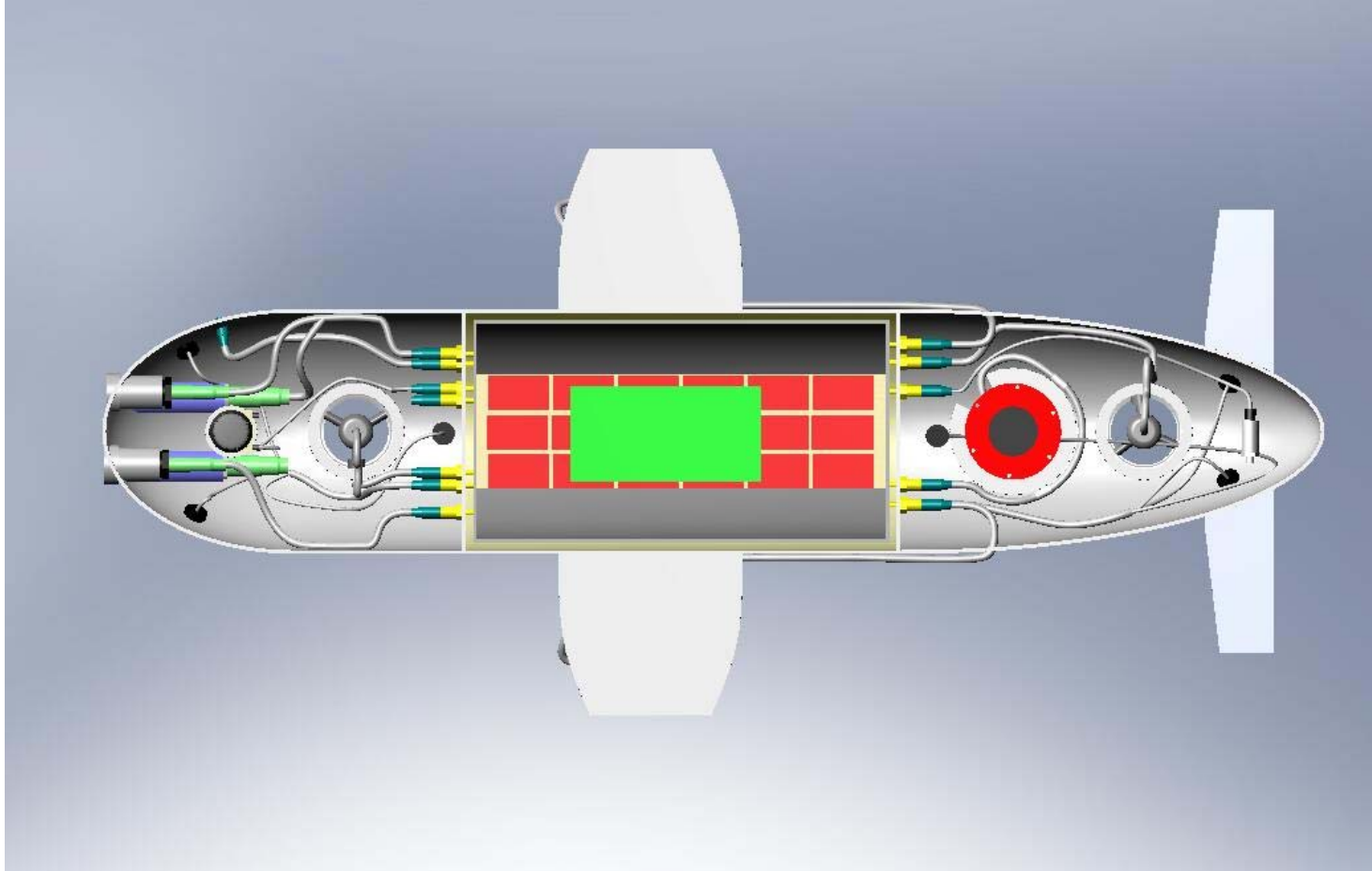
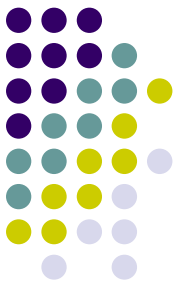
- Bu işlemler sonucu ağırlık ve kaldırma kuvveti merkezleri yaklaşık 3 cm kadar ayrılmışlardır.



# Kablolama

- Denetim elemanları ve bataryalar su geçirmez bölümde saklanırken diğer cihazlar su geçirmez olduklarından su alan bölümde bulunmaktadır.
- Su geçirmez kablolar, aracın su geçirmez orta bölümü ile su alan bölümündeki cihazlar arasındaki bağlantıyı sağlamaktadır.
- Su geçirmez konektörler ise su geçirmez bölüm ile su alan bölümü ayıran duvardan geçişi sağlamaktadır.

# Kablolama



# Üretim



Model tasarımı ve imalatı



Kalıp üretimi



Gövde üretimi



Testler



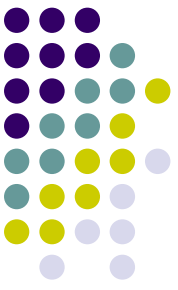
# Model



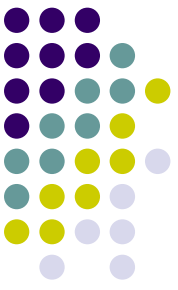
# Kalıp



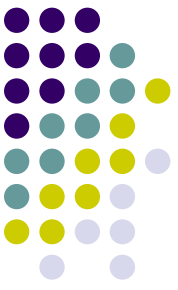
# Gövde Üretimi



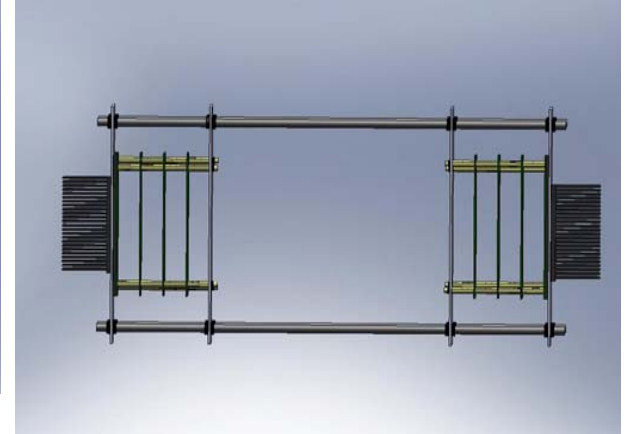
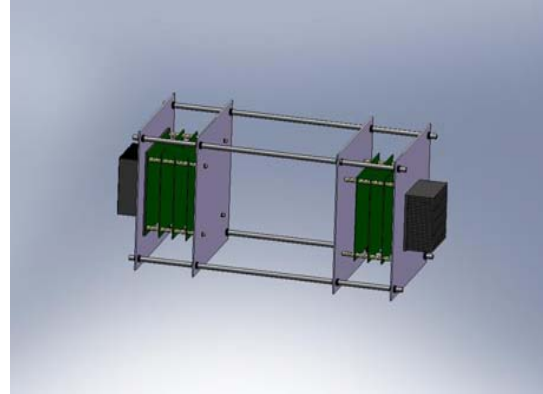
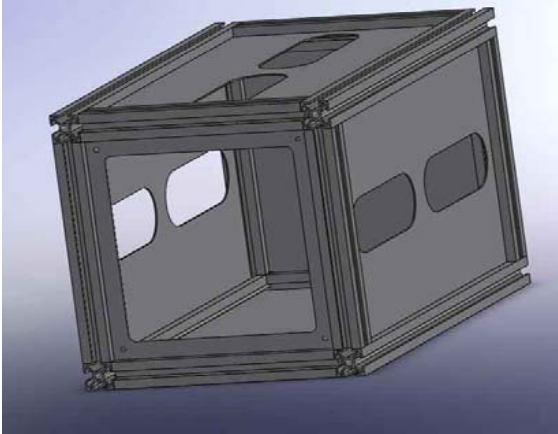
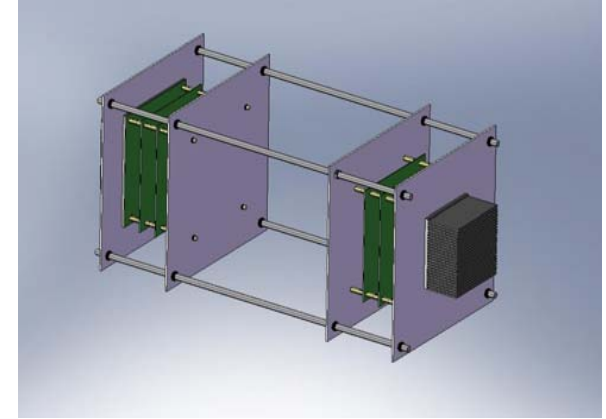
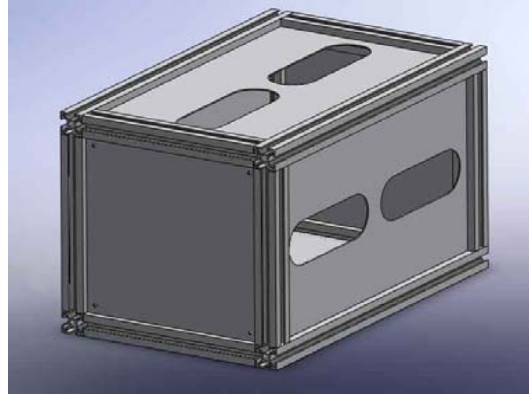
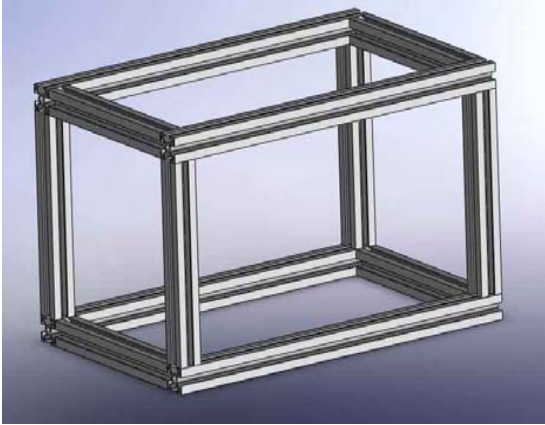
# Gövde Üretimi



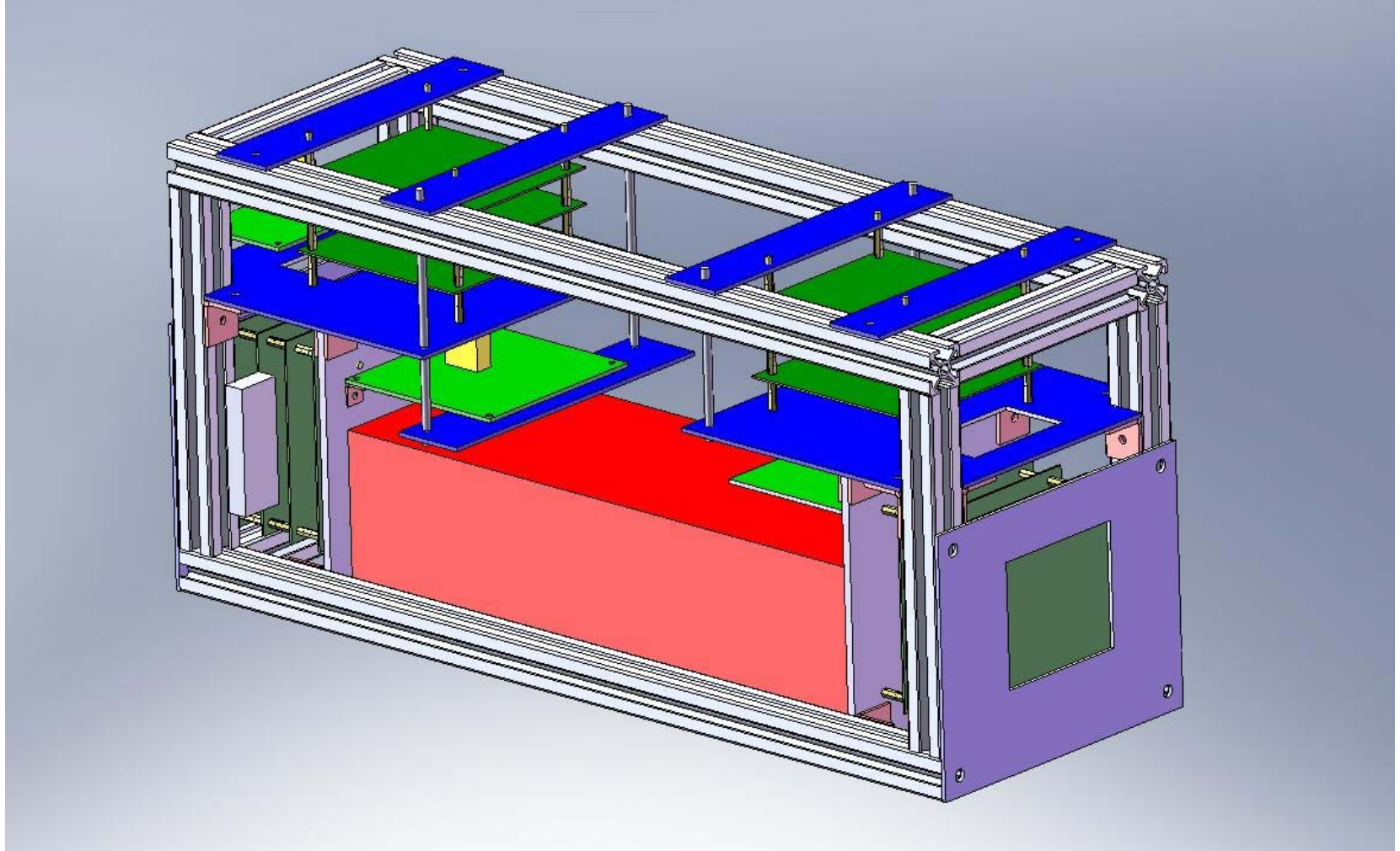
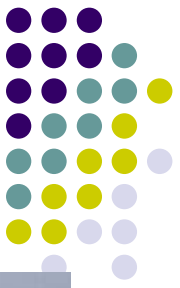
# Gövde Üretimi



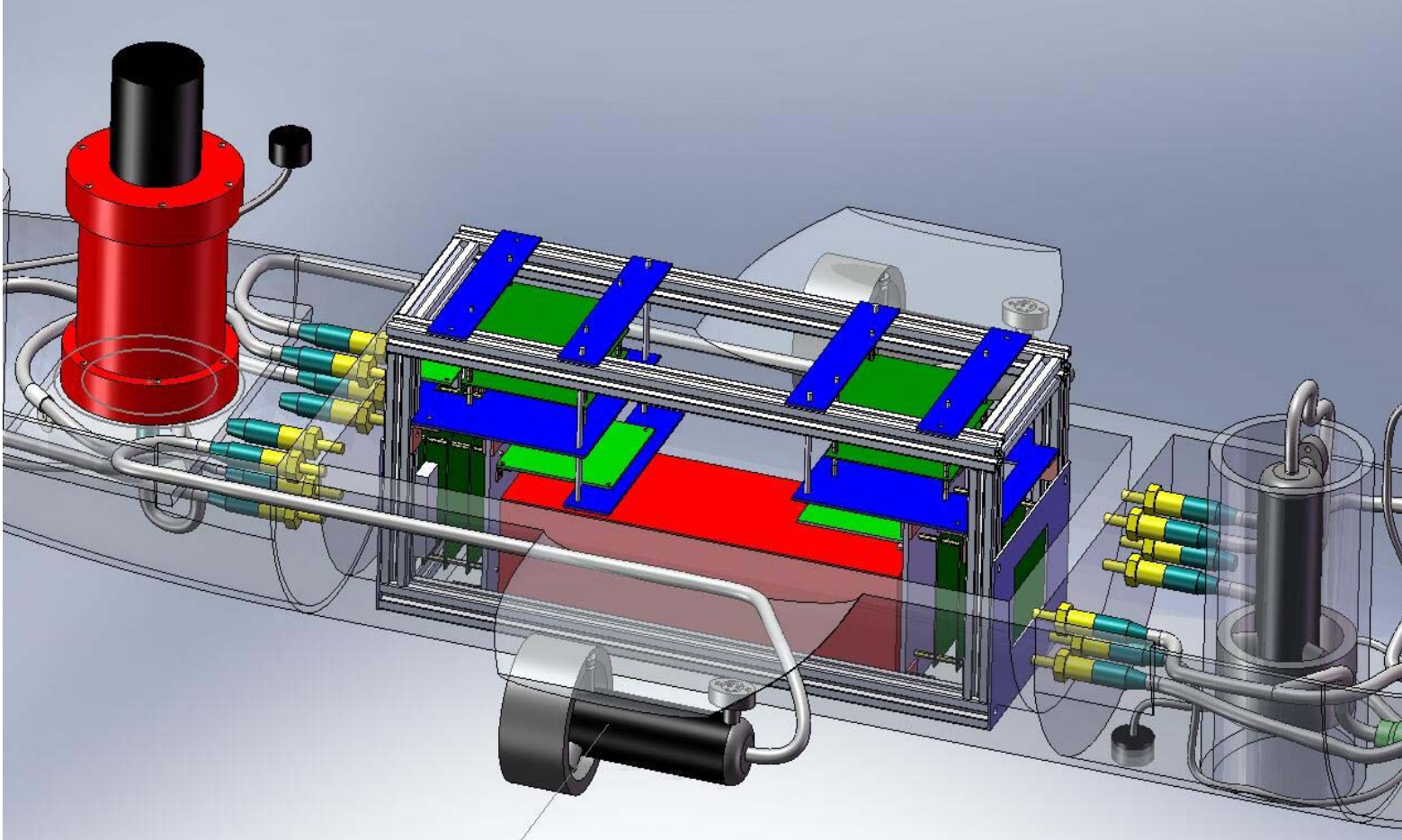
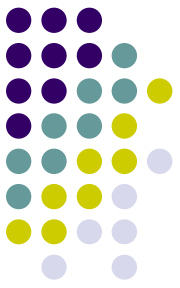
# Elektronik Kutusu Tasarımı



# Elektronik Kutusu Tasarımı



# Elektronik Kutusu Tasarımı





# Arayüz Tasarımı



UlisarGUI

**Sensor Degerleri**

Basinc Sensoru: 0 Mesafe Sensoru 2 0  
Nem Sensoru: 0 Mesafe Sensoru 4 0  
Mesafe Sensoru 1 0 Mesafe Sensoru 5 0  
Mesafe Sensoru 2 0 Mesafe Sensoru 6 0

**Denizalti Durumu**

ACIL DURUM

**Sonar Kontrol Ekranı**

Sonar Aktif

**Motorlar**

On Motor M.D: 0  
Sol Motor M.D: 0  
Sag Motor M.D: 0  
Arka Motor M.D: 0

**IMU Bilgileri**

**Kamera ve Isik Kontrol**

Isiklar Acik  
 Kamera 1 Acik  
 Kamera 2 Acik

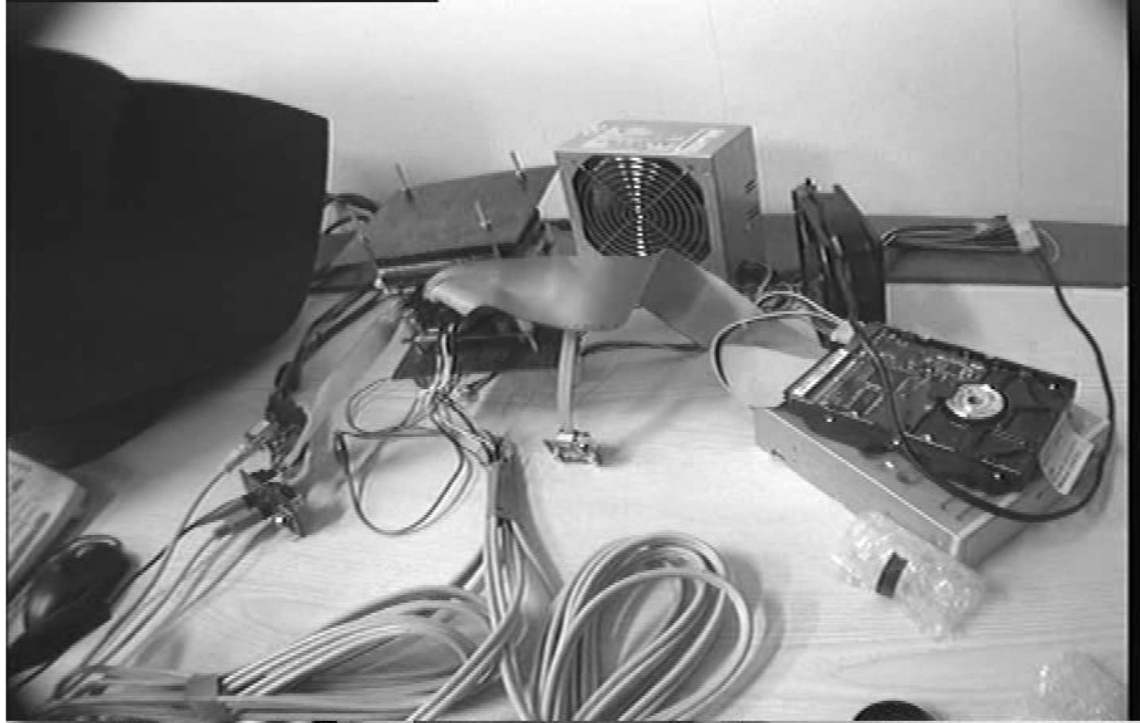
**Batarya Voltaj Seviyeleri**

1. Batarya Grubu 0 3. Batarya Grubu 0  
2. Batarya Grubu 0 4. Batarya Grubu 0

**Ag Durumu**

Kamera 1 bulunamadi  
Kamera 2 bulunamadi

# Kameralar

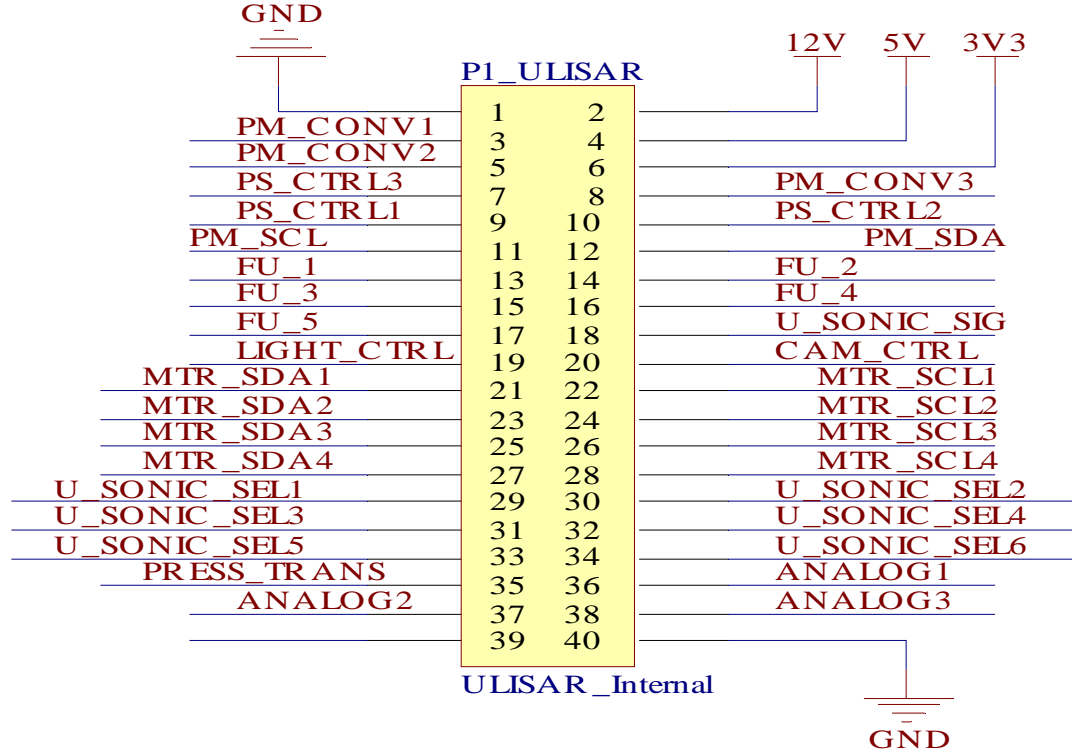




# Donanım

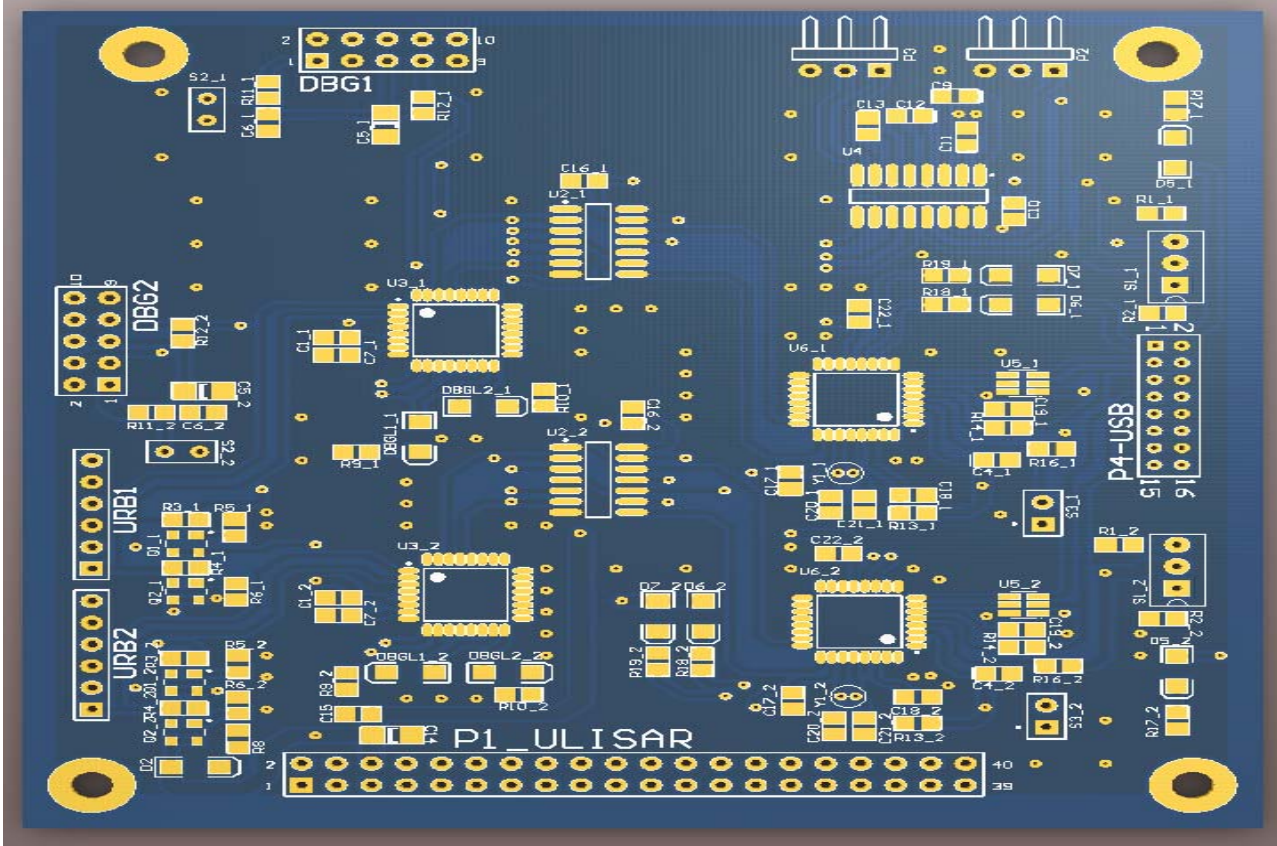
- Çok sayıda kart tasarımı,
- Karmaşık yapı,
- İmalat güçlüğü,
- Zaman alıcı çalışmalar.

# Donanım



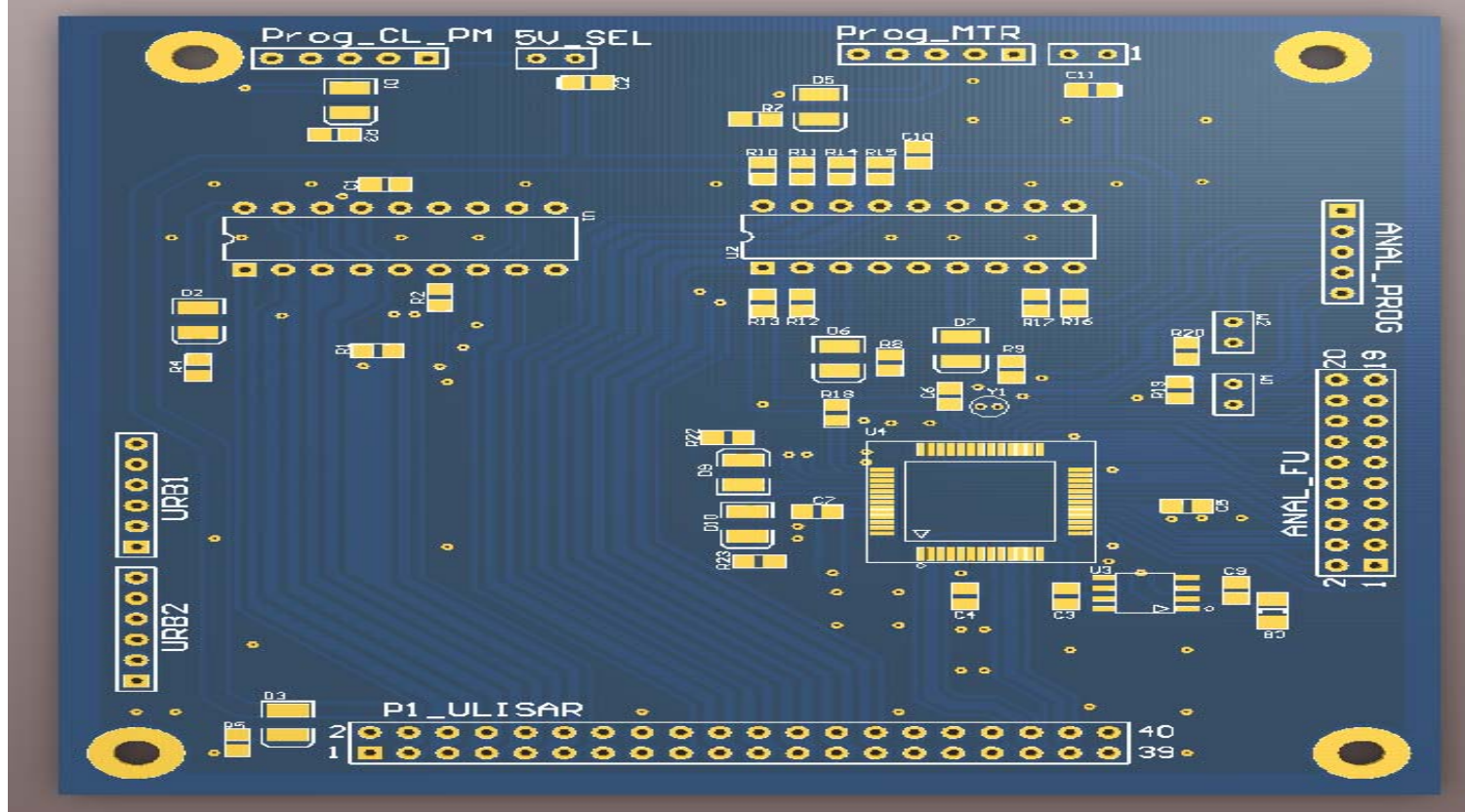
## ULISAR Konnektörü

# Donanım



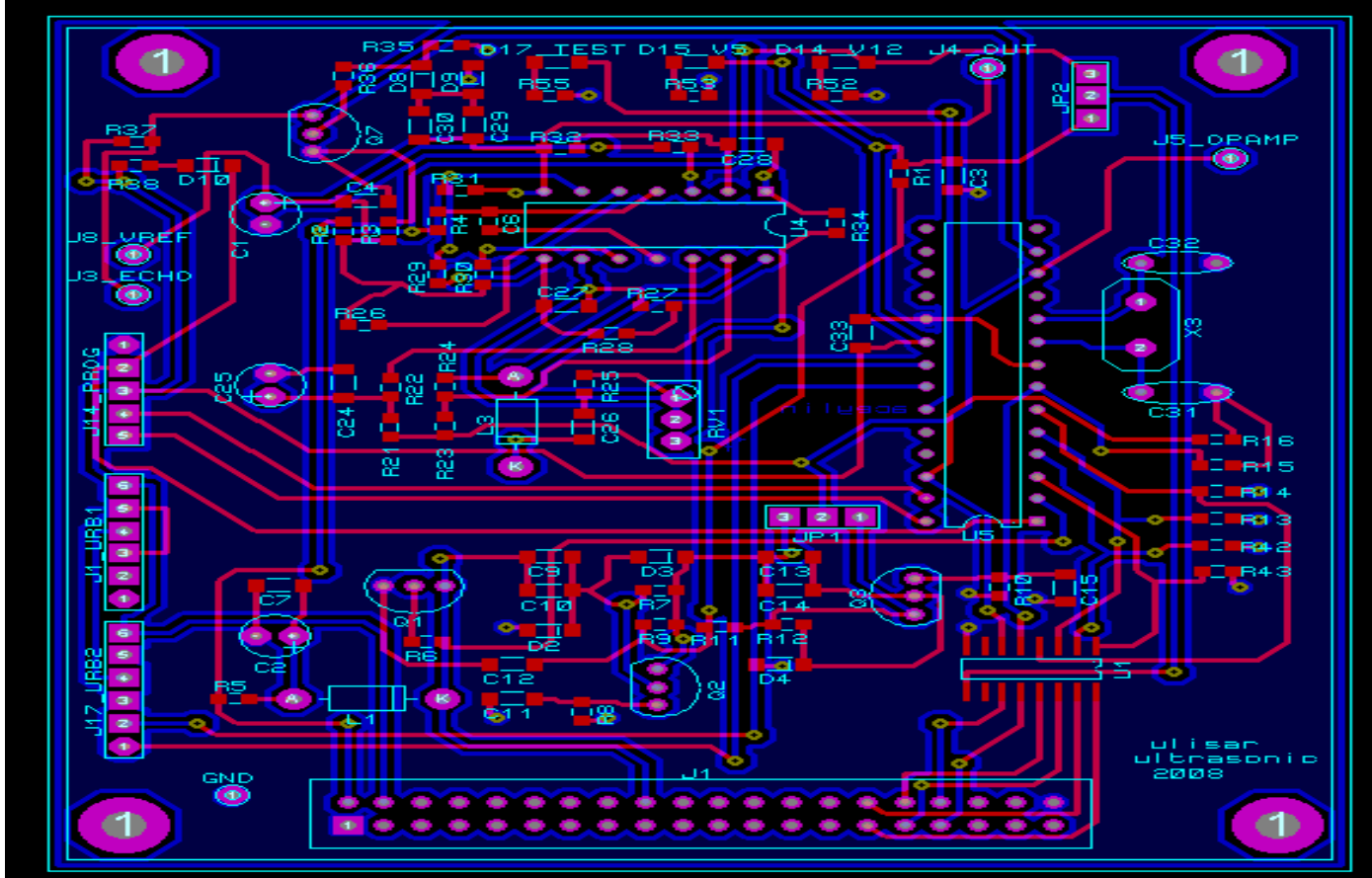
URB Köprü kart görünümü

# Donanım



Güç – Kamera- Işık – Motor – Analog Almaç Kartı Görünümü

# Donanım



## Ultrasonik Almaç Düğümü

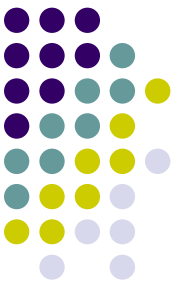




# Gömülü Yazılım Çalışmaları

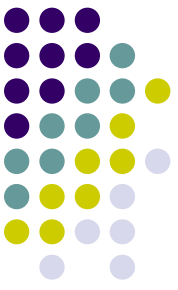
- Görüntü Bilgisayarı Yazılımı,
- Sualtı Kameralarından Görüntü Alınması,
- Görüntü Verisinin Operatör Bilgisayarına Gönderilmesi,
- İşletim Sistemi Görüntüsünün Hazırlanması ve Yüklenmesi,
- Açılış Sürecinin Özelleştirilmesi,
- Görüntü Bilgisayar Yazılımının Derlenmesi ve Yüklenmesi,
- Operatör Bilgisayar Yazılımı,
- Video Verisinin Çözümlemesi,
- Sonar Ekranının Yazılımı,
- Operatör Arayüzünün Derlenmesi ve Çalıştırılması,
- Denetim Bilgisayar Yazılımı.





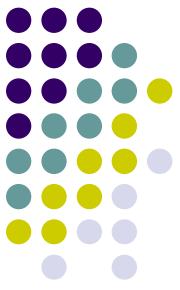
# Sonu

- Mekanik tasarım evresi projenin başarısı aısından son derece nemlidir.
- Bu evrede eřitli tasarım ğeleri gz nnde bulundurulmuř ve mhendislik alıřmaları gerekleřtirilmiřtir.
- Proje alıřmalarında makina, elektronik ve bilgisayar mhendislerinden oluřan bir disiplinlerarası proje grubu grev almıřtır.
- Gereken noktalarda analizler ve deneyler gerekleřtirilmiřtir.



# Sonuç

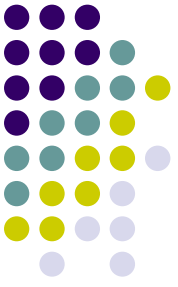
- Çok fazla yapılacak iş var.
- Navigasyon çalışmaları konusunda bilinenlerin ötesini zorlamak gerekiyor,
- Çalışmaların hiçbir zaman sonlandırılmaması gerekli,
- Bu konuda ülkemizin çok eksikliği var,
- Aynı zamanda bu konulara ihtiyaç çok fazla,
- Sadece askeri amaçlar değil sivil ihtiyaçlar düşünüldüğünde pazar çok büyük.



# Sonuç

- Projede elde edilen tecrübe ve bilgi birikiminin önümüzdeki yıllarda gerçekleşebilecek benzer projelerde faydalı olması beklenmektedir.
- Böyle bir çalışmanın sonunda görüntü işleme, sonar/kamera görüntülerinin birleştirilmesi, yol planlama, vs., gibi takip eden çalışmalar yapılabilir.

# Teşekkür



Projede maddi desteklerinden ötürü TÜBİTAK' a, elektronik kartların üretimindeki ve proje boyunca farklı konulardaki desteklerinden dolayı Yenel LTD. Şirketine ve dış gövdenin tasarımı ve üretimindeki katkılarından dolayı *Bariş Elektrik Endüstrisi A.Ş.*'ye teşekkürlerimizi sunarız.

## Dinlediğiniz için teşekkürler

# Online Calibration of Strapdown Magnetometers

Hüseyin Yiğitler\*, M. Kemal Leblebicioğlu\*\*

\* Dept. of Electrical and Electronics Engineering of Middle East Technical University,  
Ankara - Türkiye (phone: 90-312-2104403, e-mail: hyigitler@eee.metu.edu.tr)

\*\* Dept. of Electrical and Electronics Engineering of Middle East Technical University,  
Ankara - Türkiye (e-mail: kleb@metu.edu.tr)

---

**Abstract:** In this work, an online calibration algorithm for strapdown magnetometers is proposed. The proposed method is based on the observation that the magnetometer's measurements are on an ellipsoid manifold. The algorithm is attitude independent and makes use of Extended Kalman Filter.

**Keywords:** Calibration, Magnetometer Calibration, Geometric Calibration

---

## 1. INTRODUCTION

Magnetometers are widely used as heading reference devices in Automatic Heading Reference Systems (AHRS). Recently, the usage of magneto-resistive sensors in AHRS systems has become popular as they are alternative low cost solution as absolute heading measuring devices, [1]. However, the usage of these devices is limited due to the performance dependence on the environment.

Since the magnetometers measure the Earth's magnetic field, their sensitivity to neighboring electrical activities and ferromagnetic materials is very high. Due to this fact several calibration schemes have been proposed in the literature. Although, an off-line calibration of magnetometers are required to correct magnetic field anomalies due to the neighboring magnetic activities, the calibration parameters may deviate from the predetermined values when the operating environment is changed. Consequently, the effective usage of magnetometer measurements requires an online calibration algorithm.

An online calibration algorithm needs to fulfill some important requirements. To be able to use magnetometers in strapdown navigation systems, the online calibration algorithm is required to be attitude independent due to the fact that the magnetometers are used in the attitude determination algorithms. The second requirement is the low computational cost. The most important requirement is traceability of the error on the output of the calibration procedure. Certainly, there are more requirements such as convergence, efficiency and consistency, which we assume to be satisfied by all of the algorithms.

The attitude independent calibration of magnetometers has been studied in the literature, [2],[3],[4],[5],[6]. A survey of attitude independent magnetometer bias determination works until 2002 is given in [7]. These works contain the methods for estimation of magnetometer biases in spacecrafts and all

of them are used in some missions. In [7], the differences and drawbacks of the algorithms aiming to estimates the magnetometer biases are shown in a theoretical manner. It concludes that the TWOSTEP algorithm proposed in [5] is superior to the others. In [4], this method is extended to the calibration of the magnetometers. Moreover, in [3], this method is used in the recursive online calibration procedure.

Recently, some offline calibration procedures have been proposed, [8][9][10]. The problem formulations in these works are similar to each other and it highlights the geometrical basis of the subject. In [11], magnetic field disturbance compensation is integrated with the attitude estimation filter.

As will be shown later, the magnetometer calibration algorithms requires determination of magnetic bias and scale factors. The determination of magnetometer bias is not an easy task and requires minimization of a quartic function. Consequently, this problem has been studied in the literature separately. Moreover, the solution of this problem is extended to cover the solution of determination of magnetometer scale factors.

In this work, an online calibration algorithm for strapdown magnetometers is proposed, which is based on the observations that the magnetometer's measurements are composed of a magnetic bias, and the measurements are on an ellipsoid manifold because of the axis scales. As will be reasoned later, estimation of the magnetic bias and the scale factors can be treated separately, which allows developing a calibration algorithm as a combination of two different methods. Using this fact, proposed method makes use of TWOSTEP algorithm to estimate the magnetic bias and the scales are estimated using an Extended Kalman Filter.

The present work does not follow the steps introduced in [3] and [4]. Instead, the theory of the algorithm is more closely related with the works that treat the problem by geometrical

means, [8][9]. Nonetheless, this work aims to find the calibration parameters online and update the scale factors in a recursive manner satisfying the requirements specified above.

The following contains three parts. First, the error sources in magnetometers are introduced. Secondly, the problem formulation is given. And, lastly, some simulation results are demonstrated.

## 2. MAGNETOMETER MEASUREMENT ERRORS

### 2.1. Hard Iron Error

These errors are considered as the total effect of time invariant magnetic fields in the vicinity of the sensor. The magnetic field measured by the sensor is sum of the true Earth magnetic field and the constant magnetic fields due to hard iron effects. As a result, the measured magnetic field in the body frame is given by,

$$\bar{m}^b = \hat{C}_e^b \bar{m}_0^e + \bar{b}_E \quad (1)$$

where  $\bar{m}_0^e$  is the Earth magnetic field and assumed to be constant in this work,  $\bar{b}_E$  denotes hard iron effects in the vicinity of the sensor,  $\bar{m}^b$  is total magnetic field contaminated with hard iron effects in the body frame.

### 2.2. Soft Iron Error

Soft iron error is due to change in the Earth's magnetic field depending on the orientation of the sensor. These errors are represented by a matrix in  $\mathbb{R}^{3 \times 3}$ ,  $C_S$ . Denoting the transformation matrix from Earth reference frame to body frame by  $\hat{C}_e^b$ , the measured magnetic field is given by,

$$\bar{m}_{HS} = C_S \hat{C}_e^b \bar{m}_0^e + \bar{b}_E \quad (2)$$

Since  $\hat{C}_e^b$  represents the attitude of the body,  $\hat{C}_e^b$  is in  $SO(3)$ .

### 2.3. Non-Orthogonality Error

The misalignment error is due to non-orthogonal placement of three magnetic sensors. This error is represented by another transformation in  $\mathbb{R}^{3 \times 3}$ ,  $C_M$ . Adding the effects of sensor misalignment, the magnetometer measurement in body frame is given by,

$$\bar{m}_{NO} = C_{NO} \bar{m}_{HS} \quad (3)$$

In [1], it is shown that  $C_{NO}$  is in  $SO(3)$ .

### 2.4. Scaling and Bias Error

These errors are due to imperfections in the sensors. Bias is due to null shifts in the sensors and represented by a constant vector,  $\bar{b}_b$ . The scaling is due to the fact that the sensors on each axis do not have the same response. This error is represented by a positive definite diagonal matrix,  $D_S$ .

### 2.5. Wideband Noise

In addition to the other error sources, there magnetometer measurements are contaminated by wide band noise process,  $\bar{v}_k$ . In this work, this noise process is assumed to be 0 mean white Gaussian noise,

$$\bar{v}_k \sim \mathcal{N}(0, \Sigma_k) \quad (4)$$

## 3. THEORY

Due to the error sources described in the previous part, measurement of a magnetometer at time  $k$  is given by,

$$\bar{m}_k^b = D_S C_{NO} \left( C_S \hat{C}_e^b \bar{m}_0^e + \bar{b}_E \right) + \bar{v}_k \quad (5)$$

The equation (5) can be written in a compact form,

$$\bar{m}_k^b = C \left( \bar{m}_{0,k}^b + \bar{b}^* \right) + \bar{v}_k \quad (6)$$

where  $\bar{m}_{0,k}^b$  is the magnetic field of the Earth resolved in the body frame,  $C$  is a transformation in  $\mathbb{R}^{3 \times 3}$  and  $\bar{b}^*$  is the magnetic bias.

Since  $C$  is a matrix in  $\mathbb{R}^{3 \times 3}$ , it can be decomposed using *SVD* decomposition as,

$$C = T \tilde{D} V \quad (7)$$

where  $T$  and  $V$  are orthogonal,  $\tilde{D}$  is a diagonal matrix.  $V$  represents the misalignment between body frame and the frame formed by the axis of the magnetometer, [2].  $\tilde{D}$  represents the scaling on the axis of the magnetometer, and its entries are all positive.  $T$  represents the non-orthogonality in the axis of the magnetometer.

In this work, we assume that the axis of the magnetometer are orthogonal, which is usually the case for the magnetometers that have a fabrication calibration. The orthogonality of the axis is independent of the operating environment and does not change with the operating environment. Therefore,  $T$  is identity matrix, and (7) simplifies to,

$$C = \tilde{D} V \quad (8)$$

If there was no wide band noise, the magnetometer measurements would satisfy,

$$\begin{aligned} (\bar{m}_k^b - \tilde{D}\bar{b})^T \tilde{D}^{-2} (\bar{m}_k^b - \tilde{D}\bar{b}) = \\ (\hat{C}_e^b \bar{m}_0^e)^T V^T V (\hat{C}_e^b \bar{m}_0^e) = (\bar{m}_0^e)^T I (\bar{m}_0^e) \end{aligned} \quad (9)$$

where  $\bar{b} = V\bar{b}^*$ . This substitution, in fact, shows that the magnetic bias can be measured only in the misaligned magnetometer axis, which is the correct representation of the bias.

This result exhibits the nature of the components of the magnetometer measurements in the sense that, the magnetic bias and wide band noise should be in the magnetometer frame, not in the body frame. Since the magnetic field of the Earth is assumed to be constant, the left hand side of the equation is constant, which yields,

$$(\bar{m}_k^b - \tilde{D}\bar{b})^T \left[ \frac{1}{|\bar{m}_0^e|} \tilde{D} \right]^{-2} (\bar{m}_k^b - \tilde{D}\bar{b}) = 1 \quad (10)$$

Since  $\frac{1}{|\bar{m}_0^e|} \tilde{D}$  is a diagonal matrix, (10) is the equation of an ellipsoid centered at  $\tilde{D}\bar{b}$ . Consequently, the measurements of magnetometers lie on an ellipsoid manifold.

Quartic surfaces, such as ellipsoids, can be analyzed easier in projective space  $\mathbb{P}^4$ . Transformation of vectors in  $\mathbb{R}^3$  to the projective space,  $\mathbb{P}^4$ , requires representation of the vectors in homogeneous coordinates, [3]. A vector  $\bar{x} = [x_1, x_2, x_3]^T$  has a homogeneous representation  $\bar{X} = [X_1, X_2, X_3, X_4]^T$ , where,

$$x_i = \frac{X_i}{X_4} \quad \text{for } i = 1, 2, 3 \quad (11)$$

$X_4 = 1$  is substituted when obtaining a homogeneous representation of a vector. This representation is a mapping between  $\mathbb{R}^3$  and projective space,  $\mathbb{P}^4$ . In homogeneous coordinates, (10) can be written as,

$$\bar{X}^T \hat{D}^{-2} \bar{X} = 0 \quad (12)$$

where,

$$\hat{D} = \begin{bmatrix} \frac{1}{|\bar{m}_0^e|} \tilde{D} & \bar{0} \\ \bar{0}^T & -1 \end{bmatrix} \quad (13)$$

and  $\bar{X}$  is the representation of  $(\bar{m}_k^b - \tilde{D}\bar{b})$  in the homogeneous coordinates.

Under a point (projective) transformation, the quadric surface transforms to another quadric surface by,

$$\hat{S} = H^{-T} \hat{D}^{-2} H^{-1} \quad (14)$$

where  $H$  is a projective transformation.

The equation (14) implies that for every quartic surface it is possible to find an invertible transformation, that maps the quartic surface into a sphere. For the surface given in (12),

the transformation that maps the ellipsoid to the unit sphere is given by,

$$H = \begin{bmatrix} \frac{1}{|\bar{m}_0^e|} \tilde{D}^{-1} & \bar{0} \\ \bar{0}^T & 1 \end{bmatrix} \quad (15)$$

The equation (15) is an affine transformation without translation part. The translation of the measurement by an amount of the magnetic bias can also be represented by an affine transformation as,

$$H_t = \begin{bmatrix} I & -\tilde{D}\bar{b} \\ \bar{0}^T & 1 \end{bmatrix} \quad (16,a)$$

$$\bar{X} = H_t \bar{M}_k^b \quad (16,b)$$

Let  $\bar{X}'$  denote the corresponding point of  $\bar{X}$  on the unit sphere,

$$\bar{X}'^T \hat{S} \bar{X}' = \bar{X}'^T \begin{bmatrix} I & \bar{0} \\ \bar{0}^T & -1 \end{bmatrix} \bar{X}' = 0 \quad (17)$$

$\bar{X}'$  is given by,

$$\bar{X}' = H H_t \bar{M}_k^b \quad (18)$$

where  $\bar{M}_k^b$  is the Earth's magnetic field in body frame represented in homogeneous coordinates. Consequently, any point on the ellipsoid can be transformed onto a point on the unit sphere by the affine transformation,  $H H_t$ .

$H_t$  can be decomposed as,

$$H_t = \begin{bmatrix} \tilde{D} & \bar{0} \\ \bar{0}^T & 1 \end{bmatrix} \begin{bmatrix} \tilde{D}^{-1} & -\bar{b} \\ \bar{0}^T & 1 \end{bmatrix} \quad (19)$$

Then,  $H H_t$  is,

$$H H_t = \begin{bmatrix} \frac{1}{|\bar{m}_0^e|} I & \bar{0} \\ \bar{0}^T & 1 \end{bmatrix} \begin{bmatrix} \tilde{D}^{-1} & -\bar{b} \\ \bar{0}^T & 1 \end{bmatrix} \quad (20)$$

In reality, the diagonal scale matrix,  $\tilde{D}^{-1}$ , is in the form,

$$\tilde{D}^{-1} = I + D \quad (21)$$

where  $D = \text{diag}\{d_1, d_2, d_3\}$  is also a diagonal matrix.

Substitution of (21) into (20) yields,

$$H H_t = \begin{bmatrix} \frac{1}{|\bar{m}_0^e|} I & -\frac{1}{|\bar{m}_0^e|} \bar{b} \\ \bar{0}^T & 1 \end{bmatrix} + \begin{bmatrix} \frac{1}{|\bar{m}_0^e|} D & 0 \\ \bar{0}^T & 1 \end{bmatrix} \quad (22)$$

Since the entries of  $D$  are small, the second part of (22) can be neglected, which corresponds to neglecting the scale factors. In this case, the estimated bias is scaled version of the actual bias. For this case, the calibration problem can be

solved in two steps; first the bias can be found, and then the scale factors can be determined. It should be noted that the sequence is not reversible, i.e., one may not determine the scales before determining the bias as it is apparent from the fact that the error introduced by a small disturbance in bias is larger than the error introduced by small disturbance of scales. In addition,  $HH^T \neq H^T H$  implies that for two step calibration of magnetometers, the bias should be estimated over scaled estimates, which, in turn implies that the sequence of estimation is important.

Substitution of (20) and (18) into (17) yields,

$$|\bar{m}_k^b - \bar{b}|^2 = |\bar{m}_0^e|^2 \quad (23)$$

which is, essentially, the measurement model used in [2]. However, (23) only applies for the algorithms that estimate the bias recursively, which implies that the scales will be effective on the bias estimates of the TWOSTEP algorithm. However, since the scales are usually close to unity, one may approximate  $H$  as an identity matrix and try to estimate  $\bar{b}' = \tilde{D}\bar{b}$  using TWOSTEP. Although TWOSTEP is a batch estimator, in [4], a recursive version of it is introduced. In [5], the recursive version of it is tested and compared with other algorithms for real-time magnetometer calibration problem.

Substitution of (21) into (10) and adding the effects of the wide band noise, yields,

$$\begin{aligned} z_k &= |\bar{m}_k^b|^2 - |\bar{m}_0^e|^2 \\ &= -(\bar{m}_k^b)^T D^2 \bar{m}_k^b + 2(\bar{b} - \bar{m}_k^b)^T D \bar{m}_k^b + \\ &2\bar{b}^T \bar{m}_k^b - |\bar{b}|^2 + w_k \end{aligned} \quad (24)$$

where  $w_k$  is given by,

$$w_k = 2[(I+D)\bar{m}_k^b - \bar{b}]^T \bar{v}_k - |\bar{v}_k|^2 \quad (25)$$

This measurement model is used in [5] and [2]. As in these works, we assume that  $w_k$  is approximately Gaussian with mean,  $\mu_k$ , and variance,  $\sigma_k^2$ ,

$$\mu_k = -\text{Tr}(\Sigma_k) \quad (26, a)$$

$$\begin{aligned} \sigma_k^2 &= 4[(I+D)\bar{m}_k^b - \bar{b}]^T \Sigma_k [(I+D)\bar{m}_k^b - \bar{b}] \\ &+ 2\text{Tr}^2(\Sigma_k) \end{aligned} \quad (26, b)$$

Since  $D$  is diagonal, it is possible to define another diagonal matrix,  $M$ , such that,

$$D\bar{m}_k^b = M_k \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = M_k \bar{d} \quad (27)$$

where  $M = \text{diag}\{m_1, m_2, m_3\}$ . Then,  $z_k$  is given by,

$$\begin{aligned} z_k &= h(\bar{d}) + \bar{w}_k \\ &= -(\bar{d})^T M_k^2 \bar{d} + 2(\bar{b} - \bar{m}_k^b)^T M_k \bar{d} + \\ &2\bar{b}^T \bar{m}_k^b - |\bar{b}|^2 + w_k \end{aligned} \quad (28)$$

The Extended Kalman Filter (EKF) is used to estimate the scales. Since the scales are constant with respect to time, the state model is,

$$\bar{d}_{k+1} = \bar{d}_k \quad (29)$$

The measurement model is (20). The update equations of EKF are,

$$\hat{d}_{k+1} = \hat{d}_k + K_k [z_k - h(\hat{d}_k)] \quad (30, a)$$

$$P_{k+1} = [I - K_k H_k (\hat{d}_k)] P_k \quad (30, b)$$

$$K_k = P_k H_k^T (\hat{d}_k) [H_k (\hat{d}_k) P_k H_k^T (\hat{d}_k) + \sigma_k^2 (\hat{d}_k)]^{-1} \quad (30, c)$$

$$H_k (\hat{d}_k) = \frac{\partial h}{\partial \hat{d}_k} = M_k^2 \hat{d}_k + 2(\bar{b} - \bar{m}_k^b)^T M_k \quad (30, d)$$

These derivations are greatly simplified when the predetermined  $\bar{b}'$  is substituted as,

$$\bar{b} = (I+D)\bar{b}' \quad (31)$$

In fact, this substitution is equivalent to scale estimation problem on the ellipsoid centered at the origin. In this case the magnetometer measurements take the form,

$$\bar{m}_k^b = \bar{m}_k^b - \bar{b}' = \tilde{D}\bar{m}_{0,k}^b + \bar{\eta}_k \quad (32)$$

where  $\bar{\eta}_k$  is noise process that contains both the measurement noise and the error introduced by the bias estimation part.

#### 4. SIMULATION RESULTS

The proposed method is tested for artificial magnetometer measurements. Since these measurements should be on an ellipsoid, firstly the ellipsoid with true parameters is generated. In order to get sufficiently rich data, 5000 points on the ellipsoid is taken and a white Gaussian noise with predetermined covariance is added. The magnetometer is assumed to have a white Gaussian measurement error with isotropic error distribution. The standard deviation of the noise on each axis is assumed to be equal. An example of this test is illustrated in Figure 1.

In order to examine the TWOSTEP algorithm under different scales, 50 runs has been executed for measurements on the unit sphere and for the measurements on an ellipsoid, which provides Monte-Carlo like simulation.

Table 1 demonstrates the estimated parameters for the measurements on the unit sphere. Table 2 shows the



estimated bias values for an ellipsoid with scales,  $\bar{d} = [1, 1, 1, 1.2]^T$ . The bias value shown in the table is  $\bar{b}' = D\bar{b}$ , where  $\bar{b}$  is the bias values given in Table 1. The reason for this is the fact that TWOSTEP algorithm is a batch estimator, i.e., (20) does not apply.

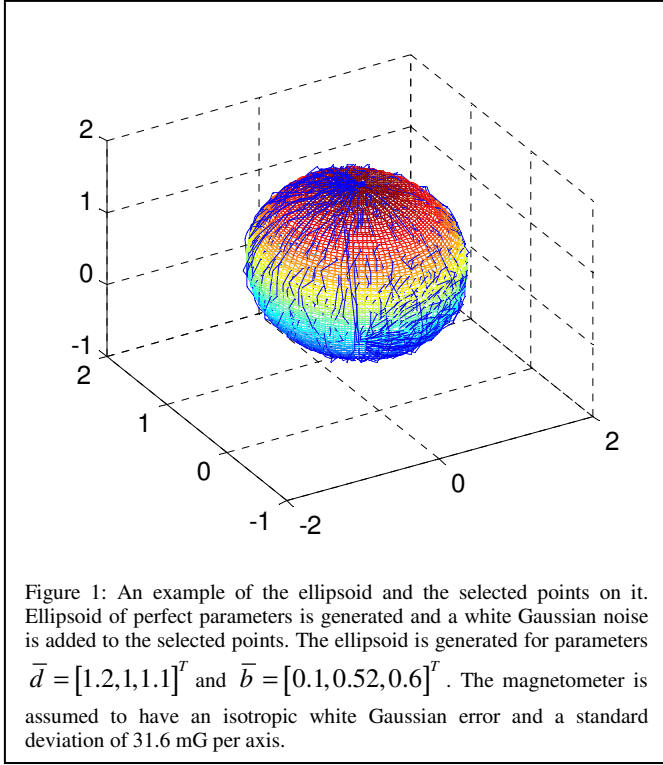


Figure 1: An example of the ellipsoid and the selected points on it. Ellipsoid of perfect parameters is generated and a white Gaussian noise is added to the selected points. The ellipsoid is generated for parameters  $\bar{d} = [1.2, 1, 1, 1]^T$  and  $\bar{b} = [0.1, 0.52, 0.6]^T$ . The magnetometer is assumed to have an isotropic white Gaussian error and a standard deviation of 31.6 mG per axis.

though it can be used as a bias estimator in separated methods.

## REFERENCES

- [1] Caruso MJ. "Application of Magnetoresistive Sensors in Navigation Systems." SAE SP-1220, pp. 15-21, 1997.
- [2] Davenport PB, Ruml WM & Welter GL. "In-flight Determination of Spacecraft Magnetic Bias Independent of Attitude ." InProc. , 1988, pp. 326-343.
- [3] Crassidis JL & Lai K. "Real-Time Attitude-Independent Three-Axis Magnetometer Calibration ." Journal of Guidance, Control, and Dynamics, Vol.28, No.1: pp. 115-120, 2005.
- [4] Alonso R & Shuster MD. "Complete Linear Attitude-Independent Magnetometer Calibration." The Journal of the Astronautical Sciences, Vol.50, No.4: pp. 477-490, 2002.
- [5] Alonso R & Shuster MD. "TWOSTEP, a Fast Robust Algorithm for Attitude-Independent Magnetometer Bias Determination." The Journal of the Astronautical Sciences, Vol.50, No.4: pp. 433-451, 2002.
- [6] Alonso R & Shuster MD. "A new Algorithm For Attitude-Independent Magnetometer Calibration." InProc. Flight Mechanics/Estimation Theory Symposium, NASA Goddard Space Flight, May 17-19, 1994, pp. 513- 527.
- [7] Shuster MD & Alonso R. "Attitude-Independent Magnetometer-Bias Dtermination: A Survey." The Journal of Astronautical Sciences, Vol.50, No.4: pp. 453-475, 2002.
- [8] Gebre-Egziabher D, Elkaim GH, Powell JD & Parkinson BW. "Calibration of Strapdown Magnetometers in Magnetic Field Domain." ASCE Journal of Aerospace Engineering, Vol.19, No.2: pp. 87-102, 2006.
- [9] Vasconcelos JF, Elkaim G, Silvestre C, Oliveira P & Cardeira B. "A Geometric Approach to Strapdown Magnetometer Calibration in Sensor Frame". 2nd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles, Killaloe, Ireland, April 2008.
- [10] Elkaim GH & Foster C. "Extension of a Non-Linear, Two-Step Calibration Methodology to Include Non-Orthogonal Sensor Axes." Journal of Aerospace Electronic Systems, Vol.44, No.3: pp. 1070-1078, 2008.
- [11] Gebre-Egziabher D, Elkaim GH, Powell JD & Parkinson BW. "A Non-linear, Two Step Estimation Algorithm For Calibrating Solid-State Strapdown Magnetometers." InProc. 8th Int. Conf. on Integrated Navigation Systems, 2001, pp. 200-209.
- [12] Roetenberg D, Luinge HJ, Baten CTM & Veltink PH. "Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation." Neural Systems and Rehabilitation Engineering, IEEE Transactions on, Vol.13, No.3: pp. 395-405, 2005.
- [13] Hartley R & Zisserman A. "Multiple View Geometry in Computer Vision." Cambridge University Press, 2000.

Table 1: Estimated bias values for measurements on the unit sphere for 50 run Monte-Carlo simulation.

Parameter	True Value (mG)	Estimated Value (mG)	Other Parameters
$b_1$	100	100.02±0.20	$\bar{d} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$
$b_2$	520	519.98±0.20	
$b_3$	600	599.99±0.14	

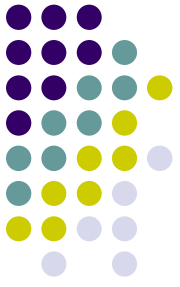
Table 2: Estimated bias values for measurements on an ellipsoid of given scales for 50 run Monte-Carlo simulations

Parameter	True Value (mG)	Estimated Value (mG)	Other Parameters
$b_1$	100	101.28±0.22	$\bar{d} = \begin{bmatrix} 1 \\ 1.1 \\ 1.2 \end{bmatrix}$
$b_2$	520	519.91±0.20	
$b_3$	620	599.95±0.14	

Although the scales are large compared with the real scales in the magnetometers, the TWOSTEP algorithm performs well,



# BİR İNSANSIZ SUALTI ARACI: ULİSAR



**M. Kemal Leblebiciođlu**  
ODTÜ, Elektrik ve Elektronik Müh. Bölümü  
[kleb@metu.edu.tr](mailto:kleb@metu.edu.tr)

# Proje Ekibi



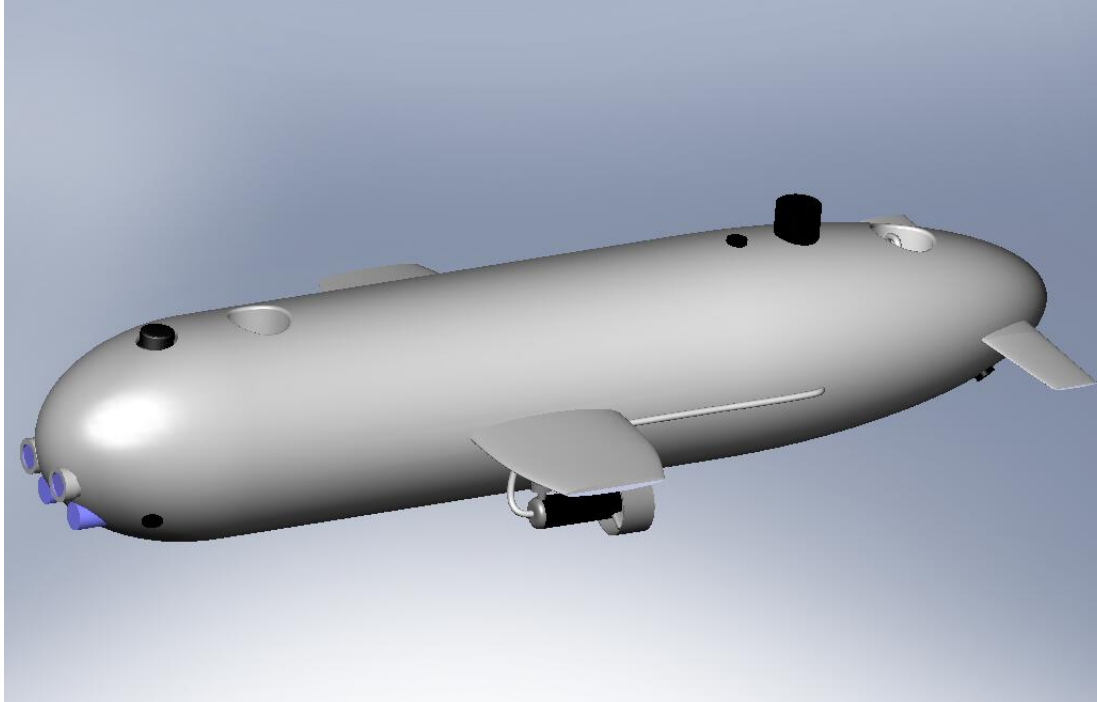
- Prof. Dr. M. Kemal Leblebiciođlu (ODTÜ, Elektrik ve Elektronik Müh. Bölümü) (Proje koordinatörü)
- Y. Doç. Dr. Afşar Saranlı (ODTÜ, Elektrik ve Elektronik Müh. Bölümü)
- Y. Doç. Dr. İlkey Ulusoy (ODTÜ, Elektrik ve Elektronik Müh. Bölümü)
- Hüseyin Yiğitler (ODTÜ, Elektrik ve Elektronik Müh. Bölümü)
- Emre Ege (ODTÜ, Elektrik ve Elektronik Müh. Bölümü)
- Üst. Kadir Isıyel (ODTÜ, Elektrik ve Elektronik Müh. Bölümü)
- Doç. Dr. Halit Oğuztüzün (ODTÜ, Bilgisayar Müh. Bölümü)
- Alper Kılıç (ODTÜ, Bilgisayar Müh. Bölümü)
- Haldun Mıdođlu (Yenel Ltd.)
- Fikret Şenel, (Barış Elektrik Endüstrisi A.Ş.)
- Prof. Dr. S. Kemal İder, (ODTÜ, Makina Müh. Bölümü)
- Prof. Dr. M. Haluk Aksel (ODTÜ, Makina Müh. Bölümü)
- Prof. Dr. Levend Parnas (ODTÜ, Makina Müh. Bölümü)
- Prof. Dr. Kemal Özgören (ODTÜ, Makina Müh. Bölümü)
- Necmettin Cevheri, (ODTÜ, Makina Müh. Bölümü)

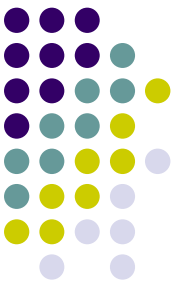
# İnsansız Sualtı Araçları



- Denizaltıları andıran sualtı robotları
- Otonom araçlar
- Bağlantı kablosu ve mürettebat gereksinimleri yok

# İnsansız Sualtı Araçları

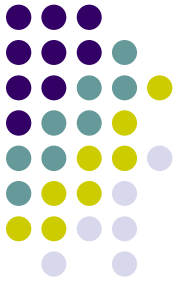




# Kullanım Alanları

- Mayın imha
- Keşif
- Sualtı kablolarının döşenmesi
- Sualtı hedeflerine ekipman taşınması
- Deniz ve okyanus suyu incelemeleri
- Petrol ve doğalgaz boruları gözlem ve bakımı
- Sualtı arkeoloji çalışmaları
- Deprem araştırmaları

# ULİSAR

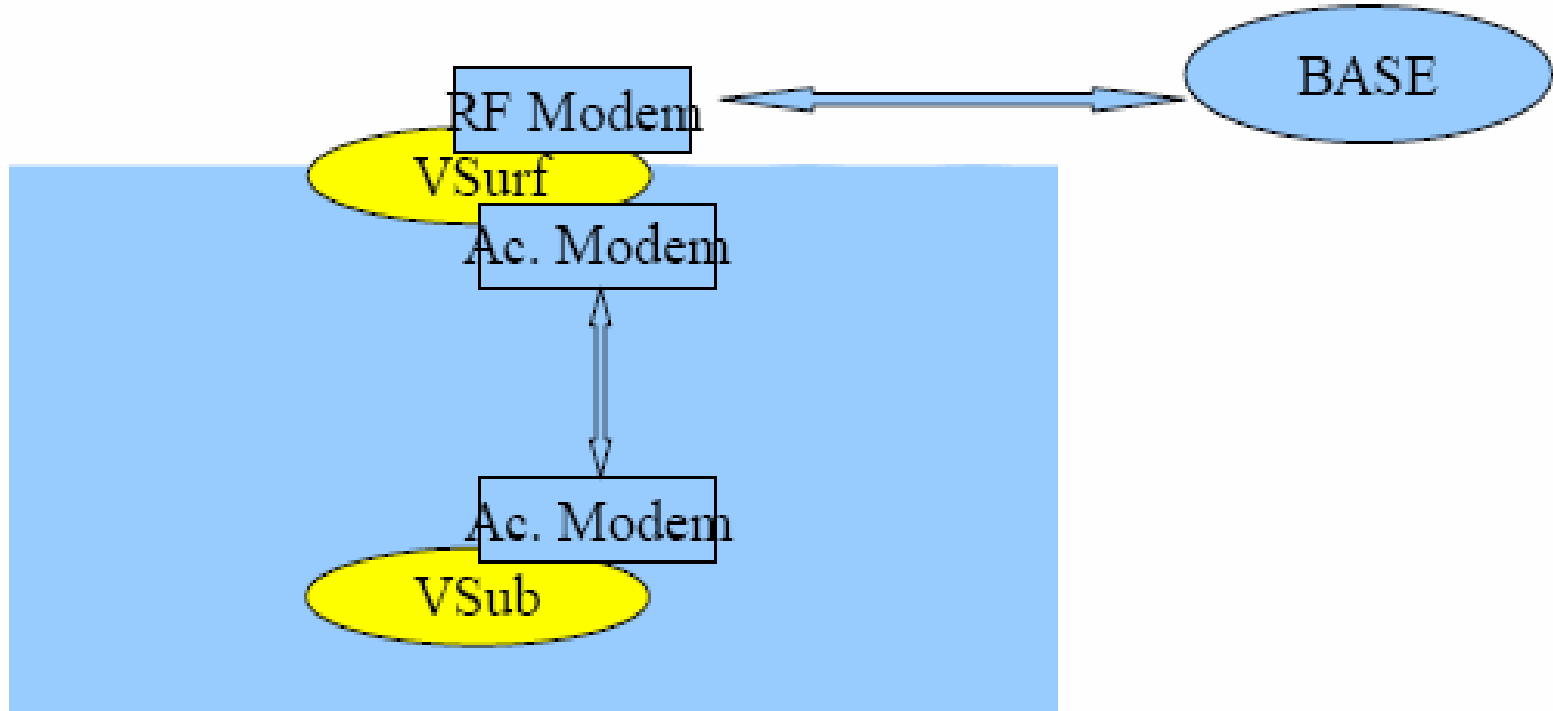
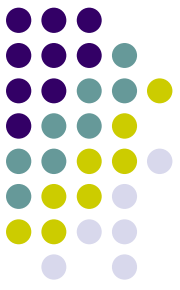


## Temel Fikir:

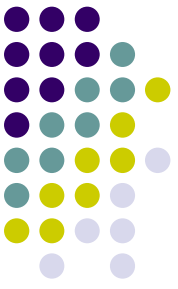
- Sualtı aracı ile beraber uyumlu olarak çalışan bir suüstü aracının varlığı,
- Aradaki haberleşme: Akustik modemkullanarak düşük rezolüsyonlu görüntü ve navigasyon bilgilerinin aktarımı,
- Suüstü haberleşmesi: RF kullanarak.



# ULİSAR



## Önerilen insansız sualtı aracı sistemi



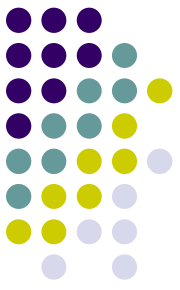
# Tasarım Öğeleri (Gövde)

- Su sızdırmaz bölme tasarımı
- Dış gövde tasarımı
- İtki ve manevra sistemleri
- Denge ve yüzerlik
- Kablolama
- İmalat yöntemi



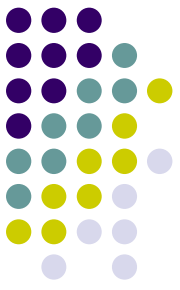
# Dış Gövde Tasarımı

- Dış gövde: aracı en dışta saran yapı
- Ekipmanın korunması, hidrodinamik yapı ve aracın şekli
- Denizaltıların aksine çeşitli şekillerde olabilirler



# Dış Gövde Tasarımı

- Dönme kararlılığı açısından simetrik yapıda
- Ekipmanı kapsayacak ölçüde uzun ve büyük çapta
- Sürüklenme ve sürtünme kuvvetlerini azaltacak ölçüde kısa ve küçük çapta
- Basınca dayanacak ve ağırlığı çok arttırmayacak ölçüde bir et kalınlığında
- Elektronik ekipmanın fazla ısınmasını engelleyecek ölçüde ısı transferine izin veren yapıda
- Modüler yapıda

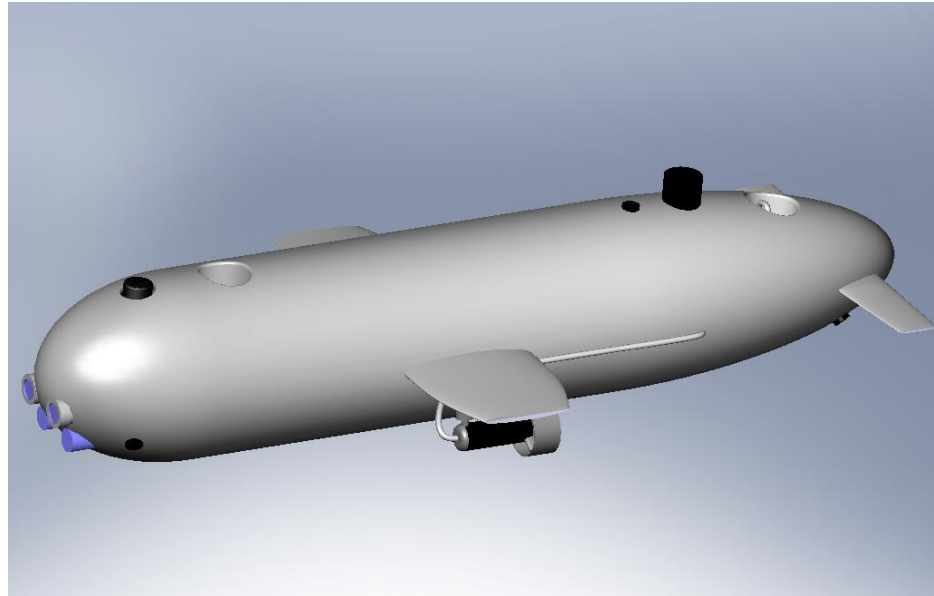


# Dış Gövde Tasarımı

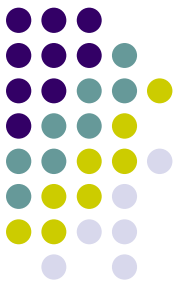
- Düşük karşı kuvvet
- Üretim kolaylığı
- Geniş veritabanı



Silindirik yapı



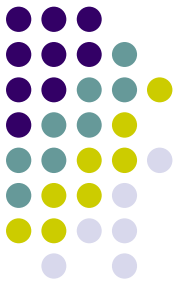
BİR İNSANSIZ SUALİT ARACI: ULISAR



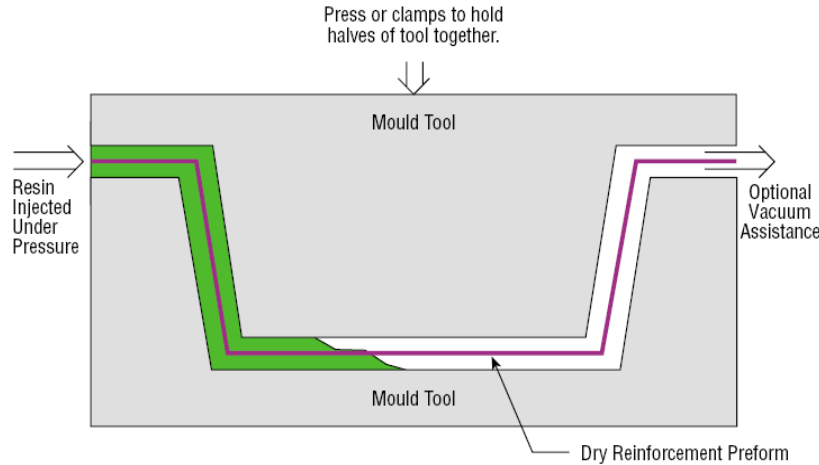
# Malzeme

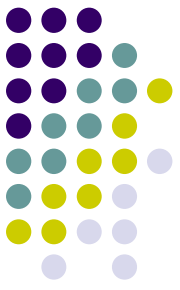
- Gövde kompozit malzemedenden yapılacaktır:
  - Düşük yoğunluklarda yüksek mukavemet
  - Tasarım esnekliği
  - Kalıplama kolaylığı
  - Çeşitli yüzey uygulamalarına izin vermesi
  - Gövde içine metal parçalar yerleştirilebilmesi
  - Yapıştırma işlemleri uygulanabilmesi
  - Barış Elektrik Endüstrisi A.Ş.' in verdiği destek
- Malzeme: Cam dokuma kumaş – epoksi reçine

# Üretim Tekniği



- Reçine Transfer Kalıplama (RTM)
  - Kalıp parçaları arasına elyaf yatırma
  - Basınç ve/veya vakum ile kontrollü reçine besleme
  - Isıl işlem ile polimerizasyon





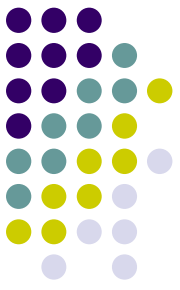
# Üretim Tekniđi - RTM

- Avantajlar:
  - - Yüksek elyaf-reçine oranı
  - - Düşük hava kabarcığı oranı
  - - Düzgün yüzey kalitesi
  - - Tasarım esnekliđi
- Dezavantajlar:
  - -Artan kalıplama masrafı
  - -Büyük parçalara uygulama zorluđu

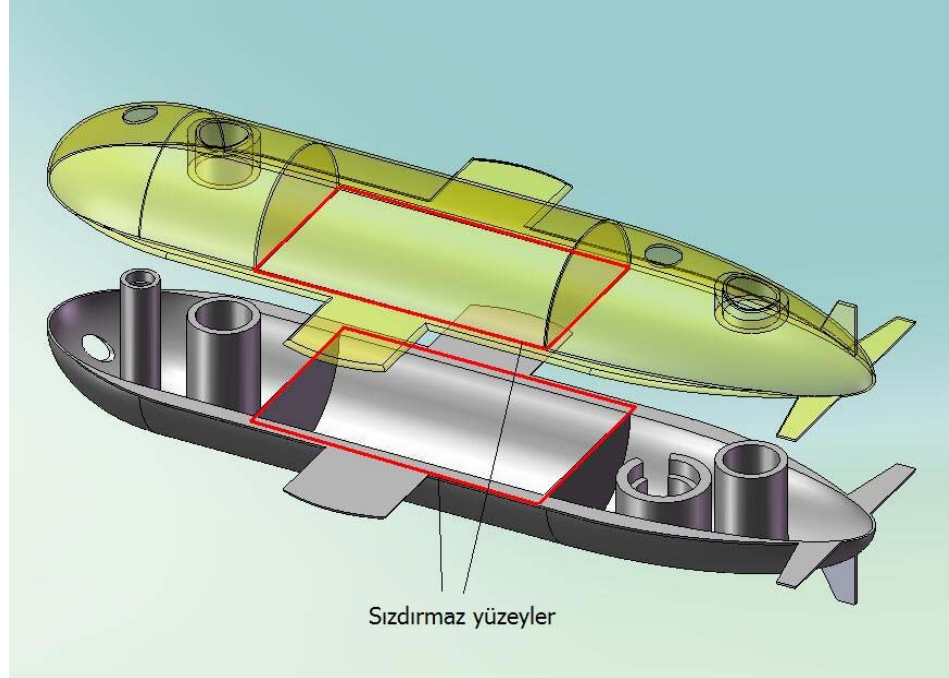


# Üretim Tekniği - RTM



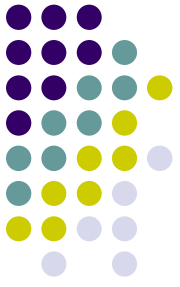


# Dış Gvde Tasarımı

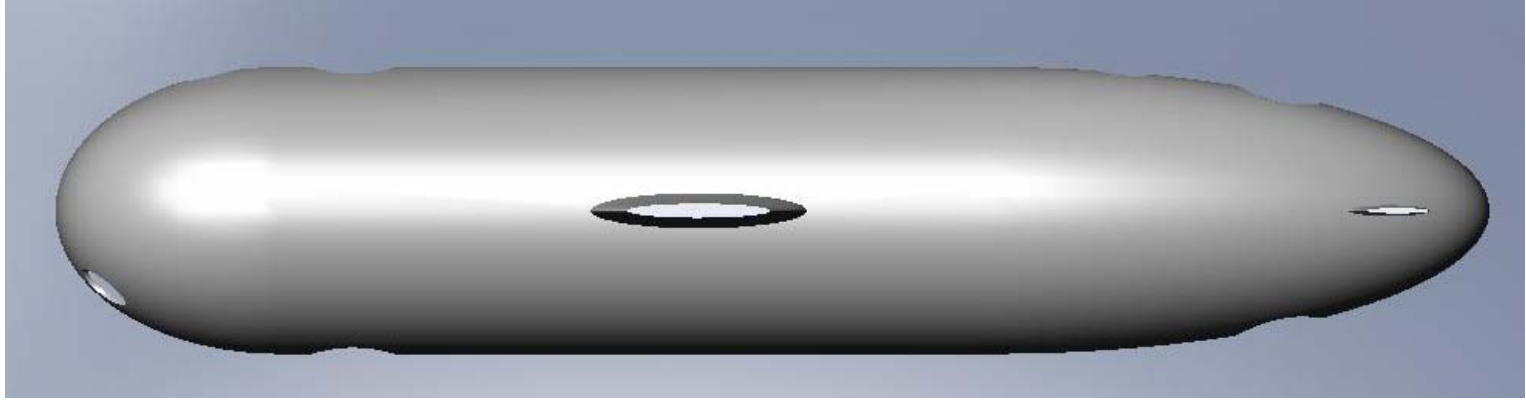


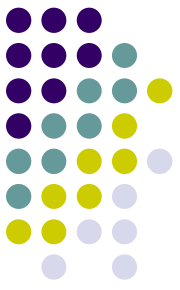
- Maliyeti dşrmek
- Yksek mukavim yapı elde etmek

# Dış G6vde Tasarımı



Hidrodinamik yapı:

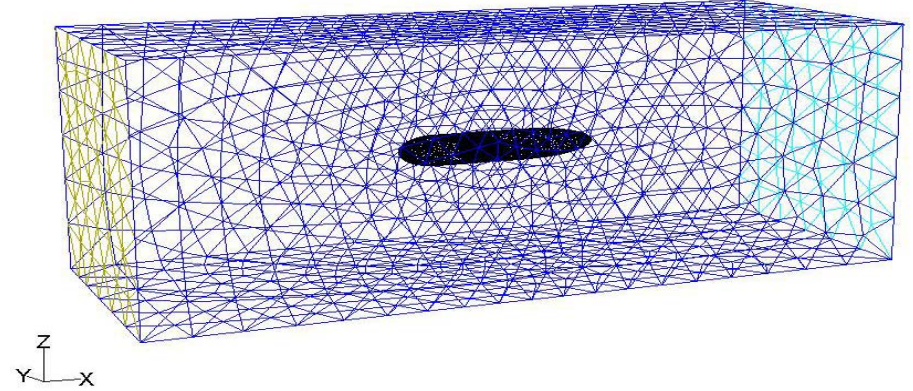




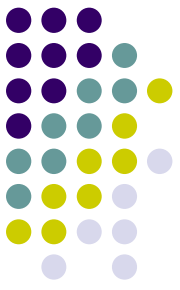
# Dış Gövdenin Tasarımı

## Hidrodinamik analizler:

- Gambit-Fluent
- Basitleştirilmiş model
- 3B, zamandan bağımsız
- k- $\epsilon$  türbülans modeli
- Sıcaklık ve özellikler sabit

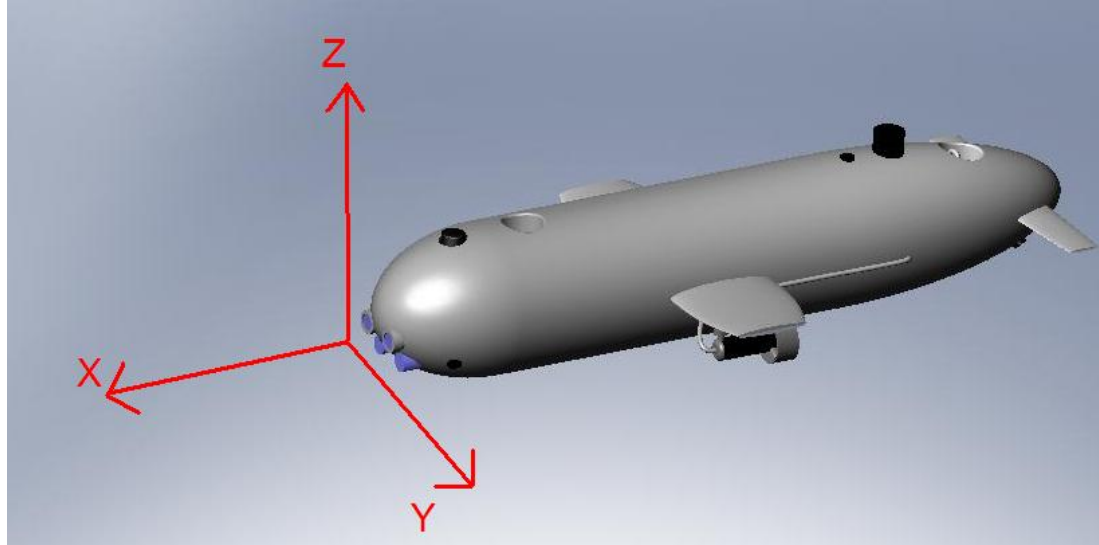


- Analizler sonucu araca etkiyen sürüklenme kuvvetleri elde edilmiş ve bu değerlerin seçilen motorların sağladığı itki değerlerini geçmediği görülmüştür.

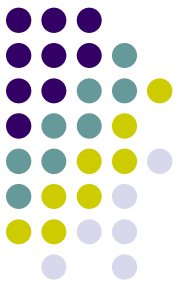


# İtki ve Manevra

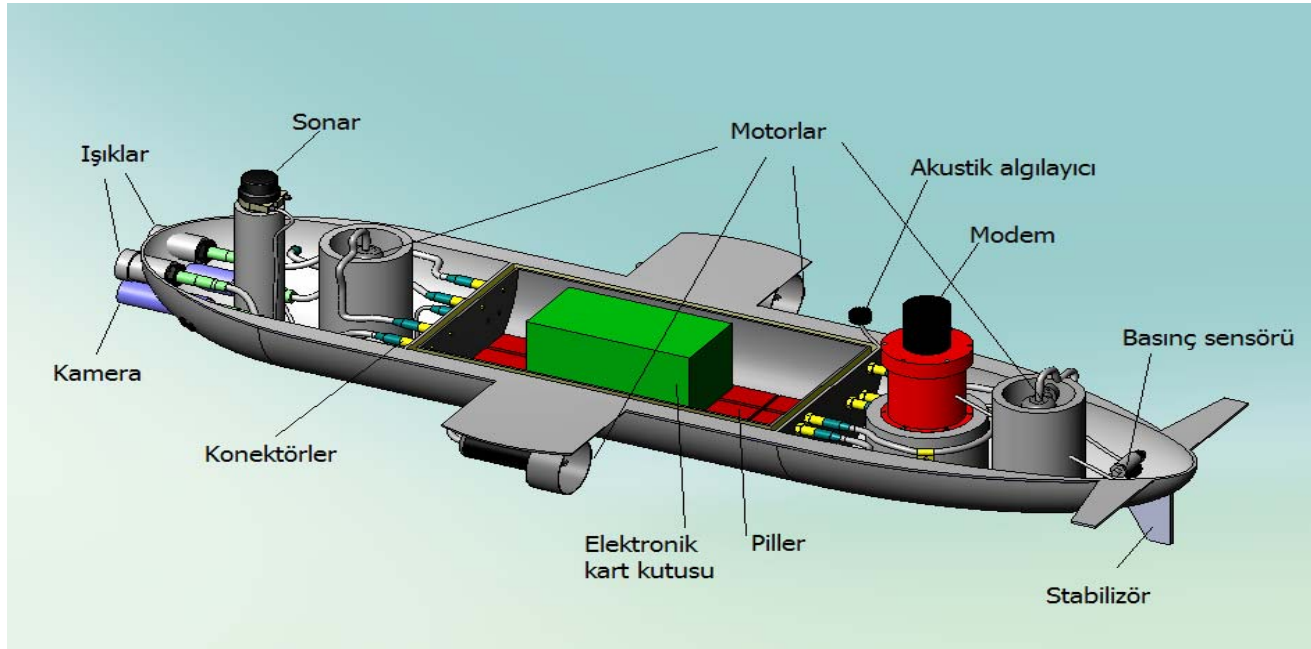
- İleri geri hareket (x eksenini boyunca)
- Yukarı aşağı hareket (z eksenini boyunca)
- Sağa sola dönüş (z eksenini etrafında)
- Yukarı aşağı dönüş (y eksenini etrafında)

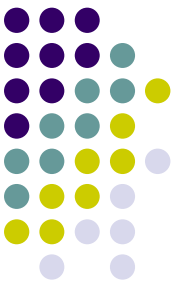


# İç Yerleşim ve Isınma Problemi



- İç yerleşimde önemli nokta: ağır parçaların alt bölümlere yerleştirilmesi

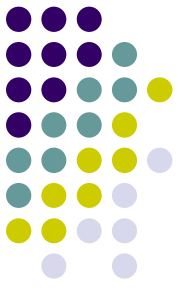




# Isınma Problemi

- Isınma probleminin öncelikli olarak sızdırmaz bölgede gerçekleşebileceği tahmin edilmektedir.
  - Denetim ekipmanlarının ve pillerin bulunması
  - Et kalınlığının fazla olması
  - Kompozit malzemenin yalıtkan özelliği
- Deneyler ve analizler gerçekleştirilmiştir.

# Isınma Problemi

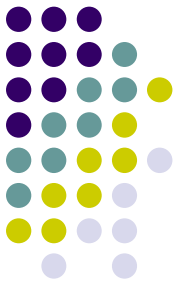


Su geçirmez bölümde pillerin haricindeki donanım elemanlarının güç tüketimleri

Donanım Eleman Adı:	Adet	Güç Tüketimi(W)	Toplam Güç(W)
Parvus CPU-1462	2	13.00	26.00
Parvus ACS-5150 DC-DC çevirici	2	4.00	8.00
Parvus CTR-1461 PCMCİ çevirici	2	0.35	0.70
Parvus CTR-1475 MPEG 4 sıkıştırıcı	1	10.00	10.00
Microstrain 3DM-GX1 IMU	1	0.78	0.78
Z-World RCM 4000 Mikrodenetleyici	1	0.10	0.10
Diğer	N/A	5.00	5.00
Kokam 2000mAh LiPo hücreler	-----	0.10	-----
Piller hariç toplam			50.58



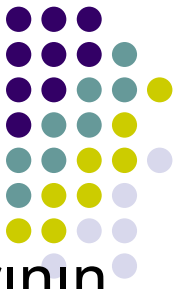
# Isınma Problemi



Pillerde harcanan güç değerleri

Adı	Eşdeğer direnci(m $\Omega$ )	Çekilen akım (mA)	Güç tüketimi(W)
Pil Paketi 1	4.20	12000.00	0.60
Pil Paketi 2	16.80	12000.00	2.42
Pil Paketi 3	16.80	12000.00	2.42
Toplam			5.44

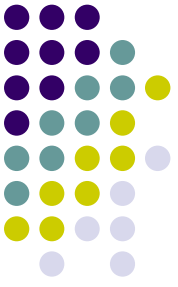
# Isınma Problemi



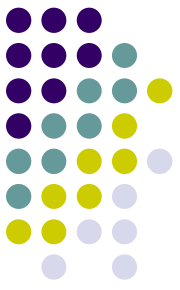
Su geçirmez bölümde bulunacak olan donanım elemanlarının çalışma sıcaklıkları

Donanım Eleman Adı:	Sıcaklık aralığı (°C)	En yüksek sıcaklık (°C)
Parvus CPU-1462	-40 - 85	80
Parvus ACS-5150 DC-DC çevirici	-40 - 85	80
Parvus CTR-1461 PCMCİ çevirici	-40 - 85	80
Parvus CTR-1475 MPEG 4 sıkıştırıcı	-40 - 85	80
Microstrain 3DM-GX1 IMU	-40 - 70	65
Z-World RCM 4000 Mikrodenetleyici	0 - 70	65
Diğer	-----	-----
Kokam 2000mAh LiPo hücreler	-20 - 60	55

# Isınma Problemi

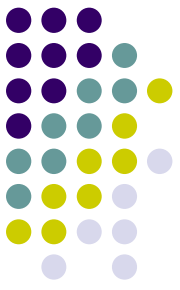


Bu durumda, eğer Diğer donanım elemanlarının çalışma sıcaklıkları  $60\text{ }^{\circ}\text{C}$  'ın üstünde olursa, sualtı aracının su geçirmez bölümünün sıcaklığı  $55\text{ }^{\circ}\text{C}$ 'ın altında olmalıdır.



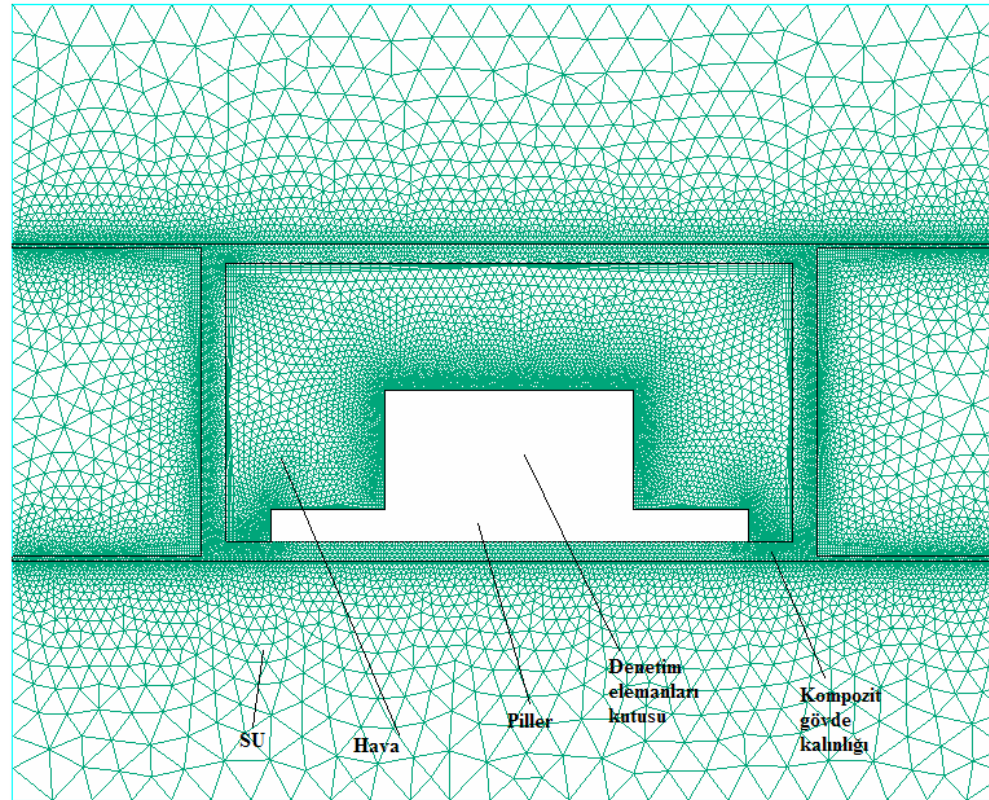
# Isınma Problemi - Analizler

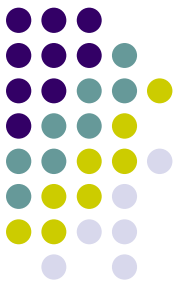
- Gambit ve FLUENT® yazılımları kullanılarak aracın çalışma süresi olarak düşünülen 2 saatlik sürelerde analiz edilmiştir.
- Zamana bağlı 3 boyutlu doğal konveksiyon ısı problemlerinin çözümü yüksek hızlarda bilgisayarlar gerektirdiğinden, problemler öncelikle 2 boyutlu çözülmüştür.
- 2 boyutlu analizlerden elde edilen sonuçlar tatmin edici olmaz ise 3 boyutlu analizlere başvurulacaktır.



# Isınma Problemi - Analizler

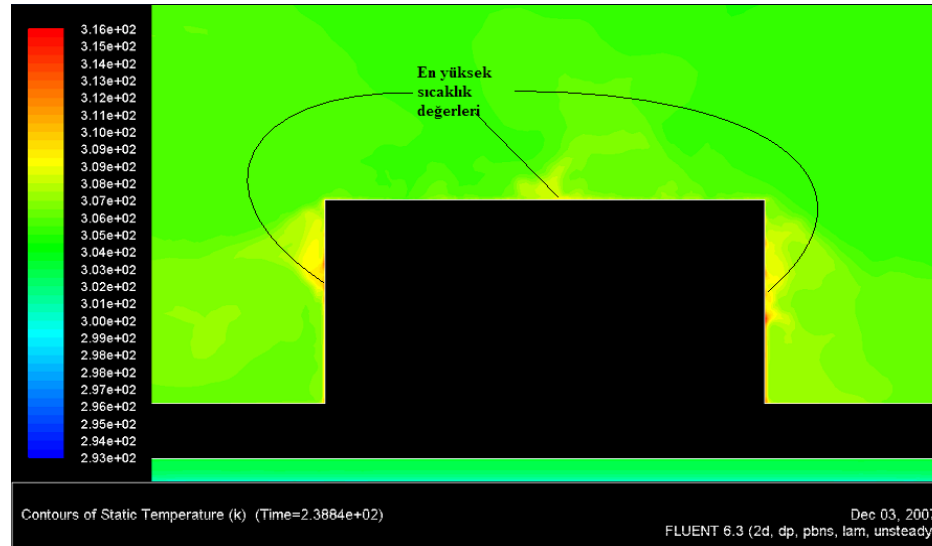
- Kritik olması beklenen sızdırmaz orta bölüm için analizler

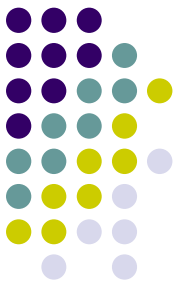




# Isınma Problemi - Analizler

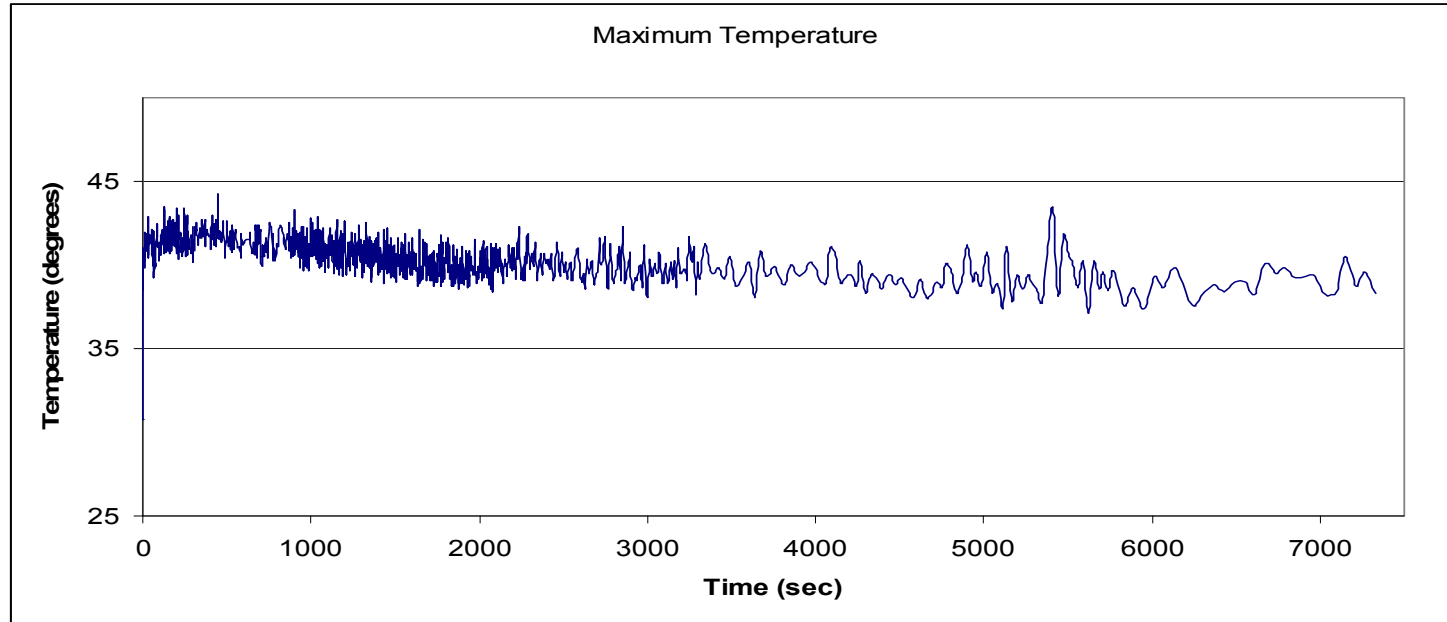
- Sınır ve başlangıç koşullar için su ve havanın alması beklenen sıcaklık değerleri kullanıldı.
- Analiz esnasında her hangi bir anda sıcaklığın, maksimum çalışma sıcaklığını geçmemesi kontrol edildi.





# Isınma Problemi - Analizler

- En kritik noktanın sıcaklık-zaman eğrisi:

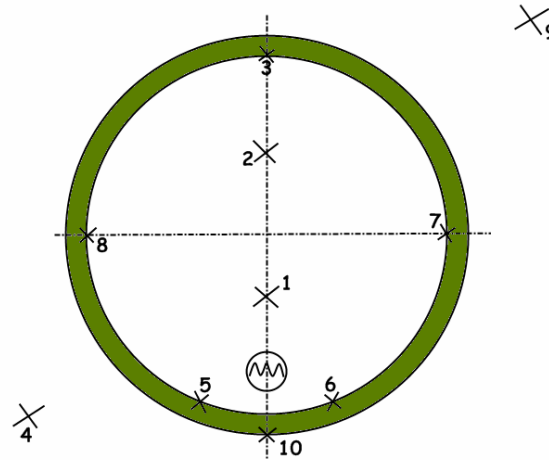


- En yüksek sıcaklık 36-37 derece arasında



# Isınma Problemi - Deneysel

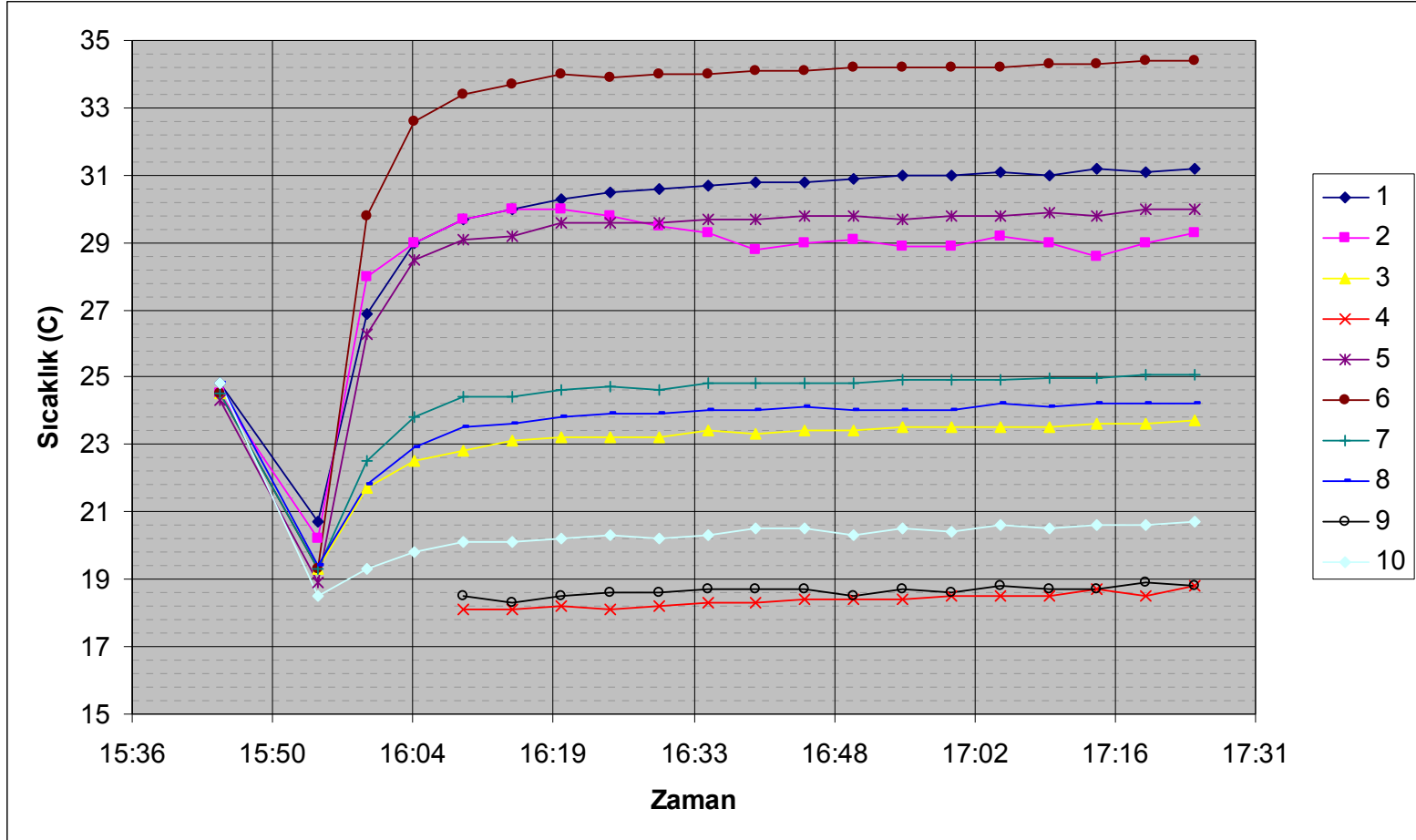
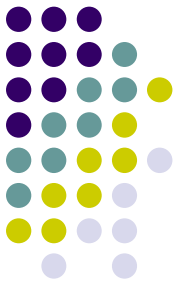
- Analizlerin doğrulanması amacıyla üretici firma tarafından yapılan ısı analiz testleri

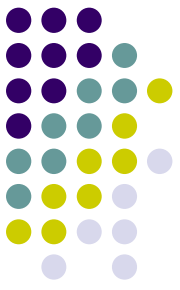


- Çeşitli noktaların sıcaklık-zaman eğrileri



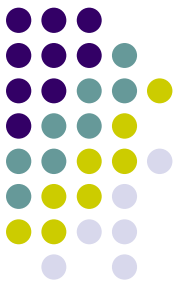
# Isınma Problemi - Deneysel





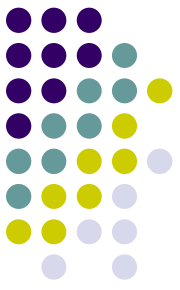
# Isınma Problemi - Sonuç

- Deney ve analiz sonuçlarının yakınlığı
- Bu sonuçlara göre araç içinde bir ısınma problemi beklenmemektedir.
- Buna rağmen araç içi ısı almaçları ile aracın çalışması sırasında ölçüm.
- Isınma problemleri yaşanması durumunda ısı borusu gibi ek çözümler gündeme gelebilir.



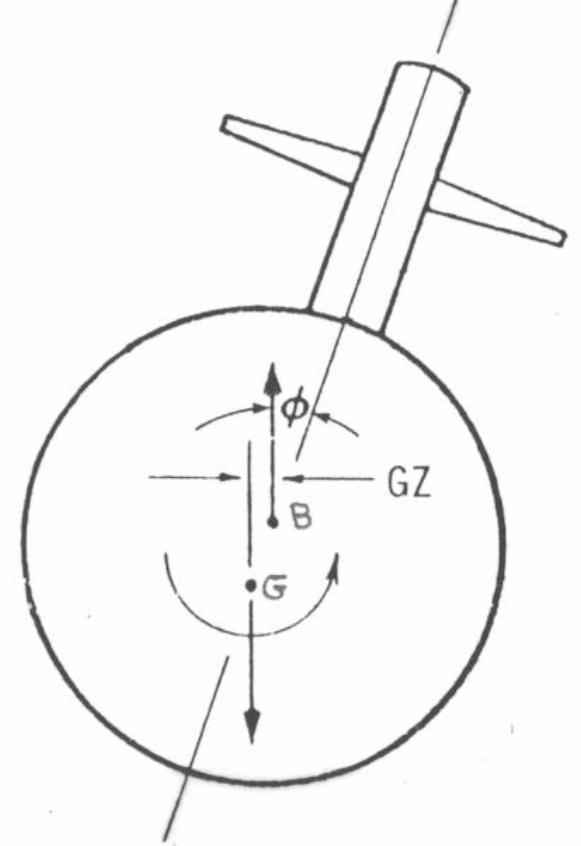
# Denge, Kararlılık ve Yüzerlik

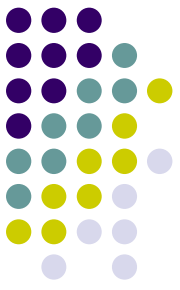
- Yüzerlik özelliğine sahip araçlar suda asılı kalabilmektedir. Bu sayede hareketsizken güç harcanması yoktur.
- Olası iletişim problemlerinde aracı kaybetmemek için araç 0.5-1 kg arasında kaldırma kuvvetine sahip olacak şekilde tasarlanabilir.
- Denge ve kararlılık ise aracın istenilen şekilde durabilmesi ve hareket edebilmesi açısından önemlidir.
- Aracın denge ve kararlılığında ağırlık ve kaldırma kuvveti merkezlerinin konumları belirleyicidir.



# Denge, Kararlılık ve Yüzerlik

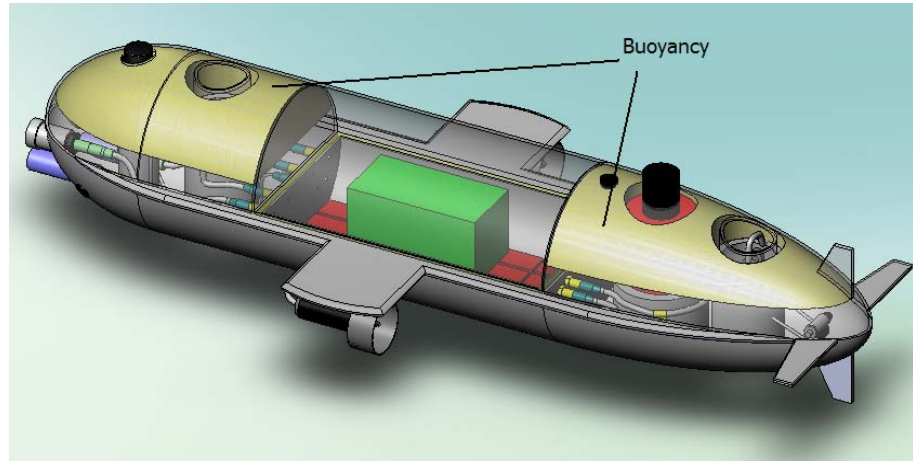
- Denge açısından iki merkezin x ve y konumları aynı olmalıdır.
- Kararlılık açısından ise kaldırma kuvveti merkezi, ağırlık merkezinin mümkün olduğunca üstünde olmalıdır
- İki merkez arasındaki bu mesafe aracın maruz kaldığı dönme etkilerine karşı düzeltici etkide bulunup kararlılık sağlamaktadır



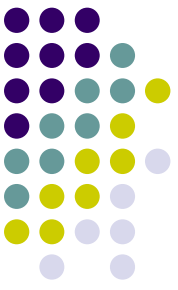


# Denge, Kararlılık ve Yüzerlik

- Araca belirtilen özellikleri sağlamak amacıyla alt bölümlere metal ağırlıklar, üst kısımlara ise yüksek yoğunluklu su emmeyen köpük eklenmektedir.



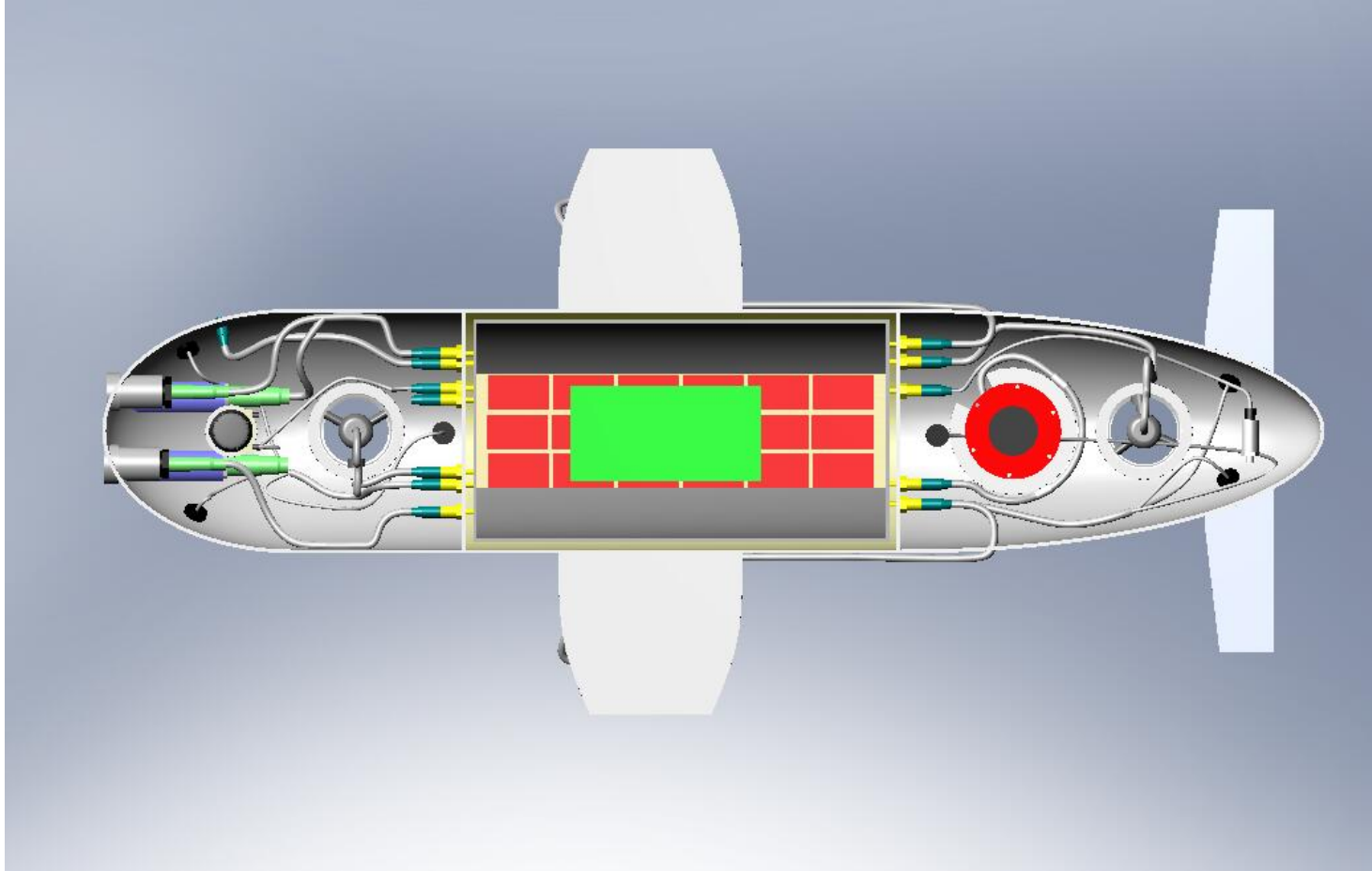
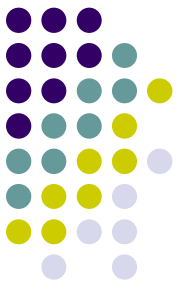
- Bu işlemler sonucu ağırlık ve kaldırma kuvveti merkezleri yaklaşık 3 cm kadar ayrılmışlardır.



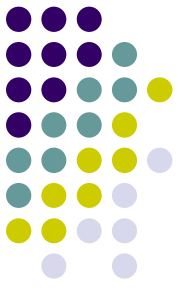
# Kablolama

- Denetim elemanları ve bataryalar su geçirmez bölümde saklanırken diğer cihazlar su geçirmez olduklarından su alan bölümde bulunmaktadır.
- Su geçirmez kablolar, aracın su geçirmez orta bölümü ile su alan bölümündeki cihazlar arasındaki bağlantıyı sağlamaktadır.
- Su geçirmez konektörler ise su geçirmez bölüm ile su alan bölümü ayıran duvardan geçişi sağlamaktadır.

# Kablolama



# Üretim



Model tasarımı ve imalatı



Kalıp üretimi



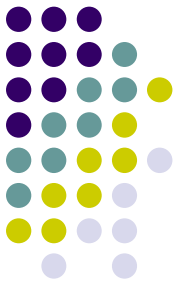
Gövde üretimi



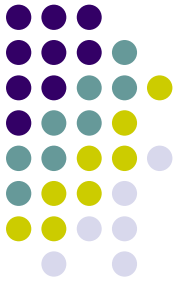
Testler



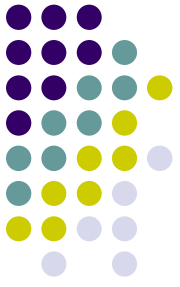
# Model



# Kalıp

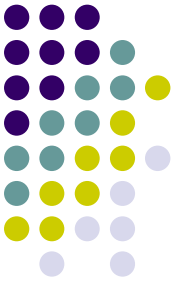


# Sonuç



- Mekanik tasarım evresi projenin başarısı açısından son derece önemlidir.
- Bu evrede çeşitli tasarım öğeleri göz önünde bulundurulmuş ve mühendislik çalışmaları gerçekleştirilmiştir.
- Proje çalışmalarında makina, elektronik ve bilgisayar mühendislerinden oluşan bir disiplinlerarası proje grubu görev almıştır.
- Gereken noktalarda analizler ve deneyler gerçekleştirilmiştir.
- Projede elde edilen tecrübe ve bilgi birikiminin önümüzdeki yıllarda gerçekleşebilecek benzer projelerde faydalı olması beklenmektedir.

# Teşekkür



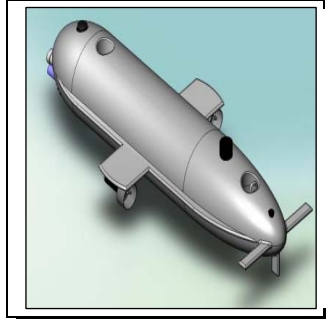
Projede maddi desteklerinden ötürü TÜBİTAK' a ve dış gövdenin tasarımı ve üretimindeki katkılarından dolayı *Bariş Elektrik Endüstrisi A.Ş.*'ye teşekkürlerimizi sunarız.

## Dinlediğiniz için teşekkürler



# **ULİSAR**

## **Ulusal İnsansız Sualtı Aracı Projesi**



**Donanım-Yazılım Raporu**

**Tarih: 30.04.2010**

# İçerik

1	Giriş	4
2	Donanım Bileşenleri Tasarımı	4
2.1	İlk Tasarım	4
2.1.1	Denetim Bilgisayarı (DB)	4
2.1.1.1	CPU Kartı	5
2.1.1.2	DC/DC Çevirici	6
2.1.1.3	Sabit Disk	6
2.1.1.4	Yoğunteker (Compact) Belleğin (Flash Memory) IDE Birimine Bağlantısı	6
2.1.1.5	Çevrebirimleri ve Denetim Bilgisayarı Arayüzleri	7
2.1.1.5.1	Rabbit 4000 $\mu$ P	7
2.1.1.5.2	Hareket Ölçüm Ünitesi (IMU) ve Denetim Bilgisayarına Arayüzü	10
2.1.1.5.3	Basınç Algılayıcısı ve Denetim Bilgisayarına Arayüzü	11
2.1.1.5.4	Sesüstü Dönüştürücüsü ve Denetim Bilgisayarına Arayüzü	11
2.1.1.5.5	Doppler Sürat Kaydedici (DVL) ve Altimetre'nin Denetim Bilgisayarına Arayüzü	12
2.1.1.5.6	Akustik Modem ve Denetim Bilgisayarına Arayüzü	12
2.1.1.5.7	İtici ve Denetleyicileri ile Denetim Bilgisayarına Arayüzleri	15
2.1.1.6	Bataryalar	15
2.1.2	Görüntü Bilgisayarı (VC)	16
2.1.2.1	CPU Kartı	16
2.1.2.2	DC/DC Çevirici	17
2.1.2.3	Çevre Birimleri ve Görüntü Bilgisayarına Arayüzleri	17
2.1.2.3.1	Sabit Disk ve Görüntü Bilgisayarına Arayüzü	17
2.1.2.3.2	Yoğunteker (Compact) Belleğin (Flash Memory) IDE Birimine Bağlantısı ve Görüntü Bilgisayarına Arayüzü	17
2.1.2.3.3	Kameralar ve Görüntü Bilgisayarına Arayüzleri	17
2.1.2.3.4	Sonarlar ve Görüntü Bilgisayarına Arayüzleri	18
2.1.2.3.5	MPEG4 Sıkıştırma Kartı ve Görüntü Bilgisayarına Arayüzü	19
2.1.2.4	Bataryalar	19
2.1.3	Elektronik Donanım Dizaynı	19
2.2	Yeni Tasarım	23
2.2.1	Genel Tasarım	24
2.2.2	Baskı Devre Kart Tasarımları	29
2.2.2.1	K1 Kartı	29
2.2.2.2	K2 Kartı	33
2.2.2.3	K3 Kartı	40
2.2.2.4	I5 Kartı	46
2.2.2.5	I6 Kartı	53
2.2.2.6	G1 Kartı	58
2.2.2.7	N4 Kartı	62
2.2.3	Güç Tüketimi	62
2.2.3.1	Donanım Elemanlarının Güç Bağlantısı	64
2.2.3.2	Pil Paketlerinin Belirlenmesi	65
2.2.3.3	Su geçirmez bölümdeki güç tüketimi	68
2.2.3.4	Su geçirmez bölümdeki donanım elemanlarının en yüksek sıcaklık değerleri	71

2.2.4	Donanım elemanlarının konektörleri	72
3	Yazılım Bileşenleri Tasarımı	80
3.1	İlk Tasarım	80
3.1.1	Sualtı Aracı Yazılım Tasarımı	80
3.1.1.1	Denetim Bilgisayarının Yazılım Tasarımı	80
3.2	Yeni Tasarım	84
3.2.1	Denizaltı Aracı Denetim Bilgisayar Yazılımı	84
3.2.1.1	Denetim Yazılımı Geliştirme Süreci	86
3.2.1.2	URB Denetim Bilgisayar Tarafı Yazılımı	87
3.2.1.3	Soyut Donanım Sınıfları	89
3.2.1.4	URB Protokolü Bilgisayar Tarafı Yazılımı Sınıf Diyagramı	94
3.2.1.5	RHexLib Kütüphanesi	96
3.2.1.6	Denetim Bilgisayar Yazılımlarının Derlenmesi	96
3.2.1.7	Denetim Bilgisayar İşletim Sistemi Görüntüsünün Hazırlanması ve Yüklenmesi	98
3.2.2	Denizaltı Aracı Görüntü Bilgisayar Yazılımı	99
3.2.2.1	Sualtı Kameralarından Görüntü Alınması	100
3.2.2.2	Görüntü Verisinin Operatör Bilgisayarına Gönderilmesi	101
3.2.2.3	İşletim Sistemi Görüntüsünün Hazırlanması ve Yüklenmesi	102
3.2.2.3.1	Açılış Sürecinin Özelleştirilmesi	103
3.2.2.4	Görüntü Bilgisayar Yazılımının Derlenmesi ve Yüklenmesi	107
3.2.3	Operatör Bilgisayar Yazılımı	108
3.2.3.1	Video Verisinin Çözümlemesi	111
3.2.3.2	Sonar Ekranının Yazılımı	114
3.2.3.3	Operatör Arayüzünün Derlenmesi ve Çalıştırılması	114
4	Karşılaşılan Problemler	115
5	Sonuç	116

# 1 Giriş

Bu raporda ULİSAR projesi kapsamında yapılan çalışmalar anlatılmaktadır. Raporda önceki dönemlerde yapılan çalışmalar, dizaynlar ve varsa değişimler sebepleriyle birlikte açıklanmıştır. Yazılım grubu kısmındaki önceki dönemlerde yazılması gerekli yazılımlar yapıldığı için bu dönem içerisinde daha çok geliştirilen donanımlarla yazılım entegrasyonu sağlanması çalışmaları yapılmıştır. Entegrasyon çalışmalarında bazı ufak tefek problemler dışında herhangi bir sorun ile karşılaşmamıştır. Ancak laboratuvar denemelerinden sonra yapılacak olan havuz denemeleri öncesinde elimizdeki motorların sürüş devrelerinde çıkan arızalardan dolayı havuz denemeleri yapılamamıştır.

## 2 Donanım Bileşenleri Tasarımı

### 2.1 İlk Tasarım

İlk olarak belirlediğimiz sistem dizaynı şu şekildedir. Denizaltı aracında iki adet bilgisayar bulunacaktır. Bu bilgisayarlar PC/104+ yapısında olacaklardır. Bilgisayarların birinde görüntü işlemleri, diğerinde ise denetim işlemlerinin gerçekleşmesi sağlanacaktır. İki bilgisayar arasında iletişim ethernet bağlantısı ile sağlanacaktır. Denizaltı aracında IMU [Hareket Ölçüm Ünitesi – Inertial Measurement Unit] (içerisinde sıcaklık ve pusula algılayıcıları de bulunmakta), basınç algılayıcısı, hız ölçer & altimetre, sonar, akustik modem, kamera, iticiler, MPEG dönüştürücü kart, aydınlatma, bataryalar ve sesüstü dönüştürücüler bulunacaktır. Denizaltı düşünülen ilk tasarımı Şekil 1 verilmiştir. İlerleyen bölümlerde sistem parçalarının görevleri belirtilmiştir.

#### 2.1.1 Denetim Bilgisayarı (DB)

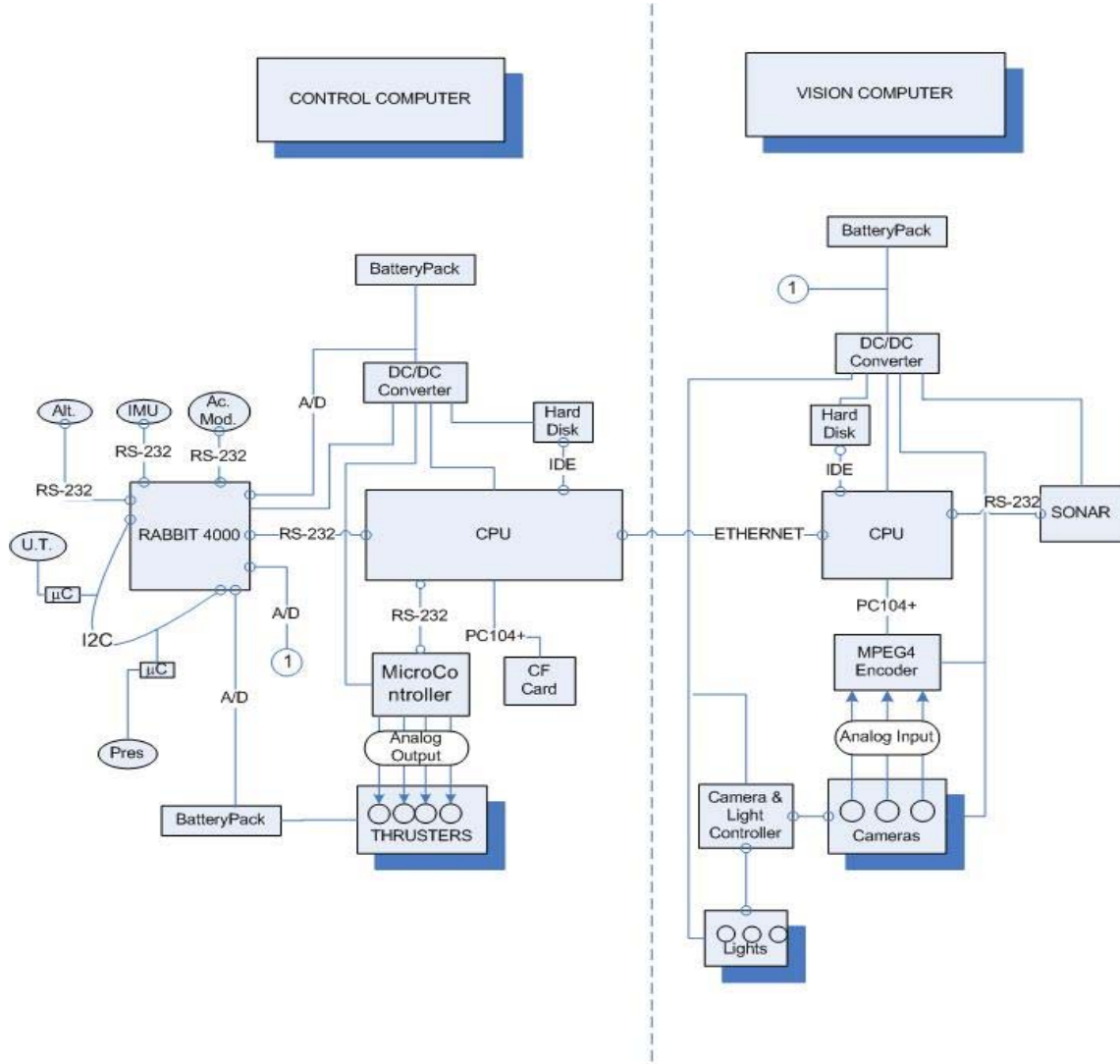
Denetim ve iletişim görevleri bu bilgisayar tarafından gerçekleştirilecektir. Algılayıcılar ve akustik modem Rabbit 4000 mikroişlemciye ( $\mu P$ ) bağlanacaktır. Operatör ile iletişim ise yarı çift yönlü (half-duplex) akustik modemle sağlanacaktır. Resim bilgisi görüntü bilgisayarından ethernet bağlantısı ile alınıp operatöre akustik modemle iletilecektir. Bant genişliğinin az olması nedeniyle siyah-beyaz ve düşük çözünürlüklü resim transferi gerçekleştirilebilecektir.



### 2.1.1.1 CPU Kartı

Denetim bilgisayarında Parvus CPU-1461 CPU kartı kullanılacaktır. Bu kart, Pentium III 800MHz işlemcisi, 6 USB 2.0 portu, 1 RS232 portu, 1 RS232/422/485 portu, 10/100Mbps ethernet portuna sahiptir. Bu kartın seçilme nedeni ise USB 2.0 desteği vermesi ve hızlı bir işlemciye sahip olmasıdır.

Algılayıcı ve haberleşme işlemleri maksadıyla Rabbit mikroişlemciye bir seri veriyolu yardımıyla bağlanılacaktır. Diğer seri veriyolu ise iticilerin denetimi maksadıyla kullanılacaktır. İticilerin denetimi analog voltaj (+/-5V) ile yapılacak ve bu işlem için ise Dijital-Analog Dönüştürücü (DAC) kart kullanılacaktır. Görüntü bilgisayarı ile iletişim ethernet bağlantısı üzerinden yapılacaktır.



Şekil 1. Denizaltı aracı sistem tasarımı

### **2.1.1.2 DC/DC Çevirici**

Denetim bilgisayarındaki PC/104'de hemen hemen tüm kartlar +5V giriş voltajına ihtiyaç duymaktadırlar. Projede kullanılacak olan batarya paketi 14.8V çıkış voltajı vereceğinden dolayı, bu voltaj DC/DC dönüştürücü kart vasıtasıyla uygun voltaja indirgenecektir. Bu işlem için ise Parvus ACS-5150 DC/DC dönüştürücü kart kullanılacaktır. Güvenlik ve güç kullanımı nedeniyle her bilgisayarın kendi dönüştürücü kartı bulunacaktır. Kartlardan biri arızalandığında bile diğer bilgisayar işlemine devam edebilecektir.

### **2.1.1.3 Sabit Disk**

Denetim bilgisayarında 20 GB 2,5 inçlik sabit disk kullanılacaktır. Bu sabit disk CPU kartın IDE'sine 2,5 inçten 3,5 inç çevirici adaptör ile bağlanacaktır. Algılayıcı bilgileri, hata ayıklama bilgisi ve diğer kayıtlar hard diskte kaydedilecektir. Algılayıcı bilgisi 100 Hz'de okunabilmesine rağmen akustik modem düşük bant genişliği nedeniyle 100 Hz'de gönderilemeyecektir, bu nedenle ileride değerlendirilebilmek için sabit diskte kaydedilecektir.

### **2.1.1.4 Yoğunteker (Compact) Belleğin (Flash Memory) IDE Birimine**

#### **Bağlantısı**

Yoğunteker bellek, işletim sisteminin ilk açılmasından itibaren gerekli yazılımın yüklenmesini sağlayacaktır. Ancak CPU kartı üzerinde yoğuntekerin bağlanacağı soket bulunmadığından, Flash kartın kullanılabilmesi için IDE birimine bağlı bir yoğunteker bulunmalıdır. Bu amaçla Parvus CTR-1461 PC104+ CardBus/PCMCIA Denetleyici kullanılacaktır. Bellek sadece ilk açılım esnasında kullanılacaktır. Bellek üzerine ne kadar çok bilgi yazılır ve okunursa, o ölçüde ömrü kısılacaktır. Bu nedenle belleğin az kullanımı ömrünü uzatacaktır. CTR-1641'in seçilme nedeni ise CPU'muza uyumunun yüksek olmasındandır.

### **2.1.1.5 Çevrebirimleri ve Denetim Bilgisayarı Arayüzleri**

Çevrebirimlerin çok fazla olması nedeniyle bunlarla iletişim büyük önem taşımaktadır. CPU'muz bütün çevrebirimlerle doğrudan ilişkili olamayacağından çözüm olarak algılayıcılarla CPU arasına başka bir mikroişlemci konacaktır. Rabbit 4000  $\mu$ P bu işlem için idealdir ancak Rabbit 4000  $\mu$ P'nin altı seri portu olmasına karşın altıdan fazla algılayıcı bulunduğundan I2C iletişim yolu kullanılacaktır.

I<sup>2</sup>C, Philips tarafından bulunan ve düşük-hızlı çevrebirimlerin anakarta, diğer gömülü sistemlere ve cep telefonlarına bağlanması için kullanılan seri bağlantı yöntemidir. 400Kbps'lik bant genişliğine sahiptir. Vsub'daki bazı algılayıcılar önce I2C yoluna bağlanacak daha sonra ise Rabbit 4000  $\mu$ P'e iletilecektir.

Bazı algılayıcıların analog çıktı vermesi nedeniyle çıktıların sayısal bilgiye çevrilmesi gerekecektir. A/D dönüştürücü olarak Analog Devices ADUC845 mikroişlemci kullanılacaktır. Analog veri sayısal dönüşürüldükten sonra I2C yoluna gönderilecektir.

#### **2.1.1.5.1 Rabbit 4000 $\mu$ P**

Çevrebirimlerle bağlantı Rabbit 4000 mikroişlemciyle sağlanacaktır. Rabbit 4000, 6 seri ve 1 ethernet bağlantısıyla güçlü bir işlemcidir. Rabbit 4000'ün çalışma zamanı 60MHz'dir. Programlamanın C diline benzemesi nedeniyle Rabbit'i programlamak çok zor değildir.

Tüm algılayıcı bilgilerinin okunması işlemciye yük bindirecektir. Algılayıcı işlemleri kesintiye uğrayabilecektir. Veri okunacak ve tek paket haline dönüştürüldükten sonra bilgisayara periyodik olarak gönderilecektir. Veri geldiğinde işlem kesintiye uğrayacak, paket sıraya alınacak ve daha sonra işlem devam edecektir. Örneğin program sıcaklık bilgisi gibi ilave başka bir bilgi isterse bu bilgiyi seçmesi gerekecektir. Bu bilgi seçildikten sonra Rabbit bunu ilgili algılayıcıdan okuyacak ve bilgisayara gönderecektir.

Rabbit 4000  $\mu$ P'e bağlanacak algılayıcılar: IMU, DVL&Altimetre, Basınç algılayıcısı, Sesüstü dönüştürücü ve akustik modem olacaktır.

Genel sistem mimarisinde Rabbit işlemci, genel olarak bütün almaç okuma işlemlerini gerçekleştirip, bu bilgileri ana denetim birimine bildirmekle yükümlüdür. Bu sebeple, bu işlemcinin yazılım mimarisi ve bu işlevin yerine getirebilmesi için gerekli olan çevresel donanım birimleri uygun olarak tasarlanmalıdır.

Almaç verilerinin okunup, uygun paketler halinde ana denetim birimine bildirilmesi genel olarak,

- Seri iletişim veri yollarından birisiyle okunan verilerin toplanması,
- Zaman etiketinin yapılandırılması,
- Kararlaştırılan veri paketlerinin yaratılması,
- Gerekli miktarda verinin depolanması,

olarak değerlendirilebilir.

### ***Almaç verilerinin toplanması***

Daha önceki raporda belirtildiği gibi, analog almaçlar uygun sayısallaştırma devreleri ve mikro denetleyiciler kullanılarak I2C veri yolu kullanılarak bu işlemciye veri göndereceklerdir. Diğer almaçlar ise hali hazırda seri iletişim veri yolları kullandıkları için onları dorudan Rabbit işlemciye bağlanacaklardır. Bu raporda sadece IMU ve GPS'in Rabbit işlemciye bağlanması anlatılacaktır.

Bu proje kapsamında, Microstrain firmasının 3DMGX1 ataletsel ölçüm birimi kullanılacaktır. Bu IMU değişken veri hızlarında, ivmeölçer, açısal hız ölçer, magnetometre ve termometre almaç verilerini RS232 standardında çıktı olarak vermektedir. Aynı zamanda, bütün verilere zaman etiketi yapıştırmakta ve ölçümlerin ölçümleme düzeltmelerini kendisi gerçekleştirmektedir.

IMU'nun çıktı frekansı ayarlanabilmekle beraber, istenilen zamanlarda da çıktı gönderebilmektedir. Başka bir ifade ile bütün ölçümleri periyodik olarak gönderebilmekle beraber, istenildiği takdirde ölçümleri aperiodyk olarak da çıktı olarak gönderebilmektedir.

Bu çalışmada, zaman etiketinin doğruluğundan emin olabilmek için aperiodyk çıktılar kullanılmıştır. IMU verileri, üretici firmanın belirlediği paketleme biçimindedir.

Bu çalışmada, IMU'dan 50 Hz hızında ölçümlere düzeltmeleri yapılmış veri toplanmıştır. Bu amaçla, IMU'ya her 20 mili saniyede çıktısı alınacak verinin kodu gönderilmiş ve veri isteme zamanı tutulmuştur. Bu işlem için Rabbit işlemcinin Timer B zamanlayıcısı kullanılmıştır.

Bu projede kullanılan GPS, U-Blox firmasının SBR-LS model GPS alması kullanılmıştır. Bu alması, 1 Hz frekansında konum ve hız verilerini RS232 veri yolunda gönderebilmektedir. Bu alması verileri NMEA standardında dizgi veri olarak veya UBX standardında ikili veri olarak çıktı verebilmektedir. Bu çalışmada, UBX standardı kullanılmıştır.

IMU verileri aperiodyk olarak, GPS verileri ise periyodik olarak okunmuştur. IMU ve GPS'ten okunan verilerin zaman etiketlemesi için Rabbit işlemcisi üzerinde 100 Hz frekansında çalışan, 4 Byte uzunluğunda bir sayaç kullanılmıştır.

Alınan verilerin içeriğini ve okuma zamanlarının gözlemlenebilmesi amacı ile MATLAB'da bir arayüz yazılımı gerçekleştirilmiştir.

## **Yazılım mimarisi**

Rabbit işlemci üzerindeki yazılım, yazılım koşma hızı düşünülerek assembly dilinde yazılmıştır.

IMU ve GPS'ten gönderilen veri paketleri birbirinden farklı olduğu için, bu iki birimin verilerini okuyan seri iletişim kanallarının is kesme işlevleri birbirinden farklıdır. Ancak bu iki işlevin çalışma prensibi birbirinin aynısıdır.

IMU ve GPS'ten gönderilen paketlerin en uzunundan daha uzun bir veri tamponu oluşturmak veri kaybını önlemek için gerekli ilk adımdır. Bu veri tamponuna veriler, seri iletişim is kesme işlevlerinde yazılmaktadır. Bu nedenle, tampona verinin yazılma zamanı önceden ön görülemez. Dolayısıyla, tampona yazılan verilerin boşaltılması işlevinin daha uygun bir biçimde yapılması gerekir.

Bu çalışmada, seri iletişim is kesmelerinin içerisinde gelen paketin tipi çözümlenmiş ve paketin uzunluğu kadar veri, tampona yazılmıştır. Bu nedenle, yaratılan tamponu tek bir dizgiden değil, uygun sayıda birden fazla dizgiden oluşturmak daha uygun hale gelmiştir. Yaratılan tampon dizgileri ise dairesel bir dizgi içerisinde tutularak okunan tamponlar bu dizginin en sonuna yerleştirilmiştir. Bu nedenle, uygun sayıda tampon kullanılarak paket kaybının önüne geçilmiştir.

Paketlerin içeriğini çözümlen birimin kaç paketin içeriğini çözümleneceği, başka bir hafıza bölgesinde yaratılan bir değişkende tutulmuştur. Bundan sonraki işlemler genel olarak paketlerin içeriğinin çözümlenmesi ve verilerin kaydedilmesidir.

Gelen paketlerin içeriği çözümlenmeden önce, alınan verinin doğruluğu denetlenmelidir. Bu amaç ile gelen verinin uzunluğu ve paketlerin içerisinde yer alan sağlama toplamı veri kullanılmıştır. Bu ölçüt sağlandığı takdirde, veriler gönderilen paketin içeriğine uygun olarak tanımlanmış yapılar içerisine yazılmaktadır. Böylece, alınan bütün veriler zaman etiketi dahil olmak üzere bir veri dizilimine yerleştirilmektedir.

Gelen ve kaydedilen veriler, ana denetim sistemine gönderilmeden önce daha önceki raporda anlatılan paketlere dönüştürülmektedir. Böylece, verilerin doğruluğu ve okuma zamanları tam olarak denetlenebilmektedir.

#### ***2.1.1.5.2 Hareket Ölçüm Ünitesi (IMU) ve Denetim Bilgisayarına Arayüzü***

IMU denetim ve seyir sistemleri için gereklidir. Bu projede IMU olarak MicroStrain 3DM-GX1 modeli kullanılacaktır. 3DM-GX1 üç açılı rate cayroyu; üç dik koordinatta DC ivme ölçer, çoğullayıcı, 16 bit A/D dönüştürücüyü birleştirmekte ve dinamik/statik çevreirim değişikliklerini gömülü işlemcisiyle hesaplayarak çıktı olarak vermektedir. Ayrıca sıcaklık bilgisini de elde etmektedir. IMU çıktısı RS-282 veya RS-485 olabilmektedir. IMU Rabbit µP'e RS-282 veriyoluyla bağlanacaktır. Çıkış frekansı 100 Hz'dir.

### ***2.1.1.5.3 Basınç Algılayıcısı ve Denetim Bilgisayarına Arayüzü***

Basınç algılayıcısı su yüzeyinden olan derinlik bilgisini verecektir. Derinlik bilgisi ise denetim ve seyir işleri için kullanılacaktır. Basınç algılayıcısının çıktısı analog olacağından bu çıktı ADUC845  $\mu$ C'e bağlanarak sayısala çevrilecek ve Rabbit 4000  $\mu$ P'e I2C yoluyla iletilecektir. Basınç algılayıcısı bilgisi de sürekli olarak sabit diske kaydedilecek ve Vsurf'e 1 Hz'le gönderilecektir.

### ***2.1.1.5.4 Sesüstü Dönüştürücüsü ve Denetim Bilgisayarına Arayüzü***

Sesüstü Dönüştürücüsü çevredeki cisimlerin uzaklığını ölçme amacıyla kullanılacaktır. Çevredeki cisimler yakınsa yüksek, uzaksa düşük voltaj çıktısı verecektir. Dönüştürücü daha çok güvenlik amacıyla çevredeki cisimlerin tespitine yarayacak ve buna göre sakınma yapılabilecektir. Cisim tespiti iki nedenle kritiktir.

Birincisi, işletmenin sualtını sadece kamera ve sonardan gelen görüntülerle görebiliyor olmasından ve bu cihazların alta ve öne doğru bakmalarından dolayı kısıtlı görüş alanına sahip olmasındandır. Bu da işletmenin yukarıyı, yanları ve arkayı görememesi sakıncasını doğurmaktadır. Dönüştürücüler ise işletmenin göremediği bölgelerde ortaya çıkabilecek cisimler hakkında işletmeni uyaracaktır. Bu durum otomatik denetim açısından da çok faydalı olacaktır. Tespit edilen cisimlerin çok yakın olması durumunda ise denizaltı aracı acil durum moduna geçecektir. Acil durum modunda ise işletmen denizaltı aracına denetim gönderemeyecek, denizaltı kendisi tehlikeden sakındıktan sonra gelen komutlara izin verecektir.

Diğer neden ise iletişim kalitesini yükseltmek maksadıyla olacaktır. Operatör ile denizaltı arasında iletişim akustik modemle sağlanacağından dolayı, araya engellerin girmesiyle iletişim kaybı oluşacaktır. Dönüştürücüler bu konuda araçlara yardımcı olacak, araçlar engeli bertaraf edebilmek için gerekli manevrayı yaparak iletişim kalitesi yükseltilecektir.

Dönüştürücü miktarının 6 ile 8 arasında olması düşünülmekte olup bu rakam araçların büyüklüğü ve dönüştürücülerin yerlerine göre kesinlik kazanacaktır. Sesüstü dönüştürücü de ADUC845  $\mu$ C'e bağlanacaklardır. ADUC845  $\mu$ C tarafından dijitalle dönüştürülen veriler Rabbit 4000  $\mu$ P'e I2C yoluyla iletilecektir.

#### ***2.1.1.5.5 Doppler Sürat Kaydedici (DVL) ve Altimetre'nin Denetim Bilgisayarına Arayüzü***

DVL yere göre sürat bilgisini ölçmek amacıyla kullanılacaktır. Sürat bilgisi ise denetim ve seyir işlemleri için çok önemlidir. DVL ile iletişim Rabbit 4000 ile RS-232 üzerinden sağlanacaktır. Bu projede kullanılması planlanan DVL, RD instrument's WorkHorse DVL'dir. Bu DVL kendi bünyesinde altimetre ihtiva etmektedir. Altimetre ise bize deniz dibinden olan yüksekliği verecektir.

#### ***2.1.1.5.6 Akustik Modem ve Denetim Bilgisayarına Arayüzü***

Akustik modem, denizaltı ile denizüstü aracı arasındaki iletişimi sağlayacağından dolayı ULISAR projesinin en önemli parçalarından biridir. Ancak günümüz teknoloji ile akustik iletişimde büyük sorunlar mevcuttur. bunların en büyüğü ise düşük bant genişliğidir. Ticari olarak bulunabilen en gelişmiş akustik modemin bant genişliği ancak 30 Kbps kadardır. Bu modem ortalama resim bilgisi göndermek için yeterlidir ancak bunun da fiyatı çok pahalıdır. Ayrıca menzil de akustik iletişimde büyük problemler yaratmaktadır. Menzil arttıkça kırılım, emilim ve diğer kayıplar artmakta böylece iletilen veri miktarında düşüş yaşanmaktadır. En iyi iletişimi sağlayabilmek için modemlerin birbirini karşılıklı görmeleri ve arada engel bulunmaması gerekmektedir.

ULISAR projesinde kullanılacak olan modem Link-Quest UWM 1000 modelidir. 7000 bps veri iletim kapasitesine sahip olan bu modem sayesinde tüm bilgiler (resim, seyir, algılayıcı bilgileri, GPS verisi, v.b.) karşılıklı iletilebilecektir. Seyir ve algılayıcı bilgilerinin bant genişliği içindeki yeri az olmasına rağmen resim ve görüntü bilgisi çok fazla yer tutmaktadır.



Bant genişliği kısıtlamasından dolayı görüntü bilgisi düşük çözünürlüklü ve siyah/beyaz olarak iletilecektir.

Projede en önemli hususlardan biri de iletişimin asla aksamaması gerektiğidir. Eğer akustik modemde uzun süreli arıza ortaya çıkarsa denizaltı aracı acil durum moduna geçecek ve bu süreçte iletişimi tekrar kurmak için belli bir süre denemelerde bulunacak fakat iletişimin tekrar sağlanamaması durumunda su sathına çıkmak için gerekli manevraları yapacaktır. Tablo 1 ve Tablo 2'de  $V_{sub}$  ile  $V_{surf}$  arasında iletilecek veri hakkında bilgi verilmiştir.

Veri Paketi Adı	Veri Tipi	Veri Uzunluğu(Byte)
Sonar	Sonar görüntüsü(Ayrık Kosinüs Transformasyonu - Cosine Discrete Transform)	100
İşlenmiş Sonar Bilgisi	Açılar (IMU)	6
	İvmeler (IMU)	6
	Manyetik Alan (IMU)	6
	Zaman (CPU Zamanı)	4
	Basınç(Basınç Algılayıcısı)	2
	Basınç Farkı (Sürat ölçümü için, Pito tüp)	2
	Dipten mesafe (Altimetre)	2
	Mesafe ölçer(Sesüstü Dönüştürücüsü)	1
	Batarya Bilgisi (Batarya monitörü)	6
	Sıcaklık (IMU)	2
Paket Bilgisi	Paket Bilgisi (Senkronizasyon, denetim biti,paket büyüklüğü,v.b.)	10
	<b>TOPLAM</b>	147

*Tablo 1. Denizaltından Denizüstü Aracına Veri iletimi*

Veri Paketi Adı	Veri Tipi	Veri Uzunluđı(Byte)
Denetim Komutları	Mod (Uzaktan Kumanda ya da Otopilot)	1
	İtki vektörü (Uzaktan Kumanda)	6
	WayPointler (Otopilot Modu)	12 x # of W.P.
	Waypoint geçiř Sürati (Otopilot Modu)	4x # of W.P.
	Kamera&Aydınlatma Komutları	1
	Sonar Komutları(Yakınlař/Uzaklař)	2
	Pozisyon düzeltme(Pozisyon+Sürat)	24
Paket Bilgisi	Paket Bilgisi (Senkronizasyon, denetim biti,paket büyüklüğü,v.b.)	10
	<b>TOPLAM</b>	44 + (W.P.)x16

*Tablo 2. Denizüstü Aracından Denizaltı Aracına Veri İletimi*

Akustik modemin veri bantı 7000bps yani 875 bayt/saniyedir. Böylece 1 baytı aktarma süresi 1.14 milisaniyedir. Ayrıca akustik modem iletim yöntemi yarı çift yönlü olması nedeniyle birim zamanda ya gönderme ya da sadece alım işlemini yapabileceğiz.

Denizaltından denizüstüne giden paket büyüklüğü 147 bayttır ve bu paket büyüklüğü standart olmasına karşın Denizüstü aracından denizaltı aracına giden paket büyüklüğü operasyon moduna göre deęişiklik göstermektedir. Operasyon moduna göre iletilecek veri bilgisi deęişiklik gösterecektir fakat biz bu paket bilgisini sabit tutarsak gereksiz bant genişlięi kullanımı olacaktır. En az paket büyüklüğü 44 bayt'tır. Her iki paketi de transfer etmek istediğimizde  $147+44=191$  bayt gerekmektedir. Bu deęeri 200 bayt kabul ettiğimizde ise toplam veri genişlięini 875 bayt olarak düşündüğümüzde bir saniyede en fazla dört paket gönderebileceğimiz ortaya çıkar. Eđer kamera bilgisini de göndermek istersek bu deęer bir saniyede 1 göndermeye tekabül eder ki bu durumda da kamera görüntüsü çok düşük kalitede olacaktır.

### ***2.1.1.5.7 İtici ve Denetleyicileri ile Denetim Bilgisayarına Arayüzleri***

Denizaltı aracında dört adet itici kullanılacaktır. Bu iticilerden ikisi tirim ve derinliği denetleyecek diğer ikisi ise dönüş açısını denetleyecektir. Denizaltı aracında kullanılacak iticiler Tecnadyne 260 modeli olacaktır. Bu iticilerin denetimi analog +/-5V giriş ile sağlanacaktır. Bu nedenle denetim için D/A dönüştürücü gerekmektedir. Bu işlemi ise Rabbit 4000 veya ADUC845  $\mu$ C'leri ile yapabiliriz.

### **2.1.1.6 Bataryalar**

Bu projede AURORRA LTD., Kokam 2000 mah-15C-Deans 4 serisi, kapalı hücreli pil paketleri şeklindeki bataryalar kullanılacaktır. Bu bataryada her paket 4 hücreli olup, 14.8V nominal, 2000 mAh çıkış verir. İtici ve denetleyicilerin güç gereksinimlerini sağlamak için, batarya paketleri yeniden düzenlenecektir. Araçlarımızın sağlıklı çalışabilmesi ve pillerin yeniden dolmuş ömürlerinin kısaltılmaması için, araçların üzerinde bir batarya takip devresi bulunacaktır. Bataryanın enerjisi belli bir seviyenin altına indiğinde, AUV, acil durum konumuna geçip, su yüzeyine doğru gitmelidir. Genel olarak, batarya takip devresinin ölçümleri, sistem işletmenine, su üstüne çıktıktan sonra iletilecektir. Batarya takip sistemi ve her bir batarya, 8 kanallı bir A/D dönüştürücüsü olan Rabbit 4000 mikro işlemcisine bağlanacaktır.

Denetleme ve görüntü bilgisayarlarının her ikisi de aynı türde ama farklı batarya paketlerine (güvenlik nedeni ile) bağlanacaktır. Böylece, paketleri veya bilgisayarların birinde bir problem olduğunda, diğer bilgisayar, çalışmaya devam edecek, sistemi acil durum konumuna getirip su üstüne çıkaracaktır.

Bataryaları doldurmak için, AURORRA LTD. Firmasının, Graupner Ultraduo 30 doldurucu ekipmanı kullanılacaktır. Bu doldurucu, bir seferde 10 hücreye kadar dolmuş yapabilmektedir. Bataryaları, uzun süre sağlıklı bir şekilde kullanabilmek için, her dolmuşdan önce bunların tamamen boşaltılması gerekmektedir. Yukarıda adı geçen doldurucu, aynı zamanda boşaltım da yapabilmektedir.

## **2.1.2 Görüntü Bilgisayarı (VC)**

Bu bilgisayar sadece görüntü ile ilgili işlerden sorumludur. Görüntü işlemleri (görüntü sıkıştırma, görüntü işleme) çok yük getirdiğinden dolayı, bu işler için, denetleme bilgisayarından ayrı bir bilgisayar ayrılması gereği ortaya çıkmıştır. ULİSAR projesinin ilk aşamalarında, denizaltı aracındaki görüntü bilgisayarı, sadece sıkıştırılmış sonar ve düşük çözünürlüklü resim bilgilerini operatöre gönderecektir. Projenin daha sonraki adımlarında, sonar ve kameradan gelen sinyaller de denetleme işlemleri için kullanılabilir. Dolayısı ile, denetim ve görüntü bilgisayarlarının kendi aralarında haberleşmesi gerekecektir.

Akustik modemin bant genişliği çok düşük olduğundan dolayı, sadece çok düşük çözünürlüklü siyah/beyaz görüntüler ve/veya düşük çözünürlüklü sonar görüntüleri gönderilecektir. Şimdilik düşük çözünürlüklü sonar görüntüleri gönderilmesi düşünülmektedir. İşletmen, sonar görüntüsünde herhangi bir ilgi çekici durum yakaladığında, bu durumun yüksek çözünürlüklü görüntüsü elde edilip sabit diske kayıt edilecektir. Hemen hemen bütün görüntü verileri sonradan işlenecektir. Bu bant genişliğinde gerçek zamanlı video dizileme zaten mümkün değildir.

### **2.1.2.1 CPU Kartı**

Görüntü bilgisayarında Parvus CPU-1461 CPU kartı kullanılacaktır. Bu kartın üzerinde Pentium III 800 MHz işlemci, 6 adet USB 2.0 bağlantısı, 1 adet RS232 bağlantısı, 1 adet RS232/422/485 bağlantısı, 10/100Mbps ethernet bağlantısı bulunmaktadır. Denetim bilgisayarı ile irtibat ethernet bağlantısı üzerinden sağlanacaktır. Sonar, görüntü bilgisayarına RS-232 ile bağlanacaktır. Kameraların çıkışı PC104+ uyumlu olan bir MPEG4 video sıkıştırma kartına bağlanacaktır. Böylece PC104+ veriyoluna sayısal video kamera çıktıları gelecektir.

### **2.1.2.2 DC/DC Çevirici**

Bu projede kullanılacak bataryalar 14.8V çıkışlıdır. Bu voltaj, DC/DC çevirici vasıtası ile uygun değerlere getirilecektir. Bu işlem Parvus ACS-5150 DC/DC çevirici kartı ile yapılacaktır. Güç tüketimini azaltmak ve sistemin dayanıklılığını artırmak için, her bilgisayarın kendi DC/DC çeviricisi olacaktır. Kartlardan biri arızalansa bile diğer bilgisayar çalışmaya devam edecektir.

### **2.1.2.3 Çevre Birimleri ve Görüntü Bilgisayarına Arayüzleri**

#### ***2.1.2.3.1 Sabit Disk ve Görüntü Bilgisayarına Arayüzü***

Görüntü bilgisayarında 40GB sabit disk olacaktır. Bu sabit disk standart bir sabit disk sürücüsü olarak çalıştırılacak ve CPU kartının IDE girişine bağlanacaktır. Kamera ve sonar görüntüleri sonradan işlenmek üzere bu sabit diske kayıt edilecektir.

#### ***2.1.2.3.2 Yoğunteker (Compact) Belleğin (Flash Memory) IDE Birimine Bağlantısı ve Görüntü Bilgisayarına Arayüzü***

Yoğunteker bellek, işletim sisteminin ilk açılmasından itibaren gerekli yazılımın yüklenmesini sağlayacaktır. Ancak CPU kartı üzerinde yoğuntekerin bağlanacağı soket bulunmadığından, Flash kartın kullanılabilmesi için IDE birimine bağlı bir yoğunteker bulunmalıdır. Bu amaçla Parvus CTR-1461 PC104+ CardBus/PCMCIA Denetleyici kullanılacaktır.

#### ***2.1.2.3.3 Kameralar ve Görüntü Bilgisayarına Arayüzleri***

ULİSAR projesinde 3 adet siyah/beyaz kamera kullanılacaktır. Haberleşme bant genişliğinin renkli görüntü alınabilmesi için yetersiz olması nedeniyle, kameralar siyah/beyaz olarak seçilmiştir. 3 kamera seçilerek görüş alanının artırılması amaçlanmıştır. Bir kameranın görüş açısı yaklaşık 45°/50°'dir. Yeterli ışığın sağlanabilmesi için 3 adet aydınlatma kullanılacaktır.

Her bir kameranın üzerine ayrı aydınlatma yerleştirilecektir. kamera ve aydınlatmaların denetimi için görüntü bilgisayarı tarafından denetlenecek ayrı bir devre tasarlanacaktır. Bu devrenin kullanımının ana nedeni ise enerji tasarrufunun sağlanmasıdır. Sadece istenen zamanlarda kamera ve aydınlatmaların kullanılacak olması ile önemli ölçüde enerji kazanımı sağlanacaktır.

Projede OutlandTech. UWC 300 düşük ışıklı siyah/beyaz su altı kameraları kullanılacaktır. Kameraların hassasiyeti 0.0003 Lux'tür. Düşük ışıklı ortamlarda hassasiyetin büyük önemi vardır. Hassasiyet değeri düşükçe kameranın başarıml oranı artmaktadır. Çözünürlük oranı ise 600 yatay çizgiyle sınırlıdır. Kameranın çıkışı analog olduğundan dolayı, gerekli sıkıştırmanın yapılabilmesi için çıkışın sayısallaştırılması gerekmektedir. Bu nedenle uygun bir MPEG4 kart kullanılacak ve görüntü bilgisayarı üzerindeki fazladan yük azaltılmış olacaktır.

Kullanılacak ışıklar ise OutlandTech. UWL 300 Düşük voltajlı aydınlatmalar olacaktır. Bu ışıkların girişleri 12 V ve güçleri 35W'dır. Kameraların, düşük aydınlatmalı ortamlarda çalışma imkanları olmasına rağmen, derinliğin artmasıyla birlikte harici ışık kaynağına olan ihtiyaç artacaktır.

#### ***2.1.2.3.4 Sonarlar ve Görüntü Bilgisayarına Arayüzleri***

Sonar su altı araçları için büyük önem taşımaktadır. Görüş alanının kısıtlı olması ve derinliğin artmasıyla birlikte ışığın azalması sonucu kameralar yeterli görüntü veremeyebilirler. Aynı zamanda su altında kameraların erimi sonar kadar fazla değildir. İyi aydınlatma ortamında kameralar en fazla 10-15 metre uzaklığı gösterebilirler fakat su altı araçlarında kullanılan sonarların erimleri 300 metreye kadar çıkabilmektedir. Sonarlar sesüstü görüntüleme teknikleri kullanırlar. Çevreyi taramak için ise ses dalgalarını kullanırlar. Sonarların aynı zamanda uzaklık bilgisi vermeleri nedeniyle, ekranlarında gördüğümüz cisimler mesafelerini de öğrenebiliriz. Araçların denetim işlemlerinde de sonar verilerinden büyük ölçüde yararlanılacaktır.

Bu projede Tritech firmasının Super SeaKing DFS sonarı kullanılacaktır. Bu sonar görüntüyü RS 232 iletişim kanalıyla sayısal olarak verecektir. Böylece sonar, görüntü bilgisayarına seri yolla bağlanacaktır. Görüntü bilgisayarına ise gelen sonar görüntüsünü sıkıştırarak ve ethernet yoluyla denetim bilgisayarına gönderecektir.

#### ***2.1.2.3.5 MPEG4 Sıkıştırma Kartı ve Görüntü Bilgisayarına Arayüzü***

MPEG4 Sıkıştırması için Parvus CTR-1472 MPEG4-Video sıkıştırma kartı kullanılacaktır. Bu kartın 4 analog girişi vardır. Karta gelen veriler zaten analog olduğundan ayrıca bir çerçeve sayısallaştırıcı karta ihtiyaç bulunmamaktadır. Aynı zamanda bu kart çerçeve sayısallaştırıcısının görevini de yerine getirebilmektedir.

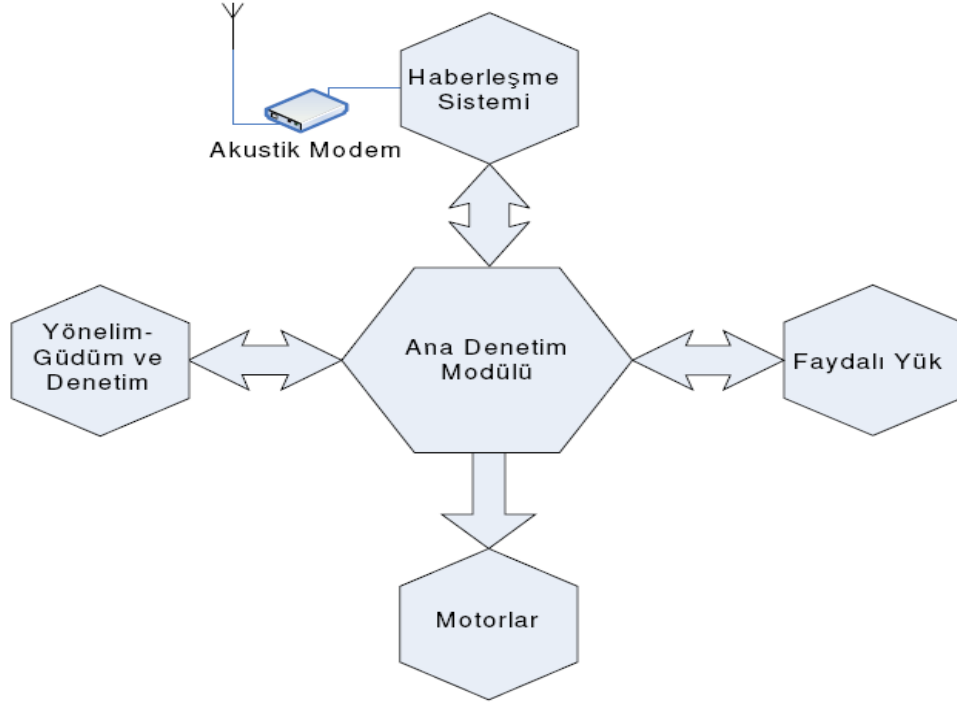
#### **2.1.2.4 Bataryalar**

Görüntü bilgisayarında kullanılan bataryalar, denetim bilgisayarında kullanılanlarla aynı olacaktır. Görüntü bilgisayarına ilaveten, aydınlatma ve sonar çok daha fazla güce gereksinim duyacaktır. Güç tüketiminin azaltılması amacıyla aydınlatma sürekli kullanılmayacak sadece gereksinim duyulan zamanlarda kullanılacaktır.

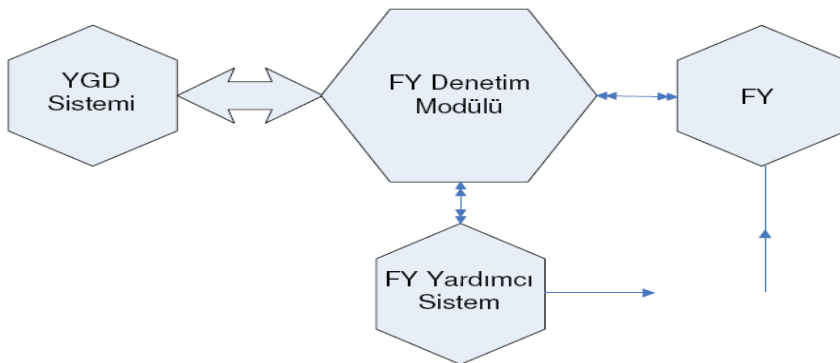
#### **2.1.3 Elektronik Donanım Dizaynı**

Su altı aracının elektronik donanımı genel olarak iki kısımdan oluşmaktadır:

1. Faydalı Yük Sistemi (FY): Bu kısımda faydalı yük ile ilgili olan bütün donanım elemanları yer almaktadır.
2. Yönelim-Güdüm ve Denetim Sistemi (YGD): Bu kısımda su altı aracının denetleme ve yönelim-güdüm sisteminin tüm donanım elemanları yer almaktadır.

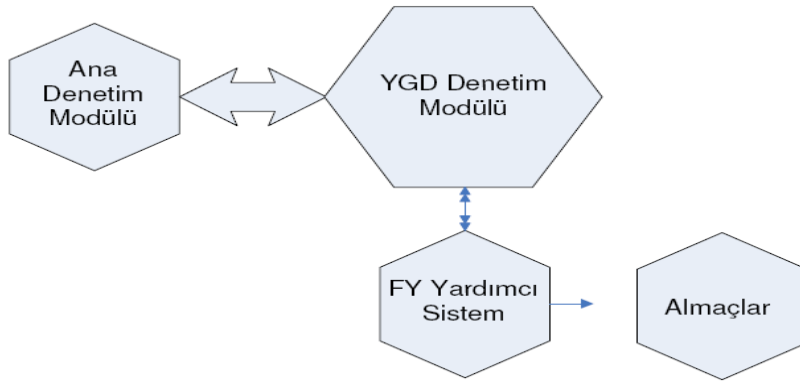


Su altı aracı ana denetim modülünden, faydalı yük sisteminden, motor sisteminden, almaç sisteminden ve akustik modemden oluşmaktadır. Ana denetim modülü, diğer sistemlerin eş zamanlı çalışmasına olanak sağlayan genel sistem denetleyicisidir. Daha önce belirtildiği gibi su altı aracı hem otomatik kumanda sistemine hem de elle kumanda sistemine sahip olması gerekmektedir. Ayrıca, otomatik kumanda modunada iken, aracın çeşitli acil durumlarda önceden belirlenmiş senaryoları uygulaması beklenmektedir. Bu nedenle ana denetim modülünde, aracın farklı modlarda çalışmasına olanak sağlayabilen karar verme mekanizması bulunmaktadır. Şekildeki ana denetim modülü hem faydalı yük denetimini hem de yönelim ve güdüm kısmını kapsamaktadır. Ancak, bu iki kısım birbirini etkilemekle beraber, faydalı yük sistemi ve YGD sistemi bağımsız iki sistem olarak da değerlendirilebilir. Bu durumda, su altı aracının faydalı yük sistemi aşağıdaki gibi gösterilebilir:

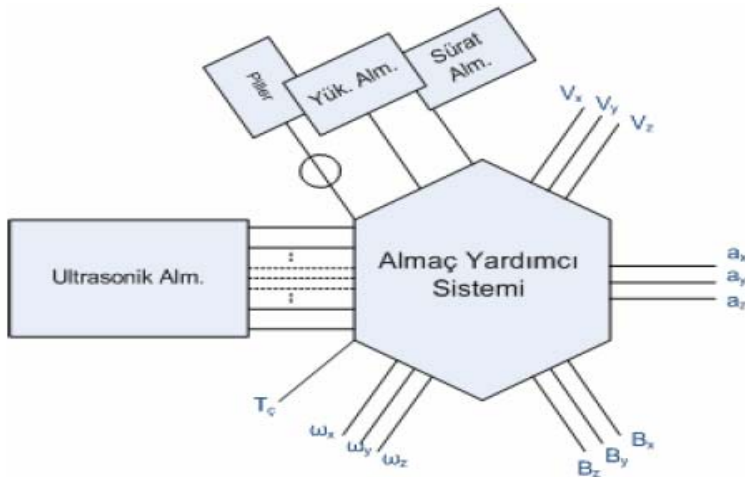




Şekilde gösterilen faydalı yük sisteminde, FY Yardımcı sistemi olarak adlandırılan alt sistem bölümü, faydalı yükün denetimi ve işlerliğini artırmak için gerekli olan donanım elemanlarını içermektedir. Örneğin, derin denizde gündüz kameralarının çalışabilmesini sağlamak için gerekli olan çeşitli ışıklandırma elemanları bu kısımda yer alacaktır. Bu elemanların denetimi yine FY Denetim Modülü tarafından yapılacaktır. Bu projede FY, gündüz kameraları ve sonar görüntüleme cihazıdır.

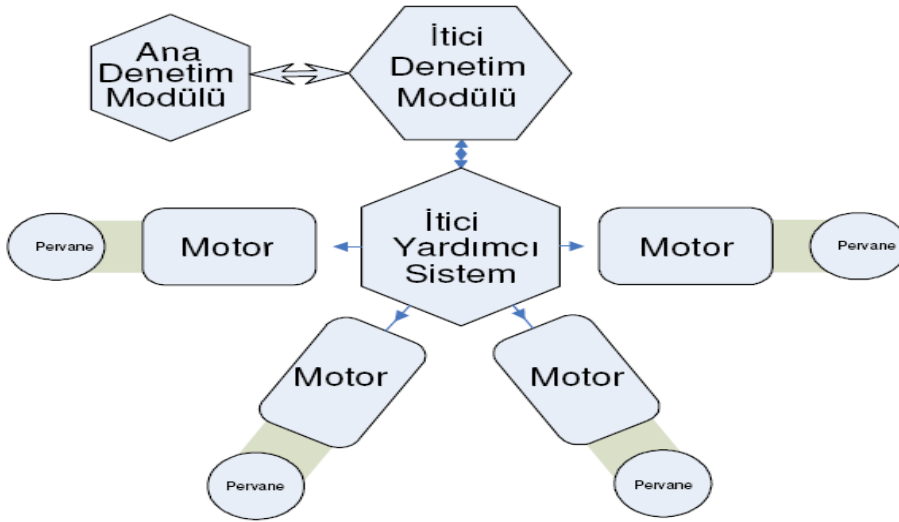


Şekilde YGD sistemi genel olarak gösterilmiştir. Bu kısmın temel işlevi su altı aracının sağlıklı biçimde istenilen pozisyona istenilen şekilde hareket etmesini sağlamak olduğu için, aracın hareketinin denetlenmesi gerekmektedir. Bu amaçla, su altı aracı donanım mimarisi çeşitli hareket ve durum almaçlarını içermesi gerekmektedir. Şekilde gösterilen “Almaçlar” sisteminin açık hali aşağıda gösterilmiştir.



İvmeölçer, açısal değişim ölçer ve manyetik alan ölçer, “Ataletsel Ölçüm Ünitesi” (IMU)’ni oluşturur. Ucuz IMU sistemleri, belirgin olmayan ve zamanla değişen hatalar yaparlar. Bu hatalar, yönelim ve güdüm sisteminde basit olmayan tahmin filtreleri kullanılmasını zorunlu

kılmaktadır. Bu tahmin filtreleri, hata özellikleri zamanla değişmeyen ve birbirinden bağımsız ölçümler yapan başka bir almaç yardımı ile çalışmaktadır. Literatürde en sık karşılaşılan uygulama DVL kullanımınıdır. Başka bir deyişle, DVL ölçümleri IMU ölçümlerini düzeltmek ve yükseklikölçerle birlikte su altı aracının konumunun belirlemek amacıyla kullanılacaktır. Ultrasonik almaçlar ise aracın çevresindeki engelleri tespit etmek amacıyla kullanılacaktır. Bu almaçlar, oldukça kritik olan engelsiz sefer kısıtının sağlanmasında görev alacaklardır. Şekilde gösterilen motorlar alt sistemi, küçük deniz altı sistemlerinde genellikle iticilerden oluşur. Her bir itici bir pervaneden ve bir tane de motordan oluşur. Kanatçıklı sistemlerinden farklı olarak, bu sistemlerde itici vektörü değiştirilerek deniz altı aracının durumu değiştirilmeye çalışılır. Bu nedenle motor sisteminde her bir eksene karşılık olacak iki tane iticiye ihtiyaç vardır. Ancak, yatay eksenlerden bir tanesinin mekanik tasarım gereği kararlı olduğu var sayılırsa itici sayısı düşürülebilir.



Bütün denetleme sistemleri sayısal olduğu için, almaçların analog çıktıları analogtan sayısala dönüştürücüler kullanılarak sayısal sinyallere dönüştürülecektir. Benzer biçimde, “İtici Denetim Modülü”nün yarattığı sayısal sinyaller, motorlara, sayısalan analoga dönüştürücüden geçirilerek analog olarak gönderilecektir.

Su altı aracı, temel olarak aşağıdaki işlevleri yerine getirmelidir:

1. Faydalı yük istenildiği şekilde kontrol edilebilmelidir.
2. Aracın durumu, her an, su üzerindeki araca gönderilebilmelidir.

3. Su altındaki araç ve faydalı yük istenildiği şekilde bir durumdan diğer duruma getirilebilmelidir. Dolayısıyla, tasarlanacak olan su altı aracı donanım ve yazılım mimarisi bu temel işlevleri yerine getirebilmelidir.

Tasarlanacak olan su altı aracının boyutları ve ağırlığı mümkün olduğunca düşük olmalıdır. Bu nedenle, kullanılacak olan herhangi bir donanım elemanı, bu elemanın bugünkü teknolojisine, boyutlarına, ağırlığına, çalışma koşullarına ve proje bütçesine bağlı olarak belirlenmelidir. Proje bütçesi kısıdından dolayı ve ilerleyen teknolojinin ağırlığı ve boyutları küçülttüğü göz önünde bulunduracak olursak, mekanik tasarımın mümkün olan en küçük boyutlarda olmayacağı sonucuna varırız. Dolayısıyla, yapılan elektronik ve faydalı yük donanım tasarımı mekanik tasarımın boyut ve ağırlık kısıtlarını belirleyecektir.

Tüm sistem yazılımının ana fikri, her şeyi mümkün olabildiğince en basit şekilde tutmaktır, çünkü kullanılması mümkün olmadığı durumlar hariç en güvenilir çözüm en basit olanıdır. Vsub bünyesinde iki ana işlem gerçekleşir, bunlar: denetim yazılımı ve görüntü işleme yazılımıdır. Bu iki uygulama birbiriyle TCP/IP iletişim kuralını kullanarak haberleşmektedir. Su altı aracında iki adet bilgisayar vardır. Bunlar denetim bilgisayarı ve görüntü bilgisayarıdır.

## **2.2 Yeni Tasarım**

ULİSAR projesinde karar verilen donanım ilk tasarım daha sonra başlayan SensoRHex projesinde de kullanılan tasarımla ortak geliştirmeler yapılması açısından biraz değiştirilmiştir. Bu değişiklik denetim ve görüntü bilgisayarı ile çevre birimlerin haberleşmesini sağlayan veriyolu üzerinde olmuştur. Yeni tasarımda çevre elemanlarla haberleşmek için Universal Robot Bus(URB) kullanılmıştır. Bu bölümde yapılan değişikliğe göre güncellenmiş donanım tasarımı verilmektedir.

## 2.2.1 Genel Tasarım

Proje kapsamında tasarlanan ve elektronik donanım kutusu içine terleştirecek olan kartlar PC104 boyutlarındadır. Diğer kartlar ise mümkün olan en küçük boyutta olacak şekilde tasarlanmıştır.

Çevresel birimler ile haberleşmenin URB veri yolu kullanılarak gerçekleştirilmesi sebebiyle, çevresel birimler gruplanarak çeşitli URB düğümleri haline dönüştürülmüştür. Aşağıda gerçekleştirilmiş olan URB düğümleri ve işlevleri verilmiştir.

Düğüm No	İşlevsel Tanımı
D1	Motor Arayüz Düğümü
D2	Ultrasonik Almaç Düğümü
D3	Kamera – Işık Denetleyici Düğümü
D4	Sıcaklık Almaçları Düğümü
D5	Analog Almaç Düğümü
D6	Güç Denetimcisi Düğümü

*Tablo 3: URB düğümleri*

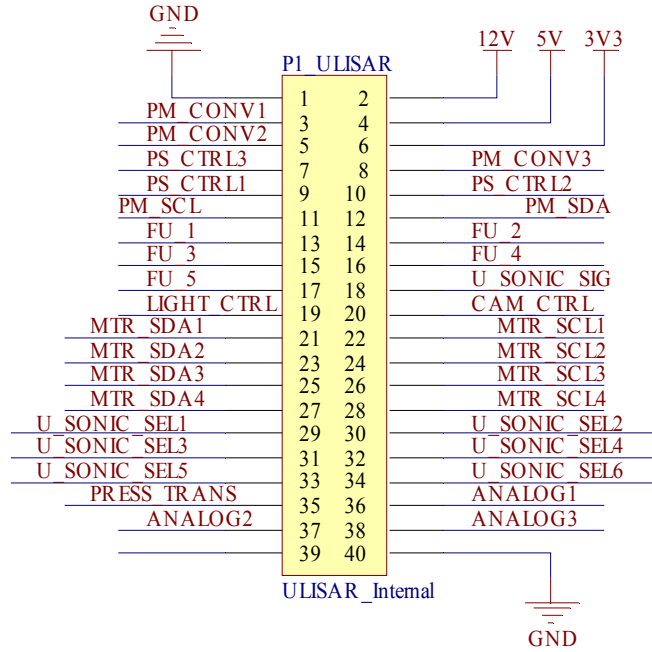
Aşağıda URB veri yoluna bağlı olmayan, ancak URB düğümlerine yardımcı kartlar verilmiştir.

Kart No	İşlevsel Tanımı
G1	Güç durumu ölçme ve güç dağıtım kartı
UF2	Ön ultrasonik almaç seçme ve kamera-ışık denetleme kartı
UR3	Arka ultrasonik almaç seçme
N4	Yönelim denetleme kartı
I5	Denetim bilgisayarına bağlı yığın arayüz kartı
I6	Faydalı yük bilgisayarına bağlı yığın arayüz kartı

*Tablo 4: URB düğümleri yardımcı kartları*

Tablo 4’de gösterilen; ancak henüz gerçekleştirilmemiş olan Yönelim Kartı, URB veri yoluna bağlı olacak olmasına rağmen deniz altının ataletsel merkezine yakın bir noktada yerleştirilme zorunluluğu nedeniyle farklı bir yerde olacaktır. URB düğümlerinin çevresel birimler ile

bağlantısının tek bir konnektör üzerinden herbir karta aktarılmasının uygun olacağına karar verilmiştir. Bu konnektörün bağlantıları aşağıda verilmiştir.



Şekil 2: ULISAR konnektörü

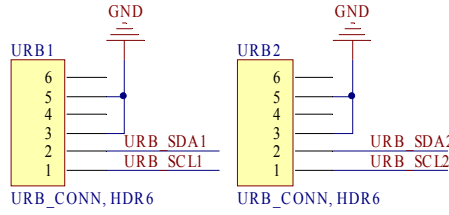
Tasarlanan tüm kartlarda, bu konnektörün ismi ve pin karşılıkları aynıdır. Bu yüzden herbir pinin işlevi aşağıdaki tabloda gösterilmiştir.

No:	Adı:	Açıklama:
•	GND	Toprak
•	12V	+12V Besleme Pini
•	PM_CONV1	Güç Durumu denetleme Çipi 1'in CONVERT pini
•	5V	+5V Besleme Pini
•	PM_CONV2	Güç Durumu denetleme Çipi 2'nin CONVERT pini
•	3V3	+3.3V Besleme Pini
•	PS_CTRL3	Güç Kesme Anahtarı 3'ün kontrol pini
•	PM_CONV3	Güç Durumu denetleme Çipi 3'ün CONVERT pini
•	PS_CTRL1	Güç Kesme Anahtarı 1'in kontrol pini
•	PS_CTRL2	Güç Kesme Anahtarı 2'nin kontrol pini
•	PM_SCL	Güç Durumu Denetleme çiplerinin I2C bağlantısı CLOCK pini
•	PM_SDA	Güç Durumu Denetleme çiplerinin I2C bağlantısı DATA pini
•	FU1	Gelecekte eklenecek URB düğümleri için rezerve edilmiş pinler
•	FU2	
•	FU3	
•	FU4	
•	FU5	
•	U_SONIC_SIG	Ultrasonik almaçların SIGNAL pini
•	LIGHT_CTRL	Işık denetleme pini (Açma/Kapama)
•	CAM_CTRL	Kamera denetleme pini (Açma/Kapama)
•	MTR_SDA1	Motor 1'in I2C bağlantısı DATA pini
•	MTR_SCL1	Motor 1'in I2C bağlantısı CLOCK pini
•	MTR_SDA2	Motor 2'nin I2C bağlantısı DATA pini
•	MTR_SCL2	Motor 2'nin I2C bağlantısı CLOCK pini
•	MTR_SDA3	Motor 3'ün I2C bağlantısı DATA pini
•	MTR_SCL3	Motor 3'ün I2C bağlantısı CLOCK pini
•	MTR_SDA4	Motor 4'ün I2C bağlantısı DATA pini
•	MTR_SCL4	Motor 4'ün I2C bağlantısı CLOCK pini
•	U_SONIC_SEL1	Ultrasonik almaç 1'in seçme pini
•	U_SONIC_SEL2	Ultrasonik almaç 2'nin seçme pini
•	U_SONIC_SEL3	Ultrasonik almaç 3'ün seçme pini
•	U_SONIC_SEL4	Ultrasonik almaç 4'ün seçme pini
•	U_SONIC_SEL5	Ultrasonik almaç 5'in seçme pini
•	U_SONIC_SEL6	Ultrasonik almaç 6'nın seçme pini

•	PRESS_TRANS	Basınç almaçı sinyal pini
•	ANALOG1	Analog almaç 1 sinyal bağlantısı (Nem almaçı 1 olarak atanmıştır)
•	ANALOG2	Analog almaç 2 sinyal bağlantısı (Nem almaçı 2 olarak atanmıştır)
•	ANALOG3	Analog almaç 3 sinyal bağlantısı
•	NC	Aktif bağlantısı yoktur
•	GND	Toprak

Tablo 5: ULISAR konektörü pin açıklamaları

Denizaltı elektronik donanımında iki adet URB köprüsünün kullanılmasına karar verilmiştir. URB köprülerine bağlı URB veri yolu konektörlerinin ikisinin de tüm kartlara aktarılmasına karar verilmiştir. Bu nedenle, tüm kartlara URB köprü I2C arayüz konektörleri yerleştirilmiştir. Bu konektörlerin her bir kart üzerinde aynı yerde bulunması sebebiyle, kart üzerindeki düğüm hangi URB köprüsüne bağlanacak ise o köprü konektörüne bağlanmaktadır. Bu konektörlerin pinleri Şekil 3 de gösterilmiştir.



Şekil 3: URB konektörleri

URB köprülerinin, düğümleri aşağıdaki gibi paylaşımlarının uygun olacağına karar verilmiştir. Aşağıdaki tabloda URB düğümlerinin hangi URB köprüsüne bağlı oldukları gösterilmiştir.

No:	Köprü 1	Köprü 2
1	Güç Denetimcisi Düğümü	Motor Arayüz Düğümü
2	Analog Almaç Düğümü	Ultrasonik Almaç Düğümü
3	Kamera – Işık Denetleyici Düğümü	Sıcaklık Almaçları Düğümü
4	Yönelim Denetimci Düğümü	

Tablo 6: URB Düğümlerinin bağlı olduğu URB köprüleri

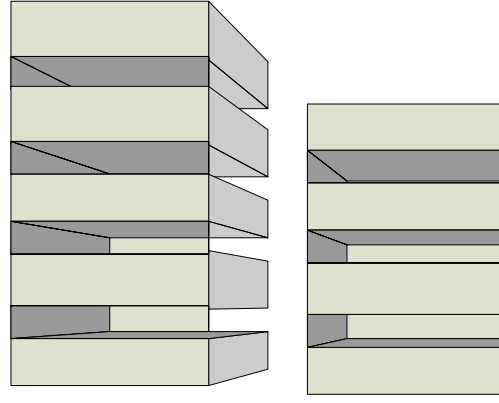
Elektronik donanım için toplam 8 kart tasarlanmıştır. Bu kartların isimleri ve hangi URB düğümlerinin üzerlerinde gerçekleştirildikleri aşağıdaki tabloda verilmiştir.

<b>Kart No</b>	<b>Adı</b>	<b>İçeriği</b>
1	K1 - URB Köprü Kartı	-2 Adet URB köprü ve 2 Adet UART-USB çevirici
2	K2 - Güç – Kamera- Işık – Motor – Analog Almaç Kartı	-Güç Denetimcisi Düğümü -Analog Almaç Düğümü -Kamera – Işık Denetleyici Düğümü -Motor Arayüz Düğümü
3	K3 - Ultrasonik Almaç Düğümü	-Ultrasonik Almaç Düğümü
4	I5 - Denetim bilgisayarına bağlı yığın arayüz kartı	-Denetim bilgisayarına bağlı tüm düğümlerin bağlantıları -Sıcaklık Almaç Düğümü
5	I6 - Faydalı yük bilgisayarına bağlı yığın arayüz kartı	-Faydalıyük bilgisayarına bağlı tüm düğümlerin bağlantıları
6	G1 - Güç durumu ölçme ve güç dağıtım kartı	- Güç durumu ölçümleme birimi - Güç kesme birimi - Güç dağıtım bağlantıları
7	UF2 - Ön ultrasonik almaç seçme ve kamera-ışık denetleme kartı	- Deniz altının ön kısmına yerleştirilecek olan ultrasonik almaç dizisinde almaç seçme birimi - Kamera ve ışık açma kapama birimi
8	UR3 - Arka ultrasonik almaç seçme	- Deniz altının arka kısmına yerleştirilecek olan ultrasonik almaç dizisinde almaç seçme birimi
9	N4 - Yönelim denetleme kartı	- Yönelim denetim sisteminin arayüz ve denetim birimi

*Tablo 7: Tasarlanan kartlar ve işlevleri*

Bu tasarım sonucunda, elektronik donanım kutusunda PC104 yığınları aşağıdaki görünümde olacaktır.





Şekil 4: PC104 yığınları

Aşağıda şu ana kadar gerçekleştirilen kartlarla ilgili tasarım ayrıntıları verilmiştir.

## 2.2.2 Baskı Devre Kart Tasarımları

### K1 - URB Bridge

#### 2.2.2.1 K1 Kartı

(URB Köprü)

PC104  
Peripherals

Ulisar'a bağlı tüm motorlar ve sensörler URB (Universal Robot Bus) veri yolu üzerinden kontrol bilgisayarına bağlanmıştır. QNX gerçek zamanlı işletim sistemini çalıştıran kontrol bilgisayarı üzerinde koşan kontrol, izleme ve sürücü yazılımları tüm bu sistem noktaları ile haberleşme gereksinimlerini yeterli düzeyde gerçekleştirmektedir.

URB protokolü, yıldız topolojisine göre tasarlanmış bir veya daha fazla köprü ünitesi içerebilmektedir. Köprü ünitesi ile kontrol bilgisayarı arasındaki iletişim USB 2.0 (12MBit hızında Full-Speed) ya da RS232 (230400kbps hızında) protokolü ile gerçekleştirilebilmektedir. Köprü devresi aynı zamanda kendisine diğer taraftan bağlı bulunan bir ya da daha fazla URB noktası ile I2C protokolü ile haberleşebilmektedir. I2C protokolü, aktif sonlandırma sistemi barındırdığı takdirde, 1Mbit hızında ve birden fazla noktanın iletişimini gerçekleştirebilmektedir. Bununla birlikte özellikle sensör ve motor URB noktalarında bulunan mikro denetleyiciler 100kbit ve 400kbit hızlarını destekledikleri için Ulisar URB alt sistemi 400kbit hızında çalışabilecek şekilde tasarlanmıştır.

Köprü ünitesi kendisine bağlı bulunan URB noktaları ile veri alışverişini "i2c master" modda gerçekleştirmektedir. Bu nedenle üniteye bağlı olan URB noktalarındaki mikro denetleyiciler

de “i2c slave” modda çalışmaktadır. Köprü ünitesi sorumlu olduğu URB noktalarına sistem mesajları ya da tanımlama mesajları ileteceği için her bir URB noktasındaki mikro denetleyici “broadcast” mesajları alabilme yeteneğinde olmalıdır.

Elektriksel olarak köprü devresi 5V gerilimi, Ulsar güç devresinden regüleli bir şekilde alıp, URB noktalarının ihtiyaç duyabileceği 3.3V gerilimini de üretebilmektedir. Bununla birlikte kontrol bilgisayarı ile haberleşmek için kullanılacak:

USB 2.0 için 5V, D+, D- ve GND

RS232 için TX, RX ve GND pinleri,

ve URB noktaları ile I2C protokolüne göre haberleşilebilmesi için I2C standart pinlerinin bulunduğu konektörler bulunmaktadır.

URB köprü tasarımı genel olarak URB geliştirme aşamasında kullanılan donanım elemanlarını içermektedir. Ancak bu projeye özgü ölçütler sebebiyle, tasarımda bazı değişiklikler yapılmıştır. Bu değişikliklerin en önemlisi köprü kartlarında UART-USB dönüştürücü yonga kullanılmıştır. Bunun en temel sebebi PC104 CPU modülünde sadece 2 adet RS232 çıkışının olmasıdır. Kullanılan UART-USB yongası, kalitesi ve kullanım sıklığı nedeniyle FTDI firmasının FT232BL ürünüdür.

URB köprü kartında, URB köprü denetimcisi SILabs firmasının C8051F342 mikro denetleyicisidir.

### Şematik Tasarım:

Ek dokümanlara bakınız. (URB\_BRIDGE\_sch.pdf)

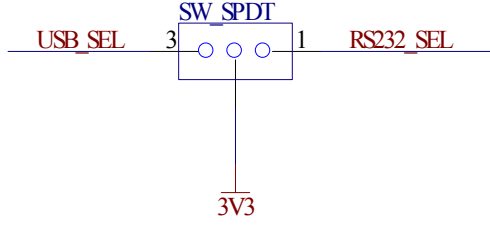
### Malzeme Listesi:

No:	Designator	Footprint	Comment	Description	Quantity	Value
1	C1_1, C1_2, C6_1, C6_2, C7_1, C7_2, C10, C16_1, C16_2, C18_1, C18_2, C22_1, C22_2	C0805	Cap	Capacitor	13	100nF
2	C4_1, C4_2	C0805+	Cap2	Capacitor	2	1uF
3	C5_1, C5_2, C14	C1206+	Cap2, Cap Pol3	Capacitor, Polarized Capacitor (Surface Mount)	3	1uF

4	C9, C11, C12, C13	C0805+	Cap2	Capacitor	4	0.1uF
5	C15, C17_1, C17_2	C0805	Cap	Capacitor	3	0.1uF
6	C19_1, C19_2	C0805	Cap	Capacitor	2	33nF
7	C20_1, C20_2, C21_1, C21_2	C0805	Cap	Capacitor	4	27pF
8	D2, D5_1, D5_2, D6_1, D6_2, D7_1, D7_2, DBGL1_1, DBGL1_2, DBGL2_1, DBGL2_2	3.2X1.6X1.1	LED2	Typical RED, GREEN, YELLOW, AMBER GaAs LED	11	
9	DBG1, DBG2	HDR2X5	Header 5X2	Header, 5-Pin, Dual row	2	
10	P1_ULISAR	HDR2X20	ULISAR_Internal	Header, 20-Pin, Dual row	1	
11	P2, P3	HDR1X3H	Header 3H	Header, 3-Pin, Right Angle	2	
12	P4-USB	HDR2X8_254	Header 8X2	Header, 8-Pin, Dual row	1	
13	Q1_1, Q1_2, Q2_1, Q2_2	SOT143B_L	BCV62	PNP General-purpose Double Transistor	4	
14	R1_1, R1_2, R2_1, R2_2, R5_1, R5_2, R6_1, R6_2, R11_1, R11_2, R12_1, R12_2	R0805	Res2	Resistor	12	1K
15	R3_1, R3_2, R4_1, R4_2, R16_1, R16_2	R0805	NOP, Res2	Resistor	6	10k
16	R8, R9_1, R9_2, R10_1, R10_2	R0805	Res2	Resistor	5	330
17	R13_1, R13_2, R17_1, R17_2, R18_1, R18_2, R19_1, R19_2	R0805	Res2	Resistor	8	0K470
18	R14_1, R14_2	R0805	Res2	Resistor	2	1K5
19	S1_1, S1_2	SW_SPDT	SW_SPDT	SPDT switch	2	
20	S2_1, S2_2	RAD-0.1	SW-PB	Switch	2	
21	S3_1, S3_2	PIN2	SW-PB	Switch	2	
22	URB1, URB2	HDR1X6	URB_CONN, HDR6	Header, 6-Pin	2	
23	Y1_1, Y1_2	CA301	XTAL	Crystal Oscillator	2	
24	U2_1, U2_2	SO14_N	M74HCT126M1R	Quad Bus Buffer (3-State)	2	
25	U3_1, U3_2	LQFP-32_L	C8051F342	C8051F34x family of mixed-signal microcontrollers	2	
26	U4	NSO16_M	MAX3232CSE	3.0V TO 5.5V, Low-Power, up to 1Mbps, True RS-232 Transceiver	1	
27	U5_1, U5_2	SOT23-6	STF201	Semtech, STF201-22 & STF201-30 USB Downstream Port Filter & TVS For EMI Filtering and ESD Protection	2	
28	U6_1, U6_2	LQFP-32_L	FT232BL	USB UART Asynchronous Serial Data Transfer Chip, LQFP-32, Pb- Free	2	

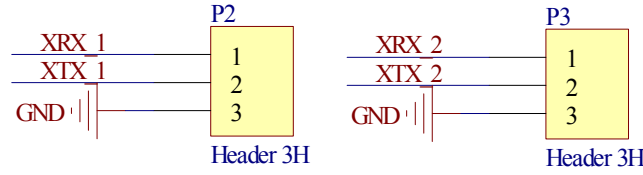
### Önemli Pin Tanımları:

URB köprüleri ile iletişim hem USB hem de RS232 veri yolları ile yapılabilmektedir. Ancak iki veri yolu biçimi aynı anda çalışmamaktadır. Hangi veri yolunun kullanılacağı ise S1\_1 ve S1\_2 jumperları ile yapılmaktadır. RS-232 veri yolunu kullanmak için 2 ile 3 numaralı pinler; USB iletişimi kullanmak için 2 ile 1 numaralı pinler kısa devre edilmelidir.



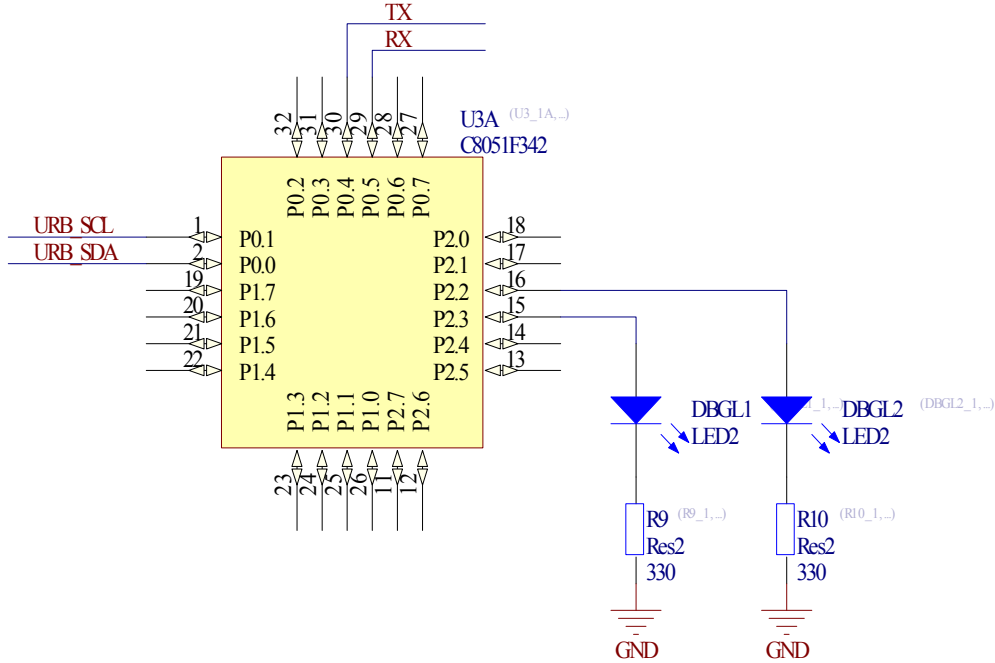
Şekil 5: Köprü iletişim veri yolu seçimi

URB Köprülerinin RS-232 çıkışları seçili olduğu durumda, aşağıda gösterilen biçimde 3 aktif yollu bir kablo ile doğrudan RS-232 bağlantısı yapılabilir.



Şekil 6: URB köprüleri RS-232 bağlantısı

Hata ayıklama işlerinde kullanılmak üzere, kartın üzerine 2 adet LED bağlantısı yapılmıştır. Aşağıda, URB köprü denetimcisinin genel kullanımlı PIN lerinin bağlantıları gösterilmiştir.

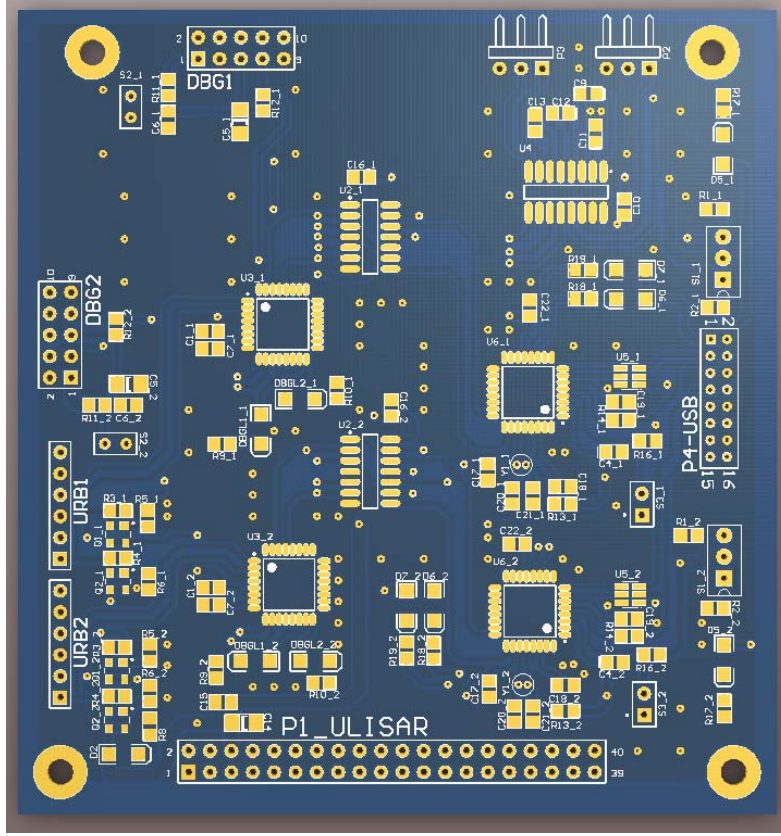


Şekil 7: C8051F342 genel PIN bağlantıları

Hata ayıklama için C8051F34x ailesi debug arayüzü gerçekleştirilmiştir.

### PCB Tasarımı:

PCB kartları elimize ulaştığında görüntüsü aşağıdaki gibi olacaktır.



Şekil 8: URB Köprü kart görünümü

PCB yerleşim planı için eklere bakınız (URB\_BRIDGE\_Assembly Drawings.pdf).

### 2.2.2.2 K2 Kartı

#### (Güç – Kamera- Işık – Motor – Analog Almaç Kartı)

Bu kart üzerinde 4 farklı URB düğümü gerçekleştirilmiştir. Bu düğümlerin işlevleri ve gerçekleştirme mikro-denetleyicileri aşağıdaki tabloda gösterilmiştir.

Düğüm	Kullanılan Mikro-denetleyici
D1 -- Motor Arayüz Düğümü	PIC1 -- PIC16F88
D3 -- Kamera -Işık Denetleyici Düğümü	PIC2 -- PIC16F88
D5 -- Analog Almaç Düğümü	AD1 -- ADUC845BSZ62-5
D6 -- Güç Denetimcisi Düğümü	PIC2 -- PIC16F88

Basit işlevli düğümlerde genel olarak PIC16F88 kullanılmıştır. Bu yonganın seçilmesinin altında yatan temel sebep az pinli paketinin, proje gereksinimlerini sağlamakta olması ve çevresel herhangi bir donanım elemanına ihtiyaç duymamasıdır.

Motor arayüz düğümünün temel işlevi, motor denetim haberleşme veri yolu olan I2C'nin URB biçimine çevirmektir. Bu nedenle bu düğümün tasarımında sadece tek bir yonga kullanılmıştır.

Kamera-Işık denetleyici düğümünün temel işlevi istenildiğinde kamera ve/veya ışıkların etkin hale getirilmesidir. Bu nedenle, bu işlev için sadece 2 adet dijital sinyale ihtiyaç duyulmaktadır. Tasarımda, her iki kamera aynı anda açılıp kapatılabilmektedir. Benzer şekilde, iki ışık kaynağı da aynı anda açılıp kapatılabilmektedir. Bu düğümün basit işlevi nedeniyle ayrı bir yonga kullanılmayarak, güç denetim yongasına bu işlev atfedilmiştir.

Güç denetimcisi düğümünün temel işlevi pil paketlerinin anlık gerilimlerini ve pil paketinden çekilen akım değerlerini okumaktır. Bu işlevi, G1 kartı üzerinde bulunan; Analog Devices firmasının ADM1191 yongası gerçekleştirmektedir. Bu yongadan, her bir pil paketi için bir adet kullanılmıştır. Bu yongaların denetlenmesi için ise K1 Kartı üzerinde PIC16F88 yongası kullanılmıştır.

Analog almaç denetimcisi olarak Analog Devices firmasının 8052 tabanlı analog mikro-denetleyicisi kullanılmıştır. URB veri yolu geliştirilmesi aşamasında da 8051 çekirdek tabanlı mikro-denetleyiciler kullanılması sebebiyle yazılım geliştirme süreci daha güvenli, hızlı ve ekonomik olacaktır. AduC845 mikro-denetleyicisi 24 bit ADC ve analog ön gereksinim devre elemanlarını kendi içerisinde içermektedir. Ayrıca analog girişlerin hem diferansiyel hem de tek giriş olarak kullanma olanağı sağlaması sebebiyle çok geniş yelpazedeki analog almaçların okunması için iyi bir seçim olmaktadır. ADC'nin referans gerilimi olarak Analog Devices'in ADR425 tip numaralı 2.5V DC referans gerilim üretici kullanılmıştır.

Şu anda kullanılan analog almaçlar, nem ve basınç almaçları olması sebebiyle, Analog almaç düğümünün Yönelim düğümü ile aracısız haberleşebilmesi için SPI veri yolu ayrı bir konektör ile Arayüz kartına taşınmıştır.

#### **Şematik Tasarım:**

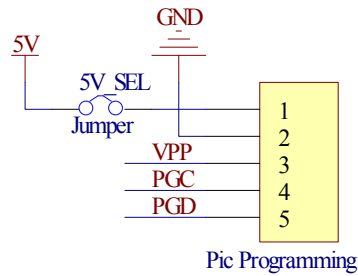
Ek dokümanlara bakınız. (PowerMonitor-CamLightCtrl\_sch.pdf)

## Malzeme Listesi:

No:	Designator	Footprint	Comment	Description	Quantity	Value
1	1, 5V_SEL, W1, W2	RAD-0.1	Jumper	Jumper Wire	4	
2	ANAL_FU	HDR2X10	Header 10X2	Header, 10-Pin, Dual row	1	
3	ANAL_PROG, Prog_CL_PM, Prog_MTR	HDR1X5	Header 5, Pic Programming	Header, 5-Pin	3	
4	C1, C10	C0805	Cap	Capacitor	2	10nF
5	C2, C11	C0805+	Cap2	Capacitor	2	1uF
6	C3, C4, C5, C6, C7, C9	C0805	Cap	Capacitor	6	100nF
7	C8	C1206+	Cap2	Capacitor	1	10uF
8	D1, D2, D3, D5, D6, D7, D9, D10	3.5X2.8X1.9	LED3	Typical BLUE SiC LED	8	
9	P1_ULISAR	HDR2X20	ULISAR_Internal	Header, 20-Pin, Dual row	1	
10	R1, R2, R10, R11, R12, R13, R14, R15, R16, R17	R0805	Res2	Resistor	10	10K
11	R3, R4, R5, R7, R8, R9, R18, R19, R20, R22, R23	R0805	Res2	Resistor	11	470R
12	U1, U2	PDIP300-18	PIC16F88-I/P	Enhanced FLASH Microcontroller	2	
13	U3	R-8	ADR421ARZ	Ultraprecision, Low Noise, 2.500 V XFET® Voltage Reference	1	
14	U4	S-52	ADUC845BS62-5	Precision Analog Microcontroller: 12MIPS 8052 Flash MCU + 10-Ch 24-Bit ADC	1	
15	URB1, URB2	HDR1X6	URB_CONN, HDR6	Header, 6-Pin	2	
16	Y1	ECS-3x8	XTAL	Crystal Oscillator	1	

## Önemli Pin Tanımları:

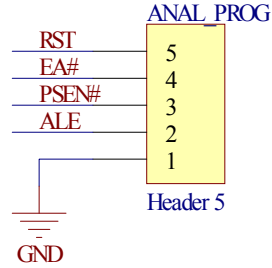
Bu kart üzerinde kullanılan tüm mikro-denetleyicilerin programlara arayüzleri kart üzerinde bulunmaktadır. PIC16F88 mikro-denetleyiciler için programla arayüzü aşağıda gösterilmiştir.



Şekil 9: PIC16F88 programlama bağlantısı

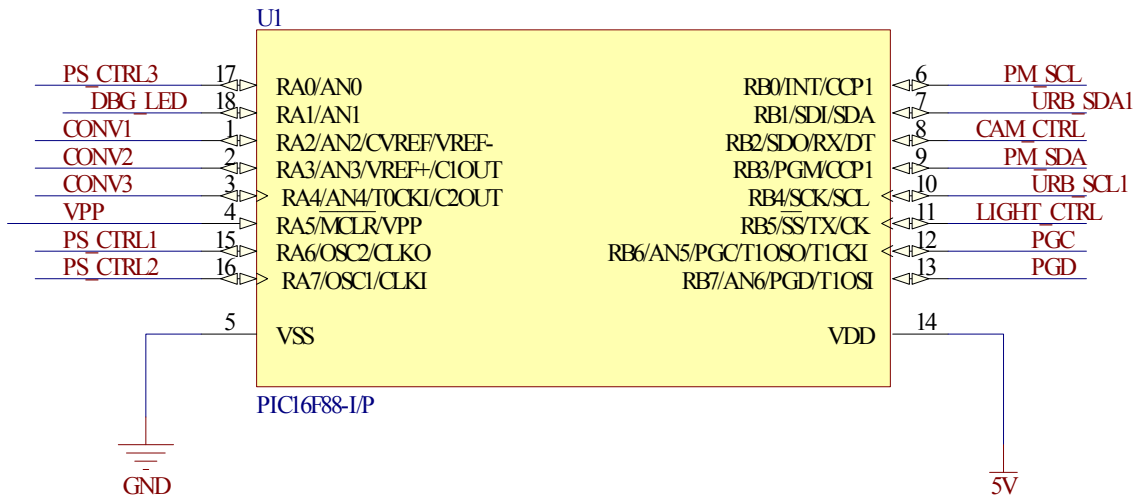
PIC16F88 yongaları programlanırken besleme gerilimi istenirse programlama arayüzünden istenirse ULIDAR konektörü üzerinden yapılabilmektedir. Yongaları programlama arayüzünden beslemek için “5V\_SEL” jumperı kısa devre edilmesi gerekmektedir.

ADuC845 mikro-denetleyicisinin programlama arayüzü ADuC845 geliştirme kitine uygun olarak tasarlanmıştır. Aşağıda pin isimleri gösterilmiştir. Bu yongayı programlayabilmek için W2 tuşuna basılması gerekmektedir. W1 tuşu ise RESET pinine bağlıdır.



Şekil 10: ADuC845 programlama bağlantısı

Güç ve Kamera-Işık denetimcisi önemli pin bağlantıları aşağıda verilmiştir.



Şekil 11: Güç ve Kamera-Işık Denetimcisi pin bağlantıları

Şekil 11’da gösterilen pin isimlerinin anlamları aşağıda verilmiştir.

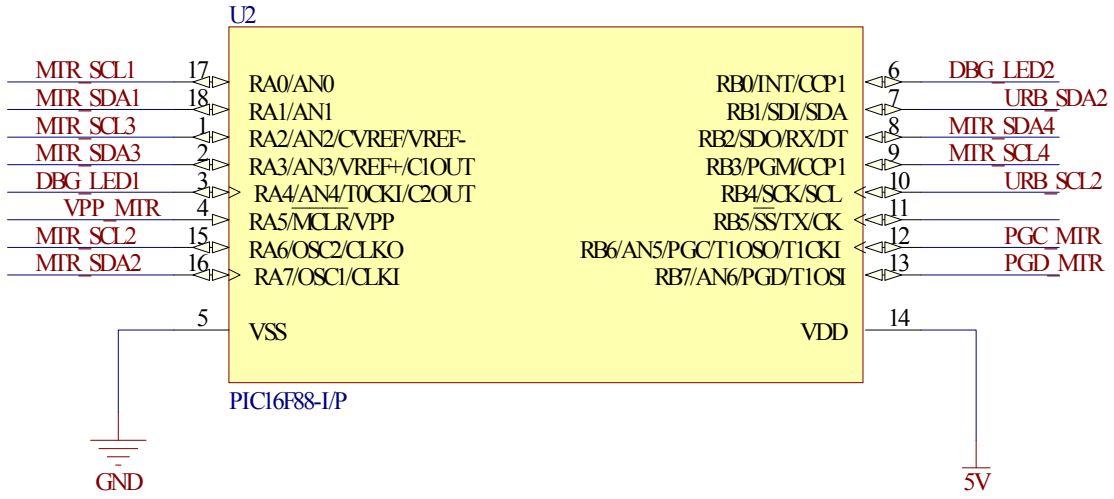
Pin Adı:	İşlevi:
PS_CTRL1	Güç denetim anahtarı 1 denetim pini
PS_CTRL2	Güç denetim anahtarı 2 denetim pini
PS_CTRL3	Güç denetim anahtarı 3 denetim pini
PM_SCL	Güç ölçüm I2C veri yolu CLOCK pini
PM_SDA	Güç ölçüm I2C veri yolu DATA pini
CONV1	Güç ölçümlene yongası 1 CONVERT (ölç) pini



CONV2	Güç ölçümlene yongası 2 CONVERT (ölç) pini
CONV3	Güç ölçümlene yongası 3 CONVERT (ölç) pini
LIGHT_CTRL	Işık denetim anahtarı denetim pini
CAM_CTRL	Kamera denetim anahtarı denetim pini
DBG_LED	Hata ayıklama LEDi

Tablo 8: Güç ve Kamera-Işık denetimcisi pin tanımları

Aşağıda Motor denetimcisi pin bağlantıları verilmiştir.



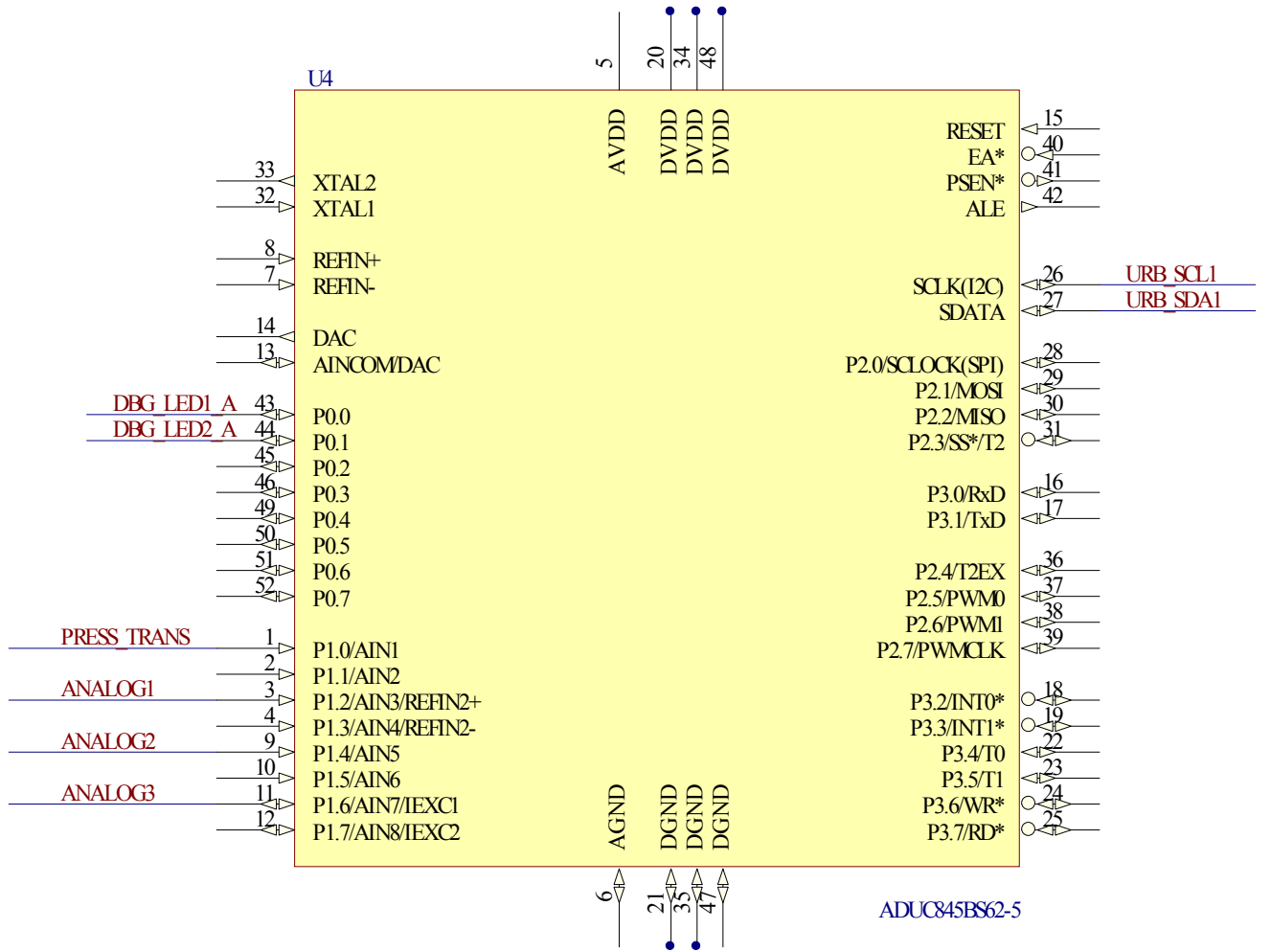
Şekil 12: Motor denetimcisi pin bağlantıları

Şekil 12’de verilen pin isimlerinin anlamları aşağıda verilmiştir.

Pin Adı:	İşlevi:
MTR_SCL1	
MTR_SDA1	Motor 1 I2C bağlantı pinleri
MTR_SCL2	
MTR_SDA2	Motor 2 I2C bağlantı pinleri
MTR_SCL3	
MTR_SDA3	Motor 3 I2C bağlantı pinleri
MTR_SCL4	
MTR_SDA4	Motor 4 I2C bağlantı pinleri
DBG_LED1	Hata ayıklama LEDi 1
DBG_LED2	Hata ayıklama LEDi 2

Tablo 9: Motor denetimcisi pin tanımları

Analog almaç denetimcisinin önemli pin bağlantıları aşağıda gösterilmiştir.



Şekil 13: Analog Almaç denetimcisi pin bağlantıları

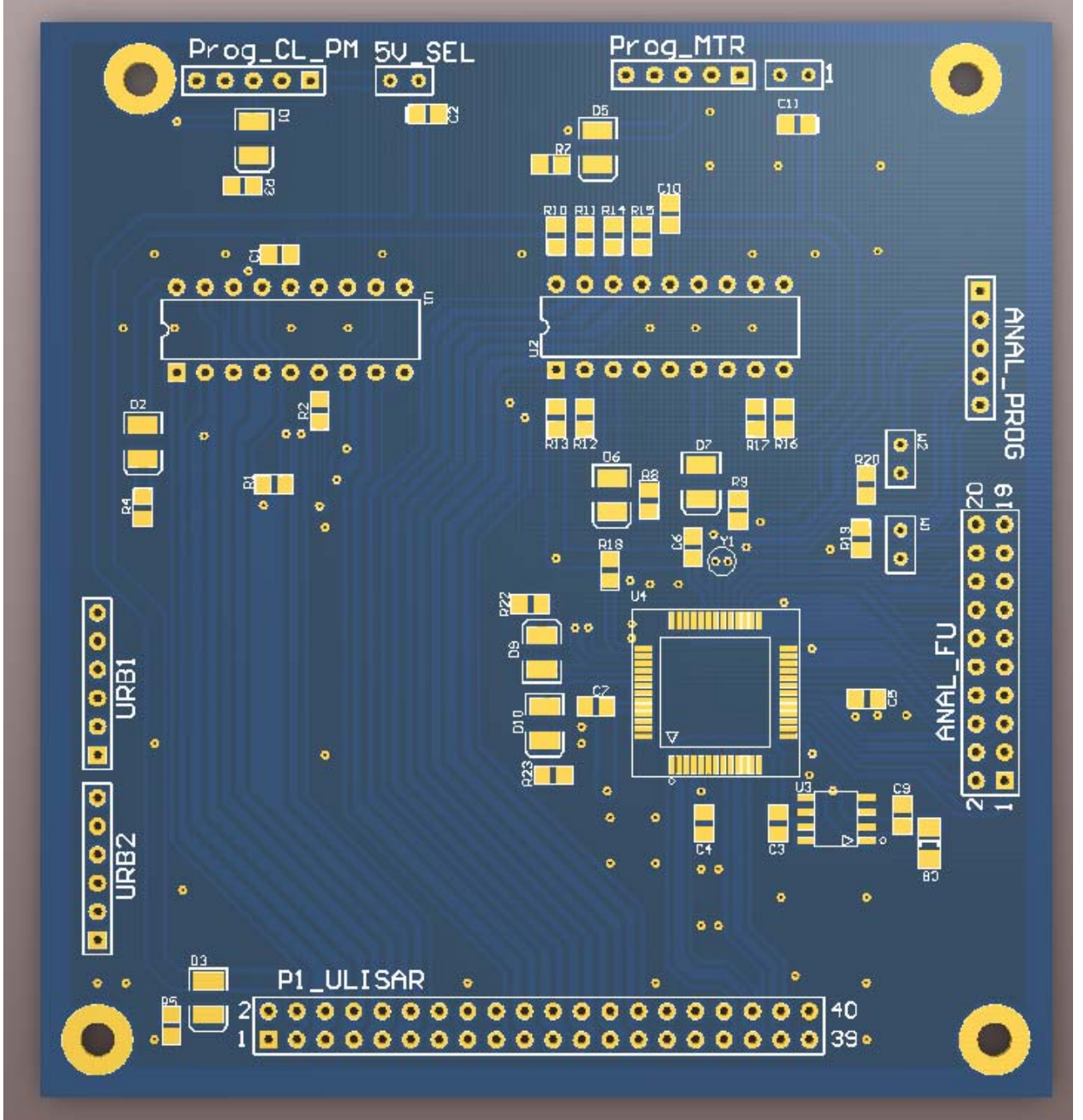
Şekil 13'de gösterilen pin isimlerinin anlamları aşağıda verilmiştir.

Pin Adı:	İşlevi:
DBG_LED1_A	Hata ayıklama LEDi 1
DBG_LED2_A	Hata ayıklama LEDi 2
ANALOG1	ANALOG bağlantı 1 (Nem Almaç 1)
ANALOG2	ANALOG bağlantı 2 (Nem Almaç 2)
ANALOG3	ANALOG bağlantı 3
PRESS_TRANS	Basınç almaç sinyali bağlantısı

Tablo 10: Analog Almaç denetimcisi pin işlevleri

## PCB Tasarımı:

PCB kartları elimize ulaştığında görüntüsü aşağıdaki gibi olacaktır.



Şekil 14: Güç – Kamera- Işık – Motor – Analog Almaç Kartı Görünümü

PCB yerleşim planı için eklere bakınız (PM\_PS\_CL\_ANAL\_Assembly Drawings.pdf).

### **2.2.2.3 K3 Kartı (Ultrasonik Almaç Dügümü)**

Engel sakınma işlemleri için ses üstü sinyallerin su altı ortamında yayılma ve yansıma özellikleri kullanılmaktadır. Öncelikle su altı ultrasonik ses frekansı olan 200 kHz ile ses üstü sinyaller üretilerek, engel olup olmadığı test edilecek yöne gönderilmekte, daha sonra ise geri yansıyan ses üstü sinyali (eko) özelliklerine bakılarak engel tanıma işlemi yapılmaktadır.

Su altı ve su üstü aracında da birden fazla olmak üzere yerleri belirlenmiş kritik noktalara yerleştirilecek modüller yardımı ile engel sakınma işlemi gerçekleştirilebilecektir. Modül bünyesinde, ses üstü sinyalin üretilmesi, gönderilmesi ve geri yansıyacak sinyalin algılanması için gönderici ve alıcı birimlerini barındırmaktadır. Sesin su altı ortamındaki yayılma hızı bilindiğinden sinyalin gönderilme ve alma arasında geçen faz farkı (süre) belirlenerek olabilecek engelin varlığı ve uzaklığı tespit edilebilmektedir.

Kullanılacak modül analog sinyaller ürettiğinden dolayı, hem engel sakınma algoritmaları için kullanılmak üzere, hem de hesaplanan yansıma süresi –dolayısıyla ölçülecek mesafe- için bu sinyallerin sayısallaştırılması ve mikro-denetleyicilere aktarılması gerekmektedir. Bu sayısallaştırma ve mikro-denetleyici için kullanılacak sinyal dönüştürme devresi aşağıda verilmiştir. Bununla birlikte, sistemde bulunacak 6 farklı almaç için sadece bir adet kontrolcü kullanılması hem işlev hem de maliyet açısından uygun görülmüştür.

Engel sakınma altsistemi bünyesinde kullanılacak bu düğüm donanımı zaman paylaşımli olarak çalışmaktadır. Böylelikle her an sadece bir almaç çalışabilmekte, böylelikle ultrasonik yankılar (sistemin kendi ürettiği) en aza indirgenecektir. Aksi halde örneğin t anında a1 almacından yayılan ses sinyalleri t2 anında a2 almacından algılanabilecek bu da hatalı ölçüme neden olabilecektir. Bu gibi problemlerin üstesinden gelmek için röle tabanlı anahtarlayıcı modül kullanılmıştır. Önceki tasarımdaki dijital anahtarlı (4066 türevi) devre ultrasonik almacın davranışından dolayı problemler içermektedir. Örneğin almacın eşdeğer devresi paralel r-l-c olarak modellendiğinde ve ultrasonik anahtarlama modülü de 200 kHz ile çalıştırıldığında almaç tarafı boost çevirici gibi çalışacaktır. Yapılan ölçümlerde de almaç giriş gerilimleri 170V civarından başlamak üzere burst sinyaller gözlenmiştir.

Bu sürümde ise elektriksel filtre ve algılama-ölçme devreleri teke indirgenmiş, almaç seçici için ise baskılı devre tipi röleler uygun görülmüştür.

Bu devrede PIC18F2550 yongasının kullanılmasına karar verilmiştir.

**Şematik Tasarım:**

Ek dokümanlara bakınız. (USONIC\_sch.pdf)

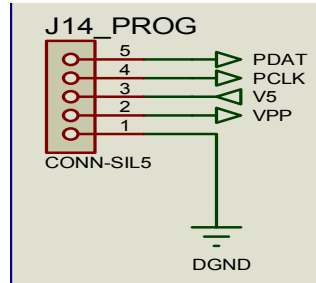
**Malzeme Listesi:**

No:	Designator	Footprint	Quantity	Value
1	R1, R4	805	2	100R
2	R2, R3, R14, R21, R42	805	5	10k
3	R5, R26, R30, R33, R52	805	5	1k
4	R6, R53, R55	805	3	470
5	R7	805	1	1.5M
6	R8, R10	805	2	100
7	R9, R11	805	2	2.4k
8	R12	805	1	750k
9	R13, R15, R16, R36, R43	805	5	10k
10	R22, R23	805	2	5.1k
11	R24	805	1	56k
12	R25	805	1	511
13	R27	805	1	2200
14	R28	805	1	1200
15	R29, R31, R32	805	3	1000
16	R34	805	1	1210
17	R35	805	1	100k
18	R37	805	1	15k
19	R38	805	1	22k
20	C1	ELEC- RAD10	1	10u
21	C2	ELEC- RAD10	1	47u
22	C3, C4, C7, C11, C12, C15, C24, C26, C33	1206	9	100nF
23	C6	805	1	100nF
24	C9, C13, C29	1206	3	1000pF
25	C10, C14, C30	1206	3	1000pF
26	C25	ELEC- RAD10	1	4.7u
27	C27, C28	1206	2	680pF
28	C31, C32	CAP20	2	15p
29	U1	SO16	1	ULN2003A
30	U4	DIL14	1	LF347
31	U5	DIL28NAR	1	PIC18F2550
32	Q1-Q3, Q7	TO92	4	BC548C
33	D2-D4, D8-D10	1206D	6	1N4148

34	D14_V12, D15_V5, D17_TEST	1206	3	DIODE-LEDA
35	GND	PIN	1	GND
36	J1	TRANS 40 DIL	1	TRANS 40 DIL
37	J1_URB1, J17_URB2	CONN-SIL6	2	CONN-SIL6
38	J3_ECHO	PIN	1	ECHOPIN
39	J4_OUT	PIN	1	OUTPIN
40	J5_OPAMP	PIN	1	OPAMP_IN_PIN
41	J8_VREF	PIN	1	VREF_PIN
42	J14_PROG	CONN-SIL5	1	CONN-SIL5
43	JP1, JP2	CONN-SIL3	2	JUMPER2
44	L1, L3	DO41	2	1mH
45	RV1	PRE-VMT	1	POTENTIOMETER
46	X3	XTAL18	1	20MHz

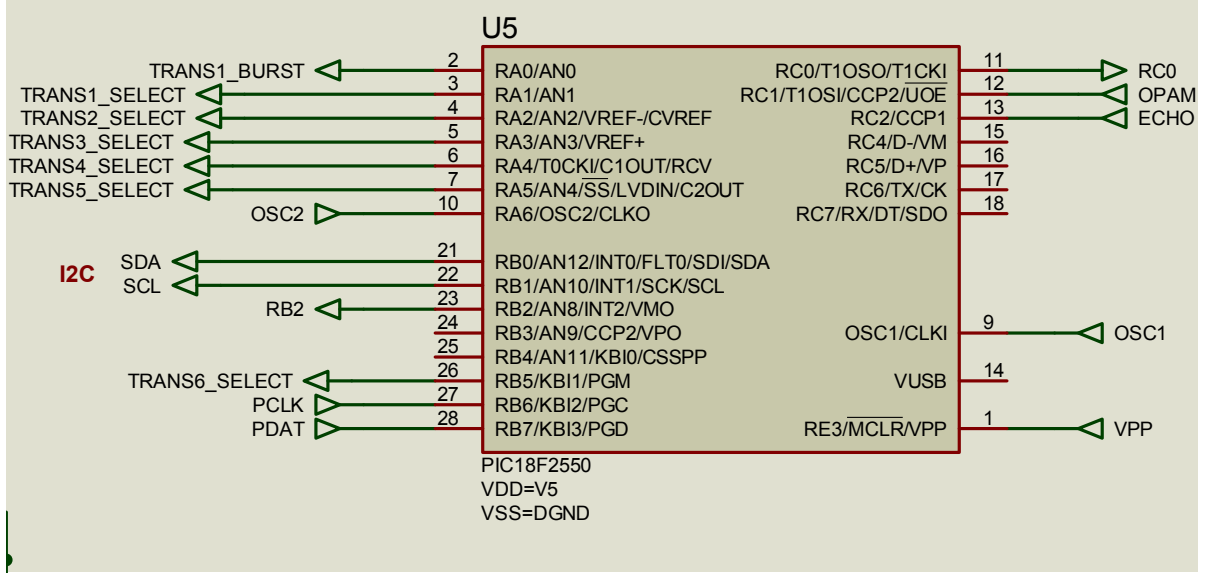
### Önemli Pin Tanımları:

PIC18F2550 yongasının programlama arayüz bağlantısı aşağıda gösterilmiştir.



Şekil 15: PIC18F2550 programlama arayüz bağlantısı

Bu kart üzerindeki mikro-denetleyicinin önemli pin bağlantıları aşağıda gösterilmiştir.



Şekil 16: Ultrasonik Almaç Denetim devresi pin bağlantıları

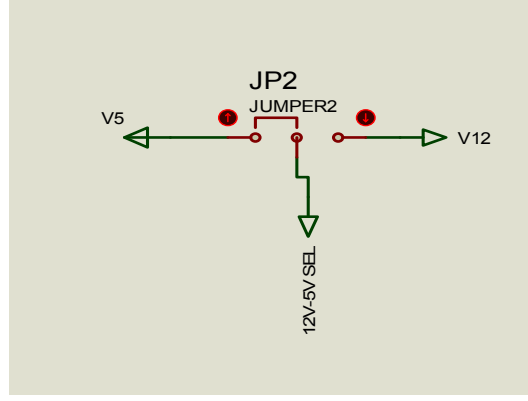
Aşağıda Şekil 16’da gösterilen pinlerin işlevleri verilmiştir.

Pin Adı:	İşlevi:
TRANS1_BURST	Ultrasonik almaç sinyal
TRANS1_SELECT	Ultrasonik almaç 1 seçme pini
TRANS2_SELECT	Ultrasonik almaç 2 seçme pini
TRANS3_SELECT	Ultrasonik almaç 3 seçme pini
TRANS4_SELECT	Ultrasonik almaç 4 seçme pini
TRANS5_SELECT	Ultrasonik almaç 5 seçme pini
TRANS6_SELECT	Ultrasonik almaç 6 seçme pini
SDA	URB I2C SDA pini
SCL	URB I2C SCL pini

Tablo 11: Ultrasonik almaç denetim devresi pin tanımları

Kullanılan Rölelerin hem 12V hem de 5V tiplerinden olabilmesi için devreye JP2 jumperi konulmuştur.

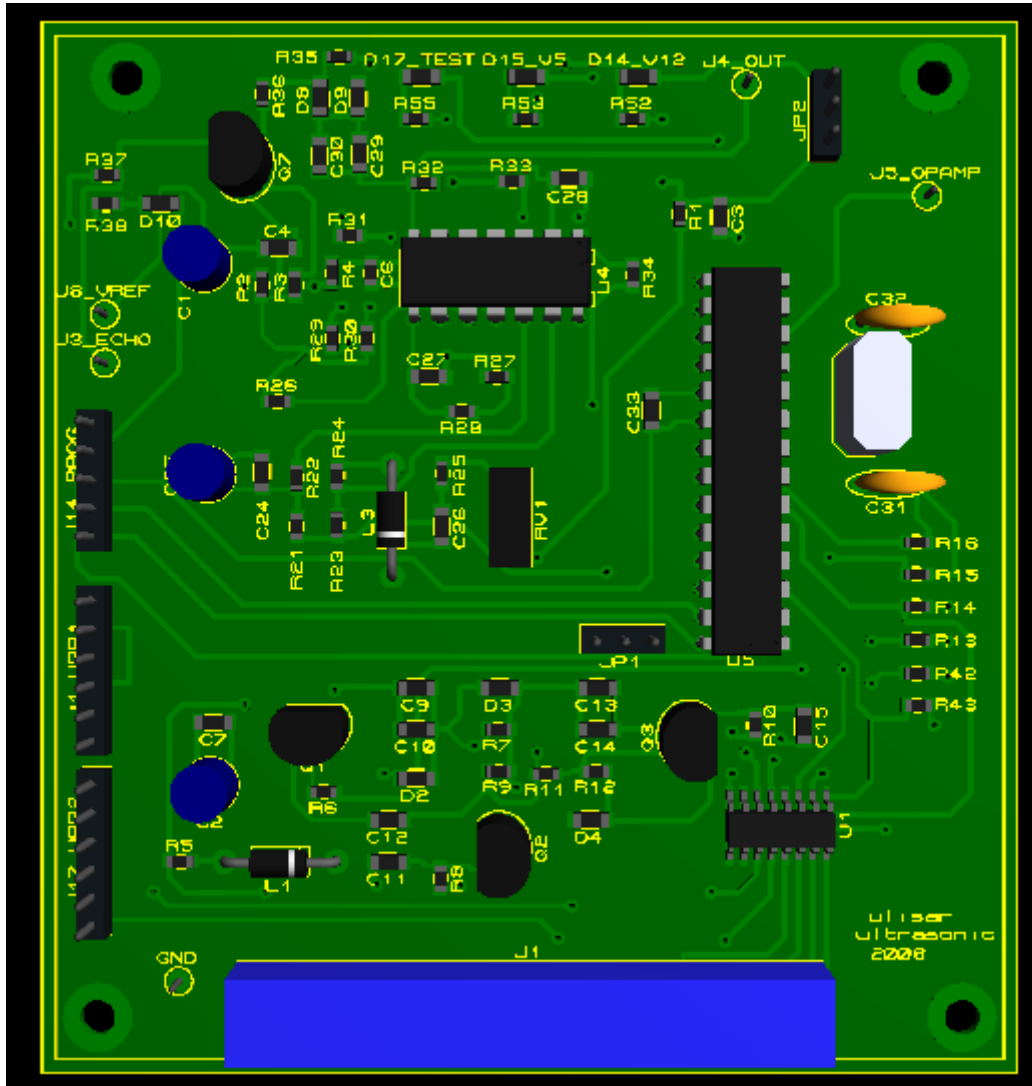




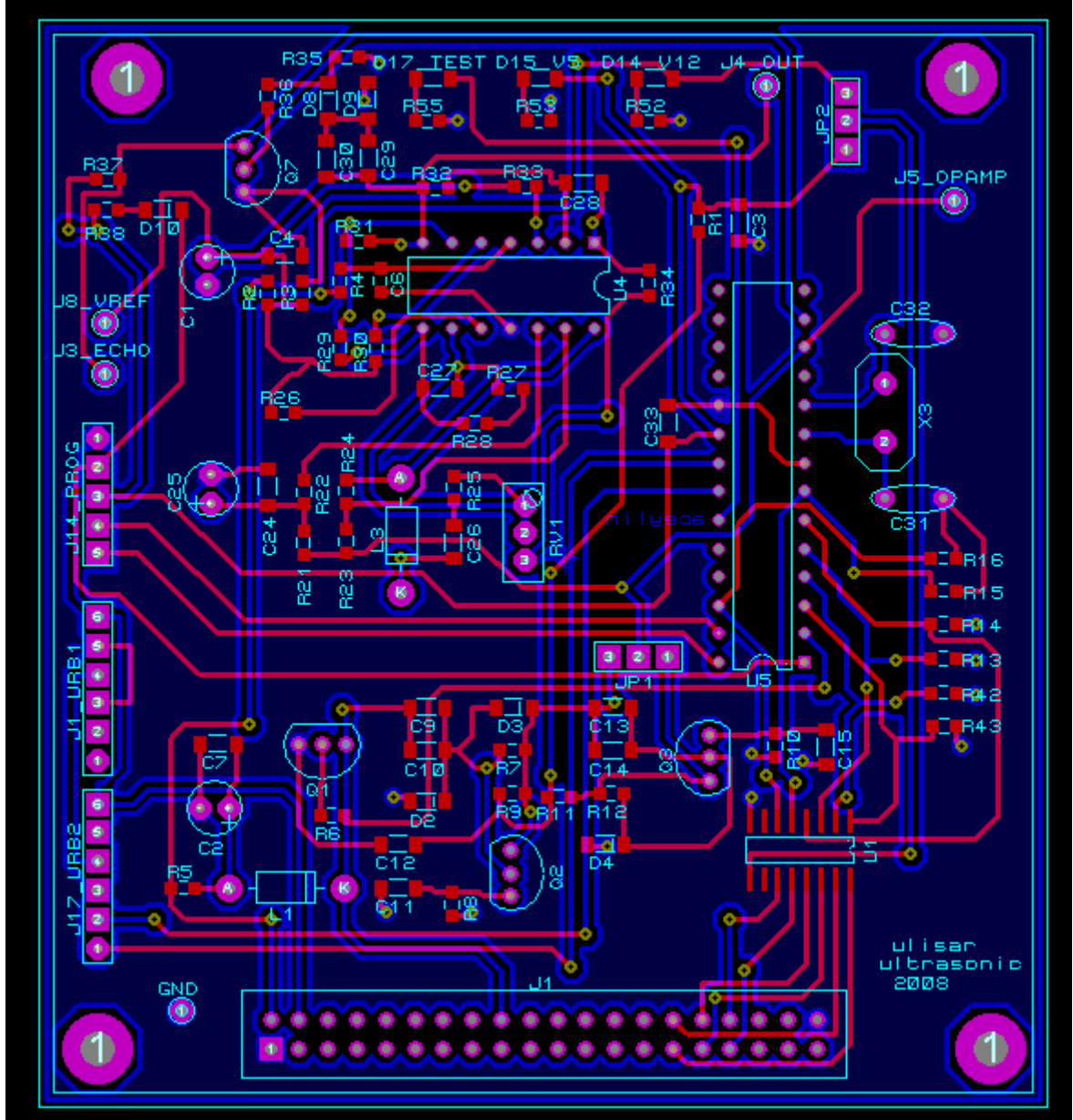
Şekil 17: 12V - 5V röle seçme jumperi

### PCB Tasarımı:

Kartlar elimize ulaştığında ultrasonik almaç denetim devresinin aşağıdaki gibi görünecektir.



PCB yerleşim planı aşağıda gösterilmiştir.



#### 2.2.2.4 I5 Kartı

(Denetim bilgisayarına bağlı yığın arayüz kartı)

Bu kartın temel işlevi ULISAR konektörü ile düğümlere taşınan veya düğümlerden alınan verilerin uygun kartlara, almaçlara veya motorlara taşınması için uygun konektör bağlantılarının yapılmasıdır. Ayrıca, bu kart üzerinde sıcaklık almaç düğümü gerçekleştirilmiştir.

Sıcaklık almacı olarak seçilen TC74A0-5.0VAT yongası I2C haberleşme veri yolunu kullandığı için, motor düğümüne benzer şekilde PIC16F88 mikro-denetleyicisi üzerinde sıcak almacı düğümünün gerçekleştirilmesi uygun bulunmuştur.

Sıcaklık almacı olarak TC74A0-5.0VAT yongasının seçilmesinin altında yatan en önemli sebep bu yonganın herhangi bir donanım elemanına ihtiyaç duymadan çalışması ve sayısal çıktı üretmesidir.

### Şematik Tasarım:

Ek dokümanlara bakınız. (INTERFACE1\_sch.pdf)

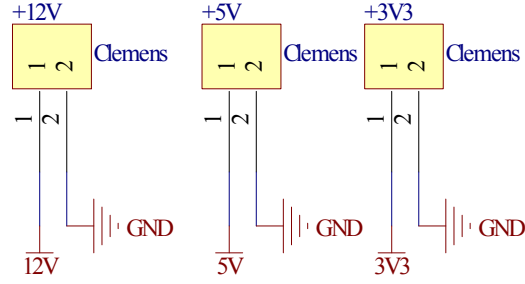
### Malzeme Listesi:

No:	Designator	Footprint	Comment	Description	Quantity	Value
1	+3V3, +5V, +12V, P_OUT_5, P_OUT_12, P_OUT_33	PowerConn2	Clemens	2 Connector Plugin Screwed 3.81mm pitch PCBN connector	6	
2	5V_SEL, W1, W2	RAD-0.1	Jumper	Jumper Wire	3	
3	ANAL_FU, NAV, PM_PS	HDR2X10	Header 10X2, Power Monitor-Switch Conn, Header 6X2H	Header, 10-Pin, Dual row	3	
4	C1	C0805	Cap	Capacitor	1	10nF
5	C2	C0805+	Cap2	Capacitor	1	1uF
6	C3, C5, C7	CAPR5-4X5	Cap2	Capacitor	3	100uF
7	C4, C6, C8	RAD-0.3	Cap	Capacitor	3	1uF
8	CPU_SERIAL	HDR2X9_200	Header 9X2	Header, 9-Pin, Dual row	1	
9	D1, D2, D3, D4, D5	3.5X2.8X1.9	LED3	Typical BLUE SiC LED	5	
10	Hum1, Hum2, Press	HDR1X3	Humidity Trans Header 3, Pressure Trans,Header 3	Header, 3-Pin	3	
11	MODEM, Prog_TMP, TMP1, TMP2, TMP3, TMP4	HDR1X5	Header 5, MODEM, Header 5, PicProgramming	Header, 5-Pin	6	
12	P1_ULISAR	HDR2X20	ULISAR_Internal	Header, 20-Pin, Dual row	1	
13	R1, R4, R5, R6, R7, R8, R11, R12, R13	R0805	Res2	Resistor	9	10k
14	R2	R0805	Res2	Resistor	1	1k
15	R3, R9, R10	R0805	Res2	Resistor	3	470R
16	U1	PDIP300-18	PIC16F88-I/P	Enhanced FLASH Microcontroller	1	
17	URB1, URB2	HDR1X6	URB_CONN, HDR6	Header, 6-Pin	2	

### Önemli Pin Tanımları:

PIC16F88 yongalarının hepsi için programlama arayüzü aynıdır. Dolayısıyla daha önce gösterilen bağlantılar bu kart için de aynıdır.

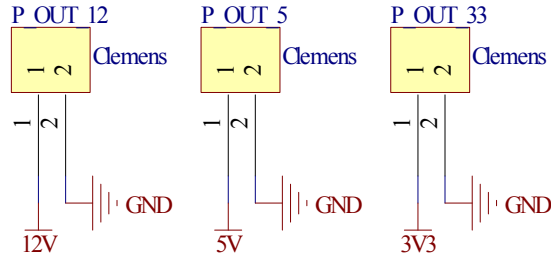
Bu kartın besleme bağlantıları aşağıda gösterilmiştir.



Şekil 18: Besleme bağlantıları

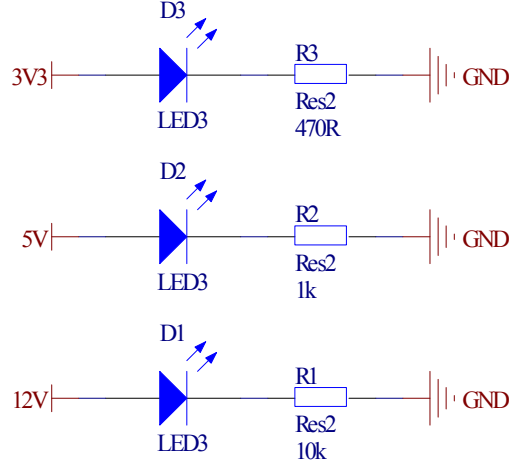
Bu konektörler, PC104 yığnında yer alan DC-DC çevirici kartının çıkış konektörlerine paralel olacak şekilde yerleştirilmiştir. Böylece elektronik donanım kutusu içerisinde besleme kabloları çok dolaşmayacaktır.

Bu kart üzerinde, gerektiği durumlarda kolayca erişilebilmesi amacıyla, besleme konektörlerinin bir bezeri daha yerleştirilerek yedek besleme çıkışları konulmuştur. Aşağıda bu bağlantılar verilmiştir.



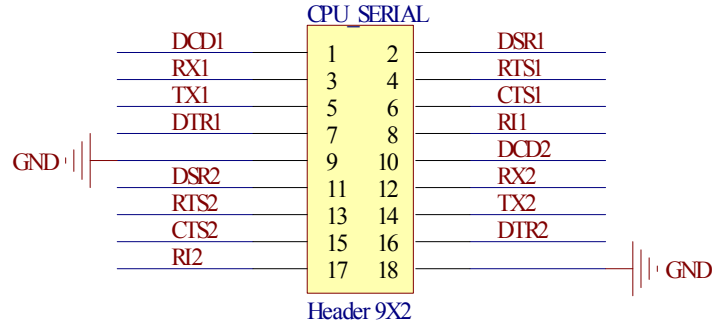
Şekil 19: Yedek besleme çıkışları

Bu kart üzerine beslemelerin takılıp takılmadığını göstermesi amacı ile 3 adet LED yerleştirilmiştir. Aşağıda bu LED ler gösterilmiştir.



Şekil 20: Besleme gösterge LED leri

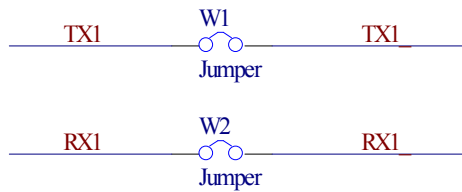
PC104 yığınının CPU biriminde yer alan RS232 konektörü (J17) doğrudan bu karta bağlanarak MODEM veya Yönelim düğümüne bağlanabilmesi amaçlanmıştır. Bu konektör aşağıda gösterilmiştir.



Şekil 21: CPU RS232 bağlantı konektörü

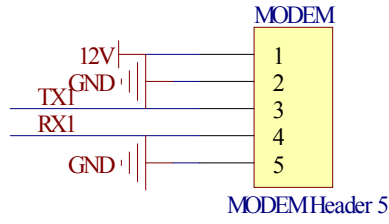
CPU\_SERIAL konektörü de kart üzerinde CPU kartında yerleştirildiği yer paralel olacak şekilde yerleştirilmiştir. Böylece 18 adet kablo kutu içerisinde kıvrılmak zorunda kalmayacaktır.

Aşağıda gösterilen jumperlar, 1 numaralı RS232 bağlantısının Yönelim düğümüne taşınmasını etkin hale getirmektedir. Bu nedenle, MODEM sisteme takılı olduğu durumda bu jumperlar **açık** devre olmak zorundadır.



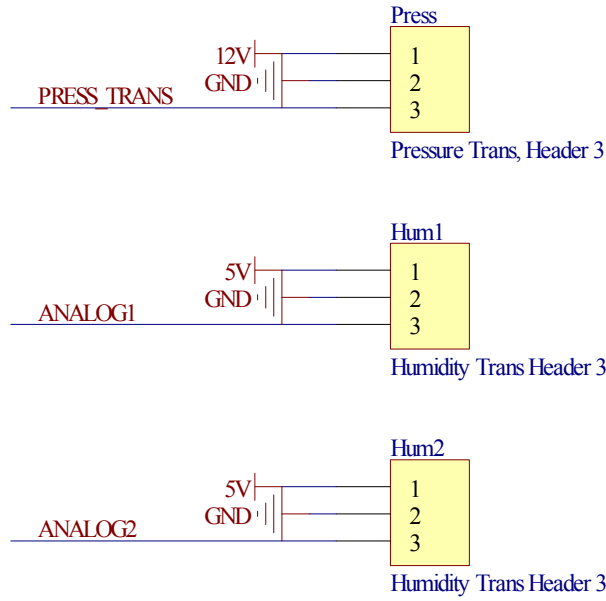
Şekil 22: RS232 1'in Yönelim düğümüne taşınmasını denetleyen jumperlar

PC104 yığını ile MODEM arasında aşağıda gösterilen konektörün olmasına karar verilmiştir.



Şekil 23: Modem konektörü

Basınç ve nem almaçları arasında aşağıda gösterilen konektörler yer alacaktır.



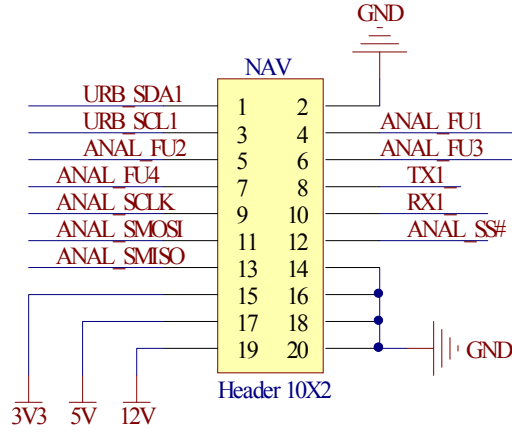
Şekil 24: Basınç ve Nem Almaçlarının konektörleri

Şekil 24’de gösterilen netlerin işlevleri aşağıda verilmiştir.

Pin Adı:	İşlevi:
PRESS_TRANS	Basınç almacı sinyali
ANALOG1	Nem Almacı 1 sinyali
ANALOG2	Nem Almacı 2 sinyali

Tablo 12: Analog Almaç konektörleri pin işlevleri

Yönelim düğümü ile PC104 yığını arasında olacak olan konektör aşağıda verilmiştir.



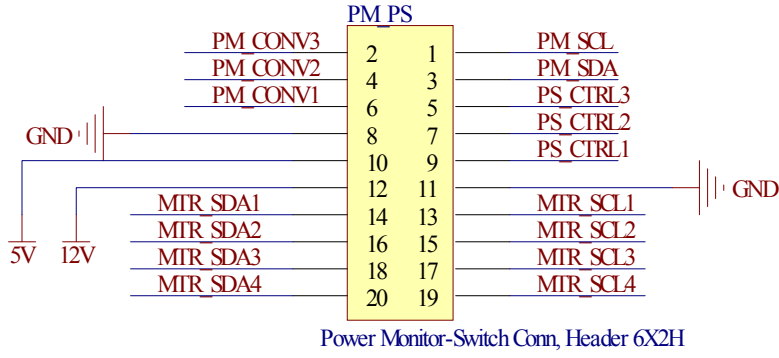
Şekil 25: Yönelim düğümü konektörü

Aşağıda Şekil 25’de gösterilen netlerin işlevleri verilmiştir.

Pin Adı:	İşlevi:
URB_SDA1	URB1 SDA pini
URB_SCL1	URB1 SCL pini
ANAL_FU1	Analog mikro-denetleyici ile Yönelim düğümü arasında sayısal pin 1
ANAL_FU2	Analog mikro-denetleyici ile Yönelim düğümü arasında sayısal pin 2
ANAL_FU3	Analog mikro-denetleyici ile Yönelim düğümü arasında sayısal pin 3
ANAL_FU4	Analog mikro-denetleyici ile Yönelim düğümü arasında sayısal pin 4
TX1_	RS232 1 TX pini
RX1_	RS232 1 RX pini
ANAL_SCLK	Analog mikro-denetleyici SPI veri yolu SCLK pini
ANAL_SMOSI	Analog mikro-denetleyici SPI veri yolu MOSI pini
ANAL_SMISO	Analog mikro-denetleyici SPI veri yolu MISO pini
ANAL_SS#	Analog mikro-denetleyici SPI veri yolu CS pini (Analog slave modunda)

Tablo 13: Yönelim düğümü konektörü pin işlevleri

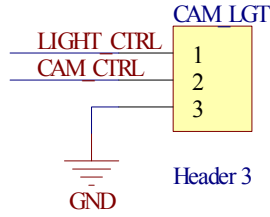
PC104 yığını ile Güç Ölçme ve Güç Dağıtım kartı arasında aşağıdaki konektör yer almıştır.



Şekil 26: Güç Ölçme ve Güç Dağıtım kartı konektörü

Bu konektörün pin işlevleri Güç ölçme ve dağıtım kartının anlatıldığı kısımda verilecektir.

Kamera ve ışık denetleme sinyallerinin diğer arayüz kartına taşımak amacı ile aşağıda gösterilen konektör konulmuştur.

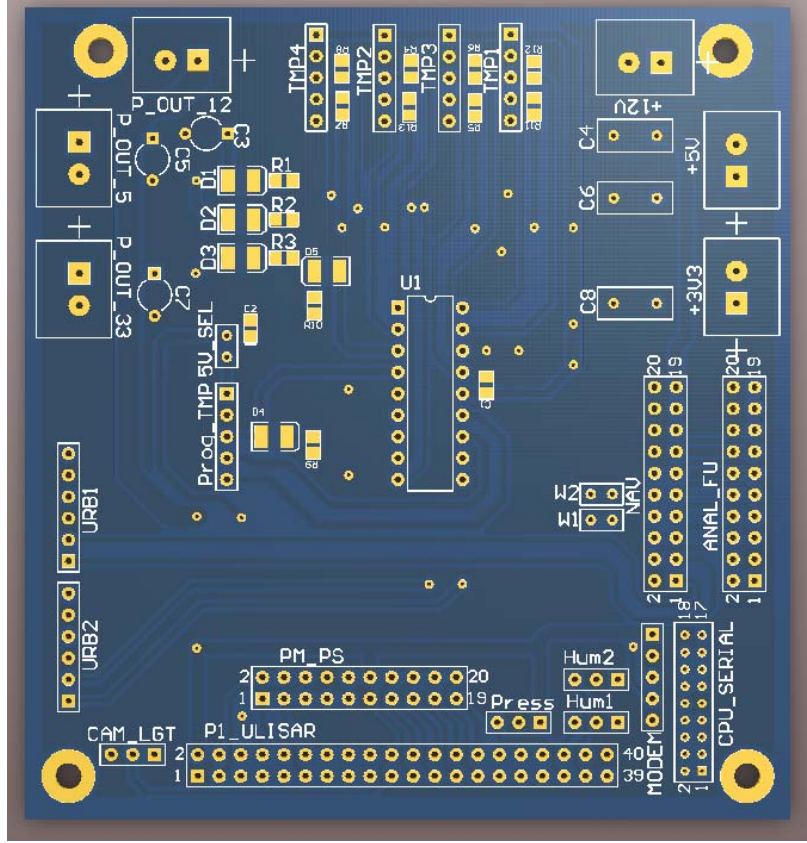


Şekil 27: Faydalı yük arayüz kartı konektörü

### PCB Tasarımı:

PCB elimize ulaştığında aşağıdaki gibi olacaktır.





PCB yerleşim planı için eklere bakınız. (INTERFACE1\_Assembly Drawings.pdf)

### 2.2.2.5 I6 Kartı

(Faydalı yük bilgisayarına bağlı yığın arayüz kartı)

I5 Kartında olduğu gibi I6 kartının da amacı Faydalı yük PC104 yığnında yer alan veya bu yığna bağlanacak çevresel birimlerin konektörlerini içermektedir. Ancak bu kartta herhangi bir düğüm gerçekleştirilmemiştir.

Faydalı yük yığnına bağlı sadece ultrasonik almaç düğümü olduğu için bu kartta sadece ön ve arka ultrasonik almaç seçme kartlarına bağlantı vardır. Faydalı yük yığnına bağlanacak SONAR ve KAMERALARın sinyal bağlantıları da bu kart üzerindedir.

#### Şematik Tasarım:

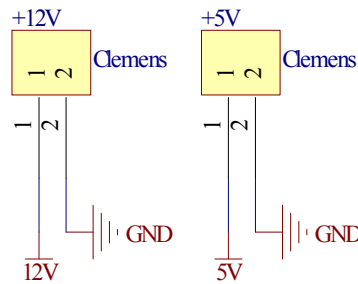
Ekteki dokümanlara bakınız. (INTERFACE2\_sch.pdf)

## Malzeme Listesi:

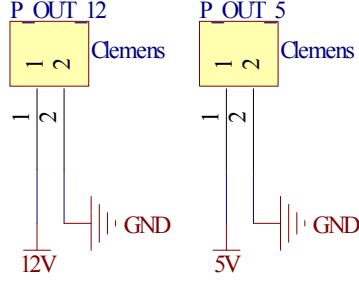
No	Designator	Footprint	Comment	Description	Quantity	Value
1	+5V, +12V, P_OUT_5,P_OUT_12	PowerConn2	Clemens	2 Connector Plugin Screwed 3.81mm pitch PCBN connector	4	
2	C3, C5	CAPR5-4X5	Cap2	Capacitor	2	100uF
3	C4, C6	RAD-0.3	Cap	Capacitor	2	1uF
4	CAM_OUT, SONAR	HDR1X4	Header 4	Header, 4-Pin	2	
5	CPU_SERIAL	HDR2X9_200	Header 9X2	Header, 9-Pin, Dual row	1	
6	D1, D2	3.5X2.8X1.9	LED3	Typical BLUE SiC LED	2	
7	F_USONIC, R_USONIC, VIDEO	HDR2X5	Front Ultrasonic Ctrl,Header 5X2, Header 5X2, Rear Ultrasonic Ctrl, Header 5X2	Header, 5-Pin, Dual row	3	
8	P1_ULISAR	HDR2X20	ULISAR_Internal	Header, 20-Pin, Dual row	1	
9	R1	R0805	Res2	Resistor	1	10k
10	R2	R0805	Res2	Resistor	1	1k
11	URB1, URB2	HDR1X6	URB_CONN, HDR6	Header, 6-Pin	2	

## Önemli Pin Tanımları:

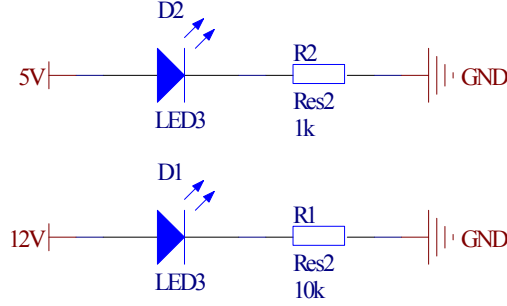
I5 Kartında verilen besleme konektörleri ve besleme gösterge LEDleri bu kart içinde 3.3V hariç geçerlidir. Bu yığına bağlanan çevresel birimlerin 3.3V istememesi sebebiyle bu karta 3.3V besleme konektörü konulmamıştır. Aşağıdaki şekillerde bu kart üzerindeki konektörler gösterilmiştir.



Şekil 28: Besleme konektörleri

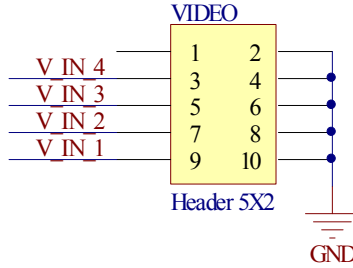


Şekil 29: Besleme çıkış konektörleri



Şekil 30: Besleme gösterge LEDleri

I5 kartında olduğu gibi, CPU kartındaki RS232 konektörü (J17) bu kart üzerinde de konulmuştur. Bu kartta I5'ten farklı olarak MPEG4 dönüştürücü kartından taşınan VIDEO giriş konektörü de yerleştirilmiştir. Bu konektör aşağıdaki şekilde gösterilmiştir.



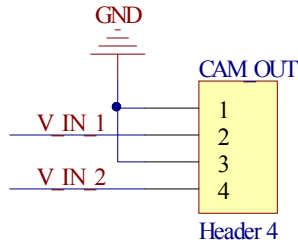
Şekil 31: VIDEO konektörü

Bu konektördeki pinlerin işlevleri aşağıda verilmiştir.

Pin Adı:	İşlevi:
V_IN_1	Video girişi 1
V_IN_2	Video girişi 2
V_IN_3	Video girişi 3
V_IN_4	Video girişi 4

Tablo 14: VIDEO konektörü pin işlevleri

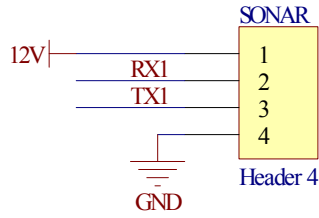
Kameralara yapılacak olan bağlantının konektörü aşağıda gösterilmiştir.



Şekil 32: Kamera bağlantıları

Kameraların sinyal kabloları doğrudan VIDEO konektörüne bağlanmıştır. Böylece, laboratuvar ortamında yapılan geliştirme çalışmaları olduğu gibi kullanılabilir.

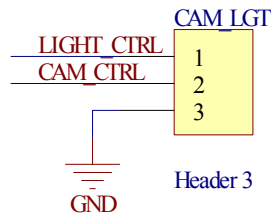
Sonara yapılacak olan bağlantının konektörü aşağıda gösterilmiştir.



Şekil 33: SONAR konektörü

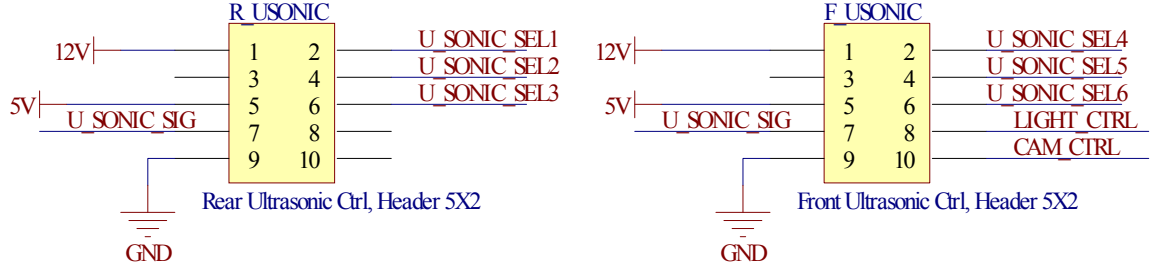
Sonar ile RS232 veri yolundan haberleşilmesi sebebiyle, CPU\_SERIAL konektörü üzerinden CPU kartının 1. RS232 çıkışı doğrudan bu konektöre bağlanmıştır.

Kamera ve ışık denetleme sinyallerinin diğer arayüz kartına taşımak amacı ile aşağıda gösterilen konektör konulmuştur.



Şekil 34: Faydalı yük arayüz kartı konektörü

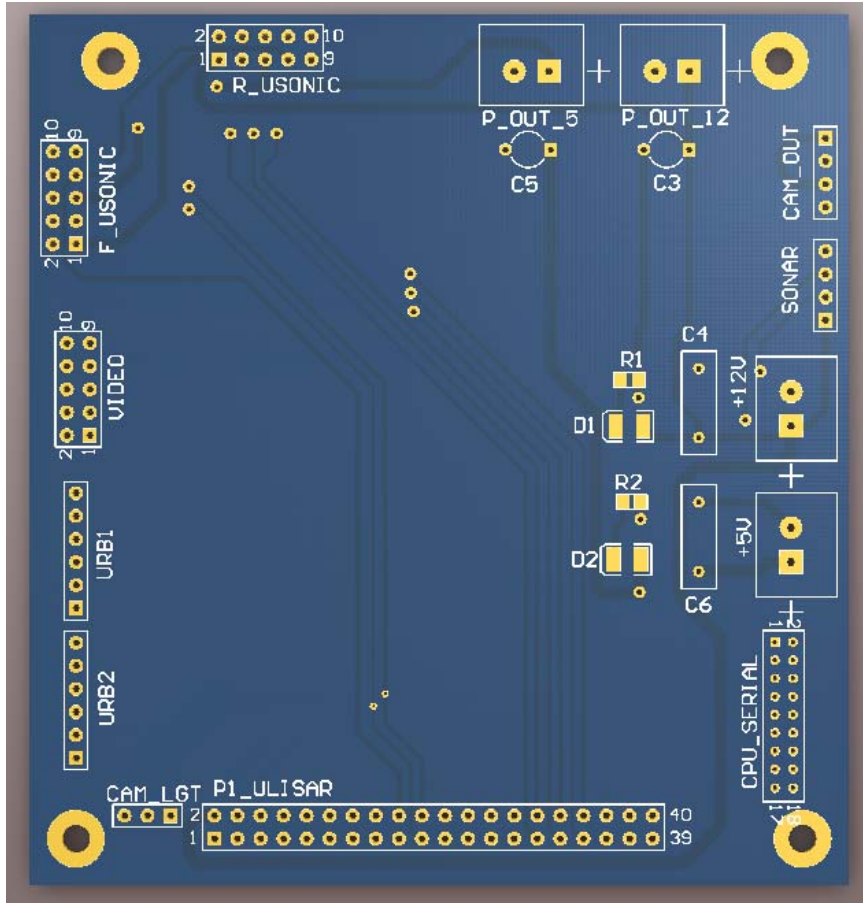
Aşağıda bu kart üzerindeki diğer konektörler gösterilmiştir. Bu konektörlerin pinlerinin işlevsel tanımları UF2 ve UR3 kartları anlatılırken verilecektir.



Şekil 35:UF2 ve UR3 bağlantı konektörleri

### PCB Tasarımı:

PCB elimize ulaştığında aşağıdaki görünümde olacaktır.



PCB yerleşim planı için eklere bakınız. (INTERFACE2\_Assembly Drawings.pdf)

## 2.2.2.6 G1 Kartı

### (Güç durumu ölçme ve güç dağıtım kartı)

Bu kartın temel işlevleri pillerin enerji seviyelerini belirlemek, anlık tüketilen enerjiyi ölçmek, pillerin çevresel birimler ile olan bağlantısını açmak/kapamak ve pillerin çıkışlarının çevresel birimlere uygun konektörler ile dağılmasını sağlamaktır.

Güç ölçme işlevini temelde Analog Devices firmasının ADM1191 yongası gerçekleştirmektedir. Bu yonganın seçilmesindeki sebepler:

- Çok geniş aralıktaki gerilim değerlerini okuyabilmesi,
- Anlık akım değerini belirleyerek çıktı olarak vermesi,
- I2C seri veri yolu ile iletişilmesi,
- Aynı I2C veri yoluna birden fazla yonganın bağlanabilmesidir.

Anlık çekilen akım miktarını ölçmek için çok küçük dirençli dirençler kullanılmaktadır. ADM1191 yongasının gereksinimleri ve pillerden çekilecek maksimum anlık akım miktarı hesaba katılarak bu direncin değerinin 0,03 OHM olmasına karar verilmiştir.

### Şematik Tasarım:

Ek dokümanlara bakınız. (PM\_PS\_sch.pdf)

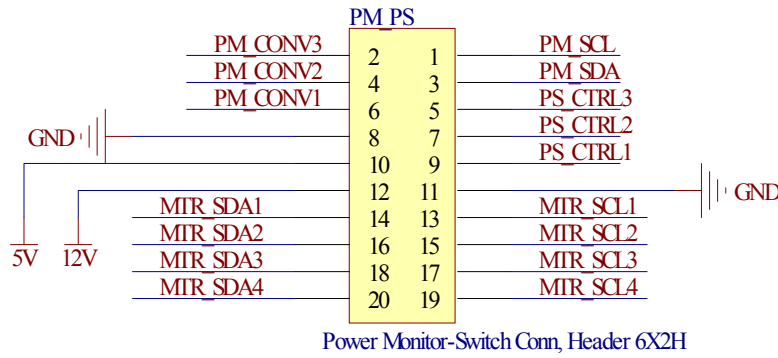
### Malzeme Listesi:

No	Designator	Footprint	Comment	Description	Quantity	Value
1	BATT_IN_1, BATT_IN_2, BATT_IN_3, PWR1, PWR2	PowerConn2	Clemens	2 Connector Plugin Screwed 3.81mm pitch PCBN connector	5	
2	D_1_1, D_2_1, D_3_1	DO-41	Diode 1N4001	1 Amp General Purpose Rectifier	3	
3	MTR1, MTR2, MTR3, MTR4	PowerConn4	Clemens	3 Connector Plugin Screwed 3.81mm pitch PCBN connector	4	
4	PM_PS	HDR2X10	Power Monitor-Switch Conn, Header 6X2H	Header, 10-Pin, Dual row	1	
5	Q_1_1, Q_2_1, Q_3_1	TO-92A	2N3904	NPN General Purpose Amplifier	3	
6	RL_1_1, RL_2_1, RL_3_1	Relay	Relay	DPDT Relay with Standart Package	3	

7	R_1_1, R_2_1, R_3_1	CurrSense	0R03	Current Sense Resistor 0.03 Ohm	3	
8	R_1_2, R_2_2, R_3_2	AXIAL-0.3	Res1	Resistor	3	2K2
9	S1	SW_SPDT	SW_SPDT	SPDT switch	1	
10	U_1_1, U_2_1, U_3_1	RM-10	ADM1191	Analog Devices, Digital Power Monitor with Convert Pin and ALERTB Output	3	

### Önemli Pin Tanımları:

PC104 yığını ile Güç Ölçme ve Güç Dağıtım kartı arasında aşağıdaki konektör yer almıştır.



Şekil 36: Güç Ölçme ve Güç Dağıtım kartı konektörü

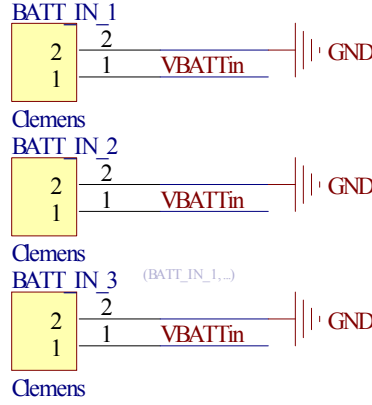
Şekil 36'da gösterilen konektör pinlerinin işlevleri aşağıda verilmiştir.

Pin Adı:	İşlevi:
PS_CTRL1	Güç denetim anahtarı 1 denetim pini
PS_CTRL2	Güç denetim anahtarı 2 denetim pini
PS_CTRL3	Güç denetim anahtarı 3 denetim pini
PM_SCL	Güç ölçüm I2C veri yolu CLOCK pini
PM_SDA	Güç ölçüm I2C veri yolu DATA pini
CONV1	Güç ölçümlene yongası 1 CONVERT (ölç) pini
CONV2	Güç ölçümlene yongası 2 CONVERT (ölç) pini
CONV3	Güç ölçümlene yongası 3 CONVERT (ölç) pini
DBG_LED	Hata ayıklama LEDi
MTR_SCL1	Motor 1 I2C bağlantı pinleri
MTR_SDA1	
MTR_SCL2	Motor 2 I2C bağlantı pinleri
MTR_SDA2	
MTR_SCL3	Motor 3 I2C bağlantı pinleri

MTR_SDA3	Motor 4 I2C bağlantı pinleri
MTR_SCL4	
MTR_SDA4	

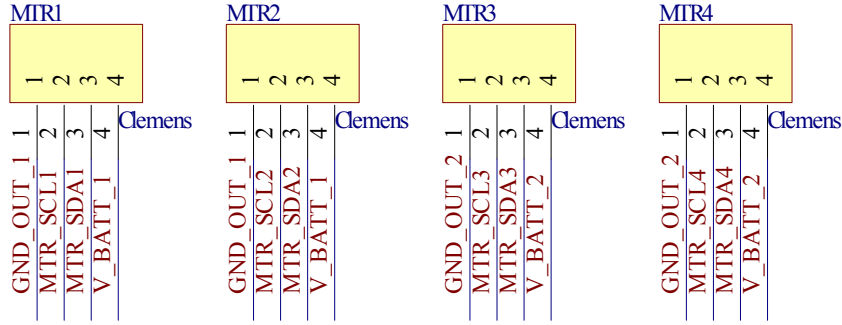
Tablo 15: Güç ölçme ve dağıtma kartı konektörü pin işlevleri

Bu karta pil paketleri arasındaki konektör aşağıda gösterilmiştir.



Şekil 37: Pil konektörleri

Aşağıda her bir motor ile yapılması gereken bağlantı gösterilmiştir.

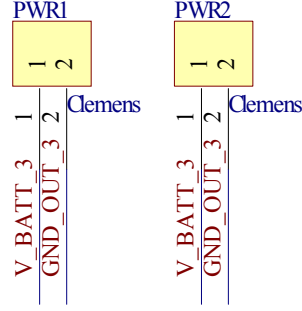


Şekil 38: Motor konektörleri

Şekil 38'de gösterilen motor konektörlerinden de anlaşılacağı üzere, 1 ve 2 numaralı motorlar 1 numaralı pil paketine; 3 ve 4 numaralı motorlar 2 numaralı pil paketine bağlanacak şekilde düşünülmüştür.

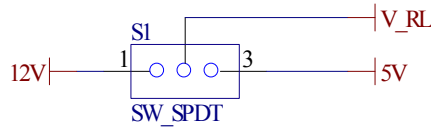
Elektronik donanımlara gidecek olan bağlantılar ise aşağıda gösterilen konektörler üzerinden yapılacaktır.





Şekil 39: Elektronik donanım besleme konektörleri

Kullanılan rölelerin tipine göre seçim yapabilmek amacı ile S1 anahtar bağlantısı yapılmıştır.

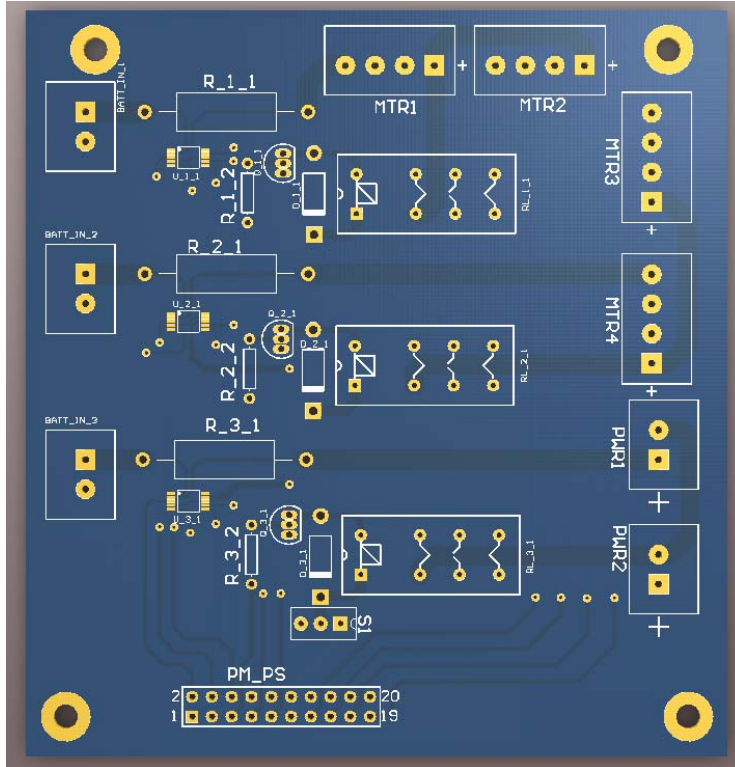


Şekil 40: Röle besleme seçme anahtarı

S1 anahtarı 1-2 pinlerini kısa devre ederse, röleler 12V ile beslenecek, 2-3 pinlerini kısa devre ederse röleler 5V ile beslenecektir.

## PCB Tasarımı

PCB elimize ulaştığında aşağıdaki görünümde olacaktır.



PCB yerleşimi için ek dokümanlara bakınız. (PM\_PS\_Assembly Drawings.pdf)

### **2.2.2.7 N4 Kartı**

#### **(Yönelim Dügümü)**

IMU dügümü aracın durum parametrelerinin tahmin edildiği ve ana denetleyici bilgisayara bildirilmek üzere URB dügümünün gerçekleştirildiği donanım birimidir. Bu nedenle bu dügümün temel olarak iki işlevi bulunmaktadır:

- Ataletsel ölçüm almaçlarının alınan bilgiler kullanılarak denizaltının durumunun tahmin edilmesi,
- Denizaltının durumunun ve ataletsel almaç değerlerinin ana denetleyici bilgisayara bildirilmesi,

Ataletsel ölçümlerin tahmin algoritmasının hesap yoğunluğunun fazla olması sebebiyle, hesaplamaların yapılacağı mikro-denetleyicinin hesap gücünün yüksek olması gerekmektedir. Yapılan yoğun incelemeler sonucunda, yazılım geliştirme süresi, donanım arayüzü kolaylığı ve dokümantasyon kalitesi açısından en uygun işlemcinin ARM mimarisinden olması gerektiği sonucuna varılmıştır. Seçilen işlemci NXP firmasının LPC2378 model ARM7 işlemcisine karar verilmiştir. Bu işlemcinin 144 pinli tipinin yeterinden çok iletişim arayüzü sağlaması, tasarımda yapılması muhtemel değişikliklere de olanak sağlayacaktır.

### **2.2.3 Güç Tüketimi**

Sualtı aracının ihtiyaç duyduğu enerji lityum polimer pil paketleriyle sağlanacaktır. Bu pillerin direnç değerleri Tablo 16: Kokam 834374H (2000mAh hücreler) pillerin direnç değerleri’de verilmiştir.

Sıcaklık(°C)	Direnç(mΩ)	Koşul
23	5.6	Tam şarj
23	6.0	Yarım şarj
23	6.3	Boş

Tablo 16: Kokam 834374H (2000mAh hücreler) pillerin direnç değerleri

Su altı aracında kullanılacak olan elektronik donanım elemanlarının güç tüketim değerleri Tablo 17’de verilmiştir.

Donanım Eleman Adı:	Akım (mA)	Voltaj (V)	Güç Tüketimi(W) Max	Adet
Parvus CPU-1462	2600	5	13	2
Parvus ACS-5150 DC-DC çevirici	N/A	8-40	6.82	2
Çıktı değerleri				
1	2000	3.3	N/A	N/A
2	10000	5	N/A	N/A
3	2000	12	N/A	N/A
Parvus CTR-1461 PCMCİ çevirici	70	5	0.35	2
Parvus CTR-1475 MPEG 4 sıkıştırıcı	1400-2000	5	7-10	1
LinkQuest Akustik Modem UWM 1000				1
Alici Modu	-----	12-24	0.75	N/A
Verici Modu	-----	12-24	1 veya 2	N/A
Uyku Modu	-----	12-24	0.008	N/A
Microstrain 3DM-GX1 IMU	65	5.2-12	0.338-0.78	1
Outland Technology UWC 300 kamera	110	12-24	1.32-2.64	2
Outland Technology UWL 300 aydınlatma	2917	12	35	2
ProWave Corp. Sesüstü Çevirici	4166.67	12	50	4
Tritech Micron DST Sonar	-----	12-50	2.5	1
Z-World RCM 4000 Mikrodenetleyci	30	3.3	0.099	1
Diğer	1000	5	10	N/A
SeaBotix SBT151 thruster	(max) 6000	24	144	4
Kokam 2000mAh LiPo hücreler	(max) 4000	3.7	0.1	-----

Tablo 17: Donanım elemanları güç tüketim değerleri

Tablo 17’de gösterilen “Diğer” (2A – 12V) donanım elemanları içerisinde motor sürücü devresi, ses üstü çevirici sürücü devresi, basınçölçer gibi daha sonra tasarlanacak olan elektronik devrelerin ve daha sonra belirlenecek olan elemanların güç ihtiyacını göstermektedir.

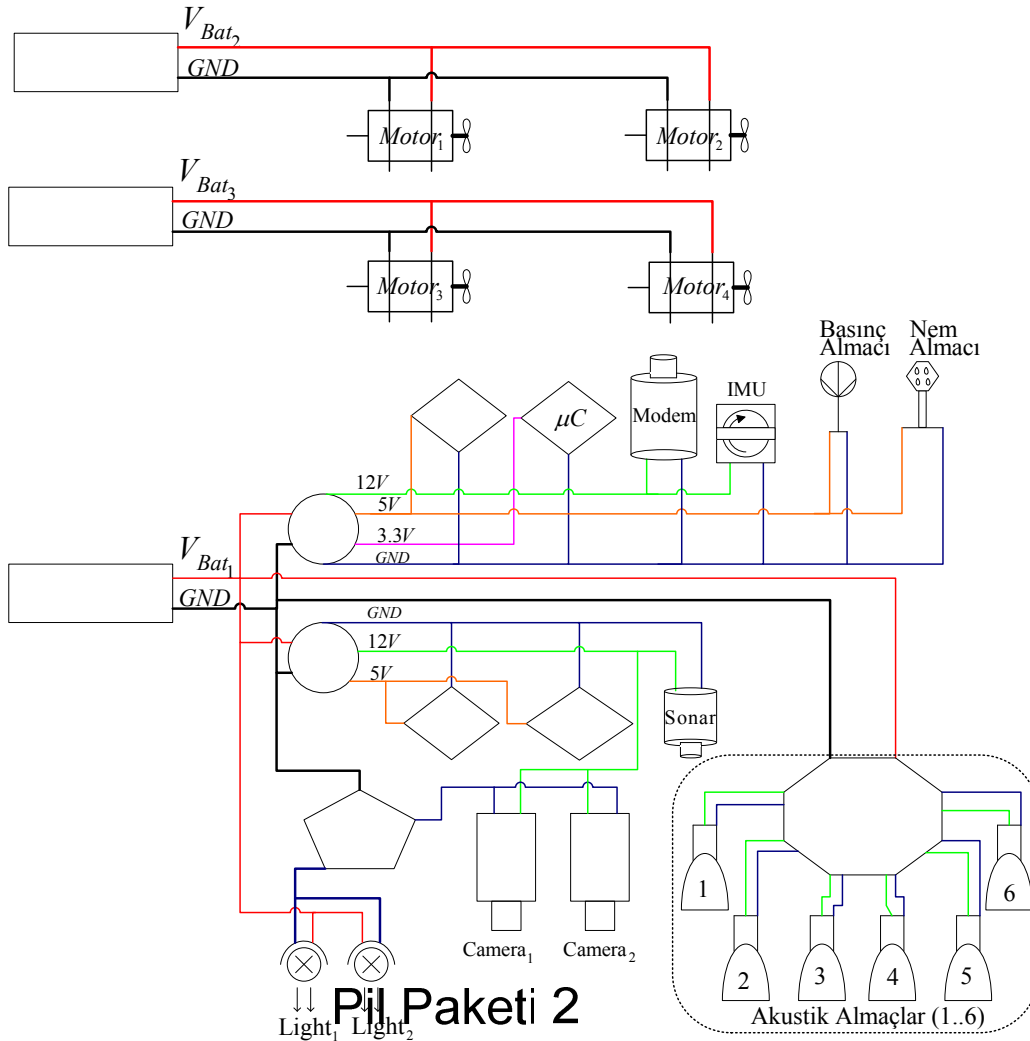
DC-DC çeviricinin çıktı değerleri Tablo 17’de gösterilmiş olmakla beraber, kullanılan DC-DC çevirici toplamda 50W çıkış gücüne sahiptir. 3,3V ve 12V çıkış voltajları ana güç

bağlantısı olan 5V kanalı üzerinden beslenmektedirler. Bu nedenle, bu çeviricinin güç hesabı yapılırken toplam gücün 50W'ı aşmaması gerekmektedir.

Kullanılacak olan Lityum Polimer pillerin içsel dirençleri bu çalışmada yok varsayılmaktadır (bu değerler çok küçük olduğu için bu varsayımın yeterince doğru sonuçlar vereceği düşünülmektedir).

### 2.2.3.1 Donanım Elemanlarının Güç Bağlantısı

Daha önce, donanım elemanlarının birbiriyle haberleşme standartlarını içeren tasarım, ULİSAR Genel Gereksinim ve Tasarım Dokümanında yer almaktadır. Donanım elemanlarının güç bağlantıları **Error! Reference source not found.**'de gösterilmiştir.



Şekil 41: Su altı aracı güç dağılımı

Tablo 17 ve Şekil 41 kullanılarak su altı aracında kullanılacak olan pil paketlerinin yapıları belirlenebilir.

### 2.2.3.2 Pil Paketlerinin Belirlenmesi

Şekil 41’de gösterilen güç bağlantısında, iki adet DC-DC çevirici kullanılmıştır. Bu nedenle, Tablo 17’de verilen güç tüketim değerlerine bakılarak her bir çeviriciden ne kadar akım çekildiği hesaplanmalıdır.

Donanım Elemanı	Akım(mA)	Voltaj(V)	Güç(W)
CPU1	2600	5	13
PCMCI çevirici	70	5	0.35
mikrodenetleyci	30	3.3	0.099
IMU	65	12	0.78
Diğer Almaçlar	1000	5	5
Modem	37.5	12	0.45
TOPLAM	3802.5	-----	19.679

Tablo 18: DC-DC çevirici 1’den çekilen toplam akım ve istenilen güç

Donanım Elemanı	Akım(mA)	Voltaj(V)	Güç(W)
CPU2	2600	5	13
PCMCI çevirici	70	5	0.35
MPEG-4 sıkıştırıcı	2000	5	10
Kameralar	220	12	2.64
Sonar	208.33	12	2.5
TOPLAM	5098.33	-----	28.49

Tablo 19: DC-DC çevirici 2’den çekilen toplam akım ve istenilen güç

Tablo 18 ve Tablo 19 değerleri bu çeviricilerin güç kapasitelerinin (en çok 50W) aşılmadığını göstermektedir. *Pil Paketi 1’den çekilen toplam akım* ise Tablo 20’de gösterilmiştir.

Donanım Elemanı	Akım(mA)	Voltaj(V)	Güç(W)
DC-DC çevirici 1	4602.5	-----	19.67
DC-DC çevirici 1’de harcanan	-----	14.8	2.68
DC-DC çevirici 2	5098.33	-----	28.5
DC-DC çevirici 2’de harcanan	-----	14.8	3.89
Aydınlatmalar	4729.73	14.8	70
Ultrasonik Çeviriciler	4166.67	12	50
Ultr.Çev.Güç Devr. Harcanan	-----	-----	0.50
TOPLAM	18597.23	-----	174.74

Tablo 20: Pil Paketi 1’den çekilen toplam akım ve istenilen güç

Tablo 20’de belirtilen ultrasonik çevirici akım değeri tek bir çevirici için verilmesinin nedeni aynı anda birden fazla çevirici çalıştırılmayacağı gerçeğidir. Yine Tablo 20’de, ultrasonik çevirici güç devresinde 0,5W güç harcanacağı kabul edilmiştir. Bu değer tahmini olmakla beraber, eğer lineer doğrultucu kullanılırsa oldukça gerçekçidir. DC-DC çeviricilerde harcanan güç, bu doğrultucuların veriminin %88 olduğu gerçeği kullanılarak hesaplanmıştır.

Şekil 41’de *Pil Paketi 1* olarak gösterilen motorlar haricindeki elektronik devre elemanlarının beslediği pil paketinin voltajı 12-40VDC arasında olması gerekmektedir. Bu durumda, pil paketlerinin oluşturmak için kullanılan Lityum Polimer pillerin, 4S ile bağlanması (pil paketlerinin 4S’lık gruplar halinde olması nedeniyle) uygun olacaktır. Paralel bağlanacak pil sayısı ise aşağıdaki formül ile hesaplanabilir:

$$N_p = \frac{P_T \cdot t}{V \cdot C} \quad (0.1)$$

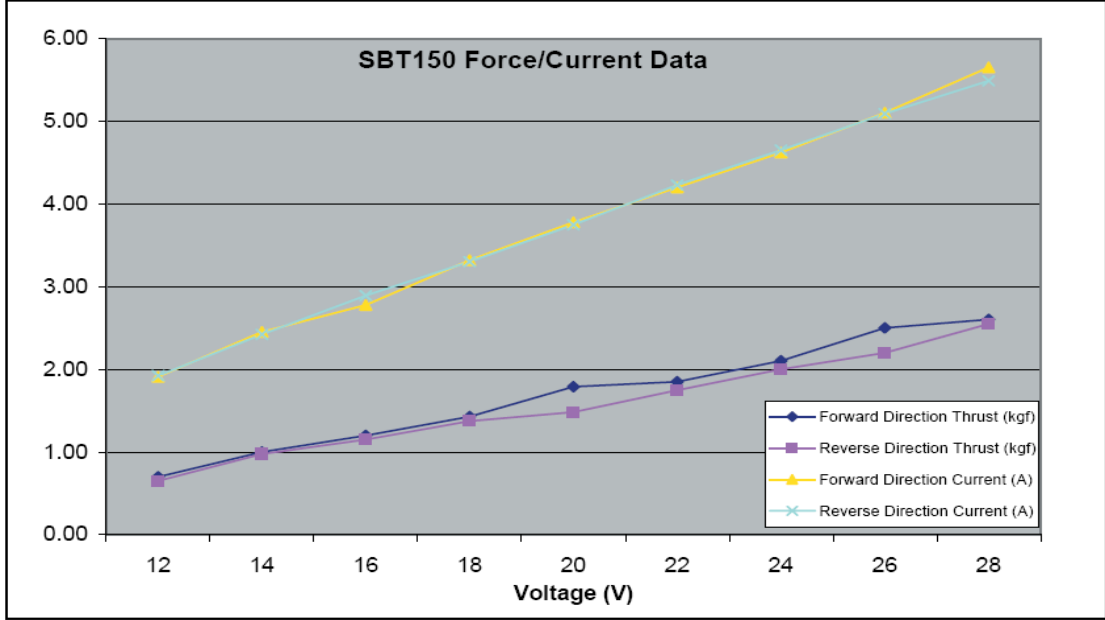
(0.1)’de  $N_p$  paralel bağlanacak pil sayısını,  $P_T$  toplam gücü,  $t$  saat cinsinden süreyi,  $V$  4S pillerin voltajını ve  $C$  de pillerin  $A \cdot sa$  cinsinden kapasitesini göstermektedir.

$$N_p = \frac{174.74 \cdot 1}{(3.7 \cdot 4) \cdot 2} = round(5.90) = 6$$

$N_p$  6 olmaktadır.

Kameralar ile sürekli olarak görüntü alınmaması planlanmaktadır. Bu nedenle, aydınlatma ve kameralar sürekli olarak çalıştırılmayacaktır. Aydınlatmalar sadece kameralardan görüntü alınacağı zaman açılacaktır. Bu tasarım ise, kamera ve aydınlatma sisteminin denetiminin uygun bir biçimde yapılmasını zorunlu kılmaktadır. Şekil 41’de gösterilen kamera ve aydınlatma anahtarının görevi anlatılan gerekliliği yerine getirmektir. Böylece, aydınlatmaların 70W’lık güç ihtiyacı süreğen olmamakta, 4S6P’lik pil paketinin 1 saatlik operasyon süresinden daha uzun bir süre gerekli gücü sisteme verebilecektir. Bu durumda *Pil Paketi 1*’in 4S6P’lik bir pil paketi olması çok uygun olmaktadır.

*Pil Paketi 2 ve Pil Paketi 3* ise birbiriyle aynı yapıya sahip olması gerekmektedir. Bunun en temel nedeni bu iki pil paketinden ikiye adet motorun beslenmesidir.Şekil 42’de itici motorların voltaja karşılık, çekilen akım değerleri ve elde edilen itki değerleri verilmiştir.



Şekil 42: İtici motorların itkinin akım ile değişimi

Motor Serial	Direction	Voltage (V)	Force (kgf)	Current Draw (A)
SBT03179	Forward	12	0.70	1.91
SBT03179	Forward	14	1.00	2.45
SBT03179	Forward	16	1.20	2.78
SBT03179	Forward	18	1.43	3.32
SBT03179	Forward	20	1.79	3.78
SBT03179	Forward	22	1.85	4.2
SBT03179	Forward	24	2.10	4.62
SBT03179	Forward	26	2.50	5.1
SBT03179	Forward	28	2.60	5.65
SBT03179	Reverse	12	0.65	1.92
SBT03179	Reverse	14	0.98	2.42
SBT03179	Reverse	16	1.15	2.89
SBT03179	Reverse	18	1.38	3.3
SBT03179	Reverse	20	1.48	3.75
SBT03179	Reverse	22	1.75	4.23
SBT03179	Reverse	24	2.00	4.65
SBT03179	Reverse	26	2.20	5.09
SBT03179	Reverse	28	2.55	5.49

Tablo 21:Şekil 42'nin oluşturulmasında kullanılan veriler

Şekil 42ve Tablo 17'de gösterilen değerler doğrultusunda ve aynı pil paketinden beslenen motorların aynı anda en yüksek itki ihtiyacının olmaması var sayımı durumunda iki adet itici motoru besleyen pik paketlerinin 8S3P olması uygun olmaktadır. Tablo 22'de tüm pil paketlerinin düzenleşimi verilmiştir.

Pil paketi adı	Düzenleşimi
<i>Pil Paketi 1</i>	4S6P
<i>Pil Paketi 2</i>	8S3P
<i>Pil Paketi 3</i>	8S3P

Tablo 22: Pil paketlerinin düzenleşimi

### 2.2.3.3 Su geçirmez bölümdeki güç tüketimi

Sualtı aracı su geçirmez bölüm su geçiren bölüm olarak ikiye ayrılabilir. Su geçiren bölümde doğrudan su ile tema edecek olan elektronik cihazlar yerleştirilirken, su geçirmez bölümde



diğer donanım elemanlarının tamamı yer alacaktır. Tablo 23’de su geçirmez bölümde bulunacak olan donanım elemanları.

Donanım Eleman Adı:	Akım (mA)	Voltaj (V)	Güç Tüketimi(W) Max	Adet
Parvus CPU-1462	2600	5	13	2
Parvus ACS-5150 DC-DC çevirici	N/A	8-40	6.82	2
Parvus CTR-1461 PCMCI çevirici	70	5	0.35	2
Parvus CTR-1475 MPEG 4 sıkıştırıcı	1400-2000	5	7-10	1
Microstrain 3DM-GX1 IMU	65	5.2-12	0.338-0.78	1
Z-World RCM 4000 Mikrodenetleyci	30	3.3	0.099	1
Diğer	1000	5	5	N/A
Kokam 2000mAh LiPo hücreler	(max) 4000	3.7	0.1	100

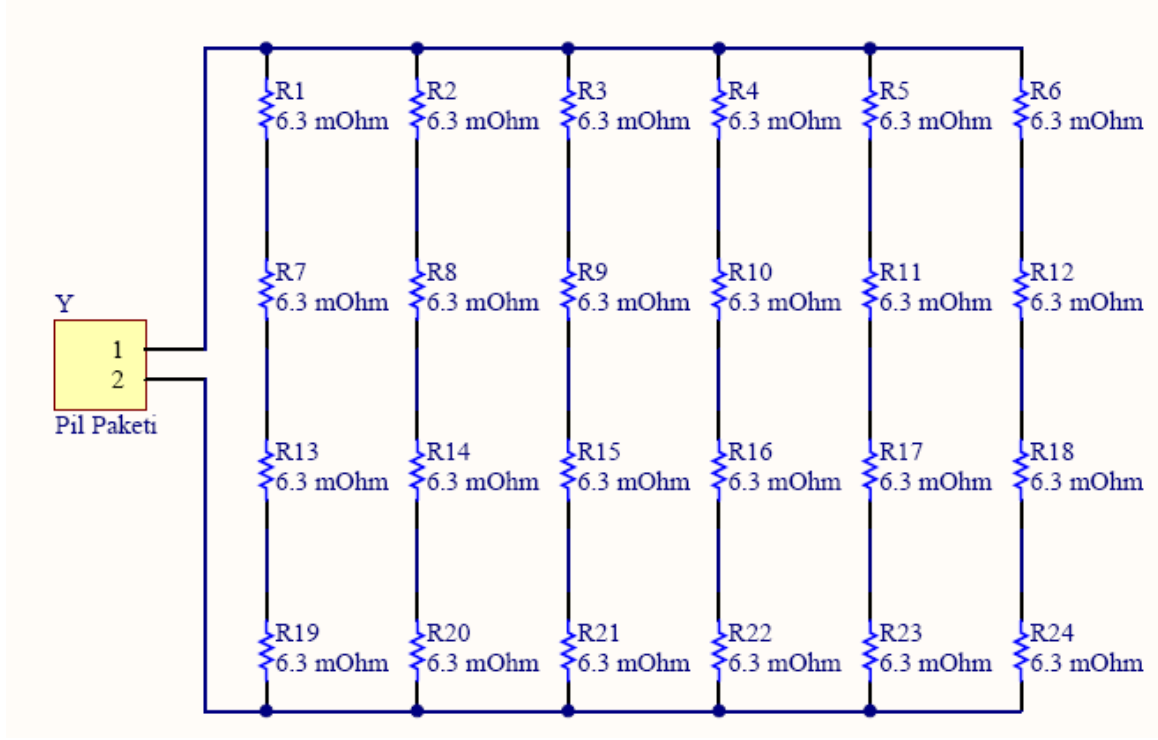
Tablo 23: Su geçirmez bölümde bulunacak olan donanım elemanları ve güç tüketimleri

Tablo 23’de gösterilen donanım elemanlarına birkaç donanım elemanı daha ilave edilecektir. Bu tabloda gösterilmeyen, ancak ilave edilmesi zorunlu olan donanım elemanları, nem ve sıcaklık çeviricileri ile arayüz kartları olarak düşünülebilir. Bu donanım elemanları, Tablo 23’de *Diğer* olarak gösterilen alanda ifade edilmiş ve gerçekteki güç tüketimlerinden çok daha büyük miktarda güç tükettikleri var sayılmıştır.

Donanım Eleman Adı:	Adet	Güç Tüketimi(W)	Toplam Güç(W)
Parvus CPU-1462	2	13.00	26.00
Parvus ACS-5150 DC-DC çevirici	2	4.00	8.00
Parvus CTR-1461 PCMCI çevirici	2	0.35	0.70
Parvus CTR-1475 MPEG 4 sıkıştırıcı	1	10.00	10.00
Microstrain 3DM-GX1 IMU	1	0.78	0.78
Z-World RCM 4000 Mikrodenetleyci	1	0.10	0.10
Diğer	N/A	5.00	5.00
Kokam 2000mAh LiPo hücreler	-----	0.10	-----
Piller hariç toplam			50.58

Tablo 24: Su geçirmez bölümde pillerin haricindeki donanım elemanlarının güç tüketimleri

Pillerde tüketilen güç ise eşdeğer dirençleri hesaplanarak bulunabilir. Bunun için, pil paketlerini paralel ve seri bağlanan ve sadece dirençten oluşan bir devre olarak düşünmek gerekir. Örneği Pil Paketi 1’in eşdeğer direnç devresi Şekil 43’de gösterilmiştir.



Şekil 43: *Pil Paketi 1'in eşdeğer direnç devresi*

Bu durumda, eşdeğer direnç değeri aşağıdaki gibi hesaplanabilir.

$$\frac{1}{R_T} = \sum_{i=1}^6 \frac{1}{\left( \sum_{j=1}^4 R_j \right)_i} \quad (0.2)$$

Sonuçta elde edilen eşdeğer direnç değeri:

$$R_{T1} = \left( \frac{6}{4 \cdot 6.3} \right)^{-1} = \frac{25.2}{6} = 4.2 m\Omega$$

Benzer şekilde Pil Paketi 2 ve Pil Paketi 3 için elde edilen eşdeğer direnç değerleri aşağıda verilmiştir.

$$R_{T_2} = R_{T_3} = \left( \frac{3}{8 \cdot 6.3} \right)^{-1} = \frac{50.4}{3} = 16.8 m\Omega$$

Pil Paketi 1'den 1C, Pil Paketi 2 ve Pil Paketi 3'den 2C akım çekilirse, pil paketleri üzerinde harcanan güç aşağıdaki gibi olur.

Adı	Eşdeğer direnci(mΩ)	Çekilen akım (mA)	Güç tüketimi(W)
Pil Paketi 1	4.20	12000.00	0.60
Pil Paketi 2	16.80	12000.00	2.42
Pil Paketi 3	16.80	12000.00	2.42
		Toplam	5.44

Tablo 25: Pillerde harcanan güç değerleri

Bu durumda, su geçirmez bölümde harcanan toplam güç **56.02 Watt** olur.

#### 2.2.3.4 Su geçirmez bölümdeki donanım elemanlarının en yüksek sıcaklık değerleri

Su geçirmez bölüme yerleştirilecek olan donanım elemanlarının çalıştıkları en yüksek sıcaklık değerleri aşağıdaki tabloda verilmiştir.

Donanım Eleman Adı:	Sıcaklık aralığı (°C)	En yüksek sıcaklık (°C)
Parvus CPU-1462	-40 - 85	80
Parvus ACS-5150 DC-DC çevirici	-40 - 85	80
Parvus CTR-1461 PCMCI çevirici	-40 - 85	80
Parvus CTR-1475 MPEG 4 sıkıştırıcı	-40 - 85	80
Microstrain 3DM-GX1 IMU	-40 - 70	65
Z-World RCM 4000 Mikrodenetleyici	0 - 70	65
Diğer	-----	-----
Kokam 2000mAh LiPo hücreler	-20 - 60	55

Tablo 26: Su geçirmez bölümde bulunacak olan donanım elemanlarının çalışma sıcaklıkları

Bu durumda, eğer *Diğer* donanım elemanlarının çalışma sıcaklıkları 60 °C 'ın üstünde olursa, sualtı aracının su geçirmez bölümünün sıcaklığı 55 °C'ın altında olmalıdır.

#### 2.2.4 Donanım elemanlarının konektörleri

Su altı aracında su ile doğrudan temas edecek olan donanım elemanları Tablo 27'de verilmiştir.

Donanım Elemanının Adı:	Konektör Üreticisi:	Konektör Modeli:
Modem	SEACON	MCBH8M
Kamera	SEACON	RMK-5-FS
Aydınlatma	SEACON	RMG-2-FS
Sonar	Tritech	-----
İtici	-----	-----
Ultrasonik Çevirici	-----	-----
Basınç Almacı	-----	-----

*Tablo 27: Donanım elemanlarının üzerindeki konektörler*

Sonarın üzerindeki konektör, sonarın üreticisi tarafından üretilmektedir. Bu nedenle, sonar ile birlikte gelen kablo ve konektör kullanılmak zorundadır.

Su altı aracında kullanılacak olan su geçirmez kablo ve konektörler, sadece aracın su geçirmez bölümü ile Tablo 27'de gösterilen donanım elemanları arasında olacaktır. Bu nedenle, su geçirmez kabloların her iki ucunda da konektör bulunması gerekmektedir. Kabloların ucuna takılacak olan konektörlerin bir tarafında Tablo 27'de verilen konektör türünün karşılığı olan konektör, diğer tarafında ise bizim tarafımızdan bu dokümanda belirlenecek olan bir konektör bulunacaktır.

Tablo 28'de Tablo 27'de gösterilen konektörlerin karşılıkları verilmiştir.

Donanım Elemanının Adı:	Konektör Üreticisi:	Konektör Modeli:	Karşılığı:
Modem	SEACON	MCBH8M	MCIL8-F
Kamera	SEACON	RMK-5-FS	RMK-5-MP
Aydınlatma	SEACON	RMG-2-FS	RMG-2-MP
Sonar	Tritech	-----	-----
İtici	-----	-----	-----
Ultrasonik Çevirici	-----	-----	-----
Basınç Almacı	-----	-----	-----

Tablo 28: *Error! Reference source not found.*'de gösterilen konektörlerin karşılıkları

Tablo 28'de verilen konektör listesinde, modem ve iticilerin konektörleri ıslak ortamlarda da bağlanabilen mikro-ıslak konektör türünden; kamera ve aydınlatma konektörleri ise lastik kaplama ile su geçirgenliği engellenmiş konektörlerdir.

#### Gövdeye bağlanacak konektörler:

Su altı aracının mekanik tasarımı, denetim donanım elemanlarının sızdırmaz bir bölmenin içerisine yerleştirilmesini gerektirmektedir. Bu nedenle, denetim elemanları ile suyla temas eden donanım elemanlarının bağlantısını, bileşik malzeme kullanılarak üretilecek olan gövdenin içerisine kalıcı olarak monte edilmiş konektörler aracılığı ile sağlanması gerekmektedir. Tablo 29'de su ile temas edecek olan donanım elemanları listelenmiştir.

Çevresel donanım elemanının adı:	Adet:
Modem	1
İtici	4
Kamera	2
Aydınlatma	2
Sonar	1
Ultrasonik alıcı	2
Basınç almacı	1

Tablo 29: Su ile temas eden donanım elemanları

Islak ortamlarda da bağlanabilen konektör tipi, güvenilirlik ve çalışma ortamında herhangi bir fazladan müdahaleye (kurutma gibi) gerek kalmadan çalışma olanağı tanıdığı için bu tarz konektörlerin kabloların diğer uçlarında kullanılması avantajlı olmaktadır. Ayrıca, SEACON firmasının MCBH serisi konektörlerinin kablounun bağlandığı uçlarında bulunan vida yolu bileşik malzeme içerisine bu konektörü sızdırmaz bir şekilde yerleştirilmesine olanak sağlayacaktır. Dolayısıyla, gövde içerisine yerleştirilecek olan ve yukarıda anlatılan su

sızdırmaz kabloların uçlarına takılacak olan konektörler SEACON firmasının mikro-wet tipi konektörleri olacaktır.

Bu konektörlerin bağlanacak oldukları donanım elemanında bağımsız olarak gövdeye yerleştirebilmek amacı ile bu konektörlerin mümkün olduğunca aynı türden olmalıdır. Tablo 30'de su ile temas edecek olan her bir donanım elemanına gidecek kablo sayısı verilmiştir.

<b>Çevresel donanım elemanının adı:</b>	<b>Kablo sayısı:</b>
Modem	7
İtici	4
Kamera	3
Aydınlatma	2
Sonar	5
Ultrasonik alıcı	2
Basınç almacı	3

*Tablo 30: Her bir donanım elemanına gidecek kablo sayısı*

Bu durumda, modem hariç bütün donanım elemanları için 6'lı konektörler kullanmak doğru bir seçim olacaktır.

#### **Bütün konektör tipleri ve sayıları:**

Su sızdırmaz kablo uçlarına bağlanacak olan konektörler ve kablo adetleri Tablo 31 ve Tablo 31: Her iki ucuna da konektör takılacak kablo adetleri (parantez içinde önerilen sipariş adetleri), konektör tipleri ve konektörlerin etkin uç sayısı

'da verilmiştir.

<b>Donanım adı:</b>	<b>Donanım elemanına takılacak uç:</b>	<b>Gövde konektörüne takılacak uç:</b>	<b>Konektör takılmış kablo adedi:</b>	<b>Konektörün etkin uç sayısı:</b>
Modem	MCIL8-F	MCIL8-M	1(2)	7
Kamera	RMK-5-MP	MCIL6-M	2(5)	3
Aydınlatma	RMG-2-MP	MCIL6-M	2(5)	2

*Tablo 31: Her iki ucuna da konektör takılacak kablo adetleri (parantez içinde önerilen sipariş adetleri), konektör tipleri ve konektörlerin etkin uç sayısı*

<b>Donanım adı:</b>	<b>Donanım elemanına takılacak uç:</b>	<b>Gövde konektörüne takılacak uç:</b>	<b>Konektör takılmış kablo adedi:</b>	<b>Konektörün etkin uç sayısı:</b>
Sonar	-----	MCIL6-M	1	5
Ultrasonik çevirici	-----	MCIL6-M	3	6
Basınç almacı	-----	MCIL6-M	2	3
İtici	-----	MCIL4-M	4	2

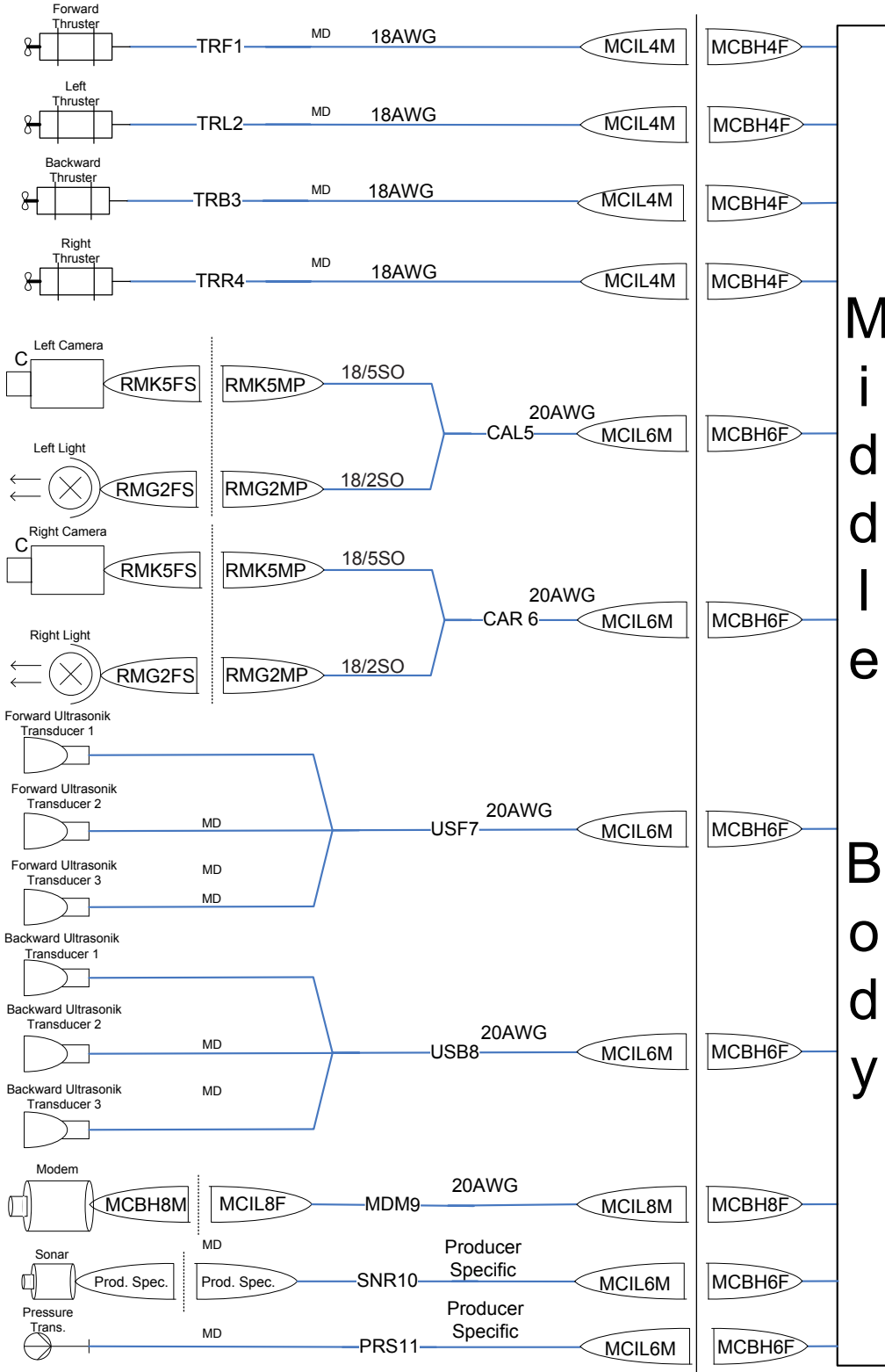
*Tablo 32: Tek bir ucuna konektör takılacak olan kablo sayısı*

Gövdeye yerleştirilecek olan konektör sayısı ise Tablo 33’de verilmiştir.

<b>Konektör Tipi:</b>	<b>Adet:</b>
MCBH8F	1
MCBH6F	7 +2
MCBH4F	4
SAPL-DS-0001-S1	1

*Tablo 33: Gövdeye yerleştirilecek konektör sayısı*

Tablo 33’de gösterilen +2 adet konektör, daha sonra gövdeye ilave edilebilecek olası almaçların bağlantısı için gerekli olabilecek yedek konektörleri ifade etmektedir. SAPL-DS-0001-S1 tipi konektör de fiber optik kablo için gerekebilecek konektördür.



Legend:

MD: Mounted on the device  
 Cable Name Cable Type

Şekil 44: Kablo bağlantı diyagramı



Şekil 44’de şu anda su altı aracında kesinlikle kullanılacak olan donanım elemanlarının konektör bağlantı diyagramı gösterilmiştir. Tüm cihazların, üzerlerindeki konektör tipleri, kablo çeşitleri ve gövdeyle yapacakları bağlantı biçimleri bu şekilden anlaşılabilir. Şekil 44’de MD cihazın üzerinde monte edilmiş olan kabloları (mounted on the device), *Prod. Spec.* ise cihazın üretici firması tarafından üretilen konektör ve/veya kabloları (producer specific) göstermektedir. Kabloların üzerinde gösterilen kısaltmalar ise kablo isimleridir. Bu isimler Tablo 34’da anlamlarıyla birlikte verilmiştir.

<b>Kısaltma</b>	<b>Anlamı</b>
TRF1	forward thruster - ileri itici
TRL2	left thruster - sol itici
TRB3	backward thruster - arka itici
TRR4	right thruster - sağ itici
CAL5	left camera&light - sol kamera ve ışık
CAR6	right camera&light - sağ kamera ve ışık
USF7	forward ultrasonic trans. - ön ultrasonik al.
USB8	backward ultrasonic trans. - arka ultrasonik al.
MDM9	modem - modem
SNR10	sonar - sonar
PRS11	pressure trans. - basınç almacı

*Tablo 34: Şekil 44’de kullanılan kablo isimleri ve anlamları*

MCBH-F serisi konektörlerin dış yüzeyi dişli olduğu için, kabloların MCIL-M serisi konektörlerin takılı olan uçlarına kilit maşonları (Locking Sleeve) gerekecektir.

Toplam konektör sayısı ise Tablo 35’de verilmiştir.

<b>Konektör Tipi:</b>	<b>Gerekli Adet:</b>	<b>Sipariş Adedi:</b>
MCIL4-M	4	15
MCBH4F	4	15
MCIL6-M	9	25
MCBH6F	9	25
MCIL8-F	1	5
MCIL8-M	1	5
MCBH8F	1	5
RMK-5-MP	2	5
RMG-2-MP	2	5
SAPL-DS-0001-S1	1	2

*Tablo 35: Toplam konektör sayıları ve önerilen sipariş adetleri*

Tablo 35’da verilen sipariş adetleri fiyatlara göre tekrar gözden geçirilmesi gerekmektedir. Ayrıca, Tablo 35’da kilit maşonlarının sipariş adetleri gösterilmemiştir. Tablo 34’de gösterilen kablo isimlerine göre, gerekli olacak konektör, kilit maşonları ve kablo tipleri Tablo 36’da özetlenmiştir.

Tablo 36’da gösterilen her bir kablo için gerekli olan konektör sayıları verilmiştir. Ayrıca, yine bu tabloda hangi konektörün bağlandığı taraf için kilit maşonunun gerektiği belirtilmiştir.

Kablo Adı:	Kol Sayısı:	Donanım Ucu Konektörü ve Sayısı:	Gövde Ucu Konektörü:	Gövdeye Yerleştirilecek Konektör:	Gerekli Kilit Maşonu Sayısı:	Kablo Tipi:
TRF1	1	-----	MCIL4M	MCBH4F	MCIL4M - 1	18AWG
TRL2	1	-----	MCIL4M	MCBH4F	MCIL4M - 1	18AWG
TRB3	1	-----	MCIL4M	MCBH4F	MCIL4M - 1	18AWG
TRR4	1	-----	MCIL4M	MCBH4F	MCIL4M - 1	18AWG
CAL5	2	Kam. RMK5MP - 1 Işık RMG2MP- 1	MCIL6M	MCBH6F	RMK5MP - 1 RMG2MP - 1 MCIL6M - 1	20AWG
CAR6	2	Kam. RMK5MP - 1 Işık RMG2MP- 1	MCIL6M	MCBH6F	RMK5MP - 1 RMG2MP - 1 MCIL6M - 1	20AWG
USF7	3	-----	MCIL6M	MCBH6F	MCIL6M - 1	20AWG
USB8	3	-----	MCIL6M	MCBH6F	MCIL6M - 1	20AWG
MDM9	1	MCIL8F - 1	MCIL8M	MCBH8F	MCIL8F - 1 MCIL8M - 1	20AWG
SNR10	1	PROD. SPEC.	MCIL6M	MCBH6F	MCIL6M - 1	20AWG
PRS11	1	-----	MCIL6M	MCBH6F	MCIL6M - 1	20AWG

*Tablo 36: Ayrıntılı konektör ve kablo tipleri*

Tablo 36 kullanılarak, SAECON ve SUASİS firmalarının yaklaşık maliyetleri aşağıdaki tablolarda verilmiştir.

Konektör Tipi:	Gerekli Adet:	SUASİS Fiyatları(\$):	Toplam:
MCIL4-M	4	44.28	177.12
MCBH4F	4	73.37	293.48
MCIL6-M	6	56.93	341.58
MCBH6F	6	96.14	576.84
MCIL8-F	1	75.90	75.9
MCIL8-M	1	69.58	69.58
MCBH8F	1	111.32	111.32
RMK-5-MP	2	59.46	118.92
RMG-2-MP	2	30.36	60.72
SAPL-DS-0001-S1	1	-----	-----
Kilit Maşonu	16	10.00	160
<b>TOPLAM:</b>			1985.46

*Tablo 37: SUASİS firmasından alınan fiyatlara göre yaklaşık maliyet (Dolar cinsinden)*

Konektör Tipi:	Gerekli Adet:	SEACON Fiyatları(€):	Toplam:
MCIL4-M	4	35.36	141.44
MCBH4F	4	46.08	184.32
MCIL6-M	6	52.40	314.4
MCBH6F	6	61.20	367.2
MCIL8-F	1	68.14	68.14
MCIL8-M	1	57.42	57.42
MCBH8F	1	75.33	75.33
RMK-5-MP	2	43.21	86.42
RMG-2-MP	2	37.55	75.1
SAPL-DS-0001-S1	1	-----	-----
Kilit Maşonu	16	7.00	112
<b>TOPLAM:</b>			1481.77

*Tablo 38: SEACON firmasından alınan fiyatlara göre yaklaşık maliyet (Euro cinsinden)*

Tablo 37 ve Tablo 38’de gösterilen maliyetlerde yedek kablolar gösterilmemiştir. Bu nedenle, kablo ismine göre yaklaşık maliyetlerin ayrı ayrı hesaplanması gerekmektedir.

## **3 Yazılım Bileşenleri Tasarımı**

### **3.1 İlk Tasarım**

Bu bölümde ULİSAR projesinin yazılım mimarisinin tanıtımı anlatılacaktır. Bu belge, tasarım detayları hakkında bilgi vermemesine karşın sistemin nasıl çalıştığı konusunda fikir verir. Yazılımın esnekliğini kısıtlayan en önemli ölçüt, akustik haberleşme bant genişliğinin kısıtlı olmasıdır. Bant genişliğinin kısıtlı olmasından dolayı sadece çok önemli bilgileri aktarma şansımız bulunmaktadır. Tüm sistem yazılımının ana fikri, her şeyi mümkün olduğunca en basit şekilde tutmaktır, çünkü kullanılması mümkün olmadığı durumlar hariç en güvenilir çözüm en basit olanıdır.

#### **3.1.1 Sualtı Aracı Yazılım Tasarımı**

Denizaltı bünyesinde iki ana işlem gerçekleşir, bunlar: denetim yazılımı ve görüntü işleme yazılımıdır. Takip edilen bölümlerde bu yazılımların yapısı açıklanacaktır. Bu iki uygulama birbiriyle TCP/IP iletişim kuralını kullanarak haberleşmektedir. Basitliğin sağlanması için, denizaltının yazılımında mümkün olduğunca az sayıda iş parçacığı olacaktır. İş parçacığı sayısı arttıkça beraberinde problemler getireceğinden bu sayede yazılımın sağlamlığı artacaktır.

##### **3.1.1.1 Denetim Bilgisayarının Yazılım Tasarımı**

Bu bilgisayarda, ana denetim yazılımı çalışmaktadır. Yazılımda ortaya çıkacak herhangi bir arıza, aracın arızalanmasına neden olacaktır. Bu bilgisayardaki yazılım zorunlu gerçek zamanlıdır. Bu işe, bilgisayardaki işlem sürecinde zamanın çok kritik olduğu anlamına gelmektedir. Yazılımın denetim döngüsü, daha önceden tespit edilmiş bir zamanda dönmelidir. Bu bilgisayarda gerçekleştirilecek işlemler sırasıyla: Algılayıcı veri okuması, Kalman filtre gerçekleştirilmesi, çözüm denetimi, seyir hesaplamaları, haberleşme işlemleri ve itici denetimleri olarak sayılabilir. Algılayıcı okuma işlemi Rabbit 4000 mikroişlemcisi tarafından yapılacaktır. Algılayıcılar okunduktan sonra, veri RS-232 yoluyla ilgili yerlere

gönderilecektir. Bu esnada ise algılayıcı okunma işlemi askıya alınacaktır. Denetim bilgisayarına bir paket ulaştığında, seri bağlantı noktası tarafından bir donanım iş kesme sinyali oluşturulur. İş kesmesi yaratıldıktan sonra, yazılım işlemlerini durdurur ve denetim ilgili ISR'a geçer. ISR ise kendi sürecini çok çabuk bitirmek zorundadır.

ULİSAR'da mevcut 3 adet işletim modu bulunmaktadır. Bunlar, Uzaktan Kumandalı Araç (RPV), Otomatik Pilot (AP) ve Acil Durum modlarıdır. RPV modunda araç bir işletmen tarafından denetlenmektedir. Otomatik pilot modunda ise araç, yüzeydeki işletmen tarafından daha önceden belirlenmiş geçiş noktalarını takip ederek ilerler. Acil durum modunda ise, RPV ve Otomatik pilot modları devre dışı kalır. Ancak acil durum ortadan kalktıktan sonra aracı diğer modlarda kullanabilmek mümkün olabilecektir. Aracın kullanım modunu belirlemek için mod düzenleyicisi kullanılacaktır. Mod düzenleyicisi ise hangi modun kullanılacağını, işletmen tarafından belirlenecek “Mod Bayrağı”na göre belirleyecektir. Bu mod bayrağı Rabbit 4000 mikroişlemcisinden gelecektir. Modun durumuna göre, durum makinası aracın durumunu belirleyecektir. Örneğin Otomatik pilotun durumları asılı kalma (hovering), dalma, yüzeye çıkma, v.b. olabilir. Her durumun kendi denetim yazılımı olacaktır.

Denetim bilgisayarı, denizaltının denetiminde, denetleme döngüsü kullanır. Denizaltıyı denetlemek için, yazılım Rabbit tarafından okunan algılayıcı bilgilerine ihtiyaç duyar. Seri veriyoluna ulaşan paket ulaştığında, ilgili seri veriyolunun ISR'ı yazılımın denetimi ele alır. ISR sadece veriyi kuyruğa yazacaktır. Denetim döngüsü bir turunu tamamlamadan önce, birden fazla veri ulaşabileceğinden sıralama işlemini kullanmaktayız. Sırala işlemi sayesinde veri kaybı ihtimali ortadan kalkar. Her veri kuyruğa yazılacaktır. Yazılım algılayıcı bilgisine ihtiyaç duyduğunda, öncelikle genel kuyruktan ilgili algılayıcının konumuna bakacaktır. Genel kuyruktan bir veri kullanımı sırasında, iş kesme sinyali oluşması ve verinin değişip hesaplamalarımızın hatalı olması ihtimaline karşı bu denetim işlemi gerçekleştirilmektedir. Kuyruk denetlendikten sonra boş konuma alınacaktır. Böylece yeni değerler kuyruğun en başından itibaren yazılmaya başlanacaktır.

Denetim döngüsünde yazılım ilk olarak algılayıcı bilgisini okumaktadır. Okuma işlemi, algılayıcı veri kuyruğunun yerel olarak işleyişinin denetlenmesine karşılık gelmektedir. Okuma işlemi tamamlandıktan sonra bu veriler Kalman filtresine girmelidir. Buraya kadar her

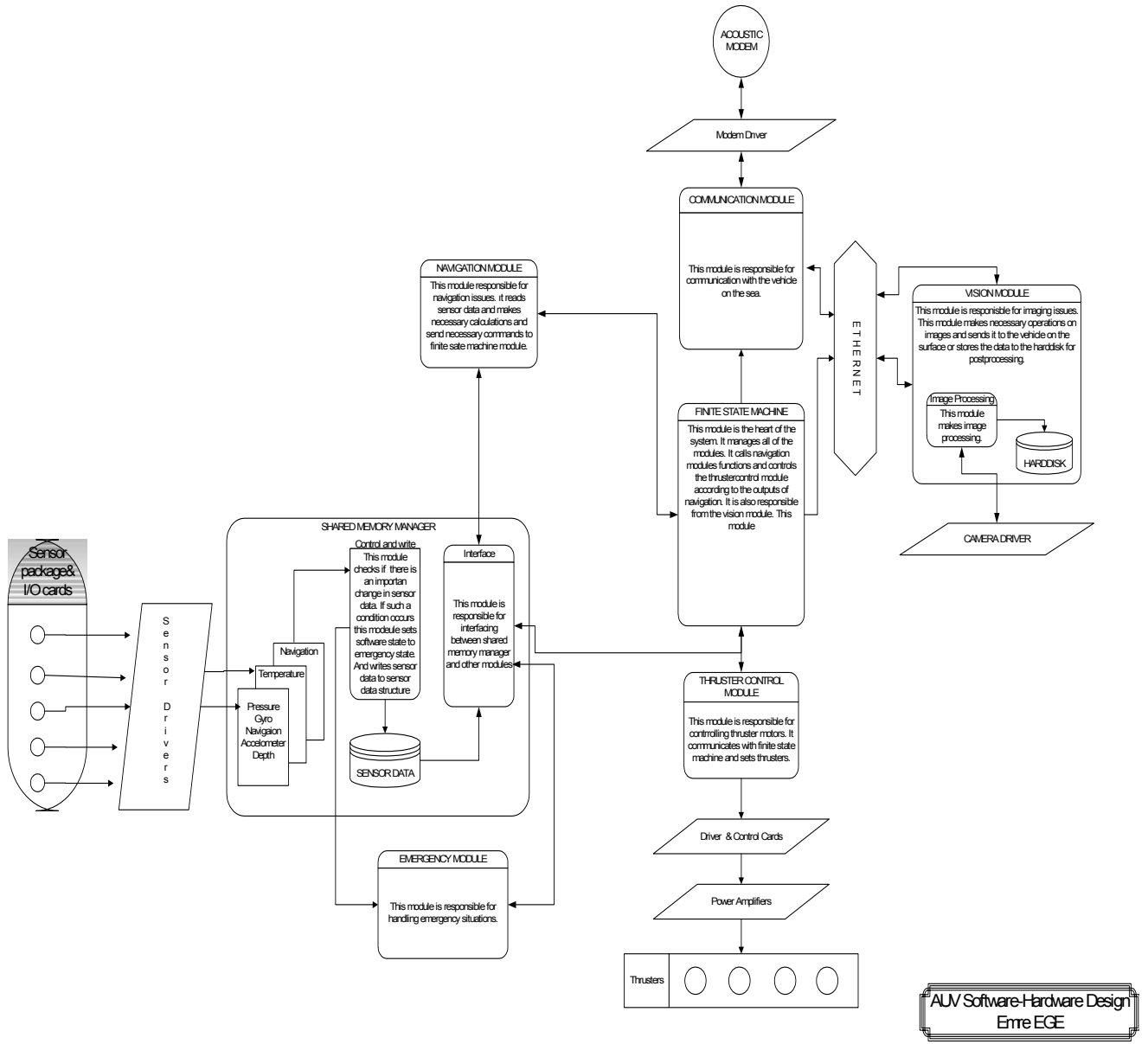
3 mod içinde işlemler aynıdır. Bu safhadan sonra moda göre işlem gerçekleştirilecektir. Sonuç olarak her moda göre, itki vektörleri hesaplanacaktır. Bu itki vektörleri ise itkilerin denetiminde kullanılacaktır. Vektörler hesaplandıktan sonra mikroişlemciye gönderilecek, oradan ise itkilere aktarılacaktır.

Tüm denetim işlemleri tek bileşenli olarak gerçekleşecektir. Ancak görüntü bilgisayarı ile haberleşmek için ayrı bir bileşene ihtiyaç duyacağız. Bu bileşen ethernet yolundan hem okuma hem de yazma yapabilecektir. Haberleşme bileşeni, okunmasına veya ethernet yoluna yazılmasına ihtiyaç duyulan yeni bir veri gelene kadar bekleyecektir.

Denetim bilgisayarı yazılım örnek kodu şu şekilde olacaktır:

```
InitOS();
InitSensors();
while(1){
    ReadSensors();
    KalmanFilter();
    Switch(ModeFlag);
    case AutoPilot:
        OperationsForAutoPilot();
    case Emergency():
        OperationsForEmergencyMode();
    case RemovePilotedVehicle:
        OperationsForRemoteControl();
    default:
    }
    SendDataToVSurf();
}
```

Anlatılan yazılım mimarisi Şekil 45 verilmiştir.



Şekil 45 Sualtı aracı yazılım mimarisi

## **3.2 Yeni Tasarım**

ULİSAR projesinde denizaltı ve operatör tarafında koşacak yazılımların daha önce geliştirilmiş ve şu anda da yine bir TÜBİTAK 1001 projesi olan SensorHex’de de kullanılan RHexLib kullanarak geliştirilmesine karar verilmiştir. Bu bölümde şu anki kullanılan yazılımlardan bahsedilecektir.

### **3.2.1 Denizaltı Aracı Denetim Bilgisayar Yazılımı**

ULİSAR projesi kapsamında kullanılacak olan denizaltı modülünde 2 adet bilgisayar bulunmaktadır. Bunların bir tanesi denizaltı aracının haberleşme, kontrol, güdüm gibi kritik işleri yapacak olan “denetim bilgisayarı”, diğeri ise sonar ve kamera görüntüleri ile ilgili işleri yapacak olan “görüntü bilgisayarı”dır. Bu bilgisayarların ve diğere çevre birimlerinin bağlantı şemaları önceki raporlarda sunulmuştur.

Denetim bilgisayarı denizaltı aracını hayati önem taşıyan işlerini yapacağı için bu bilgisayardaki koşacak olan yazılım çok kritiktir. Özellikle denetim ve güdüm gibi zaman kritik işlemler bu bilgisayarda yapılacağı için, bilgisayardaki işletim sisteminin gerçek zamanlı bir işlerim sistemi olması gerekmektedir. Gerçek zamanlı işletim sistemleri istenilen komutları aynı zamanda yapar veya eğer bir gecikme olarsa bu gecikme önceden belirlidir. Dolayısıyla gerçek zamanlılık, zaman kritik işlemlerde bulunması gereken çok önemli bir özelliktir.

Denetim bilgisayarında QNX işletim sistemi kullanılmaktadır. ULİSAR projesinin başlangıç aşamasında QNX ücretli bir işletim sistemiydi. Ancak eğitim amaçlı kullanım için belirli bir sayıda ücretsiz lisans veriliyordu. Proje devam ederken QNX işletim sisteminin ücretsiz olarak dağıtılacağı açıklandı. Şu anda denetim bilgisayarında QNX işletim sisteminin son sürümü olan versiyon 6.3.2 kullanılmaktadır.

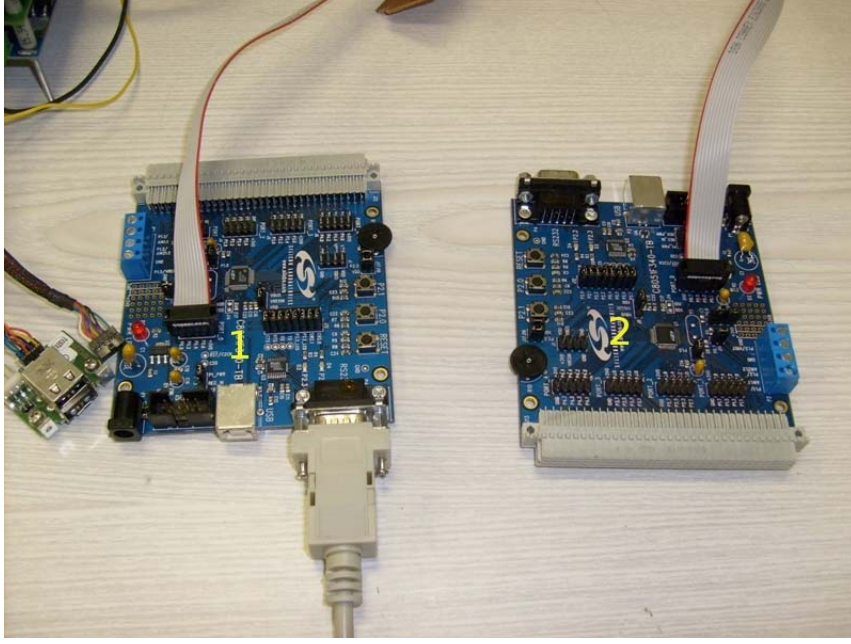
Denetim bilgisayarında aynı anda birçok işlem yapılacaktır ve bu işlemlerin gerçek zamanda yapılabilmesi gerekmektedir. Bunun yanında kullanılan yazılımın modüler olması gerekmektedir. Kullanılacak olan yazılıma istenilen özellikler yazılım bittikten sonra dahi kolay bir şekilde eklenip çıkartılabilmelidir.



Denetim bilgisayarı yazılımında yukarıda bahsedilen özelliklere sahip olan RHexLib kütüphanesi kullanılmaktadır. RHexLib yine bir TÜBİTAK 1001 projesi olan SensoRHex projesinde geliştirilmiştir. RHexLib in yapısı bölüm 3.2.1.5 de verilmiştir. RHexLib kütüphanesinde çeşitli modlar ve modüller tanımlanarak bu modların ve modüllerin istenilen şekilde ve periyotda kullanılması sağlanmaktadır.

ULİSAR ve SensoRHex projesinde ortak olan işler bulunmaktadır. Bu sebeple iki projenin ortak işleri birlikte geliştirilmektedir. Bu işlerden en önemlilerinden birisi bilgisayarın çevre birimlerle haberleşmesini sağlayan URB(Universal Robot Bus) dır. URB ile ilgili teknik bilgiler ekte verilmiştir. URB, ULİSAR ve SensoRHex proje grupları tarafından ortaklaşa geliştirilmiştir. URB yapısı sisteme kolay bir şekilde çevre birimleri ekleyip çıkarma olanağı sağlamaktadır.

URB temel olarak 3 öğeden oluşmaktadır. Bunlar bilgisayar, köprü (bridge) ve düğümdür (node). Bilgisayar denetim yazılımının çalıştığı ve çevre birimlere komut gönderen/alan kısımdır. Bridge bilgisayardan alınan komutları düğümlere yollar, düğümden aldığı bilgileri de bilgisayara yollar. Düğümler çevre birimlerin bağlı olduğu mikroişlemcilerdir. Köprüden aldığı komutları işler ve eğer bir cevap isteniyorsa cevabı hazırlayarak köprüye yollar. Şekil 46 da köprü ve düğüm resimleri bulunmaktadır. Şekilde 1 numaralı birim köprü, 2 numaralı birim düğümdür. Şekilde gözüken kartlar 8051F340 mikroişlemcisinin geliştirme kitidir. Köprü ve düğümler birbirine I<sup>2</sup>C veriyolu üzerinden bağlanmaktadır. Köprü ve düğüm arasındaki bağlantı kablosu I<sup>2</sup>C veriyoludur.



*Şekil 46: URB Köprü ve düğüm*

### **3.2.1.1 Denetim Yazılımı Geliştirme Süreci**

Bu bölümde denetim yazılımı geliştirme süreci ve şu anki durumu hakkında bilgi verilecektir. Projenin en kritik aşamalarından birisi denetim yazılımıdır. Denizaltı aracının işlevselliğinin önemli bir kısmının gerçekleştirileceği yer denetim yazılımı kısmıdır. Çevre birimlerle haberleşecek olan yazılım denetim yazılımının önemli bir kısmıdır.

Denetim bilgisayarı çevre birimlerle USB(Universal Serial Bus) üzerinden haberleşecektir. USB sürücüsünün geliştirilmesi daha uzun bir süreç olduğu için ilk aşamada RS-232 kullanılmıştır. Önceki bölümde de söylendiği üzere denetim bilgisayarı çevre birimlerle URB sayesinde haberleşecektir. Bilgisayar ve çevre birimler arasında komutları taşıyacak olan birim köprüdür. İlk olarak köprü ile denetim bilgisayarı arasında RS-232 üzerinde haberleşmeyi sağlayacak olan RS-232 sürücü yazılımı geliştirilmiştir. Bu yazılım bilgisayardan köprüye komut gönderebilmekte ve köprüden gelen bilgileri alabilmektedir. Denetim yazılımı çevre birimlerle haberleşmesini geliştirilen RS-232 sürücüsü üzerinden gerçekleştirmektedir. Daha sonra köprü kartlarına birer USB-Serial dönüştürücü çip konulmuştur.

Köprü devresi ile bilgisayar arasında USB veriyolu kullanılmıştır. USB kullanılmasının en önemli nedenlerinden birisi kontrol bilgisayarında kullanılan bilgisayarda 6 adet USB çıkışı

bulunmasıdır. Köprü devresi üzerinde USB-Serial dönüştürücü kullanılmıştır. Kontrol bilgisayarında yüklü olan QNX işletim sisteminde USB-Serial cihazının sürücüsünün adı “devc-serusb” dir. Bilgisayar açılınca işletim sistemi yüklenirken bu sürücüsünde yüklenmesi sağlanmıştır. Sürücü yüklendikten sonra /dev/ klasörünün altında kaç adet USB-Serial dönüştürücü donanımı varsa o kadar sayıda device(cihaz) dosyası oluşturulur. Bu dosyaların adı “serusbNO” dur .(Örn. serusb1, serusb2). Sürücünün yüklenmesinden sonra bu cihaz dosyaları normal seri haberleşme kullanılır gibi kullanılabilir.

Aşağıdaki bölümlerde şu anda denetim bilgisayarında kullanılan üst seviye ve alt seviye yazılımlar anlatılmaktadır.

### 3.2.1.2 URB Denetim Bilgisayarı Tarafı Yazılımı

URB bilgisayar tarafı yazılımı kullanıcıyı URB detaylarını bilmeksizin çevre birimlerle haberleşmesini sağlamak üzere tasarlanmıştır. Kullanıcı bu yazılımı kullanarak istediği çevre birime istediği komutu gönderebilmektedir. URB sürücüsü bir veya birden fazla köprüyü yönetebilmektedir. Kullanıcı hangi çevre birimin hangi köprüye bağlı olduğundan bağımsız olarak sadece çevre birimin kimlik bilgisi ile çevre birime ulaşabilmektedir. URB bilgisayar tarafı yazılımı bu özelliği sayesinde kod geliştiriciye büyük bir kolaylık sağlamaktadır.

URB bilgisayar tarafı yazılımı nesne yönelimli (object oriented) yazılım tekniği kullanılarak yazılmıştır. URB arayüzünü kullanacak olan kullanıcı URB kütüphanesindeki URBIInterface sınıfını kullanarak istediği çevre birime erişebilmektedir. URBIInterface sınıfının tanımı aşağıdaki kod parçacığında verilmiştir.

```
/*
*****
* CPU Application Programmer's Interface (API) for URB access
*****
*/
#ifndef _URBINTERFACE_HH
#define _URBINTERFACE_HH

#include <pthread.h>
#include <string.h>
#include "urb/Request.hh"
#include "urb/BusManager.hh"
#include "utils/List.h"

__URB_BEGIN_NAMESPACE
/** \ingroup URBCPUGroup
    This class provides a unified interface to URB functionality for
    all buses connected to the computer. It is designed a singleton

```

```

class (i.e. only one instance can be created), accessed through
the \ref instance method.
*/
class URBInterface {

public:

~URBInterface() {};

/** External calls to the method \ref addBusManager are used to
inform URBInterface of the presence of \ref BusManager class
instances in the system. The URBInterface then initiates discovery
of nodes (possibly located on several different buses attached to
the same manager) controlled by the new manager.
*/
void addBusManager( BusManager *manager );

/** This method goes through all BusManager instances known to the
URBInterface and attempts to locate the node with the class and
index supplied in the node info argument. If no such node is
found, NULL is returned.
*/
NodeAccessor* findNode( node_info_t *info );

/** This method returns the singleton instance of the URBInterface
class, created by the first call to this method. */
static URBInterface* instance( void );

protected:
// Protected constructor is only called from within the instance method */
URBInterface( void );

private:
utils::ManagedList<BusManager> _managers; // List of registered bus managers
utils::ListIterator<BusManager> _iter; // Iterator for the manager list
static URBInterface *_singletonInstance; // Pointer to the singleton instance
};

__URB_END_NAMESPACE

#endif

```

URB arayüz sınıfı kullanılırken başlangıçta ilk olarak `addBusManager` metodu kullanılarak köprü veya köprülerin bağlı olduğu veriyolları eklenir. Bu veriyolları RS-232, USB, PCI, vs. olabilir. Şu anda RS-232 veriyolunu kullanmak için gerekli olan kod yazılmış durumdadır. Dolayısıyla bu aşamada çevre birimlere RS-232 veriyolu kullanılarak erişilmektedir. Bir veri yolu eklendiğinde URB arayüz sınıfı o veriyolu üzerinde bağlı olan çevre birimleri (düğüm) bulur ve bu çevre birimlerin kimliklerini kendi içinde bir listede saklar. Kullanıcı bu veriyollarını ekledikten sonra herhangi bir çevre birime ulaşmak için ilk olarak URB arayüzünü sınıfının bir örneğini (instance) alır. Bu örnek `instance` metodu ile elde edilir. Daha sonra `findNode` metodunu ulaşmak istediği çevre biriminin(düğüm) kimlik bilgileri ile çağırır. Eğer verilen kimlik bilgilerine sahip bir çevre birim varsa URB arayüz sınıfı bu çevre birime erişebilmek için kullanıcıya `NodeAccessor` türünde bir işaretçi döndürür. Kullanıcı bu işaretçiyi kullanarak istediği komutu çevre birime gönderebilir.

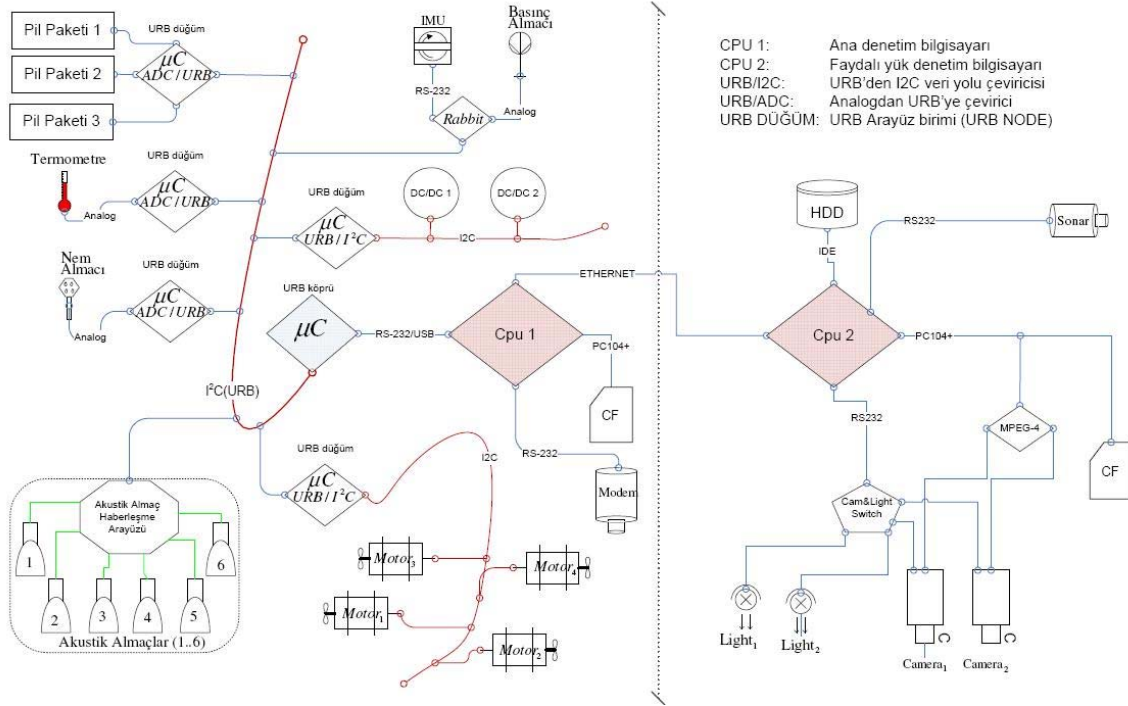
Kullanıcı bir çevre birime bir komut gönderdiğinde o çevre birimden herhangi bir cevap bekleyip beklemediğini de bildirir. Eğer bir cevap gerekli ise çevre birimden gelen cevap uygun şekilde paketlenerek kullanıcının belirlediği bir fonksiyon (callback function) çağırılır ve hazırlanan paket fonksiyona parametre olarak verilir.

URB arayüz sınıfı her ne kadar kullanıcının kolay kullanımını sağlamak üzere yazılmış olsa dahi yukarıda belirtilen işler üst seviye yazılım (örneğin kontrol veya güdüm) geliştiren kişiler için yazılım geliştirmesi sürecini uzatacak durumdadır. Dolayısıyla üst seviye yazılım geliştirenler için bir ara katman daha gerekmektedir. Bu ara katmanda RHexLib kütüphanesinde kullanılan Abstract Hardware Class(Soyut Donanım Sınıfı (SDS)) dır. SDS sayesinde üst seviye yazılım geliştirici daha önceden her bir çevre birim için tanımlanmış olan SDS ile çevre birimlere ulaşabilmektedir.

### **3.2.1.3 Soyut Donanım Sınıfları**

SDS ler soyut bir şekilde çevre birimle haberleşmek için hazırlanmıştır. Kullanıcı daha önceden belirli bir çevre birimi için yazılmış olan SDS in bir örneğini(instance) kullanarak çevre birimle haberleşir.

ULISAR projesinde birçok çevre birimi vardır. Denizaltı aracı donanım şeması Şekil 47 de verilmiştir.



Şekil 47: ULİSAR Elektronik Bağlantı Şeması

Şekil 47 de görüldüğü üzere denetim bilgisayarı birçok çevre birimle haberleşmektedir. Her bir çevre birim için birer tane SDS yazılabileceği için benzer çevre birimler için sadece bir tane SDS de yazılabilir. Örneğin denizaltı altı aracının üstünde kameralar ve sonar cihazın kapsama alanı dışında kalan bölgelerde bir engelle çarpmamak için 6 adet ultrasonik mesafe algılayıcısı bulunmaktadır. Tüm ultrasonik mesafe algılayıcıları için ayrı ayrı 6 adet SDS hazırlanabileceği gibi tek bir SDS hazırlanarak 6 alıcının numaraları ile birlikte de kullanılabilir.

ULİSAR projesi kapsamında şu ana kadar URB arayüzü ile uyumlu olarak çalışan 3 adet SDS yazılmıştır. Bunlardan bir tanesi Sanal Donanım Soyut Sınıfıdır. Bu sınıfın yazılma amacı herhangi bir donanım olmadan da yazılım geliştirecek kişinin işini yapabilmesidir. Sanal donanım soyut sınıfı temel olarak sanki bir donanım varmış gibi davranır ve kullanıcıya belirli mesajlar gönderir. Böylece kullanıcı kendi geliştirdiği yazılımın doğru çalışıp çalışmadığını bilgisayara herhangi bir çevre birimi takmadan da anlayabilir.

Yazılan ikinci SDS test düğüm soyut sınıfıdır. Test düğümü, URB protokolü geliştirilirken oluşturulmuş, içinde bazı işlevlerin gerçekleştirildiği bir yazılım bulunan 8051F340 mikroişlemcisidir. Bu mikroişlemci geliştirme kitiyle birlikte kullanılmaktadır. Geliştirme kiti ile birlikte kullanmanın temel amacı yazılan programın doğru çalışıp çalışmadığını anlamak ve eğer hatalar varsa bunları düzeltmektir. 8051F340 geliştirme kiti çevirimiçi hata ayıklama özelliğine sahip olduğundan dolayı, yazılım geliştirme aşamasında büyük bir öneme sahiptir. Test düğüm soyut sınıfı sayesinde bilgisayarda çalışan yazılımla 8051F340 geliştirme kiti üzerindeki ışıklar açılıp kapatılabilmekte, geliştirme kiti üzerinde bulunan Analog-Sayısal Çevirici (Analog-to-Digital Converter) değeri okunabilmektedir. Böylece bilgisayardan test düğümüne çift yönlü (hem yazma hem okuma) bir iletişim sağlanmış olmaktadır. Aşağıda test düğümü için yazılan soyut sınıfın tanımı verilmiştir.

```
#ifndef ULISARTESTNODE_HH_
#define ULISARTESTNODE_HH_
#include "TestNodeHW.hh"
#include "URBInterface.hh"
#include "Request.hh"
#include <stdio.h>

#define TEST_NODE_CLASS 0x45
/*
 * Ulisar Test node hardware abstract class.
 */

class UlisarTestNode : public TestNodeHW {
public:
    UlisarTestNode();
    ~UlisarTestNode() {};

    void enable( uint index ){};
    void disable( uint index ) {};
    /* This methos reads the data from ADC on the test node
     * and stores the read value into Data variable.
     */
    void readADC( void);

    /* This method returns the lastly read data.
     * This method must be called after a readADC method call.
     */
    unsigned char getData(void);

    /* This method sets or resets to leds on the test node.
     * An input argument set the led if 1, or resets if 0.
     * First input argument is led 1 command, and the second is
     * second led command on the test node.
     */
    void LedTest(int LedStatus_1, int LedStatus_2);

    /* This method resets the range sensor that is specified by the
     * input argument index.
     */
    void reset( uint index ) {};

private:
    /*
     * URBInterface instance. This variable is used to
     * access to nodes on the URB. It is set within the constructor
     */
};
```

```
urb::URBInterface * urbintf;  
  
urb::byte TestNodeID;  
};  
#endif /*ULISARTESTNODE_HH_*/
```

Üçüncü SDS ultrasonik mesafe algılayıcıları için yazılmıştır. Denizaltı aracında 6 adet ultrasonik mesafe algılayıcı bulunmaktadır. Bu algılayıcılar belirli periyotlarda denetim yazılımı tarafından okunmaktadır. Eğer belirli bir mesafenin altında bir engel algılanırsa, denetim yazılımı o engelden uzaklaşmak üzere gerekli işlemi yapmalıdır. 6 adet ultrasonik algılayıcı olduğundan dolayı 6 adet SDS yazmaya gerek yoktur. Aynı işlemi bir adet SDS yazarak da yapabiliriz. Yazılan SDS hangi ultrasonik mesafe algılayıcı ile haberleşmek istiyorsak o algılayıcıya mesaj gönderir.

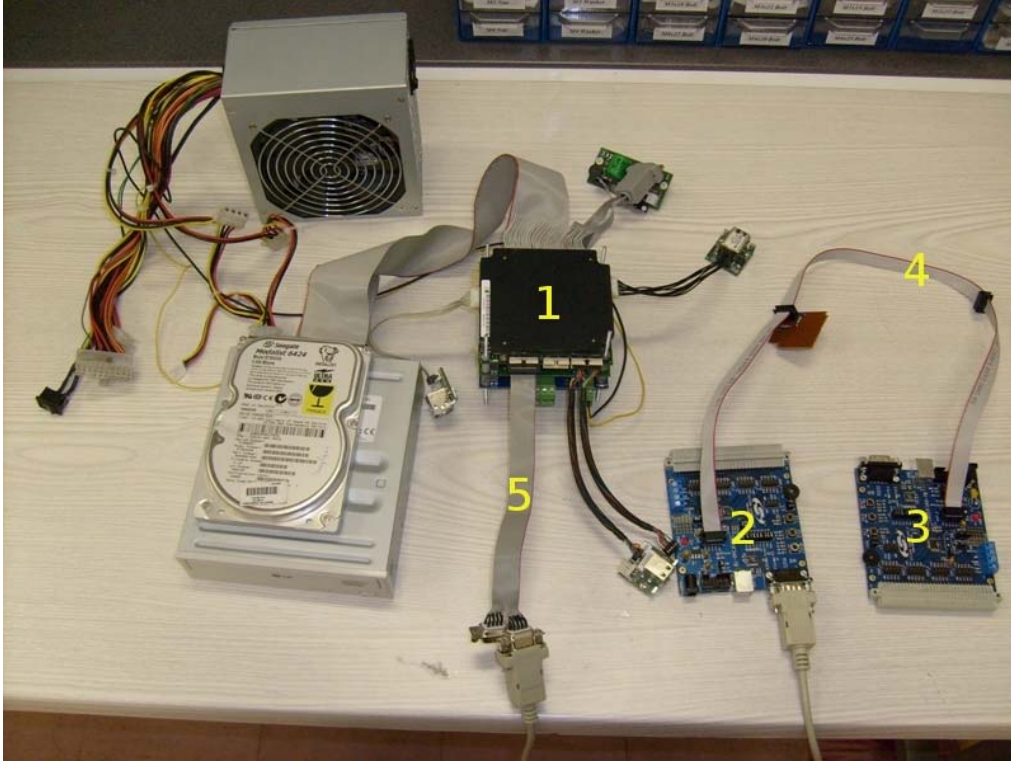
Denetim bilgisayarında çalışacak olan denetim yazılımı projenin en önemli kısımlarından bir tanesidir. Denizaltı aracının en kritik işlerinin yapılmasından sorumlu olan kısım bu yazılımın içerisinde olacaktır. Birçok işlevi bulunacak bir yazılımın geliştirilmesinin hızlı ve hata ayıklama sürecinin kısa olması için modüler bir yapıya sahip olması gerekmektedir. Dokümanın başında da belirtildiği üzere denetim bilgisayarındaki yazılım RHexLib kütüphanesi kullanılarak geliştirilmektedir. Yapılan tasarımın sorunsuz şekilde çalışabildiğini görmek için temel olarak kullanılacak olan çevre birimleri ve denetim yazılımının basit fakat tasarımın doğruluğunu ispatlayacak bir deney düzeneği kurulması gereklidir. Bu bölümde yukarıda bahsedilen tasarımın sorunsuz çalıştığının gösterildiği deney düzeneği anlatılmıştır.

Denetim bilgisayarında çalışacak olan yazılımda RHexLib kütüphanesi kullanılacağından dolayı yazılım içinde modlar ve modüller tanımlanması gerekmektedir. Hazırlanan deney düzeneği denetim bilgisayarı, denetim bilgisayarına RS-232 veri yolu ile bağlanmış köprü ve bu köprüye bağlanmış test düğümünden oluşmaktadır. Tasarım gereği eğer bu deney düzeneğini başarılı bir şekilde çalıştırabilirsek sistemin büyük oranda sorunsuz olarak çalışacağını göstermiş oluruz. Deney düzeneğinin amacı denetim bilgisayarı üzerinde çalışacak olan RHexLib kullanılarak yazılmış bir yazılımın çevre birim olan test düğümüne veri gönderip alabildiğini görmektir. Bu deney bize şu ana kadar tasarlanan her şeyin doğru çalıştığını ve bundan sonraki geliştirmelerin daha hızlı bir şekilde yapılabileceğini göstermektedir.

Denetim bilgisayarı üzerinde QNX işletim sistemi bulunmaktadır. Denetim yazılımı Linux platformunda geliştirilmekte olup cross-compilation araçlarıyla QNX işletim sistemine uygun



çalışan dosya üretilmektedir. Daha sonra oluşturulan bu dosya denetim bilgisayarına “SSH” aracılığıyla yüklenip orada çalıştırılmaktadır. Deney düzeneği için yazılan denetim yazılımında 2 adet modül bulunmaktadır. Bu modüllerden biri periyodik bir şekilde test düğümünden veri okurken diğer ise farklı bir periyotta test düğümü üzerinde bulunan LED leri yakıp söndürmektedir. Böylece RHexLib-URB entegrasyonu başarılı bir şekilde gerçekleştirilmiştir. Şekil 48 de hazırlanan test düzeninin fotoğrafları verilmiştir.



Şekil 48. Deney Düzeneği

Şekil 48 de verilen test düzeneğindeki numaralandırılmış birimlerin açıklamaları Tablo 39 da verilmiştir.

Numara	Açıklama
1	Denetim Bilgisayarı
2	Köprü
3	Test Düğümü
4	I <sup>2</sup> C Veriyolu
5	RS-232 Veriyolu

Tablo 39: Test Düzeneğinde Kullanılan Elektronik Birimler

Şekil 48de verilen test düzeneğindeki denetim bilgisayarında çalışan modüllerden biri olan ve test düğümü üzerindeki LED leri açıp kapatan modülün tanımı aşağıdaki kod parçacığında verilmiştir. RHexLib kütüphanesi her modülün update metodunu verilen periyotta çağırılmaktadır.

```
#ifndef TESTNODEMODULE_HH_
#define TESTNODEMODULE_HH_
#include "ModuleManager.hh"

/* TestNodeModule.
 * This class handles test module read,write operations.
 */
class TestNodeModule : public Module {
public:

    TestNodeModule ( int index ) : Module( "TestNode", index, false ) { };
    void init( void ) { };
    void uninit( void ) { };
    void activate( ) { };
    void deactivate( ) { };
    void update( void );
};
#endif /*TESTNODEMODULE_HH_*/
```

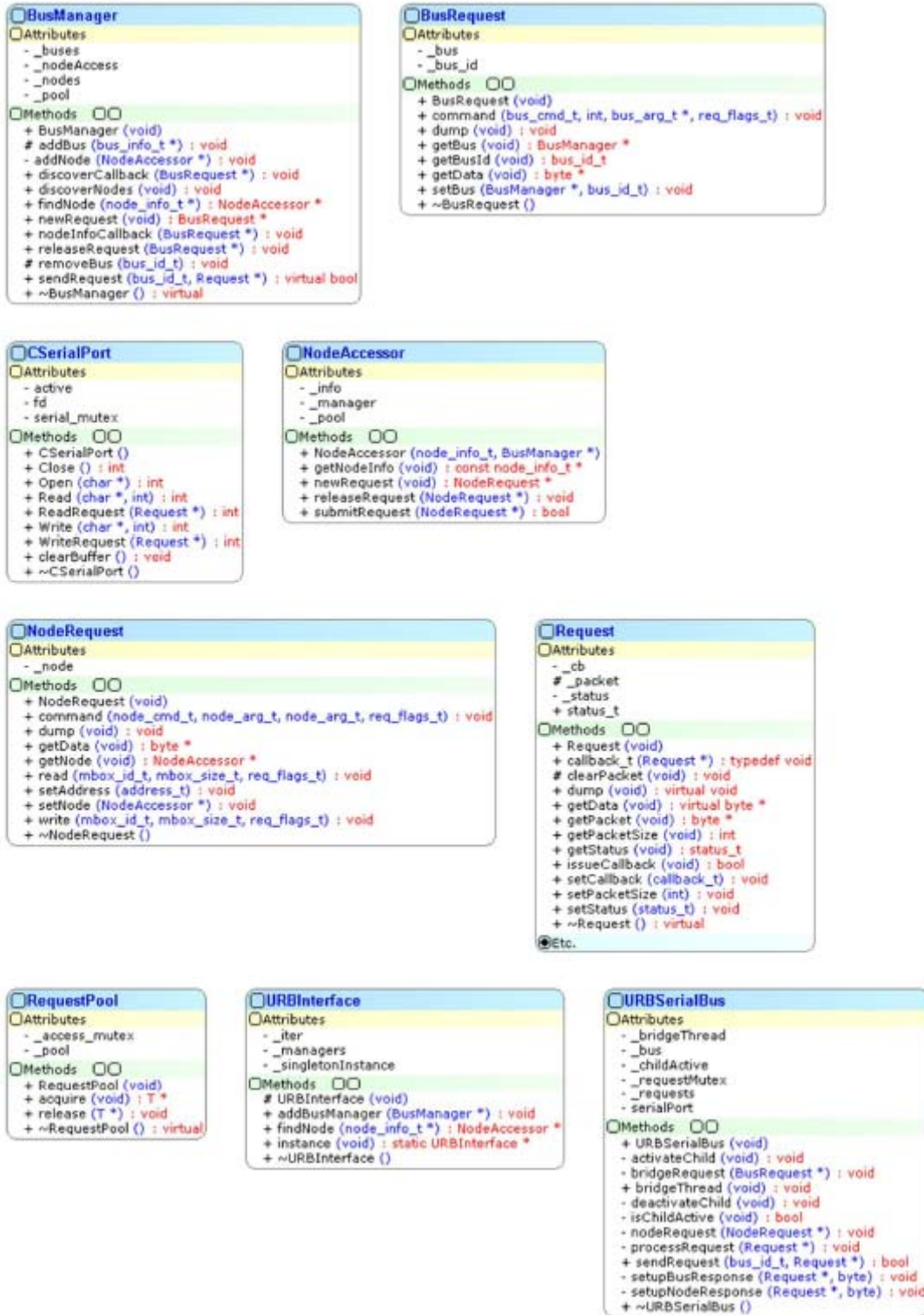
### 3.2.1.4 URB Protokolü Bilgisayar Tarafı Yazılımı Sınıf Diyagramı

Bu bölümde bilgisayar tarafın URB protokolü yazılımının sınıf diyagramları verilmiştir. Bu diyagramlarda, yazılan sınıflar ve bu sınıfların birbirleriyle olan ilişkileri gözükmektedir. URB bilgisayar tarafı yazılımının sınıflarının birbirleriyle olan ilişkileri Şekil 49 da verilmiştir. Bu şekilde hangi sınıfların hangi sınıflardan türediği gösterilmektedir.



Şekil 49 URB Bilgisayar Tarafı Yazılımındaki Sınıf İlişkileri

Şekil 50 de URB bilgisayar tarafı yazılımı sınıflarının tanımları verilmiştir.



Şekil 50: URB Bilgisayar Tarafı Yazılımı Sınıf Tanımları

### 3.2.1.5 RHexLib Kütüphanesi

RHexLib kütüphanesi RHex robotunda çalışmak üzere yazılmış yazılım parçacıkları ve modülleri topluluğudur. İçinde alt seviye donanım erişim modülleri, hareket kontrol işlevselliği ve üst seviye denetimciler mevcuttur. RHexLib kütüphanesi modüler ve yapılan donanım değişikliklerinde en az kod değişikliği yaparak çalışacak biçimde tasarlanmıştır. RHexLib kütüphanesinde aşağıdaki listede verilen parçalar bulunmaktadır. Bu parçaların işlevsellikleri yanlarında kısaca özetlenmiştir.

- Modül Yöneticisi : Periyodik görevler için durağan zamanlayıcı
- Sembol Tablosu Özellikleri: Konfigürasyon dosyalarının yönetimi ve konfigürasyon sembollerine erişim
- Veri Saklama Araçları : Çalışma zamanı veri toplama ve saklama yönetici modülü
- Durum Makinesi Altyapısı : Durum makineleri tanımlama ve çalıştırma mekanizmaları
- Donanım Erişim Altyapısı : Alt seviye donanımlara erişim soyut sınıfı

RHexLib kütüphanesindeki en önemli özelliklerden bir tanesi modül yöneticisidir. RHexLib kütüphanesi içinde modül belirli bir işi belirli bir periyotta yaptırmak için yaratılır. Belirli bir görev için yazılan modül RHexLib kütüphanesindeki “Module” sınıfından türetilir. Bir modülde temel olarak *init*, *uninit*, *activate*, *deactivate* ve *update* metodları bulunur. *init* metodu modülü çalıştırmaya başlamadan önce yapılması gereken işleri yapar. *uninit* metodu modülün çalışması sonlanacağı zaman yapılması gereken işleri yapar. *activate* ve *deactivate* metodları modülü çalıştırmaya ve durdurmaya yarar. Bir modüldeki en önemli metod *update* metodudur. Bir modül çalışırken kullanıcının belirlediği sıklıkta modülün *update* metodu çağırılır. *update* metodu içinde bir modülün ne yapması isteniyorsa onun için gereken kod yazılır.

### 3.2.1.6 Denetim Bilgisayarı Yazılımlarının Derlenmesi

Denetim bilgisayarı işletim sistemi ve çalıştırılabilir yazılımı CF karta yüklenerek çalıştırılmaktadır. ULİSAR projesi kapsamında geliştirilen denetim bilgisayarı yazılımı

aşağıda adresi verilen depoda tutulmaktadır. Bu depodan dosyaların indirilebilmesi için depo yöneticisi tarafından verilen hesap adı ve şifre kullanılmalıdır. Deponun adresi:

```
svn://pcsaranli.cs.bilkent.edu.tr/projects
```

Dosyalar depodan indirildikten sonra aşağıdaki ayarlar yapılmalıdır. ULİSAR yazılımını derlerken bazı çevresel değişkenler kullanılmıştır. Bunları her seferinde tanımlamamak için /PATH/RoboDevel altında setvars.sh adlı bir dosya ile herhangi bir konsol açtığımızda bunların otomatik olarak tanımlanması sağlanmıştır. Bunu yapabilmek için /home/user/ dizini altındaki .bashrc dosyasına ekteki komutlar eklenmelidir.

```
## optional : will default to $USER if you do not define here
export ROBODEVEL_USER=username
#
## required : where local copy will reside
export ROBODEVEL_DIR=/PATH/RoboDevel
export ULISAR_TOPDIR=/PATH/Ulisar/Underwater/ControlComputer
export URB_DIR=/PATH/urb/trunk/cpu_util
#
## required : target platform for compilation
## valid choices are linuxx86,
qnx6x86
export ROBODEVEL_TARGET=linuxx86
#
## you can individually choose robot hardware types
## if you do not specify, that robot is not compiled
export ULISAR_HARDWARE=_ULISAR_
## last line required
to source this file
. $ROBODEVEL_DIR/setvars.sh
```

ROBODEVEL\_DIR çevre değişkeni ile tanımlı klasöre gidilip “make” komutu yazıldığında çalıştırılabilir dosya oluşturulmalıdır. Çalıştırılabilir dosya /PATH/Ulisar/Underwater/ControlComputer/Demo/Supervisor dizini altında “main” adındaki

dosyadır. Bu dosya scp komutu ile denetim bilgisayarına yüklenmelidir. Buradaki PATH depodan dosyaları indirdiğimiz klasör adıdır.

Denetim bilgisayarında 1. USB ye köprü kartı üzerindeki 1. köprü bağlanmalıdır. 2. USB girişine ise köprü kartındaki 2. köprü bağlanmalıdır. Denetim bilgisayarındaki Ethernet çıkışlarından bir tanesi görüntü bilgisayarına bir tanesi ise fiberoptik-ethernet dönüştürücüye bağlanmalıdır. Denetim bilgisayarının 1. seri port çıkışına akustik modem bağlanmalıdır. Akustik modem şu anda kullanılmamakla birlikte ileride kullanılacaktır. Deneysel çalışmalarda iletişimin sorunsuz sağlanabilmesi için akustik modemin yanı sıra fiberoptik kablo bağlantısı kullanılmaktadır. Denetim bilgisayarı ile operatör bilgisayarı arasındaki bağlantı fiberoptik kablo ile gerçekleştirilmektedir. Fiberoptik kablo kullanılırken hem denetim bilgisayarının bulunduğu denizaltı aracında hem de operatör bilgisayarının bulunduğu yere Fiberoptik – Ethernet çevirici kullanılmıştır. Kullanılan çevirici Moxa firmasının IMC 101 model çeviricisidir. Bu çevirici 12-48 V arasında çalışabilmektedir. Dolayısıyla operatör bilgisayarın tarafında çeviriciyi kullanabilmek için 12 V luk güç kaynağına ihtiyaç duyulmaktadır.

### **3.2.1.7 Denetim Bilgisayarı İşletim Sistemi Görüntüsünün Hazırlanması ve Yüklenmesi**

Bu bölümde denetim bilgisayarında kullanılan işletim sisteminin CF disk için görüntüsünün oluşturulması ve yüklenmesi anlatılmıştır. QNX de işletim sistemi oluşturmak için genellikle bir script dosyası hazırlanarak bu sayesinde işletim sisteminin görüntüsü oluşturulur. Bu dosyada oluşturulacak olan görüntü dosyasında hangi modüllerin ekleneceği, hangi kütüphanelerin kullanılacağı gibi bilgiler yer almaktadır. Bu dosya çalıştırıldıktan sonra uzantısı .ifs olan görüntü dosyası oluşturulur. Oluşturulan bu görüntü dosyası daha sonra CF disk e yüklenebilir. Oluşturulan görüntü dosyasının yaklaşık olarak boyutu 850KB dir. Görüldüğü gibi çok küçük bir dosya oluşturulmaktadır. Ve bu şekilde işletim sistemi çok kısa bir zaman içerisinde(yaklaşık 8 sn) açılmaktadır. Ethernet modülünün ve diğer çalışması gereken programlar için CF diskin altında bulunan /etc/rc.d/rc.local dosyasına aşağıdaki satırlar eklenmelidir.

```
# Start the io-net driver with just the tcpip protocol
echo Starting network drivers...
io-net -ptcpip &
/bin/waitfor /dev/socket 5

# Configure the network card driver
mount -Tio-net /lib/dll/devn-pcnet.so
/bin/waitfor /dev/io-net/ip0 5

# Start the USB server
echo Starting the USB server with EHCI, OHCI and UHCI support...
io-usb -dehci -dohci -duhci &
/bin/waitfor /dev/io-usb/io-usb 5

# Force contents of /etc/net.cfg to take effect
echo Loading network configuration...
netmanager

# Run the sshd server
echo Starting sshd...
sshd
```

### **3.2.2 Denizaltı Aracı Görüntü Bilgisayarı Yazılımı**

Bu bölümde denizaltı aracında bulunan görüntü bilgisayarında yapılan çalışmalar hakkında bilgi verilmektedir.

### 3.2.2.1 Sualtı Kameralarından Görüntü Alınması

ULİSAR projesinde Outland Technology firmasından alınan UWC-300 model su geçirmez kameralar kullanılmaktadır. Bu kameralar siyah beyaz analog çıkış vermektedir. Analog çıkışı okuyabilmek için Eurotech firmasının CTR-1472 model numaralı PC/104 MPEG 4 Encoder kartı kullanılmaktadır. Bu kart en fazla 4 kameradan bilgi alabilmektedir. Alınan analog data ilk olarak sayısal bilgiye dönüştürülmekte daha sonra da elde edilen sayısal görüntü kartın üstünde bulunan MPEG4 encoder çipi tarafından MPEG4 formatında sıkıştırılarak çıkış verilmektedir. Bu özellikleri bakımından bu kart Birkaç kartla ve yazılımla yapılacak işi tek başına yapmakta buda görüntü bilgisayarında fazladan işlem yükünü engellemektedir.

CTR-1472 MPEG4 encoder kartının hem Linux hemde Windows sürücüleri bulunmaktadır. Görüntü bilgisayarında Linux işletim sistemi yüklü olduğundan Linux sürücüsü kullanılmaktadır. Bu sürücü sayesinde kart çalışmaya hazır duruma getirilebilir, çeşitli görüntü ve sıkıştırma formatları belirlenebilir ve kamera görüntüsü alınabilir. İlk çalışmalarda karttan görüntü alınmakta ve bu görüntü doğrudan operatör bilgisayarına gönderilmekteydi. Hem görüntü bilgisayarında hemde operatör bilgisayarında Linux işletim sistemi çalıştığından alınan görüntü Linux da bulunan “netcat” programı sayesinde gönderilmekteydi. Netcat programı bir soket üzerinden iki bilgisayar arasında haberleşmeyi sağlamak amacıyla kullanılmaktadır. Operatör tarafında Netcat programı sürekli dinleme konumunda bulunmakta ve görüntü bilgisayarı tarafından gönderilen görüntü bilgisini almaktaydı. Alınan görüntüyü izlemek içinse yine Linux işletim sisteminde bulunan Mplayer programı kullanılmaktaydı. MPlayer programı çalışırken girdi olarak netcat programının çıktısını almaktaydı. Mplayer programı daha sonra MPEG4 formatında kodlanmış görüntüyü decode ederek ekranda göstermektedir. Bu proje döneminde ise bu anlatılanlar kendi tarafımızdan yazılmıştır.

Görüntü bilgisayarında kamera görüntüsünü almak için ilk olarak sürücünün yüklenmesi gerekmektedir. MPEG4 encoder kartının sürücüsünü kaynak kodları proje dosyalarını saklamak için kullandığımız depoda tutulmaktadır. Geliştirilen yazılımları saklamak için SVN sunucusu kullanılmaktadır. Bu sunucu “[svn://pc.saranli.cs.bilkent.edu.tr](https://svn://pc.saranli.cs.bilkent.edu.tr)” adresinde bulunmaktadır. Sürücü dosyaları depoda “Ulisar/Underwater/VisionComputer/Drivers/MPEGEncCard” dizininde bulunmaktadır. Sürücü dosyalarını indirmek için aşağıdaki komut kullanılmalıdır.



svn checkout svn://pcsaranli.cs.bilkent.edu.tr/projects/Ulisar/Underwater/VisionComputer/Drivers/MPEGEncCard KAYDEDİLECEK\_DİZİN

Sürücü dosyaları indirildikten sonra indirilen dizine giderek “make” komutu çağırılır. Make komutu başarılı bir şekilde tamamlanırsa ctr1472.ko adında sürücü dosyası oluşturulur. Sürücüyü kullanabilmek için ilk olarak yüklenmesi gerekmektedir. Bunun için load\_ctr1472 adlı script çalıştırılır. Eğer sürücü başarılı bir şekilde yüklendiyse artık sürücü kullanılabilir. Kartı kullanmadan önce ilk olarak parametlerin uygun bir şekilde ayarlanması gerekmektedir. Parametre ayarları burada anlatılmamakta olup sürücü dosyalarının bulunduğu dizin altındaki README dosyasından öğrenilebilir. Karttan alınan görüntü çerçevesi(frame) belirli bir yapıda gönderilmektedir. Dolayısıyla görüntü çerçevesine ulaşabilmek için bu bilginin gelen veri içerisinde çıkarılması gerekmektedir. CTR-1472 kartı ile görüntü alma işlemi aşağıdaki gibi özetlenebilir.

1. Sürücüyü yükle
2. Kartın parametlerini uygun şekilde ayarla
3. Gelen bilgilerin bilgisayarda saklanacağı tampon bölgeyi karta bildir.
4. Karta kamera(lar)dan görüntü almaya başlaması komutunu gönder.
5. Karttan gelen veri içerisinde görüntü çerçevesini çıkart.
6. Çıkartılan görüntüyü kullan.

### **3.2.2.2 Görüntü Verisinin Operatör Bilgisayarına Gönderilmesi**

Görüntü bilgisayarı tarafında denetim bilgisayarında da olduğu gibi RHexLib kullanılmaktadır. Her iki bilgisayarda da RHexLib kullanılması uyum problemlerinin önüne geçmektedir. Bu amaçla RhexLib kullanılarak bir “module” yaratılmıştır. Bu modül sürekli olarak CTR-1472 kartından yukarıda anlatıldığı gibi görüntü almakta ve bu alınan görüntüyü denetim bilgisayarı üzerinden operatör bilgisayarına göndermektedir. Görüntü aktarımı esnasında RHexLib içinde bulunan RhexComm paketi kullanılmaktadır. RhexComm UDP temelli haberleşme yapan bir pakettir. Bu paket kullanılarak 2 çeşit veri transferi yapılabilir. Birincisi gönderilen mesajın karşı tarafta alınıp alınmadığı kontrol edilmez. Bu durumda veri iletiminde hata olması durumunda aynı bilgi bir daha gönderilmez. İkinci tür ise karşı taraftan gönderilen bilginin alındığına dair onay göndermesi beklenir. Onay gelene kadar bir başka mesaj gönderilmez eğer gönderilmesi gerekirse kuyruğa alınıp bekletilir. Bu iki haberleşme

türünden birincisi daha hızlıdır fakat veri kaybolması olasılığı vardır. Burada gönderilen veri görüntü çerçevesi verisi olduğundan dolayı kayıp olmasının işlevselliği durduracak kadar önemli bir veri değildir. Operatör tarafında kamera bilgisi gelirken bir kayıp olursa herhangi bir hata olmaz sadece o anki görüntü gösterilmemiş olur. Buda ihmal edilebilir bir kayıptır.

### 3.2.2.3 İşletim Sistemi Görüntüsünün Hazırlanması ve Yüklenmesi

Denizaltı aracı görüntü bilgisayarında Linux tabanlı Damn Small Linux-Not(DSL-N) adlı işletim sistemi kullanılmaktadır. DSL-N işletim sistemi DamnSmallLinux(DSL) işletim sistemi üzerine ek paketlerin yüklenmesiyle oluşturulmuştur. DSL işletim sistemi KNOPPIX tabanlı bir sistem olmakla beraber boyutu oldukça küçüktür(yaklaşık 50MB) ve gömülü uygulamalar için geliştirilmiştir. DSL-N ise DSL üzerine bazı ek paket programların eklenmesi ile meydana gelmiştir. Bir diğer fark ise DSL işletim sistemi Linux 2.4 çekirdeğini kullanabilirken DSL-N işletim sisteminin Linux 2.6 çekirdeğini kullanmasıdır. DSL-N kullanmamamızın sebebi bilgisayarda kullanılan çevresel donanım birimlerinin sürücülerinin yüklü olmasıdır. DSL-N işletim sisteminde X-windows arayüze desteği sağlanmaktadır. Ancak X-windows arayüzü sistem kaynaklarının kullanımını arttıracığından bizim uygulamamızda bu özellik kullanılmamakta sadece konsol arayüzü kullanılmaktadır. Görüntü bilgisayarı önceden de bahsedildiği üzere Compact Flash(CF) Disc den çalıştırılmaktadır. Aşağıda ilk olarak DSL-N işletim sisteminin görüntüsünün oluşturulması ve CompactFlash Diske yazdırılması anlatılacaktır. Daha sonra oluşturulan işletim sistemi görüntüsünde bazı ayarların nasıl yapılacağı anlatılacaktır.

DSL-N işletim sistemini CF karta yüklemek için ilk olarak DSL-N yi bir CD ye yüklemeliyiz. DSL-N şu adresten indirilebilir <http://damnsmalllinux.org/dsl-n/> . İndirilen görüntü dosyası CD ye yazdırıldıktan sonra, CF diskin bağlı olduğu bir bilgisayarda çalıştırılması gereklidir. DSL-N işletim sistemi açıldıktan sonra X-Window grafik arayüzü ekranına girecektir. Burada Tools->Install To Harddrive seçilir. Daha sonra açılan ekranda CF disk gösterilir ve yükleme işlemine başlanır. Son olarak “Boot Loader” seçeneği geldiğinde GRUB seçilerek yükleme işlemi sona erdirilir. Yüklemenin sona ermesiyle DSL-N işletim sistemi CF diskten çalıştırılabilir.

DSL-N işletim sistemi RAM DISK kullanarak çalışır. Yani bilgisayar açıldığında RAM harddisk gibi kullanılır ve bütün yazma/okuma işlemler RAM e yapılır. Dolayısıyla güç kesildikten sonra dosya sisteminde yapılan bütün değişiklikler kaybolur. Yapılan değişikliklerin kalıcı olması amacı ile bazı işlemler yapılmalıdır. Bu işlemler ileride anlatılmaktadır. DSL-N işletim sisteminin açılış süreci aşağıdaki verilen adreste anlatılmaktadır.

[http://www.damnsmalllinux.org/wiki/index.php/DSL\\_Boot\\_Process](http://www.damnsmalllinux.org/wiki/index.php/DSL_Boot_Process)

Yukarıdaki adreste yazıldığı gibi açılış süreci aşağıdaki sıradadır.

- Syslinux açılış yükleyicisi (bootloader) /cdrom/syslinux.cfg dosyasındaki konfigürasyona göre açılış yapar.
- /linuxrc çalıştırılır. Bu aşağıdaki işleri gerçekleştirir:
  - KNOPPIX dosya sistemi görüntüsünün yerini belirler ve /cdrom altına “mount” eder,
  - Gerekli dosya sistemi girdilerini yaratır. (Çoğunlukla /cdrom un altına sembolik linkler),
  - cloop çekirdek modülünü yükler ve /KNOPPIX altına sıkıştırılmış KNOPPIX dosya sistemini “mount” eder,
  - /ramdisk altında bir RAMDISK yaratır ve burada /home, /etc gibi dizinler oluşturur.
  - /etc/init dosyasını çağırır.
- /etc/init seçilen çalışma düzeyine(run level) göre çalışır./etc/rcS.d ve /etc/rc5.d (çalışma seviyesi 5 için) dosyaları çalıştırılır.

Sistem otomatik olarak root hesabı ile açılır ve daha sonra dsl hesabına girer ve X-Windows ortamını çalıştırır. Açılış sürecini özelleştirmek için /cdrom/syslinux.cfg dosyasına hile kodları (cheat codes) kullanılabilir.

### ***3.2.2.3.1 Açılış Sürecinin Özelleştirilmesi***

Standart DSL-N yüklemesi CF disk üzerinde 2 dosya yaratır. Bunlar /minirt.gz ve /knoppix/knoppix görüntü dosyalarıdır. /minirt.gz dosyası yaklaşık 3.5 MB büyüklüğünde bir dosyadır ve /knoppix/knoppix dosyası yaklaşık 97MB büyüklüğünde bir dosyadır. /minirt.gz görüntü dosyasının içinde başlangıç konfigürasyon dosyaları vardır. /knoppix/knoppix görüntü dosyasında ise geri kalan işletim sistemi dosyaları bulunmaktadır.

/minirt.gz görüntü dosyasının içeriği dosyayı uygun bir çalışma dizinine çıkartarak ve uygun bir “mount” noktasına (örn. /tmp) dosya sistemi olarak “mount” ederek görülebilir.DSL-N

işletim sisteminin /media/disk dizininde bulunan CF diske yüklü olduğunu varsayalım./minirt.gz görüntü dosyasını dosya sistemi olarak “mount” etmek için aşağıdaki adımlar uygulanmalıdır.

```
cd /tmp
cp /media/disk/minirt.gz .
gunzip minirt.gz
mkdir dsl_mount
mount /tmp/minirt/tmp/dsl_mount -o loop=/dev/loop3
```

Burada dikkat edilmesi gereken bir nokta “Loopback Device Support” çekirdekte kullanılabilir durumda olması gerekmektedir. “Loopback Device Support” çekirdekte kullanıma açılması için internette arama yapılabilir. Bu konuyla ilgili bir çok bilgi bulunmaktadır. /minirt.gz dosyasının içeriği incelendiğinde DSL açılış sürecinde bahsedilen /rclinux dosyasının bulunduğu görülmektedir. Bu dosya başlangıçta bazı donanım cihazlarının taramasını yapmayacak şekilde düzeltilebilir. Böylece açılış zamanında önemli bir kısalma söz konusu olacaktır. Ayrıca CF disk üzerinde bulunan GRUB dizininin altındaki grub.conf dosyasında gerekli değişiklikler yapılmalıdır.

Yukarıda da bahsedildiği gibi işletim sistemi ile ilgili dosyaların büyük bir bölümü /knoppix/knoppix adlı görüntü dosyasının içinde bulunmaktadır. Bu dosyada değişiklik yapabilmek için izlenmesi gereken işlemler aşağıda verilmiştir. Bu dosyayının içeriğini görebilmek için “compressed loopback device” desteği gerekmektedir. Bazı Linux sürümlerinde bu destek mevcuttur(Gentoo Linux). Ancak bazılarında ise bu destek bulunmamaktadır. Bunun için dosyayı mount etmeden önce çıkartmamız gerekecektir. Aşağıda anlatılan işlemler Ubuntu dağıtımı için yazılmıştır ve denenmiştir. Ayrıca bu işlemler yapabilmek için root hesabına geçmek gereklidir. Bu kısımda da gene CF diskin /media/disk altında olduğunu varsayıyoruz.

- İlk olarak görüntü dosyasını çıkartıyoruz ve daha sonra /tmp altına “mount” ediyoruz.

```
cd /tmp
extract_compressed_fs /media/disk/knoppix/knoppix > tmp.iso
```

Bu işlem biraz uzun sürebilir.

- “mount” edilen knoppix görüntü dosyası sadece okunabilir şekilde oluşturulmuştur. Dolayısıyla bu dosyayı tar/untar komutları ile başka bir yere kopyalamalı, değiştirmeli ve tekrar görüntü dosyası oluşturmalıyız. Kopyalacak yerin adını “master” yapıyoruz

```
cd /tmp
tar cvzf knoppix_orig.tgz ./knoppix_mount
mkdir master
cd master
tar xvzf ../knoppix_orig.tgz
```

- Artık burada gerekli deęişiklikleri yapabiliriz. Gerekli deęişiklikleri yaptıktan sonra tekrar sıkıştırılmış bir görüntü dosyası oluşturacağız ve bu dosyayı CF diske yükleyeceğiz. İlk olarak sıkıştırılmamış bir cdrom dosya sistemi yaratmalıyız.

```
cd /tmp
genisoimage -R -U -V "ULISAR DSL(KNOPPIX) Image" -publisher \
"ULISAR Dev Team" -cache-inodes -no-bak -pad \
-o ./knoppix_iso_image/tmp/master/knoppix_mount
```

- ISO dosya sistemi görüntüsünü yarattıktan sonra bu görüntüyü aşağıdaki komutla sıkıştıracağız. Bu işlem biraz uzun sürebilir.

```
cd /tmp
create_compressed_fs -B 65536 ./knoppix_iso_image ./knoppix_new
```

- Yeni sıkıştırılmış ISO görüntü dosyası knoppix\_new adı ile oluşturuldu. Şimdi bu dosyayı CF diske kopyalayabiliriz. Aynı zamanda orjinal knoppix dosyasının güvenlik amacı ile bir yedeğini alacağız.

```
cd /media/disk/knoppix
mv knoppix knoppix_orig
cp /tmp/knoppix_new ./knoppix
sync
```

Artık yeni knoppix görüntü dosyası kullanılabilir durumdadır. Aşağıda yeni oluşturulacak knoppix görüntü dosyasında yapılacak deęişiklikler verilmiştir.

1. İlk olarak /etc/inittab dosyasındaki çalışma düzeyini deęiştirmeliyiz. Bu sayede sistem açıldığında X-windows ortamı yerine konsol ortamına giriş yapılabilecek. Aşağıda verilen satırı bir sonraki satır gibi deęiştirmeliyiz.

```
id:5:initdefault:
id:3:initdefault:
```

2. Çalışma düzeyini 3 yaptığımız zaman bu konfigürasyon yedekleme sürecinin etkilenmesine sebep olmaktadır. Bunu düzeltmek için /.bash\_profile dosyasında dsl\_user inında çalışma düzeyi 3 e girmesini sağlayacak şekilde deęiştirmeliyiz. dls\_user inın .bash\_profile dosyasında çalışma düzeyi kontrol edilir ve X-ortamına sadece çalışma düzeyi 5 ise girmesi sağlanır. Ancak şimdi de dsl\_user na girmek için varsayılan olarak 4 seri terminal açılmaktadır. Buda önemli bir gecikmeye sebep

olmaktadır. Bunu engellemek için /etc/inittab dosyasında en sonda açılan 4 terminalden 3 ü devre dışı bırakılmalıdır.

```
1:12345:respawn:/bin/bash -login >/dev/tty1 2>&1 </dev/tty1
:2:234:respawn:/bin/bash -login >/dev/tty2 2>&1 </dev/tty2
:3:234:respawn:/bin/bash -login >/dev/tty3 2>&1 </dev/tty3
:4:234:respawn:/bin/bash -login >/dev/tty4 2>&1 </dev/tty4
```

3. Bir değişiklik de dsl\_user kullanıcısı için .filetool.lst dosyasında yapılmalıdır. Bu dosyaya sistem kapatılırken yediği alınması gereken dosyaları eklememiz gerekmektedir.Özellikle aşağıda dosyaların yedeğinin alınması şarttır.

```
/etc/ssh (sunucu anahtarlarının tekrar yaratılmasını engellemek için)
/etc/hostname (değişebilir)
/etc/shadow (şifrelerin korunmasını sağlamak için)
```

5. Sistem açılırken otomatik donanım bulmayı devre dışı bıraktığımız için ethernet donanımı modüllerini yüklememiz gerekiyor. Bunu yapmak için /opt/bootlocal.sh dosyasına aşağıdaki satırı eklemeliyiz.Bu satır olmadan network konfigürasyon komutları hata verecektir.

```
# Load the required kernel modules
echo ULISAR BOOTLOCAL: Loading the ethernet card kernel module...
```

6. Görüntü bilgisayarı ve denetim bilgisayarı birbirleri ile çapraz ethernet kablosu üzerinden bağlanmaktadır. İki bilgisayarın birbirini görebilmesi için ve haberleşme sağlayabilmeleri için durağan IP almaları gerekmektedir. Görüntü bilgisayarında durağan IP almak için aşağıdaki satırları /opt/bootlocal.sh dosyasına eklemeliyiz

```
# Configure the local ethernet interface
echo BOOTLOCAL: Statically configuring the ethernet network interface...
sudo ifconfig eth0 192.168.100.1 netmask 255.255.255.0 up
sudo ifconfig eth0 up
```

7. Görüntü bilgisayarına uzaktan bağlantı yapabilmek için SSH server uygulamasını çalıştırmalıyız. Bunun için aşağıda yazan satırları /opt/bootlocal.sh dosyasına eklemeliyiz.

```
# Start the SSH Server
echo ULISAR BOOTLOCAL: Starting up the SSH Server
sudo /etc/init.d/ssh start
```

### 3.2.2.4 Görüntü Bilgisayarı Yazılımının Derlenmesi ve Yüklenmesi

Bu bölümde üzerinde DSL-N işletim sistemi çalışan görüntü bilgisayarına gerekli yazılımın nasıl derleneceği ve yükleneceği anlatılmıştır. Görüntü bilgisayar yazılımı tutulduğu depodan aşağıdaki komut ile bilgisayara indirilmelidir. İndirme işleminin nasıl yapıldığı önceki bölümlerde anlatılmıştır. Derlemenin yapılacağı bilgisayarda gcc derleyicisi bulunmalıdır. Derlemeyi yapabilmek için görüntü bilgisayarının kaynak dosyalarının bulunduğu dizine gidilmelidir. Bu dizin aşağıda verilmiştir.

```
İNDİRME_KLASÖRÜ/Ulissar/Underwater/VisionComputer/Examples/Camera
```

Yukarıda İNDİR\_KLASÖRÜ olarak belirtilen klasör depodan indirilen kaynak dosyalarının bulunduğu klasördür. Daha sonra bu dizinde “make” komut çağırılır. Bunun sonucunda VideoServer isimli çalışan bir dosya oluşmalıdır. Bu dosya görüntü bilgisayarında çalışacak olan dosyadır. Şimdi bu dosyayı görüntü bilgisayarına yüklemeliyiz. Bunu yapabilmek için “scp” komutunu kullanacağız. Dosyayı yükleyebilmemiz içinde gerekli komut aşağıda verilmiştir.

```
scp VideoServer dsl\_user@192.168.100.1:/tmp
```

Bu komut çalıştırıldıktan sonra görüntü bilgisayarında bulunan dsl\_user kullanıcısının şifresi istenecektir. Şifreyi girdikten sonra yükleme işlemi tamamlanacaktır. Görüntü bilgisayar yazılımının çalıştırılması için uzaktaki görüntü bilgisayarına SSH ile bağlanılmalıdır. SSH bağlantısı yapmak için gerekli komut aşağıda verilmiştir.

```
ssh dsl\_user@192.168.100.1
```

Bu komut çalıştırıldıktan sonra dsl\_user kullanıcısının şifresi istenecektir. Şifre girildikten sonra görüntü bilgisayarına bağlanılmış oluruz. Bağlantı sağlandıktan sonra görüntü bilgisayar yazılımının çalışabilir dosyasının bulunduğu klasöre gitmeli ve programı çalıştırmalıyız. Bunu yapabilmek için gerekli komutlar aşağıda verilmiştir.

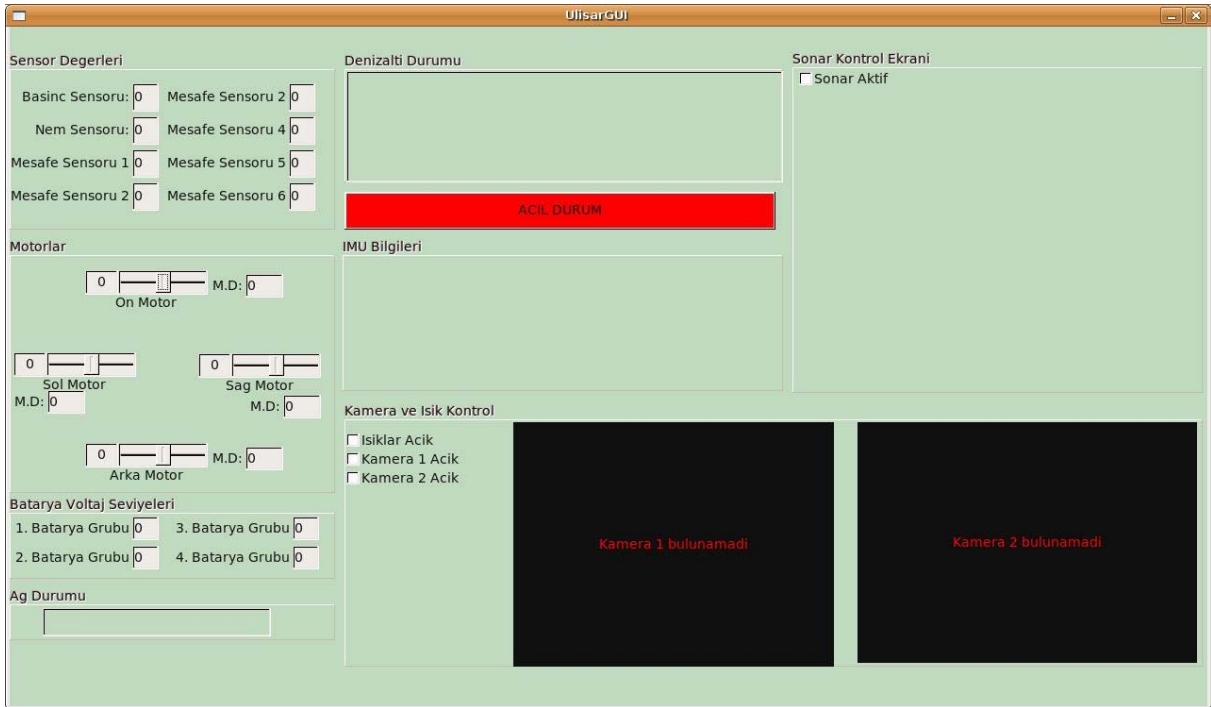
```
cd /tmp
```

```
./VideoServer
```

Programı çalıştırdıktan sonra görüntü bilgisayar kameradan aldığı görüntü bilgilerini operatör bilgisayarına göndermeye başlayacaktır.

### 3.2.3 Operatör Bilgisayarı Yazılımı

Operatör bilgisayarında işletim sistemi olarak Ubuntu 9.04 bulunmaktadır. Bu işletim sistemi kurulumu Ubuntu 9.04 imaj dosyası internetten indirilerek ardından bir CD ye yazılarak yapılabilir. İlk çalışmalarda hazırlanan operatör arayüzünün gerek görsellik gerekse de işlevsellik bakımından eksikleri bulunmaktaydı. İlk hazırlanan operatör arayüzü RHexLib arayüz hazırlama aracı kullanılarak hazırlanmıştı. RHexLib kütüphanesinin operatör kullanıcı arayüzü hazırlama aracı FLTK tabanlı bir uygulamadır. FLTK kütüphanesi oldukça profesyonel görünümlü arayüzler hazırlamaya yarayan zengin bir kütüphanedir. RHexLib kütüphanesinde de FLTK kullanıldığından yeni hazırladığımız arayüzü FLTK tabanlı olarak yazdık ve operatör arayüzüne entegre ettik. FLTK kütüphanesini kullanarak hızlı bir şekilde arayüz hazırlamaya yarayan FLUID isimli bir program bulunmaktadır. FLUID sayesinde kısa zamanda oldukça işlevsel kullanıcı arayüzleri hazırlanabilmektedir. Şekil 51 de operatör bilgisayar yazılımında kullanılan yeni operatör arayüzü verilmiştir.



Şekil 51: Operatör Arayüzü

Şekil 51 de görülen operatör arayüzündeki bölümlerin hangi amaçla kullanıldığı aşağıda belirtilmiştir.



**Sensör Değerleri:** Bu bölümde denizaltı aracında bulunan basınç sensörü, nem sensörü ve 6 adet mesafe sensörlerinin ölçtüğü değerler verilmektedir.

**Motorlar:** Bu bölümde motorların hız değerlerinin elle ayarlanması istenmesi durumunda kullanılacak olan araçlar yer almaktadır. Şekilden de görüldüğü gibi her bir motorun hızı ayrı ayrı ayarlanabilmektedir. Motor hız ayarları her bir motor için konulmuş olan kayar çubuklar tarafından yapılmaktadır. Kayar çubuklar -25 ve 25 arasında toplam 50 adımda gitmektedir. Dolayısıyla motorları 50 değişik hız adımında kontrol edebilmekteyiz. Ancak bu değerler ihtiyaç durumunda değiştirilebilir. Kayar çubukların sol taraflarında yer alan küçük bölümde kayar çubuğun o anki değeri yazmaktadır. Kayar çubukların altında M.D. ile belirlenen kısımda motorların o anki hızlarının mevcut değerleri yer almaktadır. Bu değerler denizaltı araçındaki denetim bilgisayarından sürekli olarak aktarılmaktadır. Motorların hız değerleri genelde kumandadan gelecek olan komutlara göre ayarlanacaktır.

**Batarya Voltaj Seviyeleri:** Bu bölümde denizaltı aracında yer alan LiPo pil paketlerinin voltaj değerleri yer almaktadır. Bilindiği üzere LiPo bataryalar çok dikkatli kullanılmalı ve belli bir voltaj değerinin altına düştüğünde muhakkak kullanmaya son verilmelidir. Aksi takdirde bataryalar bir daha şarj edilememekte hatta daha da kötüsü patlayabilmektedir. Bu bölümde her bir batarya paketinin o anki voltaj değeri verilmektedir. Aynı zamanda denizaltı aracında bulunan voltaj monitör devreleriyle de voltaj kontrol edilmekte eğer kritik değerlerin altına düşerse bataryanın kullanımına son verilmektedir.

**Ağ Durumu:** Bu bölümde denizaltı aracı ile haberleşmede kullanılan ağ bağlantısının durumu gösterilmektedir. Ağ bağlantısında herhangi bir kopma olursa kullanıcı bu ekranda uyarı mesajı gördükten sonra tekrar ağ bağlantısı kurmayı deneyecektir.

**Denizaltı Durumu:** Bu ekranda denizaltı araçından gelecek mesajlar yazılacaktır. Eğer denizaltı araçındaki elektronik bileşenlerden herhangi birinde bir problem olursa problemin hangi donanımdan kaynaklandığı ve ne olduğu yazılacaktır.Örneğin 3. motorda bir arıza oluşmuşsa denizaltı aracı 3. motorda bir arıza olduğu mesajı ileticek ve bu mesaj bu ekranda yazılacaktır

**Acil Durum:** Bu düğme acil durumlarda denizaltı aracının acil durum konumuna geçmesini sağlayacaktır. Acil durum anında denizaltının bütün motorları duracak, bütün elektronik ekipmanlara kendilerini kapatması mesajı gönderilecek ve bataryaların kullanımına son verilecektir.Denizaltı aracı durağan haldeyken yukarı doğru küçük bir yüzerliliğe sahip oldu için aracın yavaş bir şekilde su yüzüne çıkması sağlanacaktır.

**IMU Bilgileri:** Bu ekranda denizaltı aracında bulunan IMU(Ataletsel Ölçüm Birimi) dan gelen bilgiler yer alacaktır. IMU deniz altı aracının o andaki duruş(attitude) bilgilerini ve açısal ivme bilgilerini vermektedir.

**Kamera ve Işıl Kontrol:** Bu ekrandan kameralar ve ışıklar kontrol edilebilmekte ve ayrıca denizaltı aracında bulunan kameralardan gelen görüntüler gösterilmektedir. Araçta bulunan ışıklar oldukça fazla güç tüketmektedir. Dolayısıyla sığ sularda ışığın ihtiyaç olmadığı durumlarda ışıkların yanması operasyon süresini kısaltacaktır. Bu ekranda bulunan “Işıklar açık” kutucuğu işaretlenince ışıklar açılacak, işaret kaldırılınca da ışıklar kapanarak enerji tasarrufu sağlanacaktır. Ayrıca yine bu ekranda bulunan “Kamera 1 Açık” ve “Kamera 2 Açık” kutucuklarıyla kameraların çalışıp, durdurulması sağlanacak. Eğer herhangi bir kamera açık konuma getirilirse o kameradan gelen görüntü ilgili kamera görüntüsü penceresinde gösterilecektir. Şekil 1 de kameralara ait görüntü pencereleri siyah pencerelerdir.

**Sonar Kontrol Ekranı:** Bu ekranda araçtan gelen sonar bilgisi gösterilecektir. Yine bu ekranda bulunan “Sonar Aktif” kutucuğu aracılığıyla sonar cihazının açılıp kapatılması ayarlanabilecektir.

Şu ana kadar operatör arayüzündeki motor kontrol ve kamera kontrol bölümlerine ait yazılımlar geliştirilmiş olup bunların denizaltı aracındaki denetim ve kontrol bilgisayarı ile düzgün bir şekilde çalışması sağlanmıştır. Ayrıca araç kontrolü için kumandanın kontrol özelliğide kullanılabilir. Şekil 52’ de kullanılan kumanda ve üzerinde bulunan düğme ve kollar numaralarıyla birlikte verilmiştir.



Şekil 52: Araç kontrolü için kullanılan kumanda

Şekil 52’de numaralandırılmış olan düğme kolların işlevleri Tablo 40’ da açıklanmıştır.

Numara	İşlev
1	Kamera 1 açma-kapama
2	Kamera 2 açma-kapama
3	Işıkları açma-kapama
4	Sonar açma-kapama
5	Denizaltı aracının yönlendirilemesi. İleri-geri,sağ,sol.
6	Denizaltı aracının yunuslama(pitch) kontrolü
7	Kullanılmıyor
8	Denizaltı aracının dik olarak yukarı çıkması
9	Denizaltı aracının dik olarak aşağı inmesi
10,11	10 ve 11 numaralı düğmelere aynı anda basıldığında Acil Durum konumuna geçirmek

*Tablo 40: Kumandadaki düğme ve kolların işlevi*

### 3.2.3.1 Video Verisinin Çözülmesi

Bu bölümde operatör yazılımında görüntü bilgisayarından gelen MPEG4 formatındaki video çerçevesinin(frame) çözülmesi(decoding) işleminin nasıl yapıldığı anlatılacaktır. MPEG4 formatı günümüzde birçok videoda kullanılan kodlama yöntemidir.Linux işletim sistemi için geliştirilmiş olan ffmpeg paketinin içinde bulunan libavcodec MPEG4 de dahil olmak üzere birçok kodlama ve çözümü algoritmasını içermektedir. ULİSAR projesinin operatör yazılımında da gelen video çerçevesinin çözülmesi için libavcodec kütüphanesi kullanılmıştır. Yazılım geliştirme sırasında karşılaşılan en büyük problem libavcodec kütüphanesinin çok başarılı olmasına rağmen dökümantasyonunun çok zayıf olmasından kaynaklanmıştır. Verilen bilgilerde sadece mevcut olan bir video dosyasından nasıl bilgi okunacağı ve çözümleneceği anlatılmıştır. Ancak bizim uygulamamızda video çerçevesi bir dosyadan değil ağ üzerinden alınmaktadır. Libavcode kütüphanesinin genel olarak işleyişi ve kullanılması gereken fonksiyonlar aşağıda anlatılmaktadır.

Çözümleme işlemi için kullanılan fonksiyonlardan birkaçı şunlardır: av\_open\_input\_file, av\_find\_stream\_info, avcodec\_find\_decoder, avcodec\_open, av\_read\_frame, avcodec\_decode\_video,...Bunlardan birçoğu av\_open\_input\_file() fonksiyonuna verilen bir dosyadan veri okurlar. Burada karşımıza çıkan ilk problem libavcodec in bir dosyadan değilde hafızada belirtilen bir tampondan(buffer) nasıl bilgi okuyacağıdır. Operatör yazılımı gelen

görüntü çerçevesini aldıktan sonra tamponda bir bölgeye yazmaktadır. Dolayısıyla çözümleme işlemi bu bölgedeki veri üzerinden yapılmalıdır. Tam bu noktada ihtiyacımızı karşılamak üzere URLProtocol leri devreye girmektedir. Libavcodec kütüphanesi içinde kendi yazdığımız URLProtocol leri libavcodec kütüphanesininkileriyle(örn. url\_open, url\_write, url\_read, vs.) değiştirmemizi sağlayacak fonksiyonlar bulunmaktadır. URLProtocol leri hakkında bilgi aşağıdaki internet adresinde bulunabilir.

<http://cekirdek.pardus.org.tr/~ismail/ffmpeg-docs/structURLProtocol.html>

Libavcodec kütüphanesinin kendi içindeki fonksiyonları bir URLProtocol u tanımlayarak kendi yazdığımız fonksiyonlarla değiştirebiliriz. İlk olarak bir URLProtocol u ne işe yaradığı anlatılacaktır. Av\_open\_input\_file() fonksiyonu çağrıldığında, bu fonksiyon dosya adını inceler ve eğer önceden tanımlanmış bir URLProtocol ile başlıyorsa o dosya için uygun URLProtocol ü kullanır. Esasen bir URLProtocol ü sadece bir ön ek olarak da düşünülebilir. Örneğin http://, rtp://, pipe://, vs. Yeni yaratılacak olan URLProtocol ünün adının buf:// olduğu varsayalım.(tampondan okuma yapılacağı için.). Daha sonra aşağıda verilmiş olan kod parçasığı kullanılır.

```
// kendi protokolümüz
URLProtocol * protocol = (URLProtocol *) malloc(sizeof(URLProtocol));
protocol->name = "buf";
protocol->url_open = my_open;
protocol->url_read = my_read;
protocol->url_seek = NULL;
protocol->url_close = my_close;

register_protocol(handle->protocol);
```

Yukarıdaki kod parçasığından da görülebileceği gibi libavcodec kütüphanesine eüer bir dosya ismi buf:// ile başlıyorsa yeni yaratılan URLProtocol ün fonksiyonlarını kullanmasını bildiriyoruz. Örneğin aşağıdaki gibi bir kod parçasığı verilmiş olsun.

```
char filename[50];
snprintf(filename, 50, "buf://%p", pointerToParams);
av_open_input_file(&pFormatCtx, filename, NULL, 0, NULL);
```

Verilen örnekte dosya adı buf:// ile başladığından dolayı av\_open\_input\_file() fonksiyonu önceden tanımladığımız my-open fonksiyonunu çağırır.

```

int my_open(URLContext *ctx, const char *filename, int flags)
{
    struct Params * param;
    sscanf(filename + strlen("buf://"), "%p", &param);
    ctx->priv_data = (void *) param;
    ctx->is_streamed = 1;
    return 1;
}

```

Yukarıdaki kod parçacığında tanımlanan my\_open fonksiyonu içine URLContext tipinde bir parametre almaktadır. my\_open fonksiyonu veriyi saklamak için ctx parametresinin priv\_data elemanını kullanmaktadır. is\_streamed değeri akışın olduğunu belirtmek için 1 e eşitlenmiştir.

my\_read fonksiyonu belirtilen giriş tamponundan veriyi okur ve libavcode in tamponuna yazar.

```

int my_read(URLContext *ctx, unsigned char *buf, int buf_size)
{
    // get private data
    struct Params * params = (struct Decoder_Shared *) ctx->priv_data;
    // copy data from input buffer to ffmpeg's buffer, eg.
    count = MIN(buf_size, params->input_buffer_size);
    memcpy(buf, params->input_buffer_pos, count * sizeof(uint8_t));
    return count;
}

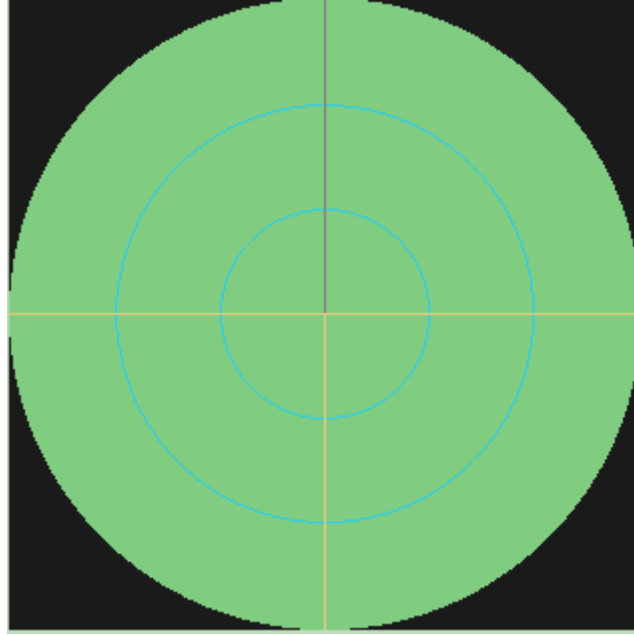
```

libavcodec kütüphanesi her veriye ihtiyacı olduğu zaman(başlatılma ve çözümüleme esnasında) my\_read fonksiyonunu çağırır.my\_write fonksiyonu is my\_read fonksiyonun yaptığı işin tam tersini yapar. Yani libavcode in çıkış tamponundan veriyi alır ve belirlenen tampona yazar.

libavcodec kütüphanesinin genel olarak çalışması yukarıda anlatılmıştır. Operatör arayüzü uygulmasında çözümüleme işlemini yapmak için ayrı bir thread çalışmaktadır. Bu thread her video çerçevesi geldiğinde o çerçeveyi alır, çözümler ve video görüntüsünü ekrana çizdirmekten sorumlu olan yazılım parçasına gönderir. Daha sonra çözümlenen video görüntüsü ekrana basılır. Bu süreç sürekli olarak kendisini tekrar eder.

### 3.2.3.2 Sonar Ekranının Yazılımı

Operatör arayüzüne eklenen yeni ekranlardan biriside sonar ekranıdır. Sonar, çalışma prensibi açısından radarlara çok benzemektedir. Ancak sualtı altı ortamında radyo dalgaları çok fazla sönümlendiğinden bunun yerine ses dalgaları kullanılmaktadır. Sonar da radar gibi 360 derece dönerek etrafa ses dalgaları yayınlar ve her dalganın (eğer varsa) dönüş süresini kullanarak dalganın yansıdığı cismin uzaklığını gönderir. Dolayısıyla sonar ekranında aynı radar ekranı gibi açısal koordinatlarda verilmiştir.Sonar ekranında bulunan pencere üzerinde çizim yapılırken OpenGL grafik kütüphanesi kullanılmıştır. Şekil 53 de örnek bir sonar penceresi görüntüsü verilmiştir. Ancak sonar bağlanmadığı için ekranda herhangi bir mesafe bilgisi bulunmamaktadır.



Şekil 53: Örnek sonar ekranı görüntüsü

### 3.2.3.3 Operatör Arayüzünün Derlenmesi ve Çalıştırılması

Operatör arayüzünün derlenmesi için kaynak kodlar depodan indirildikten sonra operator kaynak kodlarının bulunduğu dizine gidilir ve make komutu çalıştırılır.Operatör arayüzünün kaynak kodlarının bulunduğu dizin aşağıda verilmiştir.

İNDİRME\_KLASÖRÜ/Ulisar/Underwater/ControlComputer/OCUCode/UW\_GUI

Derleme başarılı bir şekilde bittikten sonra aşağıda verilen dizinde start.sh dosyası uygun parametrelerle çağırılarak arayüzün çalışması sağlanır. Operatör arayüzünün çalıştırılacağı dizin ve çalıştırma komutu aşağıda verilmiştir.

```
Ulisar/Underwater/ControlComputer/Operator/./start.sh PORT HOST_IP
```

Yukarıdaki komutta PORT denetim bilgisayarına bağlanmak için kullanılacak PORT numarası, HOST\_IP ise denetim bilgisayarının IP sini belirtmektedir. Şu anda kullanılan PORT numarası 3000 ve denetim bilgisayarının IP adreside 192.168.100.2 dir. Bu komutun ardından ekrana gelen arayüzde “Demo” modu seçildiği zaman denizaltı aracını yönettiğimiz arayüz karşımıza çıkar. Bu arayüzde artık denizaltında tanımladığımız bütün fonksiyonları kullanabiliriz. Denizaltı aracını hareket ettirmek için Joystick kullanılmaktadır.

## 4 Karşılaşılan Problemler

Son çalışmalar içinde ağırlıklı olarak entegrasyon çalışmaları yapılmıştır. Bu çalışmalar sırasında ilk problem denetim bilgisayarında kullanılan QNX işletim sisteminin bir üst versiyona geçişi sırasında yaşandı. Yeni sistemdeki network yönetim sistemi tamamiyle değiştiğinden, eski konfigürasyon dosyalarında ve işletim sistemi imaj dosyasında bazı değişiklikler yapıldı. İkinci sorun ise denetim bilgisayarın ve köprü arasında kullanılan USB-Serial sürücüsünden kaynaklandı. Kullandığımız çeviriciye uygun sürücüyü bulmak ve çalıştırılabilir duruma getirmek birkaç günümüzü aldı. Bu sorunlar dışında çalışmamızın yavaşlamasını sağlayacak herhangi büyük bir problemle karşılaşılmadı.

Sistemle ilgili olan en büyük sıkıntı motorlarımızın bir tanesi haricindeki diğerlerinin sürücü kartların havuz denemeleri öncesi bozulmuş olmasıdır. Bu aşamada motor sürücü kartlarının garanti kapsamı altında değişimi yapılacak olup daha sonra bu sürücüler yerine daha başka projelerde de kullandığımız kendi tasarımımız olan DC Brushed motor sürücüleri kullanılması düşünülmektedir.

## 5 Sonu

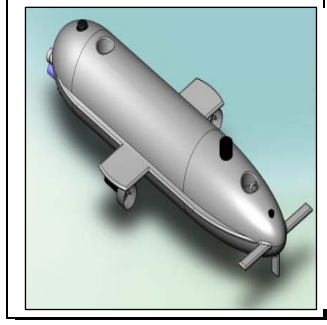
Proje kapsamında insansız bir denizaltı aracı retilmiř ve bunun tm elektronik ve yazılım kısmı proje grubu tarafından dizayn edilip retilip geliřtirilmiřtir. Denizaltı aracı řu anda bir ok iřlevi alıřıyor olup bundan sonra denizaltı aracının otonom davranıř gstermesi ynnde alıřmalara devam edilecektir.





# **ULİSAR**

## **Ulusal İnsansız Sualtı Aracı Projesi**



**Ultrasonik Almaç Donanım Tasarımı Raporu**

**Tarih: 30.04.2010**

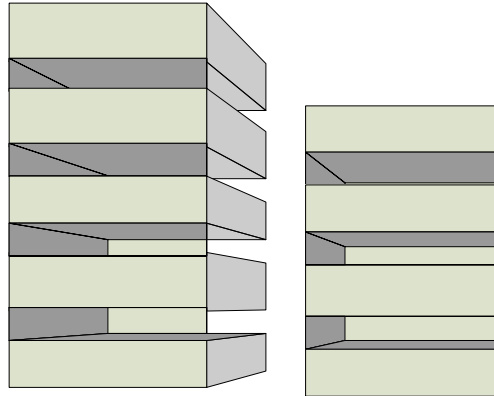
Proje kapsamında tasarlanan ve elektronik donanım kutusu içine yerleştirilmiş olan kartlardan olan ve aşağıda detaylandırılmış birimler PC104 boyutlarındadır.

Çevresel birimler ile haberleşmenin URB veri yolu kullanılarak gerçekleştirilmesi sebebiyle, çevresel birimler gruplanarak çeşitli URB düğümleri haline dönüştürülmüştür. Aşağıda gerçekleşmiş olan URB düğümleri ve işlevleri verilmiştir. Raporun bu bölümünde aşağıda koyu harflerle gösterilmiş düğümler açıklanmıştır.

Tablo 1: URB düğümleri

Düğüm No	İşlevsel Tanımı
D1	Motor Arayüz Düğümü
D2	<b>Ultrasonik Almaç Düğümü</b>
D3	Kamera – Işık Denetleyici Düğümü
<b>D4</b>	<b>Sıcaklık Almaçları Düğümü</b>
D5	Analog Almaç Düğümü
D6	Güç Denetimcisi Düğümü

Bu tasarım sonucunda, elektronik donanım kutusunda PC104 yığınları aşağıdaki görünümde olacaktır.



Şekil 1: PC104 yığınları

*15 Kartı :*

**(Denetim bilgisayarına bağlı yığın arayüz kartı)**

Bu kartın temel işlevi ULISAR konektörü ile düğümlere taşınan veya düğümlerden alınan verilerin uygun kartlara, almaçlara veya motorlara taşınması için uygun konektör bağlantılarının yapılmasıdır. Ayrıca, bu kart üzerinde sıcaklık almaç düğümü gerçekleştirilmiştir.

Sıcaklık almaç olarak seçilen TC74A0-5.0VAT yongası I2C haberleşme veri yolunu kullandığı için, motor düğümüne benzer şekilde PIC16F88 mikro-denetleyicisi üzerinde sıcak almaç düğümünün gerçekleştirilmesi uygun bulunmuştur.

Sıcaklık almaç olarak TC74A0-5.0VAT yongasının seçilmesinin altında yatan en önemli sebep bu yonganın herhangi bir donanım elemanına ihtiyaç duymadan çalışması ve sayısal

çıktı üretmesidir.

TC74A0 yongası TO-220 paketinde olup I2C protokolü ile sıcaklık bilgisini sunan bir çiptir. 2 derece hassasiyeti ile 25~85 C arasında ölçüm yapabilmektedir.

Pin No. (5-Pin SOT-23)	Pin No. (5-Pin TO-220)	Symbol	Type	Description
1	1	NC	None	No Internal Connection
2	3	GND	Power	System Ground
3	5	V <sub>DD</sub>	Power	Power Supply Input
4	4	SCLK	Input	SMBus/I <sup>2</sup> C Serial Clock
5	2	SDA	Bidirectional	SMBus/I <sup>2</sup> C Serial Data

### TC74A0 Pin açıklaması

Write Byte Format												
S	Address	WR	ACK	Command	ACK	Data	ACK	P				
	7 Bits			8 Bits		8 Bits						
Slave Address				Command Byte: selects which register you are writing to.			Data Byte: data goes into the register set by the command byte.					
Read Byte Format												
S	Address	WR	ACK	Command	ACK	S	Address	RD	ACK	Data	NACK	P
	7 Bits			8 Bits			7 Bits			8 Bits		
Slave Address				Command Byte: selects which register you are reading from.		Slave Address: repeated due to change in data-flow direction.			Data Byte: reads from the register set by the command byte.			
Receive Byte Format												
S	Address	RD	ACK	Data	NACK	P						
	7 Bits			8 Bits								
S = START Condition P = STOP Condition Shaded = Slave Transmission				Data Byte: reads data from the register commanded by the last Read Byte or Write Byte transmission.								

### TC74A0 SMBUS/I2C Protokolü

Actual Temperature	Registered Temperature	Binary Hex
+130.00°C	+127°C	0111 1111
+127.00°C	+127°C	0111 1111
+126.50°C	+126°C	0111 1110
+25.25°C	+25°C	0001 1001
+0.50°C	0°C	0000 0000
+0.25°C	0°C	0000 0000
0.00°C	0°C	0000 0000
-0.25°C	-1°C	1111 1111
-0.50°C	-1°C	1111 1111
-0.75°C	-1°C	1111 1111
-1.00°C	-1°C	1111 1111
-25.00°C	-25°C	1110 0111
-25.25°C	-26°C	1110 0110
-54.75°C	-55°C	1100 1001
-55.00°C	-55°C	1100 1001
-65.00°C	-65°C	1011 1111

TC74A0 Sıcaklık Dönüşüm Değerleri Tablosu

URB protokolü köprü ünitesinin her bir URB noktası üzerinde bulunan yazılımsal mesaj kutuları ile haberleşmesi sırasında gelen ve giden mesajların formatlarını ve görevlerini tanımlamaktadır. URB noktaları açısından bakıldığında köprüden gelen mesajların bulunacağı gelen kutusu (inbox) ve giden kutusu (outbox) tanımlanmıştır. Bu mesaj kutuları en fazla 8 adet olabilmekte, ve her köprüye en fazla 15 URB noktası bağlanabilmektedir. 0. Gelen mesaj kutusu(inbox0) köprü ünitesinin tüm URB noktalarına aynı anda (broadcast) mesaj iletmesi için ayrılmış, 0. Giden kutusu(outbox0) ise köprünün kendine bağlı bulunan URB noktalarının tanıyabilmesi için ayrılmıştır. Köprü ünitesi aşağıdaki gibi senkronizasyonu sağlayabilmek için inbox0'a belirli komutlar gönderebilmektedir.

7	6	5	4	3	2	1	0
Opcode					Reserved		
Arg[0]							
Arg[1]							

Bu komutlar NOP, SIGNAL, SYNC, SYNC\_LOCK, ACTIVATE, DEACTIVATE ve RESET komutlarıdır.

Köprü ünitesi URB noktalarından birine (2.byte son 3 bit ile tanımlanacak) aşağıdaki formatta bilgi gönderebilmektedir. I2C yapısının sağladığı avantajlar kullanılarak bu iletişim çift yönlü gerçekleştirilebilmektedir.

	7	6	5	4	3	2	1	0
	0/1	RR	UR	packetSize - 2				
1	node address			0	0	0	0/1	
2	1/0	reserved			message box ID			
3	node address			0	0	0	1/0	
4	content byte 1							
	...							
n+3	content byte n							

*K3 Kartı:*

### (Ultrasonik Almaç Düğümü)

Engel sakınma işlemi için ses üstü sinyallerin su altı ortamında yayılma ve yansıma özellikleri kullanılmaktadır. Öncelikle su altı ultrasonik ses frekansı olan 200 kHz ile ses üstü sinyaller üretilerek, engel olup olmadığı test edilecek yöne gönderilmekte, daha sonra ise geri yansıyan ses üstü sinyali (eko) özelliklerine bakılarak engel tanıma işlemi yapılmaktadır.

Su altı ve su üstü aracında da birden fazla olmak üzere yerleri belirlenmiş kritik noktalara yerleştirilecek modüller yardımı ile engel sakınma işlemi gerçekleştirilebilecektir. Modül bünyesinde, ses üstü sinyalin üretilmesi, gönderilmesi ve geri yansıyacak sinyalin algılanması için gönderici ve alıcı birimlerini barındırmaktadır. Sesin su altı ortamındaki yayılma hızı bilindiğinden sinyalin gönderilme ve alma arasında geçen faz farkı (süre) belirlenerek olabilecek engelin varlığı ve uzaklığı tespit edilebilmektedir.

Kullanılacak modül analog sinyaller ürettiğinden dolayı, hem engel sakınma algoritmaları için kullanılmak üzere, hem de hesaplanan yansıma süresi –dolayısıyla ölçülecek mesafe- için bu sinyallerin sayısallaştırılması ve mikro-denetleyicilere aktarılması gerekmektedir. Bu sayısallaştırma ve mikro-denetleyici için kullanılacak sinyal dönüştürme devresi aşağıda verilmiştir. Bununla birlikte, sistemde bulunacak 6 farklı almaç için sadece bir adet kontrolcü kullanılması hem işlev hem de maliyet açısından uygun görülmüştür.

Engel sakınma altsistemi bünyesinde kullanılacak bu düğüm donanımı zaman paylaşımli olarak çalışmaktadır. Böylelikle her an sadece bir almaç çalışabilmekte, böylelikle ultrasonik yankılar (sistemin kendi ürettiği) en aza indirgenecektir. Aksi halde örneğin t anında a1 almacından yayılan ses sinyalleri t2 anında a2 almacından algılanabilecek bu da hatalı ölçüme neden olabilecektir. Bu gibi problemlerin üstesinden gelmek için röle tabanlı anahtarlayıcı modül kullanılmıştır. Önceki tasarımdaki dijital anahtarlı (4066 türevi) devre ultrasonik almacın davranışından dolayı problemler içermektedir. Örneğin almacın eşdeğer devresi paralel r-l-c olarak modellendiğinde ve ultrasonik anahtarlama modülü de 200 kHz ile çalıştırıldığında almaç tarafı boost çevirici gibi çalışacaktır. Yapılan ölçümlerde de almaç giriş gerilimleri 170V civarından başlamak üzere burst sinyaller gözlenmiştir.

Bu sürümde ise elektriksel filtre ve algılama-ölçme devreleri teke indirgenmiş, almaç seçici için ise baskılı devre tipi röleler uygun görülmüştür.

Bu devrede PIC18F2550 yongası kullanılmıştır.

EK

**K3 Kartı Malzeme Listesi:**

No:	Designator	Footprint	Quantity	Value
1	R1, R4	805	2	100R
2	R2, R3, R14, R21, R42	805	5	10k
3	R5, R26, R30, R33, R52	805	5	1k
4	R6, R53, R55	805	3	470
5	R7	805	1	1.5M
6	R8, R10	805	2	100
7	R9, R11	805	2	2.4k
8	R12	805	1	750k
9	R13, R15, R16, R36, R43	805	5	10k
10	R22, R23	805	2	5.1k
11	R24	805	1	56k
12	R25	805	1	511
13	R27	805	1	2200
14	R28	805	1	1200
15	R29, R31, R32	805	3	1000
16	R34	805	1	1210
17	R35	805	1	100k
18	R37	805	1	15k
19	R38	805	1	22k
20	C1	ELEC- RAD10	1	10u
21	C2	ELEC- RAD10	1	47u
22	C3, C4, C7, C11, C12, C15, C24, C26, C33	1206	9	100nF
23	C6	805	1	100nF
24	C9, C13, C29	1206	3	1000pF
25	C10, C14, C30	1206	3	1000pF
26	C25	ELEC- RAD10	1	4.7u
27	C27, C28	1206	2	680pF
28	C31, C32	CAP20	2	15p
29	U1	SO16	1	ULN2003A
30	U4	DIL14	1	LF347

31	U5	DIL28NAR	1	PIC18F2550
32	Q1-Q3, Q7	TO92	4	BC548C
33	D2-D4, D8-D10	1206D	6	1N4148
34	D14_V12, D15_V5, D17_TEST	1206	3	DIODE-LEDA
35	GND	PIN	1	GND
36	J1	TRANS 40 DIL	1	TRANS 40 DIL
37	J1_URB1, J17_URB2	CONN-SIL6	2	CONN-SIL6
38	J3_ECHO	PIN	1	ECHOPIN
39	J4_OUT	PIN	1	OUTPIN
40	J5_OPAMP	PIN	1	OPAMP_IN_PIN
41	J8_VREF	PIN	1	VREF_PIN
42	J14_PROG	CONN-SIL5	1	CONN-SIL5
43	JP1, JP2	CONN-SIL3	2	JUMPER2
44	L1, L3	DO41	2	1mH
45	RV1	PRE-VMT	1	POTENTIOMETER
46	X3	XTAL18	1	20MHz

### Önemli Pin Tanımları:

Tüm elektronik kartların haberleşmesi için kullanılan genel ULİSAR konektörü üzerinde:

29., 30., 31., 32, 33. ve 34. Pinler sırasıyla, Transducer1, 2, 3, 4, 5 ve 6 için röle seçme pinleridir. 200Khz lik referans sinyal için ise 18.pin kullanılmaktadır.

Bunun yanı sıra, yine röle kartında bulunan Kamera ve Işık kontrolleri ise 20. ve 19. pinlerdedir.

Kartın ihtiyaç duyacağı güç kaynakları ise 4.pin 5V, 2.pin 12V, 1. ve 40. Pinler ise GND olarak kullanılmaktadır.

### I5 Kartı Malzeme Listesi

No:	Designator	Footprint	Comment	Description	Quantity	Value
1	+3V3, +5V, +12V, P_OUT_5, P_OUT_12, P_OUT_33	PowerConn2	Clemens	2 Connector Plugin Screwed 3.81mm pitch PCBN connector	6	
2	5V_SEL, W1, W2	RAD-0.1	Jumper	Jumper Wire	3	
3	ANAL_FU, NAV, PM_PS	HDR2X10	Header 10X2, Power Monitor-Switch Conn, Header 6X2H	Header, 10-Pin, Dual row	3	
4	C1	C0805	Cap	Capacitor	1	10nF
5	C2	C0805+	Cap2	Capacitor	1	1uF
6	C3, C5, C7	CAPR5-4X5	Cap2	Capacitor	3	100uF
7	C4, C6, C8	RAD-0.3	Cap	Capacitor	3	1uF

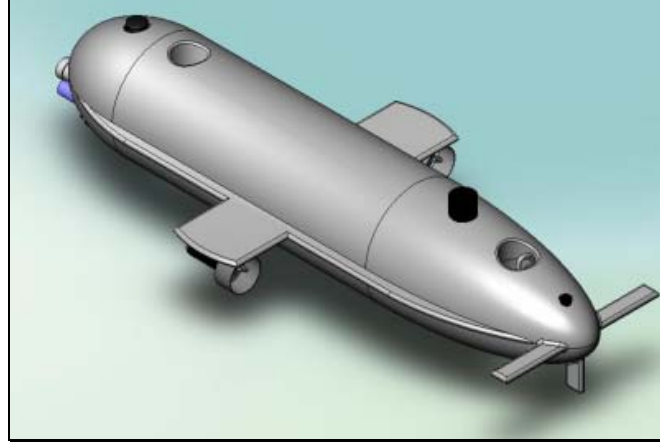
8	CPU_SERIAL	HDR2X9_200	Header 9X2	Header, 9-Pin, Dual row	1	
9	D1, D2, D3, D4, D5	3.5X2.8X1.9	LED3	Typical BLUE SiC LED	5	
10	Hum1, Hum2, Press	HDR1X3	Humidity Trans Header 3, Pressure Trans,Header 3	Header, 3-Pin	3	
11	MODEM, Prog_TMP, TMP1, TMP2, TMP3, TMP4	HDR1X5	Header 5, MODEM, Header 5, PicProgramming	Header, 5-Pin	6	
12	P1_ULISAR	HDR2X20	ULISAR_Internal	Header, 20-Pin, Dual row	1	
13	R1, R4, R5, R6, R7, R8, R11, R12, R13	R0805	Res2	Resistor	9	10k
14	R2	R0805	Res2	Resistor	1	1k
15	R3, R9, R10	R0805	Res2	Resistor	3	470R
16	U1	PDIP300-18	PIC16F88-I/P	Enhanced FLASH Microcontroller	1	
17	URB1, URB2	HDR1X6	URB_CONN, HDR6	Header, 6-Pin	2	





# ULİSAR

## Ulusal İnsansız Sualtı Aracı Projesi



### Universal Robot Bus (URB) Detaylandırılmış Tasarım Dökümanı

Tarih: 30.04.2010

# 1. Table of Contents

- 1. Table of Contents ..... 2
- 2. Revision History ..... 3
- 3. Referenced Documents ..... 3
- 4. Scope ..... 3
- 5. Introduction and Background to the Requirements ..... 3
  - a. Motivation and History ..... 3
  - b. Overview of the URB Infrastructure ..... 3
  - c. Central Processing Unit (CPU) ..... 4
  - d. The Bridge ..... 5
  - e. Hardware Nodes ..... 5
  - f. Definitions and Terminology ..... 5
- 6. Functional Requirements ..... 6
  - a. System Requirements ..... 6
  - b. Mechanical Requirements ..... 6
  - c. Hardware Requirements ..... 6
  - d. Software Requirements ..... 6
- 7. URB Nodes ..... 7
  - a. Downlink Electrical Specifications ..... 7
  - b. Downlink Communication Model ..... 7
  - c. Built-in Message Boxes ..... 7
    - i. Outbox 0: Node Configuration Information ..... 7
    - ii. Inbox 0: Node Commands ..... 8
  - d. Dowlink transactions ..... 8
    - i. Default reads ..... 9
    - ii. Default writes ..... 9
    - iii. Targeted reads ..... 9
    - iv. Targeted writes ..... 10
    - v. Compressed reads ..... 10
    - vi. Compressed writes ..... 11
- 8. The URB Bridge ..... 11
  - a. Uplink Communication Model ..... 11
    - i. Job Dispatch Packets ..... 11
    - ii. Job Response Packets ..... 12
    - iii. Node Request and Response Packets ..... 13
    - iv. Bridge Request and Response Packets ..... 13

## 2. Revision History

The following table shows the document revisions, dates and the corresponding contributing authors.

Revision	Date	Author	Comment
0.1	Mar 5, 2007		Initial revision created
0.2	Apr 4, 2007	Uluç Saranlı	Substantial section additions
0.3	Jun 14, 2007	Uluç Saranlı	URB downlink protocol details

## 3. Referenced Documents

Revision	Date	Author	Comment

## 4. Scope

The Universal Robot Bus (URB) is a communication infrastructure that facilitates easy and modular deployment of hardware modules that require real-time communications with a central control authority. This is a frequent need for small robotic systems wherein the management of various subsystems are carried out by dedicated microcontrollers and data has to flow to and from these independent computational entities.

This document details both low level design of the URB infrastructure as well as implementation details for its various subcomponents. All the relevant protocols and APIs are in the scope of this document.

## 5. Introduction and Background to the Requirements

### 5.1. *Motivation and History*

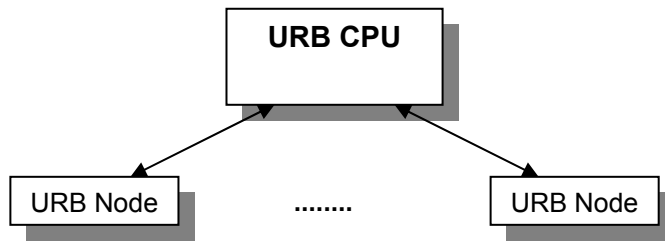
The need for a modular communications architecture with real-time properties arises from the frequent need to augment the capabilities of small mobile robot platforms intended for research purposes. In order to support data-acquisition and actuation capabilities of such a platform in an extensible way, modularity needs to be supported both in hardware and software. In many ways, motivations behind URB's design are very much in line with those of the Universal Serial Bus (USB), except that both the logical structure of the USB as well as its timeliness properties do not match very well to the requirements of real-time activities within a robotic platform.

Many of URB's components are reminiscent of components within USB, except that the overall complexity of the infrastructure is significantly lower. This is made possible by the specific context in which URB components are expected to operate. This simplicity is critical in being able to support the timely acquisition and delivery of sensor measurements and actuator commands.

The predecessor of URB is the RiSEBus, which was first developed by Haldun Komsuoğlu for the RiSE climbing robot project. Even though the initial design of URB is mostly identical to RiSEBus, later revisions of the infrastructure are expected to provide additional functionality that are not present in RiSEBus.

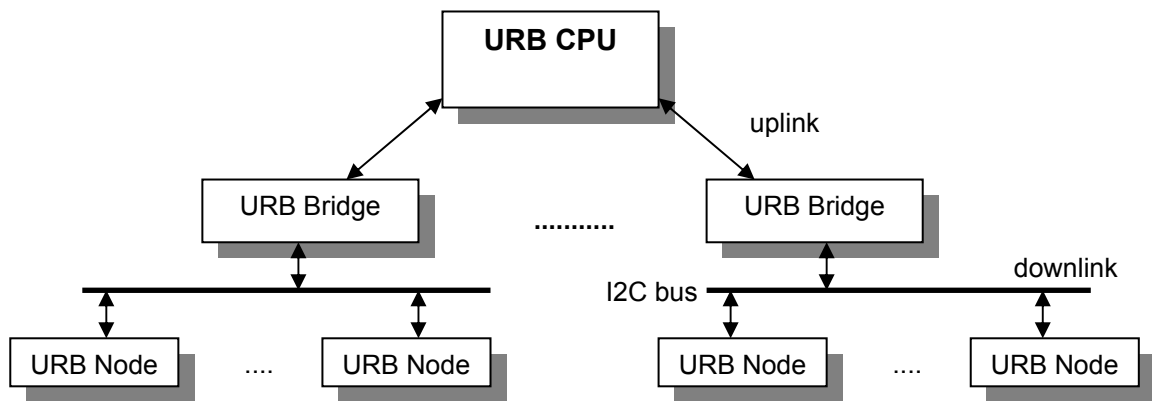
### 5.2. *Overview of the URB Infrastructure*

URB is a communications infrastructure for distributed digital control systems, facilitating hard real-time information exchanges between a set of "physically distributed" sensors and actuators (URB nodes) and a central computational unit (URB CPU).



**Figure 1: Logical topology of the Universal Robot Bus (URB)**

Logically, URB allows a central computational unit to directly communicate with a variable number of nodes, resulting in a star topology illustrated in Figure 1. This logical structure is implemented by a collection of user-level libraries both for the URB CPU and URB nodes, abstracting away from physical implementation details of the framework.



**Figure 2: Physical topology of a URB system**

In reality, however, the physical topology of the URB infrastructure, illustrated in Figure 2, incorporates "bridges" that master underlying I2C buses and manage communication requests from the URB CPU, directed to specific URB nodes in the system. Since individual I2C buses are shared among several nodes, these communication requests must be ordered and serviced accordingly by the bridge. The role of the URB bridge is very similar in spirit to that of USB host controllers. However, the most important differences between USB and URB is in their hardware topology. While URB adopts a two-tiered architecture with a variable number of bridges and their associated nodes, USB has multiple tiers separated by hubs and a fixed number of possible nodes for each tier.

A URB system consists of three types of devices: a **URB CPU** (central computational unit) which executes high level control algorithms; multiple **URB nodes** implementing sensing/actuation with basic signal conditioning; and **URB bridges** which form gateways between the PC and sets of hardware nodes. The following sections briefly describe each of these components.

### **5.3. Central Processing Unit (CPU)**

In all URB systems, the *URB CPU* is the central authority for the implementation of high level algorithms coordinating of activities for various components in the system. It contacts URB nodes to collect sensory information and apply actuation actions. In order to support this functionality, the CPU toolkit of the URB infrastructure consist of two components:

1. A standard application layer API that allows access to devices in the network as they appear in the logical topology,
2. Low level URB drivers that handle details of communicating with bridge implementations.

Note that different bridge implementations require different instances of drivers and associated API libraries. Furthermore, a URB system can employ multiple buses running in parallel controlled by

separate bridges. The application layer API on the CPU abstracts away from these details and presents a uniform collection of devices to the programmer. There are, however, some aspects of the API, which allow limited control over the behavior of individual bridges in the network to modulate their behavior. This slight deviation from the ideal star topology abstraction was introduced for flexibility in fine-tuning the timing of data acquisition.

## **5.4. The Bridge**

In a URB system, each I2C bus is controlled by a *URB bridge*, acting as a gateway between the CPU and hardware nodes on the bus. The main function of a URB bridge is to enable asynchronous communication with URB nodes in the system. Its primary responsibility is buffering data and asynchronously processing requests coming from the CPU for data exchange with hardware nodes, being configured as the master of the I2C based bus. Many different bridge implementations are possible, with the main difference being its uplink interface to the CPU.

Typically, the firmware on a URB bridge has three subsystems that operate asynchronously:

1. The uplink interfacing subsystem
2. The job control subsystem
3. The downlink interfacing subsystem.

The uplink interfacing subsystem oversees interactions with the CPU through the uplink. Its implementation depends on the type of the uplink (i.e. ISA, RS232, USB etc.). The job controller subsystem receives transactions requests and bridge control commands from the uplink and serves them in their order of priority and arrival time. Responses from completed requests are sent back to the CPU via the uplink.

In contrast, the downlink interfacing subsystem implements the I2C/SMB master functionality. It is used by the job control subsystem to contact and communicate with URB nodes, performing necessary transactions through the I2C bus.

## **5.5. Hardware Nodes**

In a URB system, *URB node* refers to all embedded devices interfacing with sensors and actuators, accessed through the I2C bus as slaves. Firmware on URB nodes typically consists of two layers:

1. The downlink interface layer, implementing basic I2C communications and downlink protocol details,
2. A task specific node application layer, dealing with the particulars of sensory data acquisition and actuator command updates.

A standard URB node library is provided for a number of I2C capable microcontrollers, facilitating the development of URB nodes to be connected to an existing system. A standard and simple API for this library ensures speedy development and compatibility for a wide variety of node designs.

## **5.6. Definitions and Terminology**

This section summarizes standard terminology used by this document and URB systems in general

- **URB CPU:** The central processing unit that coordinates sensory acquisition and actuation commands.
- **URB Bridge:** An intermediate entity that interfaces the CPU with various sensor and actuator nodes in the system
- **URB Node:** A self contained hardware entity that handles interfacing details and local processing for a collection of sensors and actuators.
- **URB System:** Any collection of CPU, bridge and node components
- **URB Uplink:** Connection between a URB CPU and a URB Bridge. There are several different uplink implementations using different communication methods such as ISA bus, RS232 and USB that can be used as the data link layer for the uplink.

- **Uplink Protocol:** Protocol for handling information exchange between the CPU and a URB bridge. Implementation details depend on the particular uplink of choice.
- **URB Downlink:** The connection between a particular URB bridge to all the nodes in its control. Currently, the I2C bus standard is used as the underlying data link layer.
- **Downlink Protocol:** Protocol for information exchange between the bridge and its associated nodes.

## 6. Functional Requirements

### 6.1. System Requirements

- **R.10 : Discovery.** The central CPU in a URB system must be able to discover types and identifying addresses for all nodes attached to the system. Discovery can be handled during initialization, so no support for hot-plugging is required.
- **R.20 : Node count.** The number of nodes attached to a URB system must be primarily limited by the available downlink and uplink bandwidths. No substantial structural limits should be imposed.
- **R.30 : Recovery.** The URB system should be able to recover from faults induced by malfunctioning nodes or bridges. Such faults should not cause a system lockup.
- **R.40 : Latency.** The latency between the time the CPU requests a piece of information from a node and the time it receives the requested information should not be more than 2ms.
- **R.50 : Synchronization.** There should be a mechanism to send a simultaneous signal to all nodes in the system to enable them to synchronize for data acquisition purposes. If it is not possible to accomplish this across multiple bridges, it should at least be possible within a single bridge.

### 6.2. Mechanical Requirements

- **RM.10 : Connector and cable standards.** The URB specification should dictate consistent cabling and connector standards for both downlink and uplink connections.

### 6.3. Hardware Requirements

- **RH.10 : Downlink speed.** The downlink for each bridge should support at least 1Mb/s raw communication speed across all associated nodes
- **RH.20 : Uplink speed.** The uplink connections should support at least 10Mb/s raw communication speed, cumulative across all bridges. This requirement may be relaxed for debug-oriented implementations such as the RS232 uplink.
- **RH.30 : Node power.** URB nodes receive power from their downlink connection. URB bridges must have the ability to turn power on and off for all the nodes associated with that bridge.

### 6.4. Software Requirements

- **RS.10 : URB node API.** A simple to use and versatile node library API should be provided.
- **RS.20 : URB node libraries.** The URB node API should be instantiated by libraries associated with a wide range of microcontrollers, including at least the SiLabs F3xx family.
- **RS.30 : URB CPU API.** A library API should be provided for the CPU side that presents the logical topology of the URB network to application programs.
- **RS.40 : URB uplink CPU drivers.** A driver layer should be provided for different uplink implementations such as ISA, USB and RS232 for different operating systems including Windows, Linux and QNX. These drivers will abstract away from implementation details of uplink connections and enable CPU libraries to implemented transparently.
- **RS.50 : URB CPU libraries.** The URB CPU API should be instantiated for a variety of operating systems including Windows, Linux and QNX. These libraries will use the uplink drivers to access the underlying uplink communication hardware in a transparent way.

## 7. URB Nodes

### 7.1. Downlink Electrical Specifications

Downlink connections require a 6-conductor cable. In addition to the I2C data and clock signals required to implement the downlink communication protocol, these connections also carry 3.3V and 5V, generated by the associated URB bridge. These can optionally be used by nodes which lack their own power supplies. Table 1 summarizes the pin assignments for URB downlink connectors.

Pin	Signal
1	I2C data line (SDA)
2	I2C clock line (SCL)
3	Signal ground (GND)
4	3.3V power (Vdd)
5	Signal ground (GND)
6	5V power (Vcc)

**Table 1: URB downlink connector pinout**

The I2C lines must be actively terminated by the bridge circuitry with 4mA current sources to enable communication speeds of up to 1Mbits/s. These current sources can be easily realized with a pair of matched PNP transistors connected between Vdd and the I2C lines. DMMT3906W chip can be used as the matched PNP transistor pair for each I2C line. Since the termination is at the bridge end, multiple parallel connections to the same downlink bus are allowed.

Nodes can optionally use the power supplied by the downlink connection but these may be noisy and must be filtered before being used by the node. Node designs must also avoid introducing ground loops by avoiding connections between the downlink ground and other ground inputs they may receive from other connections.

### 7.2. Downlink Communication Model

URB bridges communicate with its nodes using the I2C bus. The bridge is configured as the master of the bus and all nodes act as slaves. Consequently, every transaction must be initiated by the bridge.

Transactions between the bridge and its associated nodes adopt an abstraction in the form of **message boxes**. Each node is allowed to have up to **8 inboxes** for receiving input and **8 outboxes** for sending data out, each with a static fixed size selected during initialization. The bridge, being the master of the downlink connection, can request data to be read from or written to any of these message boxes based on requests coming from its uplink. Each such exchange is termed a **downlink transaction (DT)**, where a **downlink data packet (DDP)** is transmitted between the bridge and the node.

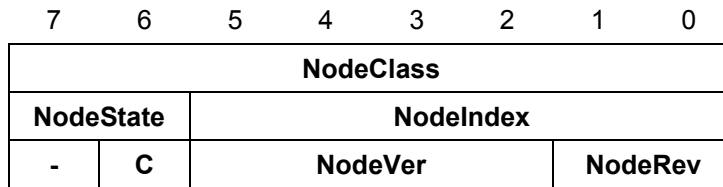
Each transaction encodes (either explicitly or implicitly) a **URB message box address (UMA)**, consisting of a 4-bit node address, a 3-bit message box id and a read/write flag. Sufficient information is provided in this address to unambiguously identify a single message box on a single URB node.

### 7.3. Built-in Message Boxes

Outbox 0 and Inbox 0 are reserved for providing node configuration information and sending URB related commands to a node, respectively. The size and layout of these message boxes are specified by the URB protocol and implemented by all relevant libraries. Various elements of the API are structured to prevent explicit access to the buffers for these message boxes so application code cannot interfere with basic URB operation.

#### Outbox 0: Node Configuration Information

The URB protocol uses OutBox 0 of each node to provide information necessary to identify the functionality of the node. As such, it is read by the discovery phase during initialization to build a catalog of which URB node addresses correspond to what kinds of functionality. The URB specification fixes the layout of Outbox 0 is as follows:



The type of functionality provided by each node is specified by its **NodeClass**. Different instances of nodes implementing the same functionality (such as multiple motor drivers connected to different motors in a robot) are distinguished by their **NodeIndex** values. These configuration parameters are selected by the node application during its initialization using specific methods provided by the node library API. Designers must ensure that no two nodes on the same downlink bus have the same class/index pair.

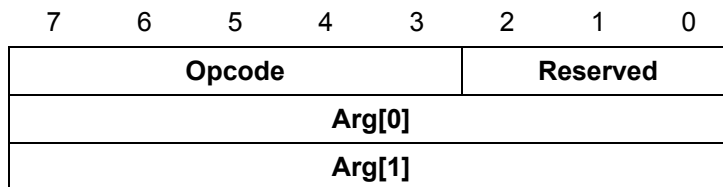
The **NodeState** field encodes the current state of the URB subsystem on the node. Possible values are: 00: Uninitialized, 01: Idle, 10: Active, 11: Reserved. Control software on the CPU can read Outbox 0 to find out about the current state of the URB subsystem on the node. More information on this can be found in the section below on Node Process Control.

The **NodeVer** and **NodeRev** fields encode the version and revision numbers for the firmware on the node. These fields can be retrieved by the control software on the CPU to ensure proper versioning of the functionality and message box formats implemented by the node.

Finally, the **C** bit specifies whether the node supports compressed downlink transactions. If it is set to 1, the URB bridge will use the compressed read and write operations instead of targeted read and writes, improving the overall communication efficiency on the downlink bus.

### Inbox 0: Node Commands

The URB protocol uses Inbox 0 of each node to send protocol commands to URB nodes. The layout of Inbox 0 is fixed as follows:



The opcode field contains one of the following commands:

Opcode	Command	Arg[0]	Arg[1]	Description
00000b	NOP	N/A	N/A	No operation
00001b	SYNC	N/A	N/A	Synchronization signal.
00100b	ACTIVATE	N/A	N/A	Activate node. Starts updates.
00101b	DEACTIVATE	N/A	N/A	Deactivate node. Stops updates.
11111b	RESET	N/A	N/A	Soft reset of the node firmware

When data is written into Inbox 0, the URB interface library decodes the command and performs the associated action. There is usually no need to directly access Inbox 0 from within application code.

### 7.4. Downlink transactions

The URB protocol implements 6 different transaction types that can take place over the downlink connection:

1. Default read: Reads from the "current outbox"
2. Default write: Writes to the "current inbox"
3. Targeted read: Reads from an explicitly specified inbox
4. Targeted write: Writes to an explicitly specified inbox
5. Compressed write: Writes to an inbox whose id is encoded as part of the I2C address



6. Compressed read: Reads from an outbox whose id is encoded as part of the I2C address
7. Broadcast write to inbox 0

The first four transaction types are used for node hardware that does not support software decoding of an I2C address. In contrast, if the microcontroller used to implement the node does support software decoding of the 7-bit I2C address, it becomes possible the more efficient, "compressed" transaction types. Finally, the broadcast write to inbox 0 is possible for both node types and can be used to send commands to all nodes at once.

Packet formats and details for each of these transaction types are explained in the following sections.

### Default reads

This type of transaction is used when the bridge wants to read from the same message box as the previous read transaction. Since the message box ID is implicitly evident from this specification, there is no need to send it explicitly through the downlink. For nodes which do not support compressed transactions, this results in substantial bandwidth savings, especially is the CPU reads explicitly from a single message box.

The first time a default read is done from a node after a hard or soft reset, data comes from Outbox 0. Consequently, the bridge can use default reads during discovery following a node reset.

The following table describes data exchanges that take place over the I2C bus corresponding to a default read transaction (from a message box of size n):

	7	6	5	4	3	2	1	0	
1	<b>node address</b>				<b>0</b>	<b>0</b>	<b>0</b>	<b>0(R)</b>	B->N
2	<b>content byte 1</b>								N->B
	...								
n+1	<b>content byte n</b>								N->B

**Table 2: I2C data transfers for default read transactions**

### Default writes

This type of transaction is used when the bridge wants to write to the same message box as the previous write transaction. Since the message box ID is implicitly evident from this specification, there is no need to send it explicitly through the downlink. For nodes which do not support compressed transactions, this results in substantial bandwidth savings, especially is the CPU reads explicitly from a single message box.

The first time a default write is done to a node after a hard or soft reset, data goes to Inbox 0.

The following table describes data exchanges that take place over the I2C bus corresponding to a default write transaction (to a message box of size n):

	7	6	5	4	3	2	1	0	
1	<b>node address</b>				<b>0</b>	<b>0</b>	<b>0</b>	<b>1(R)</b>	B->N
2	<b>content byte 1</b>								B->N
	...								
n+1	<b>content byte n</b>								B->N

**Table 3: I2C data transfers for default write transactions**

### Targeted reads

In contrast to default reads, targeted reads are structured to first explicitly transmit a specific message box ID and then perform a read operation from the newly configured message box. In fact, this transaction is nothing but a special write operation followed by a default read transaction.

The initial write operation is almost identical between targeted reads and targeted writes. The most significant bit of the first byte following the I2C address identifies whether this is to be a targeted read or targeted write operation. If this bit is 0, the transaction will be concluded and it will be followed by a default read transaction. Otherwise, as described in the next section, data bytes to be written to the targeted message box will follow.

	7	6	5	4	3	2	1	0	
1	node address				0	0	0	1(W)	B->N
2	0	reserved				message box ID			B->N
3	node address				0	0	0	0(R)	B->N
4	content byte 1								N->B
	...								
n+3	content byte n								N->B

**Table 4: I2C data transfers for targeted read transactions**

Performing a targeted read resets the "current outbox" to the outbox specified in the transaction.

### Targeted writes

In contrast to default writes, targeted writes are structured to first explicitly transmit a specific message box ID, followed by the data content. Unlike targeted reads, only a single I2C transaction is performed since the bus master does not need to change transmission modes during the transaction.

	7	6	5	4	3	2	1	0	
1	node address				0	0	0	1(W)	B->N
2	1	reserved				message box ID			B->N
3	content byte 1								B->N
	...								
n+2	content byte n								B->N

**Table 5: I2C data transfers for targeted write transactions**

Performing a targeted write resets the "current inbox" to the inbox specified in the transaction.

### Compressed reads

Some microcontrollers support software decoding of I2C addresses. This allows using the least significant three bits of the I2C address to explicitly specify a target mailbox for read or write operations, eliminating both the complexity and inefficiency of the transaction types described above.

The following table describes data exchanges that take place over the I2C bus corresponding to a compressed read transaction (from a message box of size n):

	7	6	5	4	3	2	1	0	
1	node address				message box ID			0(R)	B->N
2	content byte 1								N->B
	...								
n+1	content byte n								N->B

**Table 6: I2C data transfers for compressed read transactions**

## Compressed writes

Similar to compressed reads, compressed write transactions have the target message box ID encoded as part of the I2C address

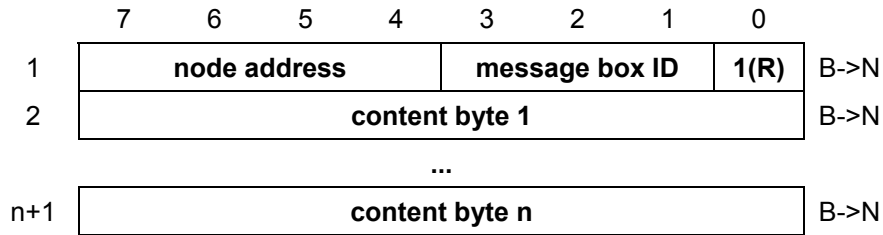


Table 7: I2C data transfers for compressed write transactions

## 8. The URB Bridge

### 8.1. Uplink Communication Model

As noted in the glossary, each uplink connection establishes a communication path between the CPU and one of the bridges. From the point of view of the bridge, the uplink is simply a bidirectional stream of bytes, structured into incoming **Job Dispatch Packets (JDP)** and outgoing **Job Response Packets (JRP)**. The incoming packets are also divided into two categories, **Normal Job Dispatch Packets (NJDP)** and **Urgent Job Dispatch Packets (UJDP)**. It can be implemented in a variety of ways, including RS232, USB, ISA bus or even SPI connections.

All packet types conform to a standard format illustrated in Table 1. Since the packet size is encoded as an integer equal to two less than the total packet size, the maximum possible total uplink packet size is 34 bytes.

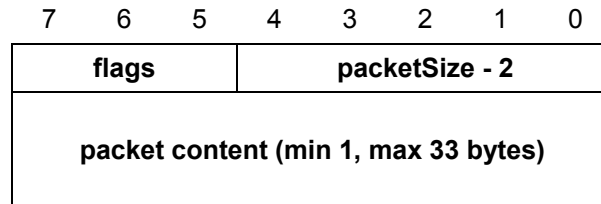


Table 8: URB uplink packet format (for both directions)

### Job Dispatch Packets

There are two different types of job dispatch packets: bridge requests and node requests. The type of issued request is determined by the **BR** flag, located most significant bit on the first packet byte. Tables Table 9 and Table 1 illustrate the detailed definitions of these packets. Two additional flags, **RR** and **UR** indicate whether a response is required for this packet and whether the job associated with this packet is urgent, respectively. Urgent packets are given priority over pending job requests, especially for uplink channels that involve parallel streams (such as a USB bridge with multiple endpoints).

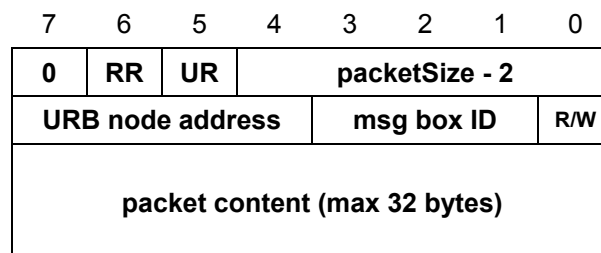


Table 9: Node request packet format (BR = 0)

Node requests initiate data transfer to and from a message box in a particular node connected to the bridge receiving the request. The second byte encodes the 4-bit URB node address, the message box ID on that particular node and a bit to select a read or write operation. As result of this encoding, URB supports up to 16 nodes per bridge as well as 8 inboxes and 8 outboxes on each node. Node requests are further discussed in Section 0.

7	6	5	4	3	2	1	0
1	RR	UR	packetSize - 2				
bridge command							
command arguments (max 32)							

**Table 10: Bridge request packet format (BR = 1)**

In contrast, bridge requests encode data and command exchanges between the CPU and the bridge. The second byte of the packet, the bridge command, indicates the nature of the request. Different bridge requests types are detailed in Section 0.

### Job Response Packets

If the **RR** field on a job dispatch packet is 1, the bridge is expected to send a response back to the CPU. Similar to dispatch packets, response packets also come in two flavors: bridge response packets and node response packets, illustrated in Tables Table 11 and Table 12. While node responses include a copy of the URB node address, message box ID and the read/write flag, responses to bridge requests include the response to the bridge request within the package content. The bridge command that generated the result is not copied back to the outgoing stream from the bridge. As a consequence, the CPU has to match responses to requests by keeping track of the order in which bridge requests were sent.

7	6	5	4	3	2	1	0
0	AR	UR	packetSize - 2				
URB node address			msg box ID			R/W	
response content (max 32 bytes)							

**Table 11: Node response packet format (BR = 0)**

7	6	5	4	3	2	1	0
1	ER	UR	packetSize - 2				
response content (max 33 bytes)							

**Table 12: Bridge response packet format (BR = 1)**

The response request (**RR**) flag of job dispatch packets is obviously unnecessary for response packets. Instead of that flag, node response packets include an "automatic response" (**AR**) flag, which indicates whether this response was generated as a result of an automatic node request issued by the bridge. This can arise in situations where the bridge is requested to periodically access a node and send the result back to the CPU. This flag is not necessary for responses to bridge requests since all

such requests have to be explicitly sent by the CPU. Instead, responses to bridge requests include an error flag **ER** and the response content includes a single byte encoding an error code.

### Node Request and Response Packets

As described above, node requests are simple direct access requests to a message box identified by the supplied URB node address and message box ID. The response content depends on whether the request is a read or write request.

For read requests, the response content contains the most recent data in the requested message box. The CPU and the addressed node must have common knowledge of the message box format. The URB bridge has no knowledge on the nature of data being transferred.

7	6	5	4	3	2	1	0
<b>0</b>	<b>AR</b>	<b>UR</b>	<b>packetSize - 2</b>				
<b>URB node address</b>				<b>msg box ID</b>			<b>0</b>
<b>message box byte 1</b>							
...							
<b>message box byte n</b>							

For write requests, the response content consists of a single byte including status bits related to the execution of the request, illustrated in Table 13. The OK bit indicates whether the write operation was successful or not.

7	6	5	4	3	2	1	0
<b>0</b>	<b>AR</b>	<b>UR</b>	<b>packetSize - 2</b>				
<b>URB node address</b>				<b>msg box ID</b>			<b>1</b>
<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>	<b>OK</b>

**Table 13: Node response packet format for write operations**

### Bridge Request and Response Packets

All bridge request packets include a command byte that indicates the type of operation that is to be carried out by the bridge. Table 14 summarizes all commands supported by the bridge, together with details on their number of arguments, response packets size and functionalities. Note that the package formats for custom downlink and uplink commands depend on the type of uplink and downlink and hence cannot be specified here. Please refer to the section or manual associated with your uplink and downlink types for more information.

One final note related to the echo command. Rather than sending a standard response packet, the echo command simply writes its argument to the outgoing bridge uplink stream. Consequently, it can potentially confuse the CPU uplink reception subsystem which depends on being able to reliably detect the starting points of response packets. Do not use the echo command unless you really know what you are doing.

<b>Bridge Command</b>	<b>#args</b>	<b>#resp</b>	<b>Description</b>
BRG_RESET_CMD	0	1	Reset the bridge and all nodes
BRG_GETVER_CMD	0	2	Send response packet with firmware version(1 byte) and revision (1 byte)
BRG_CLOCK_CMD	0	4	Send response packet with current bridge clock (4

			bytes)
BRG_LED_CMD	1	1	Set LED state and return previous state (1 byte)
BRG_PKT_COUNT_CMD	0	8	Send response packet with uplink (4 bytes) and downlink (4 bytes) packet counts
BRG_DISCOVER_CMD	0	2	Perform node discovery and report active nodes (2 bytes encoding availability of 16 nodes)
BRG_NODEINFO_CMD	1	4	Send the latest discovered class (1 byte), index (1 byte), version (1 byte) and revision (1 byte) information for the node with provided address.
BRG_NOP_CMD	0	1	No operation. Ignored by the bridge (returns 1 dummy byte)
BRG_TOKEN_CMD	0	4	Send a response packet with a special 4-byte token
BRG_ECHO_CMD	1	-	Directly echo argument through the uplink (dangerous!)
BRG_DL_GETVER_CMD	0	10	Send response packet with downlink version (1 byte), revision (1 byte) and 8 character type
BRG_DL_CUSTOM_CMD	var	var	Custom command to the downlink subsystem
BRG_UL_GETVER_CMD	0	10	Send response packet with uplink version (1 byte), revision (1 byte) and 8 character type
BRG_UL_CUSTOM_CMD	var	var	Custom command to the uplink system

**Table 14: Bridge command types, number of arguments, response content size and descriptions**

## 9. URB Evrensel Robot İçi İletişim Ağı Tasarımı- Türkçe Özet

(Universal Robot Bus (URB) Detailed Design Document)

Uluç Saranlı, Akın Avcı

### 10. Giriş

Bu dökümanın kapsamı, ULİSAR Sualtı Robot Platformu üzerinde, robot içinde değişik konumlarda bulunabilecek algılayıcı ve eyleyici ünitelerinin merkezi işlemci ile haberleşmelerini sağlayacak olan modüler ve gerçek zamanlı olarak çalışabilecek bir iletişim ağı protokolünün tasarımı ve gerçekleştirilmesi için gerekli standartların belirlenmesidir. Bu tasarım laboratuvarımızda gerçekleştirilmekte olan bir başka Tübitak projesi ile ortak bir çalışma biçiminde gerçekleştirilmekte ve birden fazla robot platformunda kullanımı umulmaktadır. Bu Türkçe özet, ana rapor ile birlikte ekli İngilizce doküman hakkında kısmi fikir vermek için hazırlanmıştır ve tüm bilgileri kapsamı beklenmemektedir.

Sualtı robot platformlarında ve diğer robotik platformlarda aktif çalışma sırasında oluşan donanım problemlerinin büyük bir yüzdesi karmaşık kablo ağlarında ve konnektörlerde ortaya çıkan sorunlar olmaktadır. Her algılayıcı ve eyleyici ünitesinin ihtiyacı olan güç ve sinyal bağlantılarının her biri merkezi işlemciye bireysel olarak bağlanacak olursa, ortaya çıkan bağlantıların karmaşıklığı çok çabuk bir şekilde kontrolden çıkabilmektedir. Bunun bir sonucu olarak robotun güvenilirliği azalmakta ve uzun süreli deneyleri gerçekleştirilmesi mümkün olamamaktadır.

URB protokolü bu sorunu temelden çözmek için robot üzerinde hem konum, hem de fonksiyon olarak birbirine yakın olan algılayıcı ve eyleyici ünitelerine olan bağlantıları, bu ünitelere yakın olan bir işlemci aracılığı ile gerçekleştirerek, ana işlemciye veri aktarımını bir I2C bağlantısı üzerinden çok az sayıda kablo kullanarak yapar. Bu sayede, karmaşık güç ve sinyal bağlantıları çok kısa tutularak, uzun mesafelerde veri akışı için gereken donanım basitleştirilmiş olur. Buna ek olarak, yerel işlemciler üzerinde yapılabilecek ön sinyal işleme veya kontrol adımları, merkezi işlemci üzerindeki yükü ciddi bir oranda azaltarak daha karmaşık algoritmaların gerçekleştirilmesine de olanak sağlayacaktır.

## 11. URB Mimarisi

### 11.1. Donanım Altyapısı

Modüler olması amacıyla URB, değişken sayıda alt birimin merkezi işlemciye bağlanabilmesini sağlayacak bir mimaride tasarlanmıştır. Tüm URB sistemlerinde üç değişik türde birim bulunmaktadır:

1. Merkezi işlemci: Bu birim, robotun davranış kontrollerini ve karmaşık algılayıcı işleme fonksiyonlarını yerine getirecek olan, hızlı ve yüksek kapasiteli bir işlemciden oluşur. PC tabanlı işlemci üniteleri bu birim için uygun olacaktır.
2. Köprü Üniteleri: Bu birimler merkeze değişik şekillerde bağlanabilen (URB, RS232, ISA vb.) ve sistemdeki alt birimler ile bağlantıyı sağlayacak olan ünitelerdir. Ünitelere giden I2C bağlantısı ile PC tarafından desteklenen bağlantı protokolleri arasında bir köprü görevi görür.
3. Alt birimler: Bu birimler, algılayıcı ve eyleyiciler ile doğrudan bağlantı kuracak olan ve değişik konumlarda bulunması uygun olan işlemci üniteleridir. Bir köprüye ait olarak çalışırlar ve I2C bağlantısında köle durumundadırlar. Merkezi işlemcinin köprü aracılığı ile gönderdiği komutları detaylarıyla yerine getirerek sonuçları geri göndermekle yükümlüdürler.

Bu mimari sayesinde, sisteme herhangi yeni bir alt birim eklemek istendiği zaman, kapasitesini doldurmamış olan bir köprüye bağlamak suretiyle bu gerçekleştirilebilir. Ayrıca protokol tarafından belirlenen otomatik keşif özelliği ile, merkezi işlemcinin bu yeni üniteyi açılışında tanınması mümkün hale gelecektir.

### 11.2. Alt Üniteler ile İletişim

Alt üniteler ile veri alışverişini kolaylaştırmak için, URB protokolü bir "mesaj kutusu" kavramını tanımlar ve kullanır. Her alt ünite, içeriği ve boyutu sabit ve uygulama tarafından seçilen sekiz adet gelen mesaj kutusu ve sekiz adet giden mesaj kutusu tanımlar. Merkezi işlemci alt ünitenin protokol adresini, mesaj kutusu numarası ve veri alışverişinin yönünü belirterek tüm bu mesaj kutularına ulaşabilir. Alt ünitenin tasarımcısı tarafından içeriği belirlenecek olan bu mesaj kutuları böylece merkezi işlemci ile tüm alt üniteler arasındaki bağlantıyı sağlamış olur.

Doğal olarak merkezi işlemci sistem ilk açıldığında hangi köprü üzerinde hangi alt ünitenin bağlı olduğunu bilemeyecektir. Modülerlik ve otomatik keşif özelliklerini sağlayabilmek için, sıfır numaralı giden kutusu protokol tarafından rezerve edilmiştir. Bu giden mesaj kutusu, alt ünitenin fonksiyonunu tanımlayan bir sınıf, aynı fonksiyona sahip üniteleri ayırtılabilmek için bir index ve ünite yazılımının versiyonunu belirten bir içeriğe sahiptir. Bu değerler alt üniteyi geliştiren programcı tarafından belirlenecek ve ünite yazılımının ilk yaptığı işler arasında sabitlenecektir. Bu mesaj kutusu daha sonra merkezi işlemci tarafından tüm URB adresleri üzerinden okunarak, hangi ünitenin hangi fonksiyona sahip olduğu otomatik olarak belirlenebilecektir.

URB bünyesinde, alt üniteler ile köprü arasındaki bağlantıda altı değişik türde veri alışverişinde bulunabilir:

1. Normal okuma: Bir önce okunan giden mesaj kutusundan veri okuma
2. Normal yazma: Bir önce yazılan gelen mesaj kutusuna veri yazma
3. Hedefli okuma: Belirli bir giden mesaj kutusundan veri okuma
4. Hedefli yazma: Belirli bir gelen mesaj kutusuna veri yazma
5. Sıkıştırılmış okuma: Hedefli okumanın sıkıştırılmış biçimi
6. Sıkıştırılmış yazma: Hedefli yazmanın sıkıştırılmış biçimi

Bu iletişim türlerinden hangisinin kullanılacağına köprü üzerinde çalışan yazılım karar verecektir. Her bir tür için protokol bünyesinde bir I2C veri alışveriş sırası belirlenmiştir ve protokole uyum açısından bu sıraya hem köprü hem de alt birimler tarafından uyulması gerekmektedir.

### 11.3. Köprü ile İletişim

Köprü ile merkezi işlemci arasındaki iletişim ise, URB standardına uygun olarak birçok değişik şekilde yapılabilir. ULİSAR robotu için planlanan USB bağlantısının haricinde, RS232 veya ISA gibi iletişim donanımlarının kullanımı da uygundur. Temelde bu bağlantının gerektirdiği temel özellik, güvenilir ve gerçek zamanlı, iki yönlü bir bağlantıdır. İdealde hem merkezden köprüye, hem de köprüden merkeze iletilen verilerin asenkron bir şekilde işlenebilmeleri en uygun durumu oluşturacaktır. Fakat USB gibi merkezi işlemci tarafından kontrol edilen bağlantıların da bu amaçla kullanılmaları mümkün olacaktır.

İletişimin içeriği, iki tür mesajdan ve köprü tarafından bunlara verilen cevaplardan oluşur. Bu mesaj türlerinden ilki, köprüye merkezi işlemci tarafından bazı temel komutların gönderilmesi için kullanılır ve alt birimler ile herhangi bir veri iletişime sebep olmaz. İkinci tür mesajlar ise, belirtilen bir veya birden fazla alt birim ile iletişimi başlatıcı ve bu veri alışverişinin gerçekleştirilmesi için gerekli tüm verileri içeren bir mesaj türüdür.

Bu iki mesaj türü için de, merkezi işlemci bir cevap isteyebilir. Her köprü komutu için verilen cevap değişik bir formata sahiptir. Buna karşın, alt birimler ile yapılması istenen okuma veya yazma işlemlerinin cevabı belirli bir standarda oturtulmuştur ve protokolü gerçekleyen tüm birim ve yazılım kütüphanelerinin bu standarda uymaları gerekmektedir.

## **12. URB Yazılım Öğeleri**

URB standart protokolünün en kolay kullanılabilmesi için, hem alt birimler, hem de merkezi işlemci için modüler ve kullanımı kolay yazılım kütüphanelerinin geliştirilmesi gerekmektedir. ULISAR robotunun üretimi ve bu robot ile araştırmalar sırasında en sık olarak yapılacak işlerden birisi de yeni algılayıcı ve eyleyici ünitelerinin sisteme eklenmeleri olacaktır. Bu çerçevede yeni tasarlanacak olan modüllerin yazılımının URB ile uyumlu olacak şekilde en hızlı bir şekilde geliştirilebilmesi için çok iyi dökümanite edilmiş ve kullanımı kolay bir kütüphanenin geliştirilmesi gerekmektedir. Bu kütüphanelerin henüz ön tasarımları yapılmış ve belirli mikroişlemciler için gerçekleştirilmesine başlanmıştır.

Doğal olarak yeni tasarlanan her alt birimin merkezi işlemci bünyesinde verileri işleyecek bir de sürücü yazılımına ihtiyaç duyulacaktır. Bu da merkezi işlemcide çalışacak olan programların URB protokol detaylarından bağımsız olarak en kısa zamanda güncellenebilmeleri için kullanımı kolay bir yazılım kütüphanesinin varlığını gerektirecektir. Bu kütüphanelerin de ana yapısı ve arayüz detaylarının tasarımına başlanmış, ön testler gerçekleştirilmiştir.

## **13. Sonuç**

Sonuç olarak, robot içi algılayıcılarla kablo bağlantısını en aza indirerek, merkezi işlemci üzerinden bu algılayıcılara en kolay bir şekilde ulaşılabilmesini sağlamak için tasarlamaya başladığımız URB standart protokolünün ön tasarımı bu dökümanda özetlenmiştir. Projenin ilerleyen safhalarında bu protokol tamamlanarak robot üzerinde kullanılmaya başlayacaktır.



## TÜBİTAK PROJE ÖZET BİLGİ FORMU

**Proje No:** 105E127

**Proje Başlığı:** ÇOK MAKSATLI ULUSAL İNSANSIZ SUALTI ARACI PROJESİ (ULİSAR)

**Proje Yürütücüsü:** Prof. Dr. M. Kemal Leblebicioğlu

**Araştırmacılar:**

Prof. Dr. Kemal İDER

Prof. Dr. Kemal ÖZGÖREN

Yüksek Müh. Haldun MİDOĞLU

Prof. Dr. Uğur HALICI

Y. Doç. Dr. İlkay ULUSOY

Prof. Dr. Levent PARNAS

Doç. Dr. Halit OĞUZTÜZÜN

Prof. Dr. Haluk AKSEL

Prof. Dr. Volkan ATALAY

Prof. Dr. Gözde BOZDAĞI

Prof. Dr. Yasemin Yardımcı ÇETİN

Prof. Dr. S. Sencer KOÇ

Y. Doç. Dr. Afşar SARANLI

Yüksek Müh. Emre Ege

Yüksek Müh. Hüseyin Yiğitler

Yüksek Müh. Alper Kılıç

Yüksek Müh. Necmettin Cevheri

Seda Karadeniz

Mine Cüneyitoğlu

Yüksek Müh. Örsan Aytekin

**Diğer Araştırmacılar:** Bu arkadaşlar resmi olarak proje çalışanları arasında gözükmeyp tezleri dolayısı ile projeye katkıda bulunmuşlardır.

Yüksek Müh. Deniz Üsteğmen Kadir İsiyel

Yüksek Müh. Yavuz Kırılı

Yüksek Müh. Murat Deniz Aykın

**Projenin Yürütüldüğü Kuruluş ve Adresi:** ODTÜ, Elektrik ve elektronik Mühendisliği Bölümü, 06531, Ankara.

**Destekleyen Kuruluş(ların) Adı ve Adresi:**

1. Barış Elektrik Endüstrisi A.Ş.
2. Yenel Ltd.

**Projenin Başlangıç ve Bitiş Tarihleri:** 1.06.2006 – 31.05.2009

**Öz (en çok 70 kelime)**

Bu proje sualtı gözleme amacı ile yapılan bir insansız sualtı aracı hakkındadır. Temel fikir olarak, sualtı aracı ile beraber hareket eden ve sualtı aracına fiber kablo ile bağlı bir suüstü aracı olması düşünülmüştür. Sualtı ve suüstü araçları aralarında fiber üzerinden haberleşirken, suüstü aracı ile ana gemi arasındaki haberleşme rf üzerinden planlanmıştır. Böylece gerçek zamanlı navigasyon mümkün olacaktır. Mayın tarama görevleri için de kullanılabilmesi için gövde kompozit malzemeden imal edilmiştir.

**Anahtar Kelimeler:** İnsansız sualtı aracı (uuv), navigasyon, kompozit malzeme, fiber optik kablo, sürücüler, alıcılar, sonar, akustik modem, kamera, projektör.

**[Fikri Ürün Bildirim Formu](#) Sunuldu mu?** Gerekli Değil

Fikri Ürün Bildirim Formu'nun tesliminden sonra 3 ay içerisinde patent başvurusu yapılmalıdır.

**Projeden Yapılan Yayınlar:**

1. H. Yiğitler, K. Leblebicioğlu, "Online Calibration of Strapdown Magnetometers", 9th IFAC Symposium on Robot Control, Nagaragawa Convention Center, Gifu, Japan, September 9-12, 2009.
2. N. Cevheri, F. Şenel, K. İder, H. Aksel, L. Parnas, "Bir İnsansız Sualtı Aracı: ULİSAR Gövde Tasarımı", SAVTEK 2008, Savunma Teknolojileri Kongresi, Cilt I, s. 549-556, 26-27 Haziran 2008, ODTÜ, 2008.
3. N. Cevheri, H. Aksel, "Hydrodynamic and Thermal Analysis of an Autonomous Underwater Vehicle (in Turkish)", 2008 FLUENT Users Conference, Bilkent, Ankara, October 7-9, 2008.
4. K. Leblebicioğlu, "ULİSAR: İnsansız Bir Sualtı Aracı", 2008 TOBB Ekonomi ve Teknoloji Üniversitesi, Çok Erkinli ve Haberleşmeli Dinamik Kontrol Sistemleri Çalıştayı, 1 Temmuz,

2008.

**Projeden Doğrudan Yapılan Tezler:**

1. K. Isiyel, "Autopilot Design and Guidance Control of ULISAR UUV (Unmanned Underwater Vehicle)", Yüksek Lisans Tezi, ODTÜ, 2007.
2. N. Cevheri, "Computer Aided Engineering of an Unmanned Underwater Vehicle", Yüksek Lisans Tezi, ODTÜ, 2010.