INTELLIGENT METHODS FOR DYNAMIC ANALYSIS AND NAVIGATION OF AUTONOMOUS LAND VEHICLES

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

BY

HÜSEYİN BURAK KAYGISIZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF
DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JULY 2004

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 Doctor of Philosophy.

Prof. Dr. Mübeccel Demirekler
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 Doctor of Philosophy.

Prof. Dr. İsmet Erkmen	Prof. Dr. Aydan Erkmen
Co-Supervisor	Supervisor

Examining Committee Members

Prof. Dr. Erol Kocaoğlan (METU, EE)

Prof. Dr. Aydan Erkmen (METU, EE)

Assoc.Prof. Dr. Ozan Tekinalp (METU, AE)

Assist. Prof. Dr. Yakup Özkazanç (Hacettepe, EE)

Dr. A. Pınar Koyaz (TUBİTAK-SAGE)

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 : Hüseyin Burak Kaygısız

Signature :

ABSTRACT

INTELLIGENT METHODS FOR DYNAMIC ANALYSIS AND NAVIGATION OF AUTONOMOUS LAND VEHICLES

Kaygısız, Hüseyin Burak
Ph.D., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Aydan ERKMEN
Co-Supervisor: Prof. Dr. İsmet ERKMEN

July 2004, 182 pages

Autonomous land vehicles (ALVs) have received considerable attention after their introduction into military and commercial applications. ALVs still stand as a challenging research topic. One of the main problems arising in ALV operations is the navigation accuracy while the other is the dynamic effects of road irregularities which may prevent the vehicle and its cargo to function properly. In this thesis, we propose intelligent solutions to these two basic problems of ALV. First, an intelligent method is proposed to enhance the performance of a coupled global positioning/inertial navigation system (GPS/INS) for land navigation applications during the GPS signal loss. Our method is based on using an artificial neural network (ANN) to intelligently aid the GPS/INS coupled navigation system in the absence of GPS signals. The proposed enhanced GPS/INS is used in the dynamic environment of a tour of an autonomous van and we provide the results here. GPS/INS+ANN system performance is thus demonstrated with the land trials. Secondly, our work focuses on the identification and enlargement of the stability region of the ALV. In this thesis, the domain of attraction of the ALV is found to be patched by chaotic and regular regions with chaotic boundaries which are extracted using novel technique of cell mapping equipped with measures of fractal dimension and rough sets. All image cells in the cellular state space, with their individual fractal dimension are classified as being members of lower approximation (surely stable), upper approximation (possibly stable) or boundary region using rough set theory. The obtained rough set with fractal dimension as its attribute is used to model the uncertainty of the regular regions. This uncertainty is then smoothed by a reinforcement learning algorithm in order to enlarge regular regions that are used for chassis control, critical in ALV in preventing vibration damages that can harm the payload. Hence, we will make ALV work in the largest safe area in dynamical sense and prevent the vehicle and its cargo.

Keywords: Inertial Navigation System (INS), Global Positioning System (GPS), Artificial Neural Network (ANN), Autonomous Vehicles, Aided Navigation, Chaos, Nonlinear Analysis.

ÖZ

OTONOM KARA ARAÇLARININ DİNAMİK ANALİZİ VE SEYRÜSEFERİ İÇİN AKILLI METODLAR

Kaygısız, Hüseyin Burak
Doktora, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Aydan ERKMEN
Ortak Tez Yöneticisi: Prof. Dr. İsmet ERKMEN

Temmuz 2004, 182 sayfa

Otonom kara araçları (OKA) askeri ve ticari kullanımlara girmesinden sonra çok fazla ilgi çekmeye başlamışlardır. OKA'lar hala önemli bir araştırma konusu olmaya devam etmektedirler. OKA uygulamalarında karşılaşılan ana sorunlardan biri sistemin operasyonu sırasında karşılaşılan seyrüsefer hassasiyeti iken bir diğeri yoldan gelen ve araç ile yükünün çalışmasını engelleyebilecek dinamik etkilerdir. Bu tezde, OKA'nın bu iki sorunu için akıllı çözümler önerilmiştir. Öncelikle, küresel konumlama sisteminden (KKS'den) sinyal alınamadığı durumlarda tümleştirilmiş küresel konumlama/ataletsel seyrüsefer sisteminin (KKS/ASS) kara sistemlerindeki başarımını yükseltmek için bir akıllı yöntem önerilmiştir. Yöntemimiz KKS sinyalinin olmadığı zamanlarda KKS/ASS sistemine yardım etmek için kullanılan bir yapay sinir ağına (YSA) dayanmaktadır. Önerilen YSA yardımlı KKS/ASS sistemi dimaik bir testle denemiş ve sonuçlar burada verilmiştir. Böylece, KKS/ASS+YSA sisteminin başarımı yer denemeleri ile gösterilmiştir. Bundan sonra, çalışmamız OKA'nın kararlılık bölgesinin belirlenmesi ve genişletilmesi üzerine yoğunlaşmıştır. Bu tezde, OKA'nın çekim bölgesinin kaotik ve kesirli boyut ile pürüzlü kümeler kullanılan hücreden hücreye dönüşüm ile ortaya çıkarılan kaotik sınırlara sahip düzenli alanlardan oluştuğu bulunmuştur. Pürüzlü küme teorisi kullanılarak, hücresel durum uzayındaki bütün hücreler, kesirli boyutlarına göre alt yaklaşımın (kesinlikle kararlı), üst yaklaşımın (olasılıkla kararlı) veya sınır bölgesinin üyeleri olarak asınıflandırılmıştır. Elde edilen pürüzlü küme düzenli bölgedeki belirsizliği modellemek için kullanılmıştır. Bu belirsizlik, araç şasisi denetimi için kullanılacak ve titreşimden oluşacak yük bozulmalarını önleyecek düzenli bölgeyi genişletmek amacı ile bir zorlamalı öğrenme yöntemi kullanılarak düzgünleştirilmiştir. Bu sayede, OKA'nın dinamik anlamda olabilecek en geniş güvenli alanda çalışması ve araç ile yükün korunması sağlanmıştır.

Anahtar Kelimeler: Ataletsel Seyrüsefer Sistemi (ASS), Küresel Konumlama Sistemi (KKS), Yapay Sinir Ağı (YSA), Otonom Araçlar, Yardımlı Seyrüsefer, Kaos, Doğrusal Olmayan Analiz.

To My Parents, My Brother and My Wife

As a small gift for all your support and love

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor Prof. Dr. Aydan Erkmen for her continuous support, encouragement, guidance, contributions during my Ph.D. study. Her detailed comments and constructive criticism have significantly improved the work.

Also, I would like to express my deepest appreciation to my co-supervisor, Prof. Dr. İsmet Erkmen for his support, discussions and wonderful friendship throughout my Ph.D. study. Especially, his support during the conferences, were unforgettable.

I wish to express my deepest thanks to Asst. Prof. Dr. Yakup Özkazanç for the depth of his support, constructive ideas and his invaluable suggestions. I have gained a lot from the discussions we had.

I am grateful to all of my colleagues at TÜBİTAK-SAGE, especially to all the fellows in Guidance and Control Group. My work and discussions with them created an ideal working environment for my research. Special thanks are given to Yüksel Subaşı for his last long friendship and for all the enjoyable conversations.

TÜBİTAK-SAGE, which supported my research and supplied the test equipments is greatly acknowledged.

Finally, I owe all to my lovely parents, my brother and my wife for their unlimited and unconditional love, support, guidance, care and encouragement. My dear mother, I have always felt your endless love and best wishes with me. My dear father, your kind guidance, cooperation, and constructive criticism have been always enlightened me in establishing my life and career. My dear brother, your support, help and encouragement have given me great confidence througout my life.

I would also like to thank to my wife Aybike, for her everlasting love, encouragement and patience. I have always felt her endless support with me.

Last but not the least, thanks to my little son Metehan who has been the joy of my life after joining our family.

TABLE OF CONTENTS

CHAPTER

APPENDICES

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

LIST OF SYMBOLS

SYMBOL	
A	amplitude of impressed vibration
a, b	distance of c.g. from back and front tires
$\mathbf{B}, \mathbf{F}, \mathbf{G}$	control gain, state dynamics, noise gain matrices
$\mathbf{C}_b^n, \mathbf{C}_n^b$	body to navigation and navigation to body transformation matrix
c_{i1}, c_{i2}	damping constants of front and back tires
d_c, d_L, d_i, d_{cor}	fractal dimensions
$d_{jp}, w_{ji}^{[s]}$	desired output and synaptic weight of neural network
$\mathbf{f}^b$	specific force expressed in body frame
$\mathbf{g}, \mathbf{g}^n$	gravity and gravity expressed in navigation frame
$\mathbf{H}, \mathbf{\Phi}$	Kalman filter measurement and state transition matrices
k_{i1}, k_{i2}	spring constantsof fron and back tires
L, λ, h	lattitude, longitude, height

m	vehicle mass
R_N, R_E	north and east radii of earth at the current location
R_n	regularity number
$\mathbf{P}, \mathbf{Q}, \mathbf{R}$	Kalman filter state, input and measurement covariance matrices
$\mathbf{u}, \mathbf{w}$	control input and driving noise vectors
$\mathbf{v}, \mathbf{p}$	velocity and position vectors
v_n, v_e, v_d	north, east and down velocities
$x, \delta x$	state and error state
$\mathbf{x}, \delta\mathbf{x}$	state and error state vectors
y, θ	displacement and pitch angle of the vehicle chassis
$\mathbf{z}$	Kalman filter measurement
Z_i	ith cell
α, ω	phase and frequency of vibration
λ_i	lyapunov exponent
$\mathbf{\Omega}$	earth's rotation rate
$\mathbf{\Omega}_{nb}^{b}$	skew-symmetric matrix of rotation of $\boldsymbol{\omega}_{nb}^{b}$

$\boldsymbol{\omega}_{ib}^{b}$	rotation of body w.r.t. inertial frame expressed in body frame
$\boldsymbol{\omega}_{nb}^{b}$	rotation of body frame w.r.t. navigation frame expressed in body frame
$\boldsymbol{\omega}_{ie}^{n}$	rotation of earth frame w.r.t. inertial frame expressed in navigation frame
$\boldsymbol{\omega}_{en}^{n}$	rotation of navigation frame w.r.t. earth frame expressed in body frame

CHAPTER 1

INTRODUCTION

1.1 Motivation

Autonomous vehicles (AVs) have received considerable attention after their introduction into the military and commercial applications such as all terrain tracking, agriculture, exploration and surveillance. AVs still stand as a challenging research topic due to their high versatility and dynamical complexity, where one of the major problems arising in AV operations is the navigation coupled to the dynamic effects of road irregularities which may prevent the vehicle and its cargo such to function properly. Nowadays, sophisticated navigation systems have been developed and applied to military and commercial vehicles based on the integration of global positioning system (GPS) and inertial navigation system (INS). For the last 20 years, inertial navigation system has been coupled with global positioning system in different levels of integration and has been the research focus in space missions and military arena towards enhancing their accuracy providing high frequency and reliable navigation data. On the other hand, the increase of mass production capabilities and the emergence of new technologies for the inertial sensors triggered the use of low cost GPS/INS systems in wide spread civil application areas such as automotive, robotics and more specifically unmanned autonomous vehicles. In the future, the applications are expected to spread further over the commercial area for autonomous land vehicles for environment monitoring/agriculture/mineral exploration. However, low cost GPS/INS systems applied to autonomous vehicles exhibit a major problem stemming from their nature. It has been shown that the integrated navigation system yields very accurate navigation solutions when GPS signals are available, but not when the GPS signal is lost: INS, consisting of low cost sensors, drifts very fast during the GPS signal loss and thus prevents the host vehicle to function properly.

Another problem in AV operation is the dynamic effects of road irregularities which may produce chaotic vibrations within the vehicle cabin and cause hardware damage. Thus, autonomous military and civil vehicles should be coupled to a control in nonchaotic regions of the state space. It is well known that highly nonlinear systems state spaces are contaminated by uncertainty and unpredictability. The state space reveals itself having very confined regions good for control among patches of chaotic regions, all separated by the “predictability plasma”. The “plasma” as connoted in this thesis is vast broken area in state space where possibly chaotic and possibly controllable states are mixed leading to high unpredictability. Confining the operational state space of the vehicle to only the deterministically controllable region is a weak solution since it means loss of richness of information in control. Human beings acting as a controller can often take risks and also use possibly controllable stable states together with the deterministically controllable stable ones. This is a major ability of experience-based risk taking that learning human systems have. In this study, we aim at developing a similar human inspired risk-taking learning system which optimally increases the operational state space of an AV by mending regions in its state space that are contaminated by uncertainty during its navigation on a highly irregular terrain and at developing a more reliable GPS/INS system with the aid of intelligent techniques for AV navigation.

1.2 Problem Characteristics

Modern autonomous land vehicles designed to function in missions such as environment monitoring, automotive navigation, field surveillance, agriculture, utilize GPS integrated low cost INS for navigation purposes due to its cost effectiveness and relatively constrained accuracy: The integrated navigation system yields very accurate navigation solutions for a low maneuvering vehicle only when operating in an environment where there is a continuous access to GPS signals.

In real applications, an autonomous vehicle entering into a tunnel, a downtown area with high buildings, a canyon or a forest is incapable of receiving the GPS signal that is so critical for navigation accuracy. Moreover, GPS jamming can prevent the system to function properly. Consequently, under such events vehicle navigation depends only on INS that consists of low cost sensors and drifts very fast during the GPS signal loss. Thus autonomous vehicles can be lost in the most critical part of a mission such as a rescue operation in a canyon or a disaster area.

In order to circumvent the problem, higher grade inertial sensors or additional aiding sensors other than GPS can be used. The first approach clearly leads to a very expensive solution, which may not be always possible or desirable. The second means additional sensors, not only increasing again the cost, but laso bringing their own sensitivity to noise and adding complexity to the system.

In order to handle such a problem in a cost effective manner, GPS/INS behavior patterns should be generated in order to provide position estimation for an effective adaptive and intelligent support to the INS in cases of GPS signal loss. Since navigation accuracy is high during GPS signals, it is a period where the GPS/INS behavior patterns can be learned generating an experience memory of the integrated navigation system. When GPS is silent, this memory can then act as a model reference such that the previously learned pattern can be used to estimate position data for the INS in order to prevent the system from drifting.

Another problem for ALV stems from the dynamical environment that the vehicles are exposed to. During the operation of the ALV, its chassis frequently undergoes oscillations bearing uncertainty; possibly leading to chaotic behaviors which may damage its cargo so may prevent it from functioning properly during the mission. An intelligent control that handles uncertainty and suppresses the vibration is then critical for the health of mission. However, such a control mechanism requires the precise investigation of the chaos map of the states related to the oscillatory motion, in order to decrease the uncertainty originating from these oscillations. Decreasing uncertainty will thus tend to the enlargement of regular regions used for control. Regular regions of the vehicle/road system need then to be generated. Uncertainty contamination of other regions should be detected and measured. A smoothing technique is needed to be developed in order to extract useful data in the contaminated area. The precise map of ALV on the irregular road state space to be used by the controller should then be the regular regions enlarged by decontaminated data that would prevent the ALV cargo from damage during its travel.

Inspiration from nature has shown in recent studies that control under uncertainty benefits from information richness, which can be achieved in our problem by enlarging controllable uncertainty secured regions using the decontaminated data.

1.3 Objective, Methodology and Contributions

The objective of this thesis is to develop an intelligently aided Inertial Navigation System (INS)/Global Positioning System (GPS) for use in tactical and civil applications and detecting and avoiding possible onset of chaos in the system.

The demand for high performance integrated navigation systems which can be used on tactical missiles, robots, automobiles and unmanned vehicles and maintain the high reliability and accuracy even in the absence of GPS signal, has been the main focus of research works. This thesis work aim at providing continuously accurate Artificial Neural Network (ANN) aided GPS/INS integrated navigation system for tactical and commercial applications. The ANN is trained online when GPS signal is on to learn GPS/INS integration patterns and apply them in the case of GPS signal loss. The trained ANN then uses to complete the observations with estimated ones, when there is no real observation (GPS loss), acting therefore as an intelligent estimator.

The methodology developed in this thesis is further implemented on an ALV with a low cost integrated navigation system. The real-time architecture is provided and verified by experiments using real data.

The intelligent structure aid to the GPS/INS for increased performance even under GPS signal loss using an ANN is investigated for different scenarios around the METU campus in real time tests.

After developing the intelligent navigation structure, we have focused on the detection and avoidance of the onset of chaos. This is achieved using a new approach combining fractal theory and rough set theory in order to define the regular regions of nonlinear systems in a more precise manner. This approach conducts global stability analyses by cell to cell mapping, determines the boundary region of the attraction domain using rough set and fractal dimension concepts and smooths uncertainty in the boundary region via an environment guided learning algorithm. In order to avoid the problem of chaos and supply regular navigation to the vehicle cargo cabin, a global stability analysis is run on the system employing cell to cell mapping and the response of the vehicle to different road conditions is investigated. Moreover, the uncertainty stemming from the nature of chaotic region is smoothed as the roughness in the regular stability domain, modelled by rough set using reinforcement learning.

The main contributions of this thesis to the general field of navigation and ALV dynamic can be summarized as follows:

- Development and implementation of a GPS integrated tactical grade inertial navigation system. Such a real-time implementation is important in understanding the practical issues and characteristics of the system. System characterization is achieved by real-time field tests.

- Development of an inertial navigation system aided by artificial neural network structure providing position/position difference estimates. This system requires no external measurement and yields a low cost accurate navigation solution in the absence of GPS signal. The performance of the intelligent navigation system is demonstrated by real-time tests and all pros and cons of the system are analyzed.

- Analysis of vehicle/road interaction for irregular roads.

- Development of a new approach to the identification of the chaotic boundaries of regular (periodic and quasiperiodic) regions in nonlinear systems, using cell mapping equipped with measures of fractal dimension and rough sets. The fractal/rough set model developed is a new approach brought to the area of the uncertainty modelling in chaotic systems.

1.4 Content of the Thesis

Chapter 2 introduces the well known implementation of inertial systems using global positioning system. This integration is achieved by the implementation of an extended Kalman filter in a loosely coupled GPS/INS structure. The existing GPS/INS system performance is also illustrated through experimental results.

Chapter 3 develops our proposed approach, an intelligent ANN enhancement to a GPS/INS system with an analysis for an optimum neural network architecture equipped with an efficient learning algorithm. This chapter dwells with the generation of an optimum multilayer perceptron (MLP) structure with relevant number of layers and perceptrons, and an appropriate learning algorithm and discusses the most suitable performance based parameter of the neural network. This chapter thus includes performance analysis based on system parameter changes and presents the sensitivity analysis of the neural network.

Chapter 4 gives the experimental results for the intelligent navigation system proposed.

Chapter 5 presents a model for the ALV on irregular road and analyses global stability according to changing road conditions.

Chapter 6 conducts a state space analysis of chaotic and uncertain vehicle dynamics. Attraction domains of stable behaviors are generated and their boundary domains are determined. This chapter also introduces our model of the roughness in stability of this domain using rough set and the fractal dimension concepts. Moreover, we smooth uncertainty roughness in this boundary region via a reinforcement learning algorithm so as to reach the largest regular region possible for efficient control.

Chapter 7 concludes the thesis.

CHAPTER 2

THE EXISTING WORKS: GPS/INS INTEGRATION

2.1 Introduction

An inertial navigation system (INS) is basically formed of a cluster of accelerometers used to sense acceleration components in three dimensions and a computer. The orientation of the accelerometers with respect the reference frame is determined using rotation sensors (e.g., gyroscopes). Accelerometers measure the specific force component of acceleration instead of total acceleration due to basic physical limitations. In order to determine the total acceleration gravitational acceleration is added to the accelerometer outputs where gravity is calculated in the navigation computer as a function of INS position. Two basic INS approaches exist to generate the specific force, thus acceleration, from the accelerometers and determine the position with respect to a reference frame; these are the gimbaled and the strapdown approaches.

In the gimbaled approach, accelerometers are mounted on a gimbaled platform that remains aligned with the reference frame. This alignment is achieved by constantly actuating the gimbals with respect to the transition of the reference frame. Thus the accelerometers are directly integrated to provide velocity and position in the reference frame.

In the strapdown approach, accelerometers and gyroscopes are mounted directly to the body of vehicle. Rotation rates measured by the gyroscopes are constantly used to update the transformation between body and reference frames. Accelerometer measurements are then passed through this transformation to obtain the acceleration in the reference frame. Thus the transformed acceleration is integrated to provide velocity and position.

In general, the difference between a strapdown and gimbaled system can be seen as a tradeoff between mechanical and computational complexity. A gimbaled configuration requires less computational process since the accelerometers are maintained in the reference frame. However, this requires a high mechanical complexity. From a performance point of view, gimbaled systems are better since the strapdown systems are exposed to full vehicle rotation rate, thus higher dynamic range. As a result, higher dynamic ranges introduce larger scale factor and nonlinearity errors and increase noise on the sensor data. However, the advantages of strapdown systems in weight, size, power and more importantly cost forced navigation systems to turn from gimbaled to strapdown mechanization in modern applications. Thus, modern aircrafts, marine vessels, missiles and autonomous vehicles are equipped with strapdown inertial navigation systems. In this thesis work, we have also implemented strapdown mechanization on an autonomous vehicle.

Strapdown system development commenced in late 1950's. The first flight-operational hardware was flown in 1966 (Garg (1978)). Thus, the concept of strapdown inertial navigation has emerged with the increasing computational capacity more than thirty years ago. The development and deployment of the first strapdown systems are discussed in (Powell (1963), Quasius (1963), Bessen (1964)).

The main effort in the strapdown systems have been the design of computation process in a limited computational environment. Dealing with rapid maneuvers and high vibrational environment has required sensors with high bandwidth and new computational techniques in order to compensate the resulting errors. As a result, the computational process is splitted into high and low speed sections as seen in (Savage (1984)). High speed, low amplitude motions and the need for compensation of their effects have started the research on coning (attitude) and sculling (velocity) techniques. Various coning compensation algorithms are derived in order to decrease the attitude errors stemming from vibration. Some of these techniques can be found in (Miller (1983), Jiang (1992), Tazartes (1997), Ignagni (1990), (1996), Salychev (1998)). Sculling compensation has been another critical algortihm for accuracy and Ignagni (1998) has shown the duality between the optimal coning and sculling techniques. As a result, the strapdown system and related computational techniques have matured over last forty years.

During this period, another challenging subject has been the integrating of inertial systems with external aid sensors. As the applications required, low cost and more accurate systems, an intensive research have been carried out resulting new types of integration techniques. The dominant method used in integrated navigation involves the application of Kalman Filter theory to the navigation (Brown (1992), Lewantowicz (1995), Philips (1996), Titterton (1997)). However, other techniques such as nonlinear filters (Vik (2001)) and neural networks are also applied. While Tekinalp and Özemre (2001) has employed artificial neural networks for calibrating inertial sensors and initializing inertial navigation system. Chiang et. al. have achieved to integrate DGPS with INS using neural networks.

The main system used in order to provide external aid to the inertial systems has been the Global Positioning System (GPS) because of its low cost and complementary features. The GPS/INS integration is implemented using different architectures which are given in (Greenspan (1995)) with their own advantages and disadvantages.

Having reviewed the relevant literature on GPS and INS systems, focus will be directed to implementing a strapdown inertial navigation system and integrating it to global positioning system in this chapter.

2.2 Strapdown Inertial Navigation System

As stated previously, a strapdown inertial navigation system consists of body mounted accelerometers, gyroscopes, a computer (navigation/attitude computer) and instrument electronics such as power conditioning, input/output interface. The data flow between the components and overall system are given in Figure 2-1.

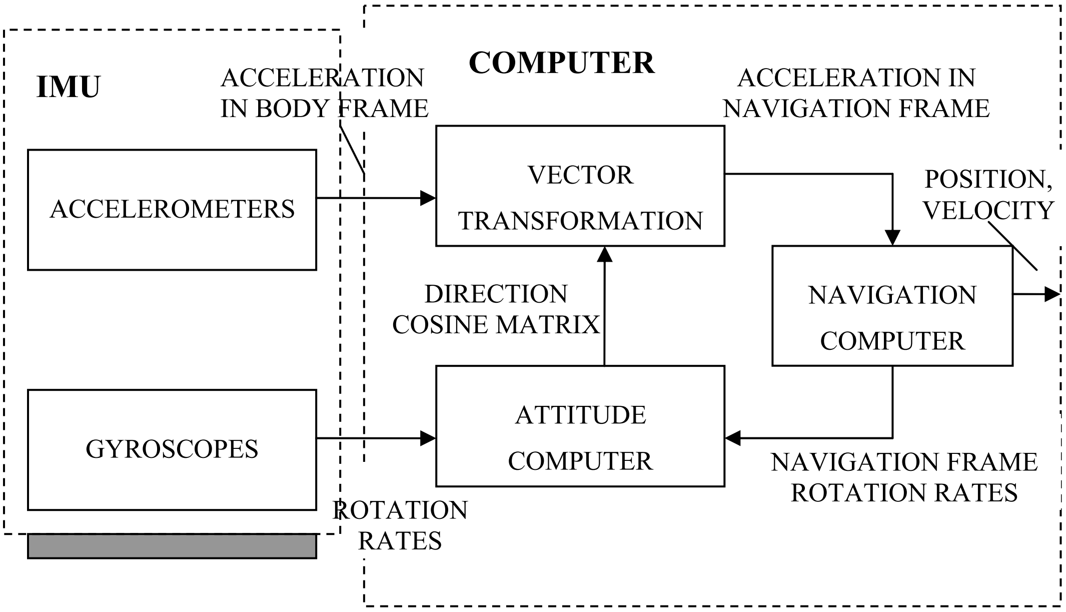

Figure 2-1 Strapdown Inertial Navigation System

As seen in the figure, accelerometers and gyroscopes which are composed with an electronic processor to form an inertial measurement unit (IMU), measure the specific force and the rotation rates of the body frame with respect to the inertial frame (non-rotating fixed frame). Rotation rates measured by the gyroscopes are used in attitude computation which yields a direction cosine matrix used to transform acceleration vectors between the body and navigation frames. The transformed acceleration vector is used to calculate position and velocity. Navigation frame rotation rates required for attitude computations are also calculated in navigation computer.

As a result, inertial navigation system is composed of an inertial measurement unit providing acceleration and rotation measurements, and a computer implementing coordinate frame transformations and navigation calculations. The detailed description of inertial measurement unit, coordinate frames and navigation equations of the inertial navigation system developed in the balance of this thesis work, is given in the subsequent sections.

2.2.1 Inertial Measurement Unit

As described above, inertial navigation system depends on the measurement of acceleration which can be integrated sequentially to provide position and velocity. Furthermore, strapdown inertial systems require rotation information to determine the orientation of body acceleration with respect to a navigation frame.

The required acceleration and rotation information is provided by an inertial measurement unit. The inertial measurement unit constructed for the thesis work incorporating a tri-axial accelerometer and 3 single axis gyroscopes (Figure 2-2), supplies velocity and angle increments of the body at 400 Hz to the processing card with a Motorola MC68340 (24.1 MHz) where sensor error compensation algorithms run and resulting compensated data output at 100 Hz rate fed into the navigation computer using a dual port RAM.

The inertial sensor which measures acceleration is known as accelerometer. Accelerometers are classified as either mechanical or solid-state with respect to their form of construction. These devices use gyroscopic mass unbalance technology or vibratory technology to measure specific force. Accelerometer that we make use throughout the thesis is a solid-state vibratory technology tri-axial accelerometer of LITEF known as B-290 (Figure 2-2).

Every axis of the tri-axial accelerometer also known as pendulous accelerometers has a pendulum, translational proof mass. When a specific force is applied, the pendulum is loaded and a torquer acting on the pendulum minimizes the deflection of the pendulum. The force applied by the torquer is a measure of the acceleration sensed.

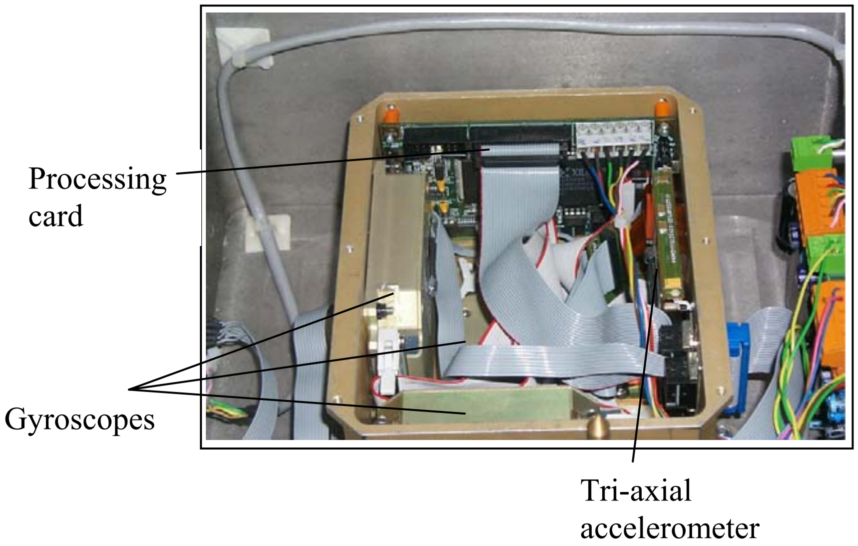

Figure 2-2 Inertial Measurement Unit

Table 2.1 B-290 Triad Specifications (Litef 2002a).

LITEF B-290 Accelerometer	
Bias repeatability	3 mg (1σ)
Bias stability	0.2 mg (1σ)
SF repeatability	3000 ppm (1σ)
SF stability	200 ppm (1σ)
Resolution	50 μg
Axis misalignment	20 mrad

B-290 accelerometer is classified as a low-grade, low-cost sensor compatible with tactical applications and has performance specifications given in Table 2.1 (Litef 2002a).

Rotation information required for strapdown navigation applications are provided by sensors called gyroscopes. Gyroscopes are used in various applications such as platform/sensor stabilization, autopilot feedback and navigation to sense angular rate about some predefined axis. Gyroscopes are manufactured based on either mechanical or optical technology for providing this rate information: Classical mechanical gyroscopes use the inertial properties of a rotor spinning at high rates to determine the angular changes whereas optical gyroscopes exploit interferometric methods to sense angular motion. In this thesis work we use optical gyroscopes known as fiber optic gyroscope (FOG).

Fiber optic gyroscopes (FOG) work based on the “Sagnac Effect” concept. FOG consists of fiber-optic coils, lithium niobate optics chip, broadband light source and photodetector (Lefevre (1993)). In FOG, the light source sends two light beams to a fiber coil, each traveling in opposite directions (clockwise and counterclockwise) with respect to a measuring point. Unless the coil rotates, the beams fly over the same path and land on the measuring point at the same time. Otherwise a time difference occurs between their arrivals and two regions occur, one dark and the other semi-dark on the readout device, due to the phase difference between two light beams. The light intensity of these regions is directly related to the applied rotation rate and is output as an electrical quantity (current or voltage) proportional to the rotation after a modulation phase.

In the applications of this thesis we make use more specifically of LITEF μFORS6 gyroscope for which performance specifications is given in Table 2.2 (Litef 2002b).

Solid state accelerometers and fiber optic gyroscopes of our application is chosen for their wide usage for navigation purposes in tactical missiles, tactical guidance of unmanned land/air/undersea vehicles and robotics, because of their relatively high accuracy and low cost (Gamble (2001), Kaygısız (2003)).

Table 2.2 µFORS-6 Specifications (Litef 2002b).

LITEF µFORS-6 Gyroscope	
Bias repeatability	6°/h (1σ)
Bias stability	3°/h (1σ)
SF repeatability	2000 ppm (1σ)
SF stability	500 ppm
Axis misalignment	±10 mrad (max.)

2.2.2 Coordinate Frames

Precise definition of Cartesian coordinate systems is crucial to the inertial navigation since sensors provide measurements and the system calculates position with respect to some defined coordinate which are generally in navigation system calculations, taken as orthogonal and right handed. These coordinate frames in strapdown navigation systems are:

Body Frame: This coordinate frame has its origin at the center of the navigation system. X_b-axis points towards the roll, Y_b-axis is towards the pitch axis according to right handed system and Z_b-axis is pointing down (yaw axis).

Inertial Frame: Newton has defined the inertial frame as a frame which does not accelerate or rotate. The Newtonian equations of motion are defined based on this frame. Inertial frame is impossible to realize in practice and the best approximation to such a frame is according to the distant stars which can be assumed motionless.

In the navigation systems, the inertial frame is described with respect to the earth. Such a frame is called Earth Centered Inertial (ECI) frame. In the ECI frame, the origin is at the center of the earth, X_I-axis is positive towards the intersection point of Greenwich meridian and equator at t_0, Z_I-axis lies along the North Pole at t_0, Y_I-axis is defined to complete a right-handed orthogonal coordinate system. Inertial sensors provide measurements of body motion with respect to inertial frame.

Earth Centered Earth Fixed Frame (ECEF): The coordinate system is a right-handed earth fixed system (Figure 2-3). As seen in the figure, the origin is the earth center of mass, Z_e-axis has a direction toward North Pole and X_e-axis is directed through the intersection of Greenwich and the plane passing through the origin and normal to the Z_e-axis. Y_e-axis completes a right-handed earth fixed coordinate system. This frame rotates with the earth and coincides with ECI frame every 24 hours. This coordinate system approximates the earth as an ellipsoid which is known as WGS 84 ellipsoid. Any position in this coordinate system can also be defined by the geodetic parameters of latitude, longitude and height. The parameters of the corresponding ellipsoid can be found in literature (DoD (2000)).

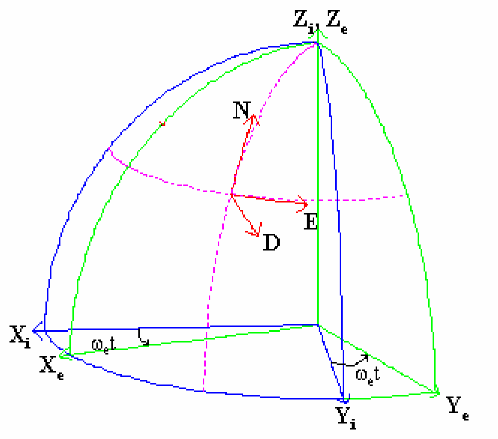

Figure 2-3 Coordinate Frames

Local Level (North-East-Down) Frame: The local-level frame has its origin at the location of navigation system and the axes lie along north, east and local normal to the ellipsoid, down (*N, E, D* axes in Figure 2-3). The angular rate of the local-level frame with respect to earth, known as the transport rate, is determined by the motion of the navigation system with respect to the earth. In this thesis work, this frame for our navigation computations and the local-level mechanization described in the coming subsection is applied in the implementation of the strapdown system in our ALV application.

2.2.3 Local-Level Frame Mechanization

In local-level frame mechanization, the velocity is expressed in the local geographic coordinate system as north-east-down in order to maintain the navigation accuracy over large distances on earth. This type of mechanization computes the position of the vehicle in terms of latitude, longitude and height and provides the velocity information in north, east and down coordinates. Here, the rate of change of the vehicle speed with respect to earth is expressed as:

$$\frac{d}{dt}\mathbf{v}_e\,|_n = \frac{d}{dt}\mathbf{v}_e\,|_i - (\boldsymbol{\omega}_{ie} + \boldsymbol{\omega}_{en}) x \mathbf{v}_e \tag{2.1}$$

where $\boldsymbol{\omega}_{ie}$ is the rate of earth with respect to inertial frame and $\boldsymbol{\omega}_{en}$ is the rate of navigation frame with respect to earth.

Substuting the following relation

$$\dot{\mathbf{v}}_e^i = \mathbf{f} \ - \boldsymbol{\omega}_{ie} x \mathbf{v}_e + \mathbf{g} \tag{2.2}$$

into (2.1), one obtains

$$\frac{d}{dt}\mathbf{v}_e\,|_n = \mathbf{f} - (2\boldsymbol{\omega}_{ie} + \boldsymbol{\omega}_{en}) x \mathbf{v}_e + \mathbf{g} \tag{2.3}$$

Expressing (2.3) in navigation axes, navigation equation becomes

$$\dot{\mathbf{v}}_e^n = \mathbf{C}_b^n \mathbf{f}^b - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) x \mathbf{v}_e^n + \mathbf{g}^n \tag{2.4}$$

where $\mathbf{f}^b$ is the specific force measured by the accelerometers, $\mathbf{g}^n$ is the gravity vector represented in navigation frame and $\mathbf{C}_b^n$ is a direction cosine matrix used to transform body acceleration vector into navigation axes. This matrix is updated with the following equation:

$$\dot{\mathbf{C}}_b^n = \mathbf{C}_b^n \boldsymbol{\Omega}_{nb}^b \tag{2.5}$$

where $\boldsymbol{\Omega}_{nb}^b$ is the skew symmetric matrix of $\boldsymbol{\omega}_{nb}^b$, rate of body frame with respect to navigation frame expressed in body frame, and

$$\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - \mathbf{C}_n^b (\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \tag{2.6}$$

$\boldsymbol{\omega}_{ib}^b$ is the rotation of body with respect to inertial frame which is the output of gyroscopes. The other terms are the earth's and navigation frame's rotation. Earth's rotation with respect to inertial frame is found in navigation frame as:

$$\boldsymbol{\omega}_{ie}^n = \begin{bmatrix} \Omega \cos L & 0 & -\Omega \sin L \end{bmatrix}^T \tag{2.7}$$

and craft rate, which is navigation frame's rotation with respect to earth is calculated as:

$$\boldsymbol{\omega}_{en}^n = \begin{bmatrix} \dfrac{v_e}{R_E + h} & \dfrac{-v_n}{R_N + h} & \dfrac{-v_e \tan L}{R_E + h} \end{bmatrix}^T \tag{2.8}$$

where the velocities are north and east velocities, R's stand for north and east radius of the earth at the vehicle location, h is height, L is lattitude and Ω is earth's sidereal rotation rate.

The update algorithms we make use to implement the equations described in this section are given in Appendix in detail.

2.2.4 Vibration Effects and Compensation

Vehicle vibration has a harmful effect on the performance of inertial navigation systems and on control since inertial sensors have a limited bandwidth and reject the motion vibration at high frequencies. Vibrations at high frequencies cannot be detected with the accelerometers and gyroscopes and cause small changes in attitude and acceleration from the actual ones. This contributes to the navigation system errors and degrades the system performance (Bortz (1971), Savage (1984)).

This problem can be circumvented by introducing a low pass filter to the sensor outputs. However, low pass filters introduces time lag to the system and prevents the sensors to sense actual motion. In order to prevent information loss, a mechanical filter should be introduced to the physical environment. This is achieved by using vibration absorbers. It is important to ensure that the resonance frequency of the vibration absorber does not coincide with a region where the vibration is effective. Attenuating the high frequency vibrations, we should still deal with leftover vibration. If not dealt properly, remaining vibration causes error on attitude and velocity algorithms. The effect of vibration on attitude is termed as coning. Coning is defined as the cyclic motion of one axis due to rotational motion of the other two axes. The error on attitude due to coning is compensated with the algorithms known as “coning compensation”. Coning compensation is applied on the inertial sensor outputs and in the inertial measurement unit in this work. Inertial sensors provide measurements at 400 Hz and these measurements are coning compensated to supply 100 Hz angle increment data to the strapdown system. Degradation due to vibration also occurs in velocity algorithms. This effect of vibration is known as “sculling”. Sculling is the degradation on acceleration readings due to translational motion of an axis during a rotational motion of the other axes. This error is compensated in inertial measurement as in coning case and this compensation is known as “sculling compensation”.

The problem in coning and sculling compensation means to determine algorithms that well maintain high performance under vibrational dynamical conditions. The algorithms selected in our strapdown system are 4th order algorithms since inertial measurements are at 400 Hz and strapdown system works at 100 Hz and are described in the Appendix.

The harmful effect of vibration on the performance of inertial navigation systems is compensated using vibration absorbers and compensation techniques described. However, the problem of chaotic vibrations on ALV control can only be circumvented by decontaminating state space regions into mostly stable ones (this will be the novel approach that we develop in Chapter 6).

2.3 GPS/INS Integration

As stated before, strapdown navigation systems rely on rate and acceleration measurements and initial position and velocity information to provide position and velocity. However, errors on sensor readings caused by bias, scale factors, thermal/magnetic effects, other nonlinearities and initialization offsets cause an accumulation in navigation errors. Since an inertial navigation system is a dead reckoning system, any lack of precision is passed from one evaluation to the next and navigation solution drifts with time. Thus, the accuracy of strapdown systems are predominantly governed by the accuracy of the sensors and improved accuracy can be achieved through the use of more accurate sensors. However, employing more accurate sensors clearly lead to very expensive solutions which are not affordable for many applications.

An alternative approach is to use an additional information source in order to improve the accuracy of the inertial navigation system. This is the principle of an aided inertial system where one or more navigation system outputs are compared to the corresponding outputs of an external system and fed into an optimum filter to generate corrections to the navigation system as seen in Figure 2-4. There exist different types of sensors used as external system to improve the accuracy of the inertial navigation such as Doppler radar, baro-altimeter, radar altimeter, terrain map, airspeed indicator and miscellaneous radio navigation aids. In most modern integrated navigation applications, external aid sensor utilized is a type of satellite radio navigation with low cost, high accuracy and global coverage known as global positioning system (GPS). GPS integrated INS systems have become the major navigation tool in recent years for applications such as automotive, robotics and unmanned autonomous vehicles. Current technological trend makes these integrated navigation systems a part of our daily life.

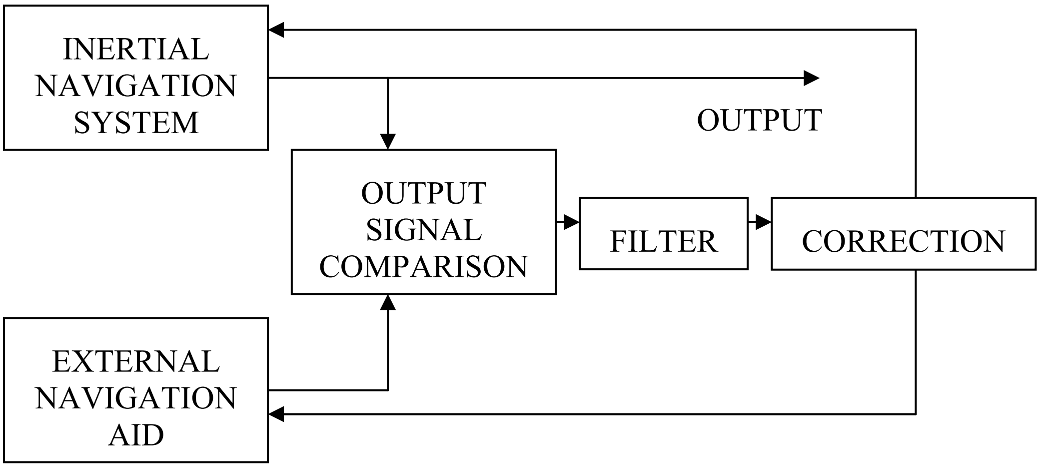

Figure 2-4 Aided Navigation System Concept

In this thesis work, GPS is also used as an external aid sensor due to its high accuracy and low cost as well as its high frequency of usage in commercial products. GPS integrated INS navigation systems as it is also considered in this thesis work, consist of three main components, namely the inertial measurement unit (IMU), a GPS receiver and a navigation computer. The data flow between the components and overall system are given in Figure 2-5.

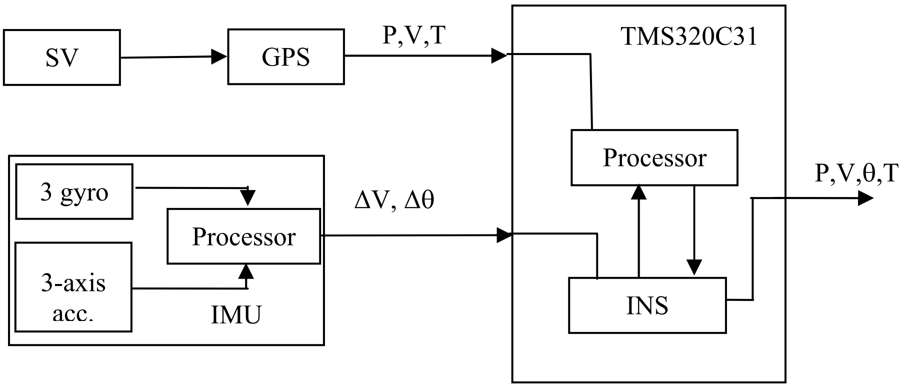

Figure 2-5 GPS/INS system components

The details of IMU, INS and the navigation computations are given in the previous sections and GPS is described in the following section.

2.3.1 Global Positioning System

Our work is based upon an INS that utilizes a global positioning system (GPS) as the additional source of navigation information in order to improve its performance.

Global Positioning System (GPS) is a satellite based radio navigation system providing 3 dimensional position and velocity information to the user. The system consists of earth located receivers that communicate with 24 satellites in six orbits such that at least 4 satellites are visible from any point on the earth. Each GPS satellite transmits navigation and range data on L1 (1575.42 MHz) and L2 (1127.60 MHz) frequencies. Military receivers exploit both frequencies while civil receivers can only access to L1 codes (Parkinson (1995), Kaplan (1996)).

A receiver located on the earth needs to monitor at least four satellites in order to find out its 3 dimensional position and receiver clock bias. A user receiver entering into a tunnel, a downtown area with high buildings, a canyon or maneuvering with high dynamics can be incapable of receiving the GPS signals and would fail to supply location data (Brown (1992)).

The GPS receiver has a one-way communication with each satellite which send navigation messages together with the transmit time. The difference between the receiving time and the transmit time multiplied by the speed of light gives the distance between the receiver and the satellite, pseudorange. Using four or more pseudorange measurements, a receiver can solve its 3 dimensional position and clock bias. GPS receivers also acquire Doppler measurements which are related to the receiver velocity. Doppler measurements can be approximated as pseudorange rate in low dynamics environment. GPS receiver employs the external measurements of GPS pseudorange (ρ) and GPS pseudorange rate ($\dot{\rho}$) in order to solve its position and velocity.

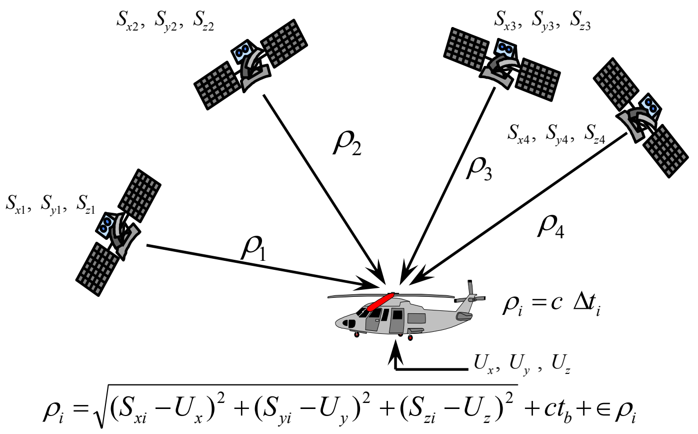

Figure 2-6 Receiver-satellite orientation

Figure 2-6 shows a receiver and its 4 satellite positioning and orientation. There, t_b is the receiver clock bias, c is the speed of light, $\in_\rho$ is the all errors (multipath, troposphere…) other than clock bias. As seen in the figure, the difference between the time at which the signal is sent and that when the signal is received, is multiplied by the speed of the signal, which is the speed of light, to yield the pseudorange value, ρ_i. The pseudorange value would then be

$$\rho = \sqrt{(x_e - x_i)^2 + (y_e - y_i)^2 + (z_e - z_i)^2} + ct_b + \in_\rho \tag{2.9}$$

where subscript *e* denotes the receiver and *i* is for the satellite *i*. The coordinate system in the given formula is Earth Centered Earth Fixed system. Pseudorange rate without any error terms can be defined as

$$\dot{\rho} = \frac{\partial \rho}{\partial t} \tag{2.10}$$

Acquiring the external measurements of pseudorange and pseudorange rate, GPS receiver incorporates an Extended Kalman Filter algorithm to process these external measurements for the estimation of position and velocity.

Most GPS receivers employ one of the two common dynamic models in their extended Kalman filter structures. These are Position-Velocity (PV) and Position-Velocity-Acceleration (PVA) models (Kaygısız (2000)).

PV dynamic model is generally realized in low dynamical environments where receiver velocity remains nearly constant. The velocity is modelled as random walk with position being naturally the integral of velocity. PVA dynamic model is generally realized in high dynamical environments where the receiver experiences sudden velocity changes. Here acceleration is modelled as Markov processes while clock bias and frequency are again modelled as random walk, as in the PV model.

2.3.2 Integration Architectures

GPS/INS systems have been the subject of extensive research and application in the recent years. Studies have generally focused on techniques and architectures in order to unify the GPS and INS systems into one more accurate navigation system.

There exist two basic GPS/INS integration techniques in the literature. These are nonlinear filtering (Vik (2001)) and extended Kalman filtering (Lewantowicz (1995), Philips (1996)). Generally, in the real time applications, the extended Kalman filtering is used in GPS/INS systems. This is also what we considered in the balance of this thesis work.

Another important issue in the integration process is the architecture employed. There are three main architectures that we will give a quick overview (Figure 2-7-Figure 2-9) with their own advantages and disadvantages (Greenspan (1995)).

Separate architecture: GPS updates the INS position and velocity states with resets. It is the cheapest and easiest architecture that can be applied. It binds the velocity and position errors of the INS but cannot correct the IMU errors.

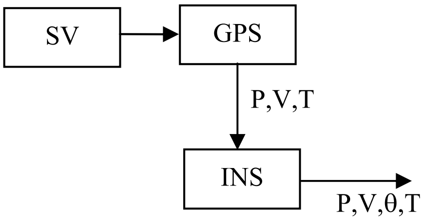

Figure 2-7 Separate integration

Loosely-coupled architecture: It estimates INS error states and feedback these errors in order to update the INS and IMU errors. This architecture gives the advantage of having a higher grade INS even if the GPS signal is lost.

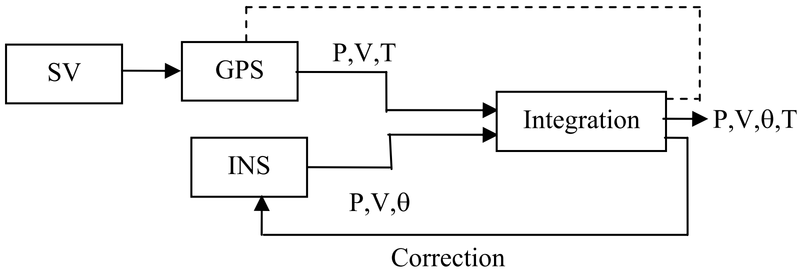

Figure 2-8 Loosely coupled integration

Tightly-coupled architecture: Integrated system continues the work when the visible satellite number is less than four (loosely coupled systems need at least four satellites). It corrects the GPS errors and provides a better sensor calibration. This architecture employs a single Kalman filter unlike the loosely coupled one which uses cascaded Kalman filter approach. The main disadvantage is the complexity of the system.

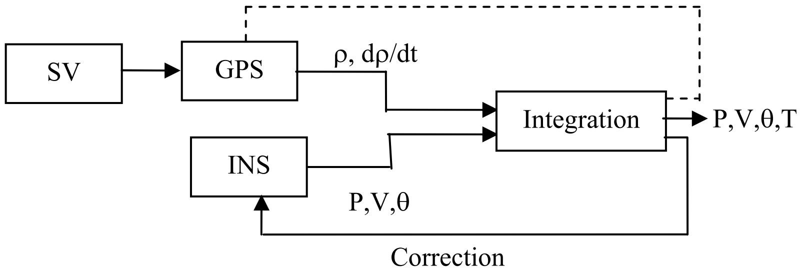

Figure 2-9 Tightly coupled integration

This chapter will provide the mathematical structure of a loosely coupled one among the three architectures defined, since this is the architecture utilized in our navigation system.

2.3.3 Loosely Coupled GPS/INS

The Kalman filter technique given in Appendix D, is assigned as the integration filter of Figure 2-8 and performs two basic actions in the GPS/INS system: 1) When the GPS measurement is not available, system uses INS/IMU error model and the current navigation parameters in order to update the state covariance matrix, $\mathbf{P}$, giving the uncertainty levels on the current state vector. This matrix is propagated through time according to the current state vector. 2) When the GPS measurement is available, the Kalman filter performs its second action known as update, in which it combines the GPS measurement with current error estimate and updates the Kalman filter states and the corresponding covariance matrix, $\mathbf{P}$. The updated error states are then fedback to the INS and IMU where the system errors are corrected.

Referring to Appendix D, the Kalman filter that we use in our application can be split into two main parts: propagation and update. These steps are given as:

Propagation

$$\hat{\mathbf{x}}_k(-) = \mathbf{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1}(+) \tag{2.11}$$

$$\mathbf{P}_k(-) = \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}(+)\mathbf{\Phi}^T{}_{k-1} + \mathbf{Q}_{k-1} \tag{2.12}$$

Update

$$\mathbf{K}_k = \mathbf{P}_k(-)\mathbf{H}_k{}^T[\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k{}^T + \mathbf{R}_k]^{-1} \tag{2.13}$$

$$\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + \mathbf{K}_k[\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k(-)] \tag{2.14}$$

$$\mathbf{P}_k(+) = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{P}_k(-) \tag{2.15}$$

where **x** is the Kalman filter state vector, **z** is the measurement vector, **Φ** is the state transition matrix, **H** is the measurement matrix, **K** is the Kalman gain matrix,

P is the state covariance matrix, **R** is the measurement covariance matrix, **Q** is the process noise covariance matrix. Within propagation part, there is no state propagation since the error states are fedback into the integrated system at every update time.

Since a slow propagation can yield degradation in high dynamical environments, in the system we realize, the Kalman propagation of the covariance matrix is done at 10 Hz. In addition, Kalman update is done for every GPS measurement (1 Hz, in our application) when the GPS signal is available.

2.3.4 Error State Model

The propagation step of the Kalman filter based upon the uncertainty levels stemming from the navigation system, depends on the error state dynamics of the INS. Error state vector of the inertial system includes position, velocity, attitude errors in addition to the accelerometer and gyroscope errors. The inertial sensors own numerous error sources, all of which cannot be defined as error states in the Kalman filter because of computational load and observability problems. Thus error states for inertial sensors are determined according to the mission needs, hardware constraints, sensor performance and observability analysis. We employed accelerometer bias and gyro drifts as inertial sensor error states of the Kalman filter. Consequently, a 15-state Kalman filter is designed in order to integrate INS and GPS systems. In order to construct such a structure, a set of differential equations known as error state model of the navigation system should be derived. The derivation is given below.

2.3.4.1 Attitude Error

Direction cosine matrix, $\mathbf{C}_b^n$, holds the attitude information in between the body and navigation frames. True direction cosine matrix, $\mathbf{C}_b^n$, is related to the computed direction cosine matrix, $\tilde{\mathbf{C}}_b^n$, as follows:

$$\tilde{\mathbf{C}}_b^n = (\mathbf{I} - \boldsymbol{\varepsilon}\times)\mathbf{C}_b^n \tag{2.16}$$

In the above relation *I* is a 3x3 identity matrix and

$$\boldsymbol{\varepsilon}\times = \begin{bmatrix} 0 & -\varepsilon_z & \varepsilon_y \\ \varepsilon_z & 0 & -\varepsilon_x \\ -\varepsilon_y & \varepsilon_x & 0 \end{bmatrix} \tag{2.17}$$

$(\mathbf{I} - \boldsymbol{\varepsilon}\times)$ represents the transformation matrix between the true and computed coordinate frames. $\varepsilon_x, \varepsilon_y, \varepsilon_z$ are the attitude errors of the navigation system in roll, pitch and yaw channels, respectively. Rearranging (2.16) one obtains,

$$(\boldsymbol{\varepsilon}\times) - \mathbf{I} = -\tilde{\mathbf{C}}_b^n \mathbf{C}_b^{nT} \tag{2.18}$$

Differentiating the above equation, one obtains how the attitude errors propagate in time.

$$(\dot{\boldsymbol{\varepsilon}}\times) = -\dot{\tilde{\mathbf{C}}}_b^n \mathbf{C}_b^{nT} - \tilde{\mathbf{C}}_b^n \dot{\mathbf{C}}_b^{nT} \tag{2.19}$$

Inserting direction cosine update equation given into (2.19) we obtain

$$(\dot{\boldsymbol{\varepsilon}}\times) = -\tilde{\mathbf{C}}_b^n \tilde{\boldsymbol{\Omega}}_{ib}^b \mathbf{C}_b^{nT} + \tilde{\boldsymbol{\Omega}}_{in}^n \tilde{\mathbf{C}}_b^n \mathbf{C}_b^{nT} - \tilde{\mathbf{C}}_b^n \boldsymbol{\Omega}_{ib}^{b\,T} \mathbf{C}_b^{nT} + \tilde{\mathbf{C}}_b^n \mathbf{C}_b^{nT} \boldsymbol{\Omega}_{in}^{n\,T} \tag{2.20}$$

By rearranging this equation, we get

$$\begin{aligned}(\dot{\boldsymbol{\varepsilon}}\times) &= -\left[\mathbf{I} - \boldsymbol{\varepsilon}\times\right]\mathbf{C}_b^n\left[\tilde{\boldsymbol{\Omega}}_{ib}^b - \boldsymbol{\Omega}_{ib}^b\right]\mathbf{C}_b^n + \tilde{\boldsymbol{\Omega}}_{in}^n\left[\mathbf{I} - \boldsymbol{\varepsilon}\times\right]\mathbf{C}_b^n\mathbf{C}_b^{nT} - \left[\mathbf{I} - \boldsymbol{\varepsilon}\times\right]\mathbf{C}_b^n\mathbf{C}_b^{nT}\boldsymbol{\Omega}_{in}^n \\ &\cong -\mathbf{C}_b^n \delta\boldsymbol{\Omega}_{ib}^b \mathbf{C}_b^{nT} + \boldsymbol{\varepsilon}\times\boldsymbol{\Omega}_{in}^n - \boldsymbol{\Omega}_{in}^n\boldsymbol{\varepsilon}\times + \delta\boldsymbol{\Omega}_{in}^n\end{aligned} \tag{2.21}$$

Converting the above relation into vectoral form, differential equation for the attitude error, $\boldsymbol{\varepsilon} = \begin{bmatrix}\varepsilon_x & \varepsilon_y & \varepsilon_z\end{bmatrix}^T$, is

$$\dot{\boldsymbol{\varepsilon}} = -\boldsymbol{\omega}_{in}^{n} \times \boldsymbol{\varepsilon} + \delta\boldsymbol{\omega}_{in}^{n} - \mathbf{C}_{b}^{n}\delta\boldsymbol{\omega}_{ib}^{b} \tag{2.22}$$

In this relation $\delta\boldsymbol{\omega}_{ib}^{b} = \tilde{\boldsymbol{\omega}}_{ib}^{b} - \boldsymbol{\omega}_{ib}^{b}$ represents the gyroscope errors. $\delta\boldsymbol{\omega}_{in}^{n} = \tilde{\boldsymbol{\omega}}_{in}^{n} - \boldsymbol{\omega}_{in}^{n}$ is composed of two parts and can be given as $\delta\boldsymbol{\omega}_{in}^{n} = \delta\boldsymbol{\omega}_{ie}^{n} + \delta\boldsymbol{\omega}_{en}^{n}$.

$$\begin{aligned}
\delta\boldsymbol{\omega}_{ie}^{n} &= \begin{bmatrix} -\Omega\sin L\delta L \\ 0 \\ -\Omega\cos L\delta L \end{bmatrix} \\
\delta\boldsymbol{\omega}_{en}^{n} &= \begin{bmatrix} \dfrac{\delta v_e}{R_E+h} - \dfrac{v_e}{\left(R_E+h\right)^2}\delta h \\ -\dfrac{\delta v_n}{R_N+h} + \dfrac{v_n}{\left(R_E+h\right)^2}\delta h \\ -\dfrac{\delta v_e \tan L}{R_E+h} + \dfrac{v_e \tan L}{\left(R_E+h\right)^2}\delta h - \dfrac{v_e}{\left(R_E+h\right)^2\cos^2 L}\delta L \end{bmatrix}
\end{aligned} \tag{2.23}$$

2.3.4.2 Velocity Error

Velocity error is the difference between true velocity and computed velocity. True velocity equation of a vehicle is given as:

$$\dot{\mathbf{v}}_{e}^{n} = \mathbf{C}_{b}^{n}\mathbf{f}^{b} - (2\boldsymbol{\omega}_{ie}^{n} + \boldsymbol{\omega}_{en}^{n})x\mathbf{v}_{e}^{n} + \mathbf{g}^{n} \tag{2.24}$$

Computed velocity of a vehicle in navigation frame can be given as:

$$\dot{\tilde{\mathbf{v}}}_{e}^{n} = \tilde{\mathbf{C}}_{b}^{n}\tilde{\mathbf{f}}^{b} - (2\tilde{\boldsymbol{\omega}}_{ie}^{n} + \tilde{\boldsymbol{\omega}}_{en}^{n})x\tilde{\mathbf{v}}_{e}^{n} + \tilde{\mathbf{g}}^{n} \tag{2.25}$$

By differencing true and computed velocity, we obtain the following differential equation.

$$\begin{aligned}
\delta\dot{\mathbf{v}}_{e}^{n} &= \dot{\tilde{\mathbf{v}}}^{n} - \dot{\mathbf{v}}^{n} \\
&= \tilde{\mathbf{C}}_{b}^{n}\tilde{\mathbf{f}}^{b} - \mathbf{C}_{b}^{n}\mathbf{f}^{b} - (2\tilde{\boldsymbol{\omega}}_{ie}^{n} + \tilde{\boldsymbol{\omega}}_{en}^{n})x\tilde{\mathbf{v}}_{e}^{n} + (2\boldsymbol{\omega}_{ie}^{n} + \boldsymbol{\omega}_{en}^{n})x\mathbf{v}_{e}^{n} + \tilde{\mathbf{g}}^{n} - \mathbf{g}^{n}
\end{aligned} \tag{2.26}$$

Inserting $\tilde{\mathbf{C}}_b^n = (\mathbf{I} - \boldsymbol{\varepsilon}\times)\mathbf{C}_b^n$ into the above equation and after some manipulations, velocity error equation becomes,

$$\delta\dot{\mathbf{v}}_e^n = \mathbf{C}_b^n\mathbf{f}^b \times \boldsymbol{\varepsilon} + \mathbf{C}_b^n\delta\mathbf{f}^b - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n)x\delta\mathbf{v}_e^n - (2\delta\boldsymbol{\omega}_{ie}^n + \delta\boldsymbol{\omega}_{en}^n)x\mathbf{v}_e^n + \delta\mathbf{g}^n \quad (2.27)$$

where $\delta\boldsymbol{\omega}_{ie}^n = \tilde{\boldsymbol{\omega}}_{ie}^n - \boldsymbol{\omega}_{ie}^n$, $\delta\boldsymbol{\omega}_{en}^n = \tilde{\boldsymbol{\omega}}_{en}^n - \boldsymbol{\omega}_{en}^n$, $\delta\mathbf{f}^b = \tilde{\mathbf{f}}^b - \mathbf{f}^b$, $\delta\mathbf{v}^n = \tilde{\mathbf{v}}^n - \mathbf{v}^n$ and $\delta\mathbf{g}^n = \tilde{\mathbf{g}}^n - \mathbf{g}^n$.

2.3.4.3 Position Error

The following differential equation is employed to update position in a local-level strapdown mechanization using

$$\begin{bmatrix} \dot{L} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \dfrac{v_n}{R_N + h} \\ \dfrac{v_e}{(R_E + h)\cos L} \\ -v_d \end{bmatrix} \quad (2.28)$$

By differentiating this equation and neglecting the small terms, position error equation becomes

$$\begin{bmatrix} \delta\dot{L} \\ \delta\dot{\lambda} \\ \delta\dot{h} \end{bmatrix} = \begin{bmatrix} \dfrac{\delta v_n}{R_N + h} - \dfrac{v_n}{(R_N + h)^2}\delta h \\ \dfrac{\delta v_e}{(R_E + h)\cos L} - \dfrac{v_e}{(R_E + h)^2\cos L}\delta h + \dfrac{v_e \tan L}{(R_E + h)\cos L}\delta L \\ -\delta v_d \end{bmatrix} \quad (2.29)$$

where δL, $\delta\lambda$ and δh the north position, east position and height errors, respectively.

2.3.4.4 Sensor Errors

In addition to these 9 states standing for inertial navigation system errors, other 6 states are used in order to model the accelerometer and gyro bias errors, as a first-order Markov process:

$$\delta\dot{\mathbf{f}} = -\frac{1}{\tau_a}\delta\mathbf{f} + \mathbf{w}, \delta\dot{\boldsymbol{\omega}} = -\frac{1}{\tau_g}\delta\boldsymbol{\omega} + \mathbf{w} \tag{2.30}$$

where **w** is white noise. The time constants of the Markov process, τ_a and τ_g, , are set to 12,000 seconds based on experimental observations of the inertial sensors.

Consequently, state vector **x** in the 15-state Kalman filter is $\begin{bmatrix}\delta L & \delta\lambda & \delta h & \delta v_n & \delta v_e & \delta v_d & \varepsilon_x & \varepsilon_y & \varepsilon_z & \delta f_x & \delta f_y & \delta f_z & \delta\omega_x & \delta\omega_y & \delta\omega_z\end{bmatrix}^T$.

Using the given differential equations, one can create the continuous state transition matrix, **F**, for the error states, which is then discretized for the real time application using first order Taylor series approximation yielding $\mathbf{\Phi} = \mathbf{I} + \mathbf{F}\Delta t$. Δt is chosen as 0.1 seconds so the covariance

$$\mathbf{P}^{-} = \mathbf{\Phi}\mathbf{P}^{+}\mathbf{\Phi}^{T} + \mathbf{Q} \tag{2.31}$$

is propagated at 10 Hz. At this part, there is no state propagation since the states are fedback into the integrated navigation system at every update time in order to correct the system errors, as mentioned earlier.

2.3.5 Measurement Model

Loosely coupled approach employs the position and velocity difference between INS and GPS solutions as the measurement. The measurement model of the integrated system is given as

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v} \tag{2.32}$$

where

$$\mathbf{z}=\begin{bmatrix} L_I-L_G \\ \lambda_I-\lambda_G \\ h_I-h_G \\ v_{nI}-v_{nG} \\ v_{eI}-v_{eG} \\ v_{dI}-v_{dG} \end{bmatrix}$$, $\mathbf{H}=\begin{bmatrix}\mathbf{I}_{6x6} & \mathbf{0}_{6x9}\end{bmatrix}$, $\mathbf{x}_{15\text{x}1}$ is the state vector and $\mathbf{v}_{6\text{x}1}$ is the measurement noise vector representing GPS position and velocity measurement noise. In the above equations, the indices *I* stand for INS and *G* for GPS. The uncertainty effect of the measurement noise is modeled at measurement noise covariance matrix, $\mathbf{R}_{6x6}$, defined in the following section.

2.3.6 Noise Statistics

GPS/INS integration is performed via an Extended Kalman filter. The performance, convergence and sensitivity characteristics of the filter depend on the modelling accuracy. A dynamic model consists of two main components: state transition matrix given in Section 2.3.4, and covariance matrices. The state transition matrix holds the information of how a dynamical system behaves in time while covariance matrices represent the uncertainty on the states, measurements and inputs of the system. Having derived the transition matrix in Section 2.3.4, this section will focus upon the covariance matrices used in the Kalman filter, which are the initial state covariance matrix $\mathbf{P}_{0,}$ measurement covariance matrix **R** and process noise covariance matrix **Q**.

2.3.6.1 Initial State Covariance Matrix, $\mathbf{P}_0$

The initial conditions on the state covariance matrix elements are assigned based on the uncertainty level of the initial alignment process which is achieved by a theodolite (a kind of azimuth angle measurement system), GPS position and velocity measurements, and accelerometer and gyroscope long term repeatability characteristics. Consequently, initial state covariance matrix **P**$_0$ is formed from the standart deviations given in the GPS, sensor and theodolite specifications used for initial alignment. The GPS provides position solution with a standard deviation of 7 meters in horizontal channels and 10 meters in height channel, while it provides velocity with a standard deviation of 0.1 m/s. On the other hand, initial attitude is determined using a theodolite system with a maximum error of 1 mrad.

As a result, the transformed standard deviations for position are $\sigma_L = \dfrac{7}{R_N + h}\ rad$, $\sigma_\lambda = \dfrac{7}{(R_E + h)\cos L}\ rad$, $\sigma_h = 10$ m. Velocity and attitude uncertainty in all dimensions are $\sigma_V = 0.1\,\text{m/s}$, $\sigma_{att} = 1\ \text{mrad}$, respectively. Accelerometer and gyroscope long term repeatability are given as $\sigma_{acc} = 3\ \text{mg}$, $\sigma_{gyro} = 6\ °/\text{hr}$, respectively. Assigning the related variances to the corresponding diagonal elements, the initial state covariance matrix **P**$_0$ is generated for the GPS/INS system.

2.3.6.2 Measurement Covariance Matrix, R

Measurement covariance matrix represents the GPS measurement noise. Therefore, it is based on the GPS position and velocity error characteristics. Exploiting the GPS specifications, the **R** matrix is

$$\mathbf{R} = \begin{bmatrix} \sigma_L^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_\lambda^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_h^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_v^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_v^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_v^2 \end{bmatrix} \tag{2.33}$$

where the transformed standard deviations for position are $\sigma_L = \dfrac{7}{R_N + h}\ rad$, $\sigma_\lambda = \dfrac{7}{(R_E + h)\cos L}\ rad$, $\sigma_h = 10$ m and velocity uncertainty is $\sigma_V = 0.1$ m/s due to GPS position and velocity output uncertainty given before.

2.3.6.3 Process Noise Covariance Matrix, Q

The entries of the **Q** matrix are produced based on the sensor specifications given in Table 2.1 and

Table 2.2. **Q** matrix elements denote the effects of sensor scale factor, misalignment, random walk and bias. The process noise covariance matrix can be constructed from 3x3 submatrices as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_p & \mathbf{Q}_{pv} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{Q}_{pv} & \mathbf{Q}_v & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_{att} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_g \end{bmatrix} \tag{2.34}$$

The critical issue in generating the **Q** matrix is to consider the effects of the error sources, which are not included as the Kalman filter states. This is achieved by introducting the effects of these errors into the related states available in the Kalman filter, hence in the Q matrix.

Attitude and velocity error dynamics of the system is

$$\begin{aligned} \dot{\boldsymbol{\varepsilon}} &= .... - \mathbf{C}_b^n \delta\boldsymbol{\omega}_{ib}^b \\ \delta\dot{\mathbf{v}}^n &= .... + \mathbf{C}_b^n \delta\mathbf{f}^b \end{aligned} \tag{2.35}$$

Process noise covariance on the velocity resulting from the neglected scale factor, misalignment and random walk is formulated as

$$\mathbf{Q}_v = E\left\{\mathbf{C}_b^n \delta\mathbf{f}^b \delta\mathbf{f}^{b,T} \mathbf{C}_b^{n,T}\right\}\Delta t = \mathbf{C}_b^n E\left\{\delta\mathbf{f}^b \delta\mathbf{f}^{b,T}\right\}\mathbf{C}_b^{n,T}\Delta t$$

$$= \Delta t \sigma_{sa}^2 \mathbf{C}_b^n \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}^T \mathbf{C}_b^{n,T} + \Delta t \sigma_{ma}^2 \mathbf{C}_b^n \begin{bmatrix} f_y + f_x \\ f_x + f_z \\ f_x + f_y \end{bmatrix}\begin{bmatrix} f_y + f_x \\ f_x + f_z \\ f_x + f_y \end{bmatrix}^T \mathbf{C}_b^{n,T} + \Delta t \begin{bmatrix} \sigma_{ra}^2 \\ \sigma_{ra}^2 \\ \sigma_{ra}^2 \end{bmatrix} \quad (2.36)$$

where f's are the components of the specific force, σ_{sa}^2 , σ_{ma}^2 , σ_{ra}^2 denotes scale factor, misalignment and random walk variances of the accelerometer and Δt is the propagation time interval of 0.1 seconds. Likewise noise covariance on the attitude stemming from scale factor, misalignment and random walk is

$$\mathbf{Q}_{att} = E\left\{\mathbf{C}_b^n \delta\boldsymbol{\omega}^b \delta\boldsymbol{\omega}^{b,T} \mathbf{C}_b^{n,T}\right\}\Delta t = \mathbf{C}_b^n E\left\{\delta\boldsymbol{\omega}^b \delta\boldsymbol{\omega}^{b,T}\right\}\mathbf{C}_b^{n,T}\Delta t$$

$$= \Delta t \sigma_{sg}^2 \mathbf{C}_b^n \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}^T \mathbf{C}_b^{n,T} + \Delta t \sigma_{mg}^2 \mathbf{C}_b^n \begin{bmatrix} \omega_y + \omega_x \\ \omega_x + \omega_z \\ \omega_x + \omega_y \end{bmatrix}\begin{bmatrix} \omega_y + \omega_x \\ \omega_x + \omega_z \\ \omega_x + \omega_y \end{bmatrix}^T \mathbf{C}_b^{n,T} + \Delta t \begin{bmatrix} \sigma_{rg}^2 \\ \sigma_{rg}^2 \\ \sigma_{rg}^2 \end{bmatrix} \quad (2.37)$$

where ω's are the components of the rotation rate, σ_{sg}^2 , σ_{mg}^2 , σ_{rg}^2 denotes scale factor, misalignment and random walk variances of the gyroscope and Δt is the propagation time interval. σ_{sa}^2 , σ_{ma}^2 , σ_{ra}^2 and σ_{sg}^2 , σ_{mg}^2 , σ_{rg}^2 are calculated using the performance specifications in Table 2.1 and

Table 2.2 and IMU test results performed. Accelerometer scale factor variance is found as $\sigma_{sa}^2 = \left(\frac{3007}{1{,}000{,}000}\right)^2$ by combining the long term and short term components given at the specifications. Gyroscope scale factor variance is calculated in a similiar manner as $\sigma_{sg}^2 = \left(\frac{2062}{1{,}000{,}000}\right)^2$. During the IMU tests, sensor axis misalignments are compensated and decreased to 0.2 mrad (1-sigma) value. Accelerometer and gyroscope noise levels are 50 µg and 0.0076 °/s at 64 Hz. These noise levels are represented by σ_{ra}^2 and σ_{rg}^2 , respectively where

$\sigma_{ra}^2 = 3.7\text{e}-9(\text{m/s})^2/\text{s}$ and $\sigma_{rg}^2 = 2.7\text{e}-10\,\text{rad}^2/\text{s}$. The sensor measurement noise σ_{ra}^2 also contributes to the position and position-velocity components of **Q** matrix as

$$\mathbf{Q}_p = \left(\frac{\Delta t^3}{3}\right)\sigma_{ra}^2 \begin{bmatrix} \frac{1}{(R_N+h)^2} & 0 & 0 \\ 0 & \left(\frac{1}{(R_E+h)\cos L}\right)^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.38}$$

and

$$\mathbf{Q}_{pv} = \left(\frac{\Delta t^2}{2}\right)\sigma_{ra}^2 \begin{bmatrix} \frac{1}{(R_N+h)} & 0 & 0 \\ 0 & \frac{1}{(R_E+h)\cos L} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.39}$$

As given in Section 2.3.4, accelerometer and gyroscope bias states are modeled by a Gauss-Markov process with a time constant of 12,000 seconds. Corresponding noise variances of these states, $\mathbf{Q}_a$ and $\mathbf{Q}_g$, are calculated from IMU tests, as follows:

$$\begin{aligned} \mathbf{Q}_a &= \sigma_a^2 \mathbf{I}_{3x3} \\ \mathbf{Q}_g &= \sigma_g^2 \mathbf{I}_{3x3} \end{aligned} \tag{2.40}$$

where $\sigma_a^2 = 2.22\text{e}-9(\text{m/s}^2)^2$ and $\sigma_g^2 = 1.4\text{e}-16\ \ (\text{rad/s})^2$.

Having determined system error behavior, measurement models, noise statistics and related covariance matrices, we have the necessary tools to apply GPS/INS integration filter and to estimate the navigation system errors.

2.3.7 Correction

The estimated errors are fed back to the inertial sensors and inertial navigation system after every update step of Kalman filter at 1 Hz. Corrections to the sensor and navigation states at kth step are realized as:

$$\begin{aligned}
&\mathbf{p} = \hat{\mathbf{p}} - \delta\mathbf{p} \\
&\mathbf{v}^n = \hat{\mathbf{v}}^n - \delta\mathbf{v}^n \\
&\mathbf{C}_b^n = \left(\mathbf{I} - \boldsymbol{\varepsilon}\mathbf{x}\right)\hat{\mathbf{C}}_b^n \\
&\Delta\mathbf{w} = \Delta\hat{\mathbf{w}} - \Delta t_m \sum_{i=1}^{k} \delta\mathbf{f}_i \\
&\boldsymbol{\alpha} = \hat{\boldsymbol{\alpha}} - \Delta t_m \sum_{i=1}^{k} \delta\boldsymbol{\omega}_i
\end{aligned} \tag{2.41}$$

where $\mathbf{p}$, $\mathbf{v}^n, \mathbf{C}_b^n$, $\Delta\mathbf{w}$, $\boldsymbol{\alpha}$, denote respectively position and velocity, direction cosine matrix, velocity increment output of accelerometer and angle increment output of gyroscope. The terms with hat are computed (actual) parameters, while the others represent true parameters. Δt_m, the output interval of the inertial sensors, is 1/400 seconds in this implementation. Kalman filter directly provides all the correction terms in (2.41).

2.4 Sensitivity Analysis

As stated in Appendix D, Kalman filter is an optimal estimator of system states under the assumption that system model is linear contaminated by white Gaussian system and measurement noises.

However, the filter requires exact models of dynamics, measurements and the exact knowledge of process noise covariance (**Q**), measurement noise covariance

(**R**) and initial state covariance (**P₀**) in order to provide optimal gains (**K**), thus optimal estimation. Otherwise, the filter estimations will either be suboptimal or diverge. As a result, filter sensitivity to covariance matrices and dynamic models becomes critical for performance.

In our GPS/INS system, we make approximations to determine the covariance matrices and hence the Kalman gain, **K**. Thus, the sensitivity of the filter to process noise covariance (**Q**), and initial state covariance (**P₀**) is critical for our system. Since the measurement noise covariance (**R**) is created based on the exact values given in (Novatel, (1995)), filter sensitivity to measurement covariance is not considered in this thesis work.

In order to determine if our nominal covariance matrices **Q**, and **P₀** are adequate, a short test is conducted. The position errors of the INS are recorded for 200 seconds during the test. These position errors are then compared to Kalman filter estimated position variances given by,

$$\mathbf{P}_k = \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}\mathbf{\Phi}_{k-1}^T + \mathbf{Q}_{k-1} \tag{2.42}$$

The analysis is repeated with 0.1x**Q**, 10x**Q**, 0.1x**P₀** and 10x**P₀** (Figure 2-10 and Figure 2-11).

Figure 2-10 gives the INS position error in north, east and down channels with the Kalman filter estimated error variances. The Kalman filter is initiated with 0.1x**P₀**, **P₀** and 10x**P₀** in order to estimate the error variance and it is observed that the nominal values estimates the uncertainty better than 0.1x**P₀**, and 10x **P₀** initiated estimators.

Figure 2-11 give the INS position errors with the Kalman filter estimated error variances which possess process covariance values of 0.1x**Q₀**, **Q₀** and 10x**Q₀**. It is seen from the figure that the estimated error variance does not change for any choice of process covariance. Thus, the filter is insensitive to the modeling errors in process covariance matrix.

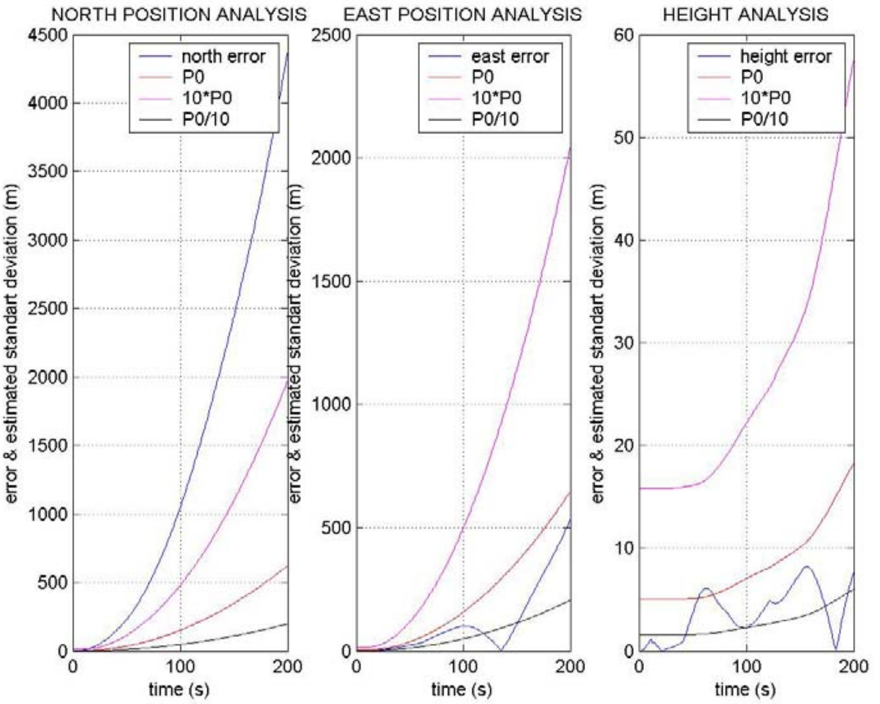

Figure 2-10 Position error standard deviations vs. position error

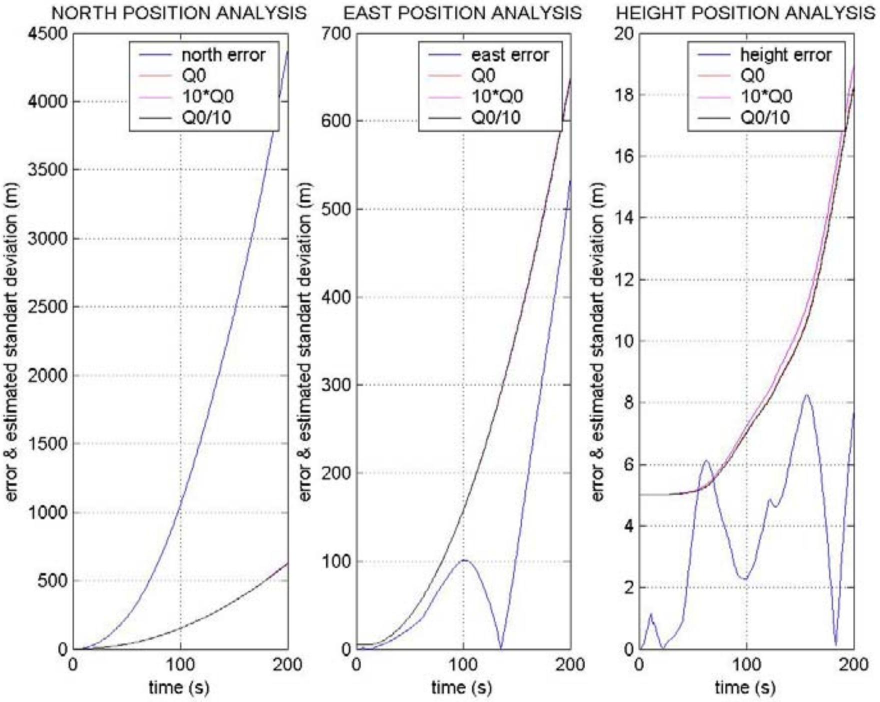

Figure 2-11 Position error standard deviations vs. position error

We conclude that the filter is sensitive to the initial state covariance while it shows no sensitivity to process noise covariance errors. Moreover, it is seen in Figure 2-10 and Figure 2-11 that our nominal **Q** and $\mathbf{P_0}$ matrices given in Section 2.3.6 are adequate for capturing the true error propagation.

2.5 Some Practical Problems: Latency and Lever Arm Effect

There exist two main issues that have to be considered in the implementation of an integrated navigation system. These are latency and lever arm. Latency can be described as the time lag between the GPS signal receive time and actual INS time. Lever arm is the positional difference between the IMU location and GPS antenna location relative to where the GPS position and velocity solutions are provided.

GPS measurement is received with a time lag from the current INS navigation solutions. Depending on the GPS receiver employed, this lag can exceed hundreds of milliseconds and causes high drifts for a highly dynamical vehicle. In an airborne vehicle flying at 900 km/h, a 10 msec lag can cause 2.5 meters of error. The mentioned time offset is the first source of the latency in the GPS/INS system. The graphical description of the latency is given in Figure 2-12. In order to circumvent the problem stemming from the time offset, navigation data is stored in memory within a specified time window until the valid GPS data is received. INS position and velocity solutions corresponding to the GPS data validity time, t_{gps}, are found by interpolating navigation solutions at time t_k and t_{k+1} since the previous navigation solutions are buffered in memory. And a precise measurement at t_{gps} is then obtained.

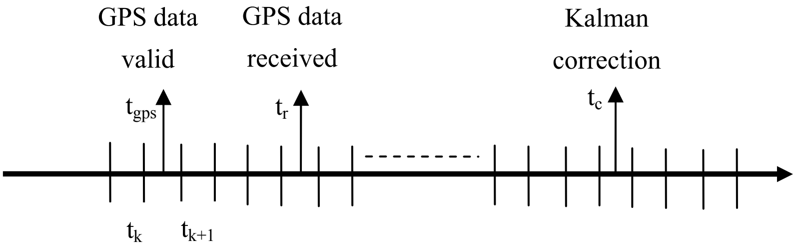

Figure 2-12 Graphical representation of the data latency

Another source of the latency is the time passing after the GPS measurement is acquired until the Kalman correction is made. This time difference prevents the filter to relate the measurements with corresponding error states that propagate within this time difference.

The measurement at t_{gps}, which is the GPS measurement validity time, can be expressed as

$$\mathbf{z}\left(t_{gps}\right) = \mathbf{H}\mathbf{x}\left(t_{gps}\right) \tag{2.43}$$

In order to relate it to the error states at the correction time, t_c, this measurement can be rearranged using the state transition matrix as

$$\mathbf{z}\left(t_{gps}\right) = \mathbf{H}\mathbf{\Phi}\left(t_{gps}, t_c\right)\mathbf{x}\left(t_c\right) \tag{2.44}$$

Thus, the measurement acquired at GPS validity time, t_{gps}, is connected to the error states at the correction time, t_c. The new measurement matrix taking the lag between the measurement and the correction into account is defined as

$$\mathbf{H}_c = \mathbf{H}\mathbf{\Phi}^{-1}\left(t_c, t_{gps}\right)\mathbf{x}\left(t_c\right) \tag{2.45}$$

Using the new measurement matrix, $\mathbf{H}_c$, the erroneous effect of the time offset between the measurement and the correction is compensated.

The other problem in the practical implementation is the GPS antenna lever arm. GPS provides position and velocity of the antenna location; however IMU position and velocity is required to correct erroneous INS solutions. Otherwise, the integrated system solutions will shift from the true trajectory. Thus, GPS measurements at the antenna are translated into the IMU location (Figure 2-13).

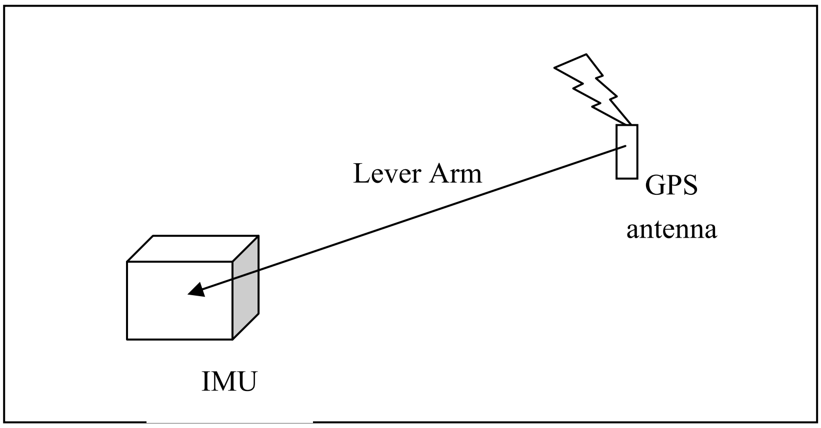

Figure 2-13 GPS antenna lever arm

The lever arm vector between GPS antenna and IMU is expressed in body coordinates as follows:

$$\mathbf{LA}^b = \begin{bmatrix} LA_x \\ LA_y \\ LA_z \end{bmatrix} \tag{2.46}$$

The lever arm effect on the GPS position measurement is

$$\mathbf{p}_{LA} = \begin{bmatrix} \dfrac{1}{(R_N + h)} & 0 & 0 \\ 0 & \dfrac{1}{\left(R_E + h\right)\cos L} & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{C}_b^n \mathbf{LA}^b \tag{2.47}$$

and the effect on the velocity measurement is the velocity induced by the rotation of the vehicle and can be expressed as

$$\mathbf{v}_{LA} = \mathbf{C}_b^n \left(\boldsymbol{\omega}^b x \mathbf{LA}^b \right) \tag{2.48}$$

(2.47) and (2.48) are added to the position and velocity measurements in order to prevent the shifts from true trajectory and provide the navigation solutions with respect to IMU location.

2.6 GPS/INS Performance in Real Time Applications

In this section, we will demonstrate the performance of GPS/INS integration highlighting its shortcomings that have built the motivation of this thesis work and have guided the development of our proposed method that will be introduced in Chapter 3.

2.6.1 Experimental Setup

Since real time tests are performed on the ALV that we use to demonstrate our proposed approach, we will introduce our ALV experimental setup in this portion of the thesis. This section provides the details of the real time system which is an autonomous land vehicle equipped with the hardware used in this thesis work. This system is used throughout the thesis for real time implementation, demonstrations and experimental tests, that we document it here in the thesis report.

The vehicle tests were carried out in the campus of Middle East Technical University populated with trees and buildings. A 600 second test is conducted on a closed path in the campus site incorporating as much as possible disturbances and diversity demonstrating properly the navigation performance. The campus site test trajectory is given in Figure 2-17. The campus path circles around a 1500mx400m area and generates a closed path of nearly 5 kilometers long where the main portion remains in the north direction.

The vehicle given in Figure 2-14 is the test vehicle used during the land tests conducted in the balance of this thesis work. The vehicle contains inertial navigation system, a power unit, GPS and two computers. A laptop computer is used to store GPS and IMU data and a desktop computer is employed in system performance monitoring.

More specifically, the hardware contains a GPS/INS integrated navigation system employing a tactical grade IMU, navigation computer and a GPS receiver manufactured in TÜBİTAK-SAGE. The SAGE-IMU contains the three µfors-6 gyroscopes and one B-290 three-axis accelerometer already introduced in Section 2.2.1 in an orthogonal arrangement as shown in Figure 2-15. The GPS/INS system containing TMS320C31 microprocessor for navigation computation and a Novatel 3151R receiver is mounted on the test vehicle for navigation purposes (Figure 2-16). Raw IMU data were logged at 400 Hz, while GPS data were logged at 1 Hz. Both GPS and IMU data were stored through a serial port on the laptop computer.

The vehicle starts the test from stand still and experiences velocities around 70 km/h in the campus test. The vehicle travels in a closed path on the campus site test trajectory (Figure 2-17) where half of the trajectory is populated with trees and buildings which is problematic for GPS signals and other half is an open space. During the test, GPS/INS navigation solutions were recorded in order to conduct follow up performance analyses of the integrated navigation system. The

Figure 2-14 The van is used as a test vehicle for GPS/INS system. It contains the GPS antenna and GPS/INS unit. It also houses laptop and desktop computers for monitoring and real time data collection.

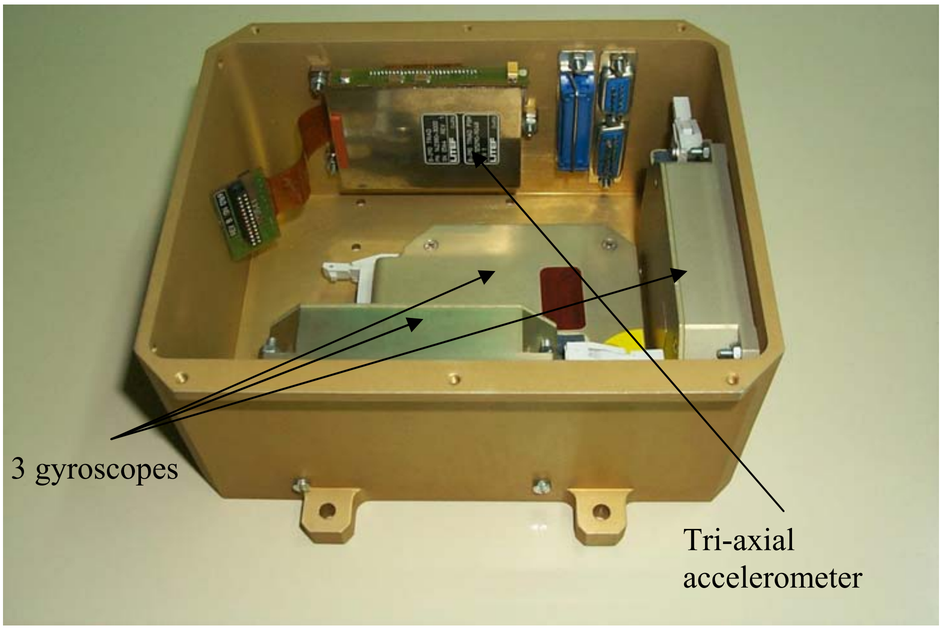

Figure 2-15 The SAGE IMU contains three gyroscopes and one three-axis accelerometer in an orthogonal arrangement.

Figure 2-16 GPS/INS system mounted on the test vehicle for navigation purposes and real time data collection.

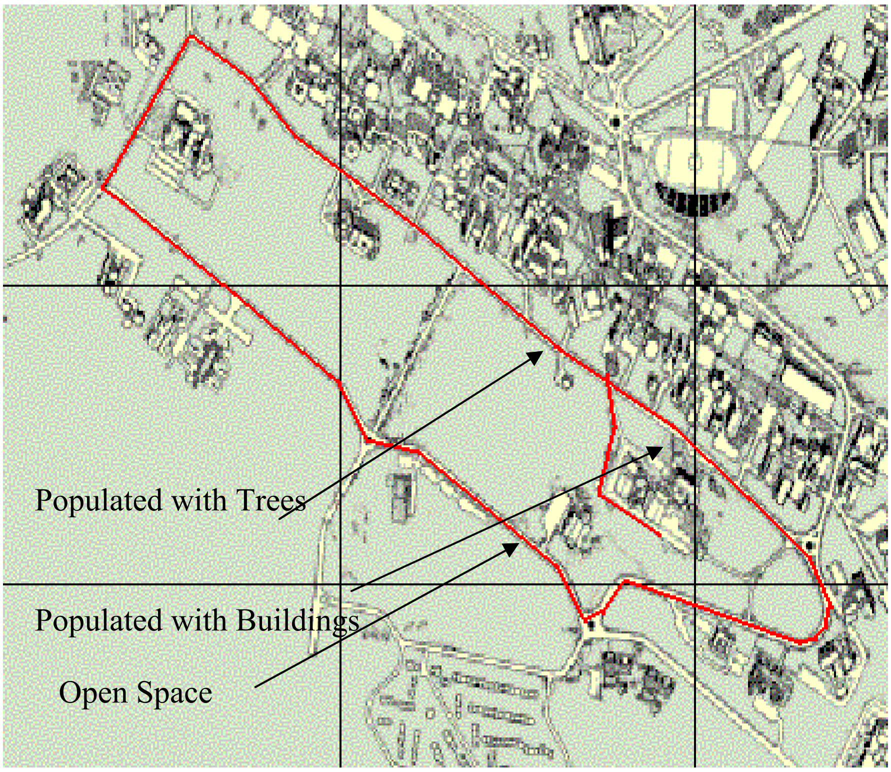

Figure 2-17 The Middle East Technical University campus. The land test is conducted on a closed loop trajectory given in the figure.

desktop computer has been used to monitor real time navigation solutions and overall performance status of the whole system.

2.6.2 Performance Test Results

The experimental results of the tactical grade loosely coupled GPS/INS system are presented in this section. The test is conducted on the path in the Middle East Technical University campus, previously explained. Test results illustrate the performance of our navigation system over a variable landscape with changing inclination and height, with and without GPS signals. Figures describing the test results have been generated and discussed where dotted lines denote the inertial navigation system solutions and solid lines show the GPS/INS solutions.

Figure 2-18 shows the GPS/INS and INS solutions solely for positional changes in the horizontal plane. As seen in the figure, “INS only” solution wanders out of the figure limit due to sensor errors at the first 100 seconds of the total 600 seconds test. Figure 2-19 presents the vertical positions provided by the inertial and the integrated systems. Inertial system solution again drifts away during the test. Figure 2-20 and Figure 2-21 present the GPS/INS velocity and attitude solutions. As seen in the figures, velocity and attitude solutions are not drifting because of GPS aid. The accelerometer bias estimates given in Figure 2-22, are seen to converge within the 300 seconds of the test. The z-axis bias converges to the real value as a result of high observability in this channel. From the gyroscope drift estimates given in Figure 2-23, it may be seen that only the z-axis drift converges throughout the test since z-axis is highly observable in the land tests due to the high rotations occurring in this axis. x-axis and y-axis are slightly observable as result of low rotation rates that these axes are exposed to. Figure 2-24 and Figure 2-25 present the Kalman corrections throughout the test and demonstrate that the filter relies on INS position and GPS velocity rather than on GPS position solution and INS velocity solutions due their high uncertainty.

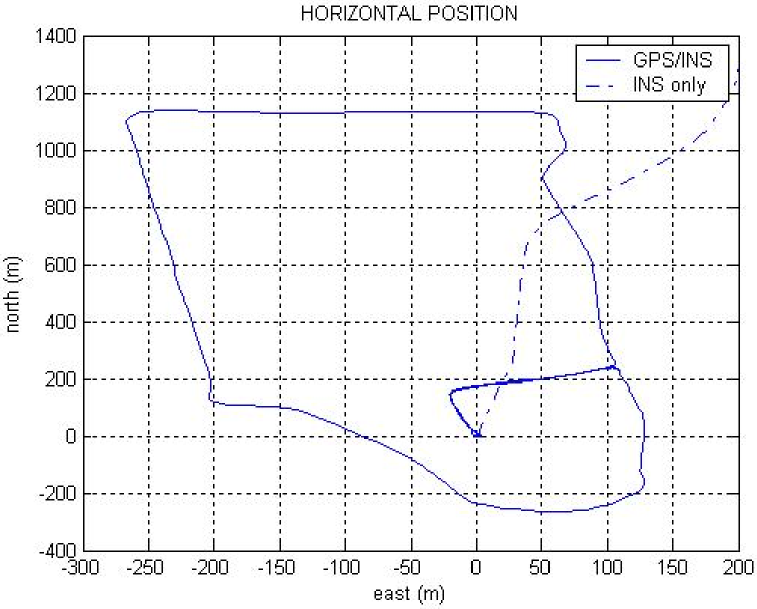

Figure 2-18 INS solution and integrated INS/GPS solution for horizontal channel. INS solution drifts very fast due to the bias and alignment errors of sensors.

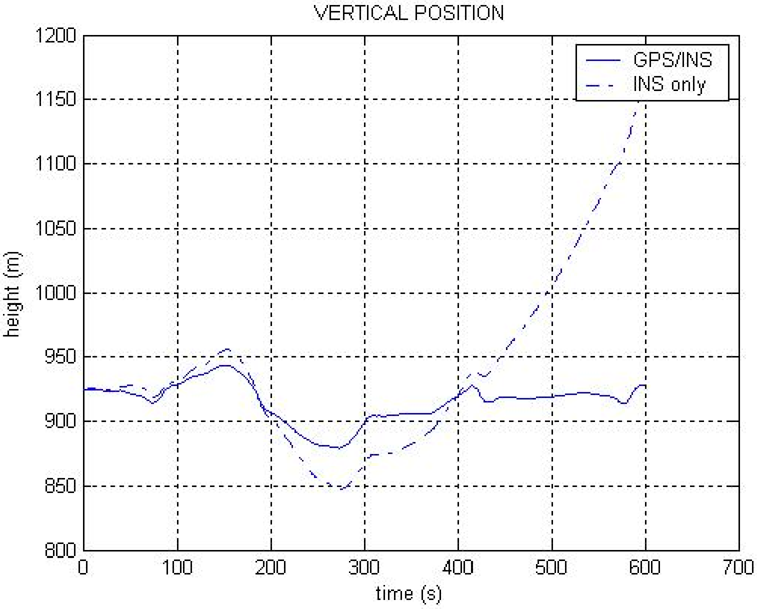

Figure 2-19 Test route profile by GPS/INS and INS only

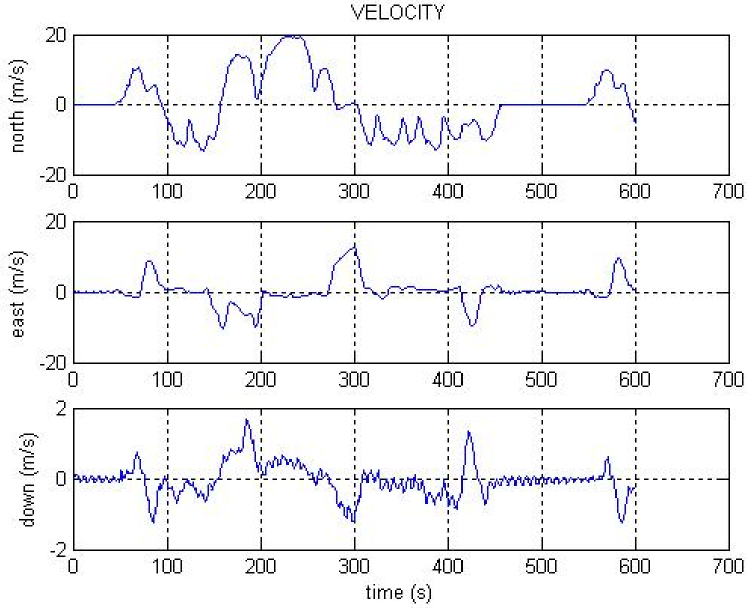

Figure 2-20 GPS/INS integration prevents velocity from wandering.

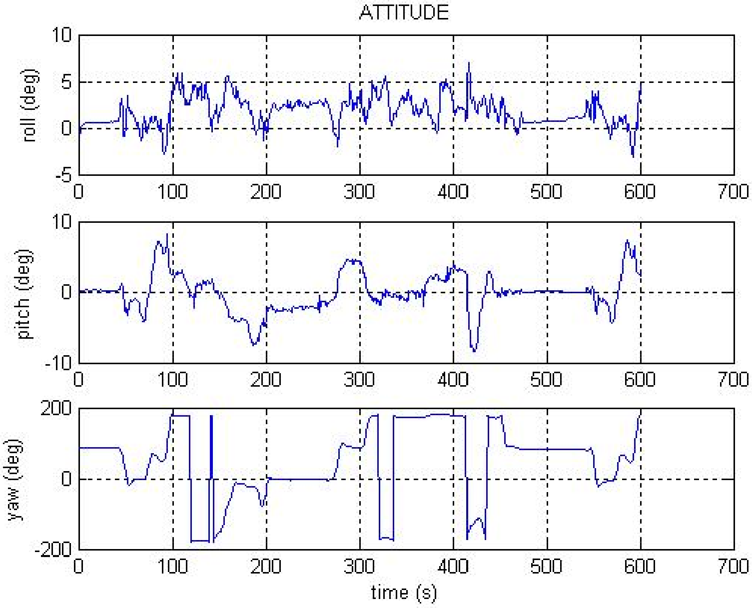

Figure 2-21 Roll and pitch angles stays under 10º in the land tests. Yaw angle sweeps ±180º area.

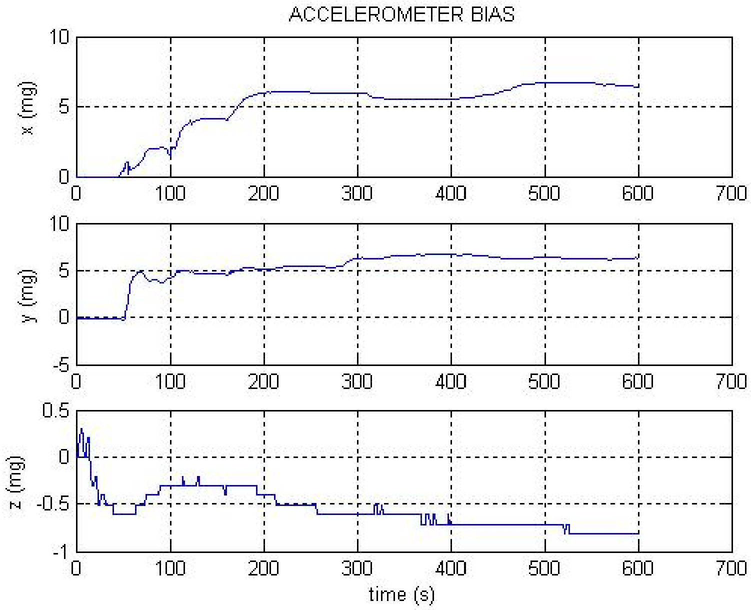

Figure 2-22 Accelerometer long-term bias values are estimated and converged at the end of first 300 seconds.

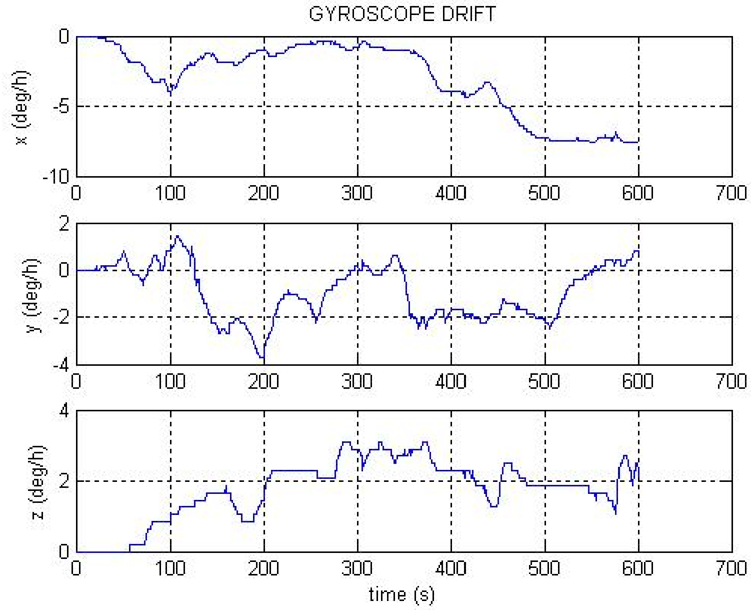

Figure 2-23 z-axis gyroscope long-term drift is estimated as a result of observability. x and y-axis gyroscope drifts are poorly observable.

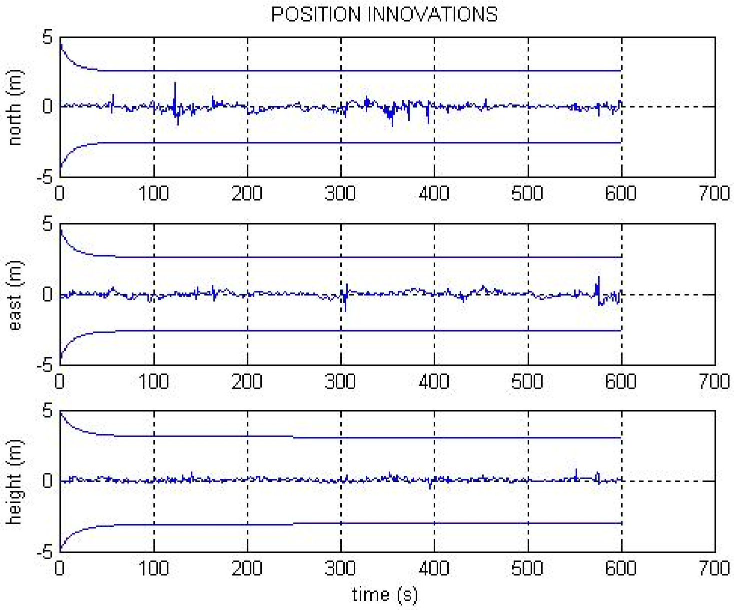

Figure 2-24 The position uncertainty of GPS/INS system is about 5 meters (1σ) in steady state. The innovations show that the filter relies on INS solutions due to high uncertainty of GPS position solution.

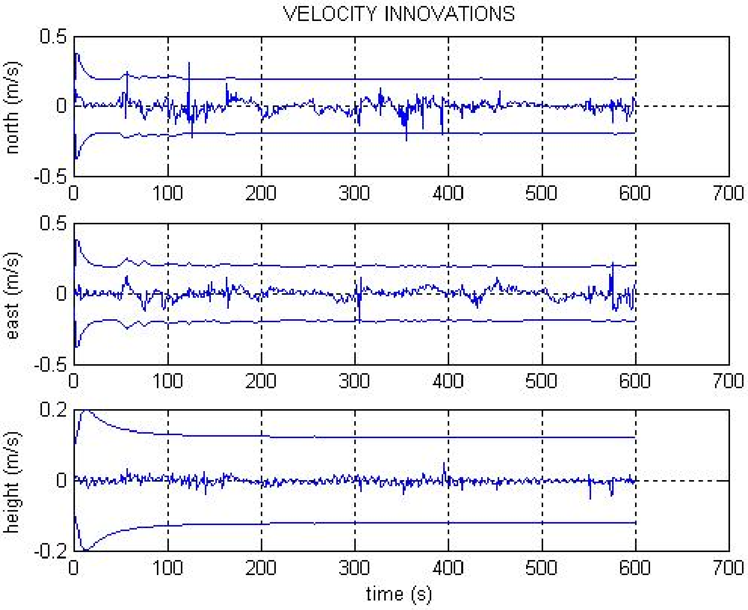

Figure 2-25 The horizontal velocity components use GPS velocity solutions more than the vertical channel due to the high uncertainty of INS velocity solution at this channel and low observability of horizontal bias terms.

2.7 Summary

The GPS/INS system proved to provide very accurate, reliable and robust navigation results. GPS aiding to correct position, velocity, attitude, accelerometer bias and gyroscope drift yields accurate navigation data.

INS only system performance relying on the erroneous rate and acceleration measurements is shown to be far from being accurate enough during the test runs. "INS only" position solution wandered at the first 100 seconds and became unacceptable for any navigation purpose.

Consequently, it is shown that the GPS/INS system begins to depend solely on INS, consisting of low cost sensors and drifts very fast out of its trajectory during the GPS signal loss. This is a significant problem for the autonomous vehicles depending on these low cost integrated navigation systems and can cause the loss of vehicles.

An intelligent method is proposed and developed in this thesis work. The proposed method is described in Chapter 3 of our thesis report.

CHAPTER 3

OUR PROPOSED SYSTEM: THE ANN AIDED GPS/INS INTEGRATED NAVIGATION

3.1 Introduction

Strapdown navigation systems rely on the erroneous rate and acceleration measurements to provide position and velocity. Thus, the inertial navigation system based upon such technology drifts with time without bound and the accuracy of strapdown systems are governed by the accuracy of the sensors. In order to improve the accuracy, a GPS receiver is successfully integrated to the navigation system as described in Chapter 2. The shortcomings of this integration that has motivated our work have been summarized at the end of the previous chapter. Although when the GPS signal is on the integrated system yields high performance navigation solution, it fails to receive GPS signals in a tunnel, a downtown area with high buildings, and a canyon or during a maneuver with high dynamics and would yield degraded navigation solutions. In order to circumvent the problem, possible approaches can be to employ higher grade inertial sensors or additional aiding sensors other than GPS. These approaches are not feasible in low cost applications since both leads to very expensive add-on solution that increase the complexity of the system. In order to handle the problem, we propose a system aiding the integrated GPS/INS system in a cost effective manner with an artificial neural network (ANN) architecture (Kaygısız (2003)).

This chapter provides the theoretical and practical aspects of the ANN aided GPS/INS system. The main contributions of this chapter are:

- Development of the general system architecture that integrates the neural network with the GPS/INS system.

- Determination of the optimum neural network architecture in a way compatible with the real-time hardware capability and performance requirements.

- Identification of the sensitivity of neural network usage in the enhancement of the GPS/INS system performance.

- Development of two different (position and position difference) aiding structures.

This chapter will use the land vehicle test setup and data given in Section 2.6 of Chapter 2 in order to provide the necessary comparative grounds for the system performance analysis required in the development phase of our proposed system.

3.2 System Architecture

A detailed block diagram of proposed GPS/INS+ANN system is shown in Figure 3-2. The inertial measurement unit (IMU) that we actually use is a triad of silicon accelerometers and fiber optic gyros mounted orthogonally inside a cubical case in order to measure specific force and rotation of body with respect to the inertial frame. This was fully introduced in Section 2.2.1 of Chapter 2. As we recall, gyros have bias values of 6 °/hr, while for accelerometers, biases are around 3 mg. IMU data are sampled at the rate of 100 Hz for autonomous land vehicle (ALV) test where the sampling rate is directly related to the highest dynamic frequency in the vehicle. The sampled data are then sent to the INS in order to generate the navigation output as described in Chapter 2.

GPS generates position and velocity outputs every second with bounded error that is less than 15 m for position and 0.05 m/s for velocity in actual ALV tests. The Kalman update is triggered at every GPS measurement (*1 Hz*) using the difference between GPS and INS solutions as input. Hence, the Kalman filter generates the corrections for diminishing the INS and IMU errors in order to keep the overall GPS/INS output within a bounded uncertainty.

An Artificial Neural Network (ANN) is integrated to the GPS/INS system as the main part of our proposed system and operates in two different phases namely the training and the prediction phases.

After every update triggered by GPS measurements, the ANN is trained using the GPS/INS states as input and position difference between the current and the previous update position as the output.

Whenever the GPS signal is absent, ANN leaves its training phase and every second estimates the position difference. Vehicle position is then calculated as the added value of the cumulative of the position differences occurring every second and the last GPS/INS position found when the GPS signal was last available. Alternatively, the position difference of every second may also be provided to Kalman filter. This calculated position or position difference is then used in the place of the nonexistent GPS position as input to the Kalman filter. Our approach helps to slow down drastically the drifting of the system when compared to the classical GPS/INS structure.

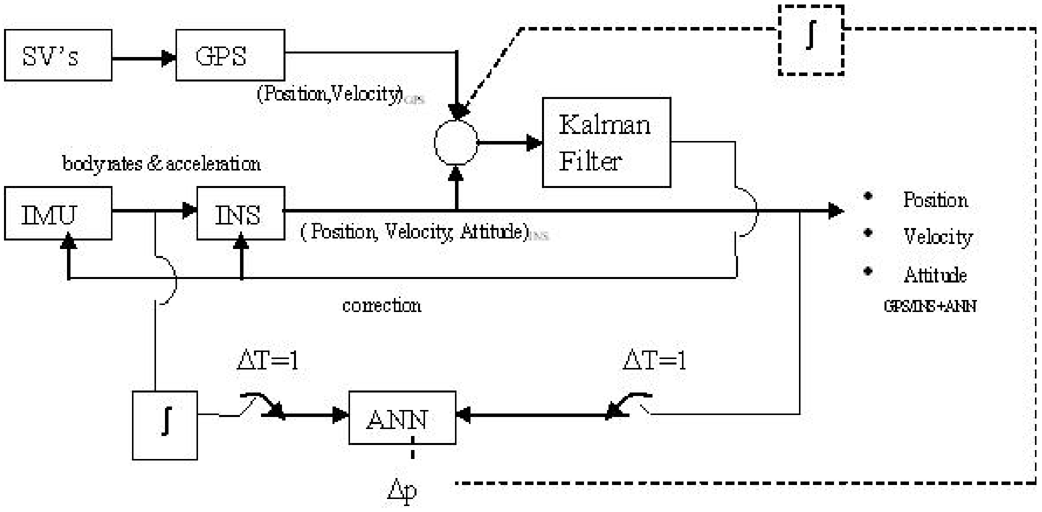

Figure 3-1 System Block Diagram

A detailed block diagram of GPS/INS+ANN in training phase is shown in Figure 3-2. After every update triggered by GPS measurements, a database is created for the ANN training phase by the recording of the previous updated states of GPS/INS and the cumulative of IMU velocity increments and angle increments of the last second as input. The position difference between the current and the previous positions are the recordings for the outputs of the ANN during its training until the loss of GPS signal.

The inputs to the ANN that are the previous velocity, attitude and summation of velocity and angle increment measurements are determined from the strapdown equations given in Chapter 2.

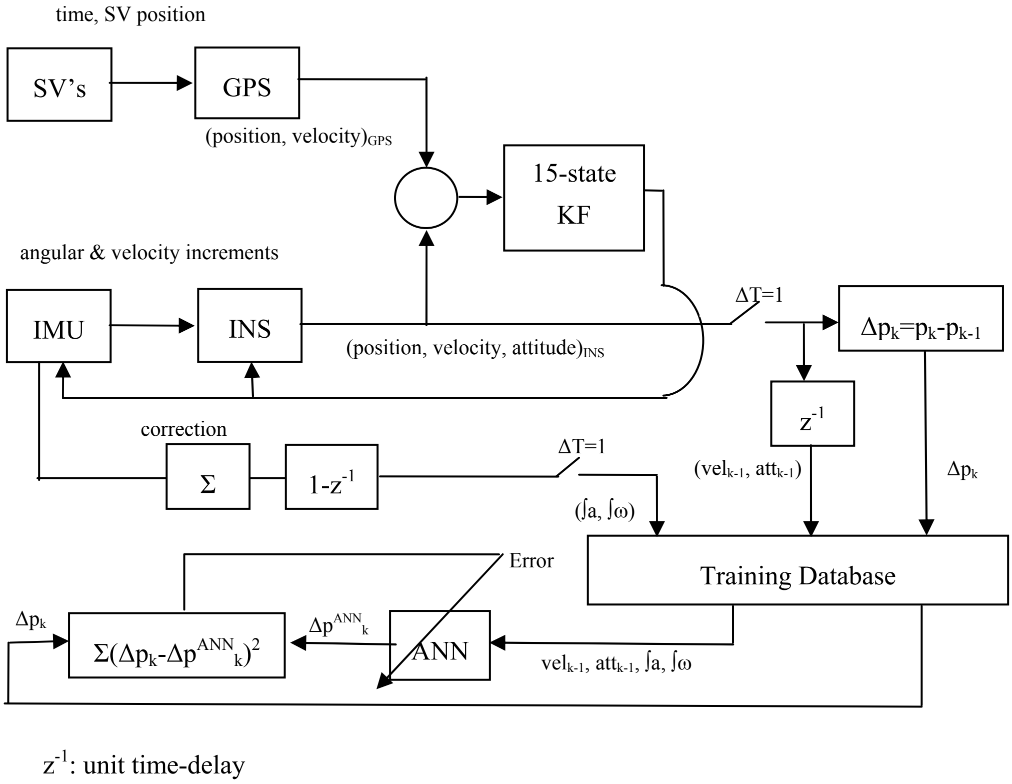

Figure 3-2 System block diagram (training phase)

The position difference $\Delta\mathbf{p}_k$ is a three dimensional vector with north position, east position and height as components, function of velocity, attitude states of the navigation system and the integral of acceleration (**a**) and rotation rates (ω) which appears as summation of angle and velocity increments:

$$\begin{aligned}\Delta\mathbf{p}_k &= \begin{bmatrix}\Delta L\\ \Delta\lambda\\ \Delta h\end{bmatrix} = \mathbf{p}_k - \mathbf{p}_{k-1} \\ &= f(\mathbf{v}_{k-1}, \mathbf{\Phi}_{k-1}, \int_{t_{k-1}}^{t_k} \mathbf{a}, \int_{t_{k-1}}^{t_k} \boldsymbol{\omega})\end{aligned} \tag{3.1}$$

The variables of the given function are inputs to the ANN of the navigation system. As the GPS signal is lost, the training phase starts and the forward and backward computations are iteratively repeated by injecting recorded data recursively into the network until the performance criterion of the minimum output mean square error is met.

Two different architectures can be exploited for intelligent aiding mechanism as mentioned before. Aiding may be achieved either by feeding ANN-predicted positions or ANN-predicted position differences. These two aiding mechanisms have shown similar performance characteristics as will be found in Chapter 4 since both mechanisms use the same information source: “Position aiding mechanism” uses the cumulative of ANN predicted position differences while “position difference aiding mechanism” directly uses predicted position differences to aid the navigation system. The main difference between these two mechanisms lies in the complexity. While “position aiding mechanism” requires a 18-state Kalman filter and a position calculation process, “position difference aiding mechanism” directly feed ANN outputs, which are the position differences to the 15-state Kalman filter. These two mechanisms and corresponding Kalman filter structures are described in detail in Section 3.4 and Section 3.5.

The block diagram of the prediction phase with position aiding is given in Figure 3-3. Whenever the performance criterion of mean squared error, which is reaching the global minimum of mean square error, is met, ANN leaves its training phase and estimates every second the position difference. These estimates are added to the last GPS/INS (GPS is on) position and a position estimate is obtained as:

$$\mathbf{p}_k = \mathbf{p}_l + \sum_{i=l}^{k} \Delta\mathbf{p}_i \tag{3.2}$$

where *l* is the last time when the GPS/INS together outputs a position solution for the vehicle. After *l*, GPS does not function and ANN starts to estimate the position. As a result, vehicle position is calculated as the cumulative sum of the position differences occurring every second, added to the GPS/INS position when the GPS signal was last available. This estimate is then used instead of the missing GPS position (Figure 3-3).

The block diagram of the prediction phase with position difference aiding is given in Figure 3-4. Whenever the performance criteria are met, ANN leaves its training phase and every second estimates $\Delta\mathbf{p}_i$, the position difference. These estimates are fed to the navigation system instead of the missing GPS position information (Figure 3-4).

Training, topology and the sensitivity analyses for these two architectures are common since they both employ the same neural network architecture in the training phase. As stated before, the main difference lies in the complexity of the Kalman filter structures employed for integration.

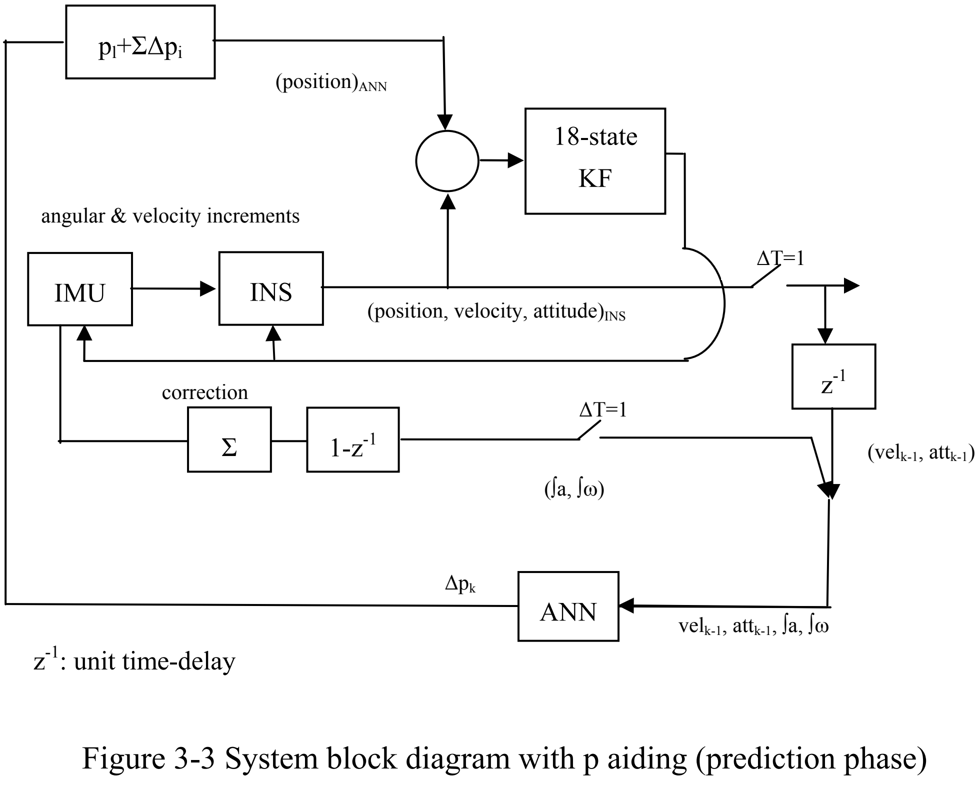

Figure 3-3 System block diagram with p aiding (prediction phase)

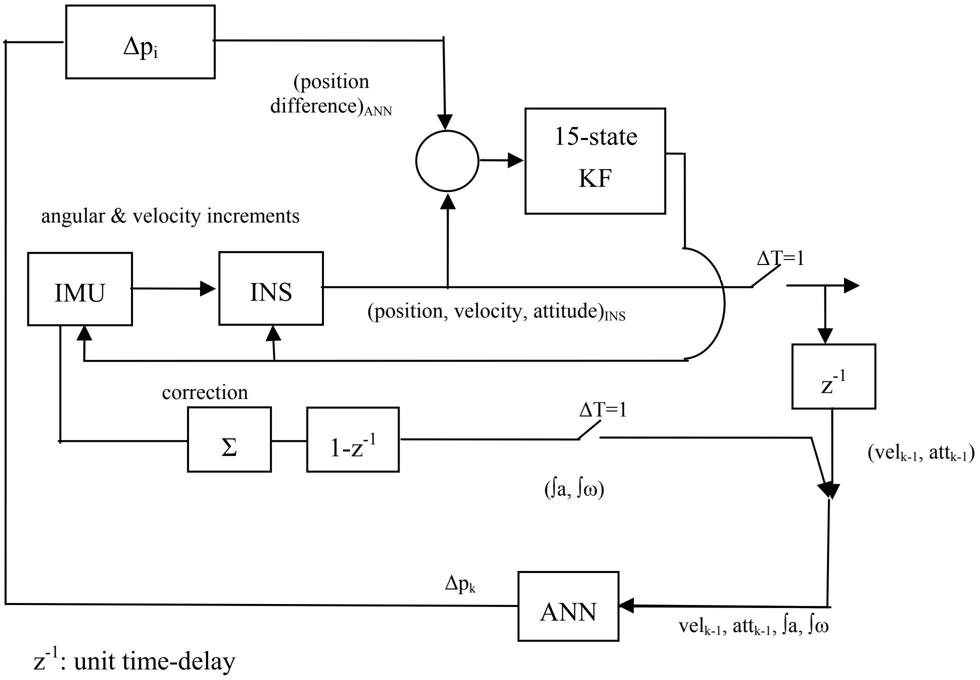

Figure 3-4 System block diagram with Δp aiding (prediction phase)

3.3 Neural Network Structure

The multilayer perceptron (MLP)-based intelligent structure shown as ANN in Figure 3-2, Figure 3-3 and Figure 3-4 is basically an artificial neural network composed of three layers which are the input (preprocessing), hidden and output (post processing) layers. The hidden layers are composed of three parallel MLPs, each for learning navigation data of 3 different dimensions. The internal structure of the system is shown in Figure 3-5.

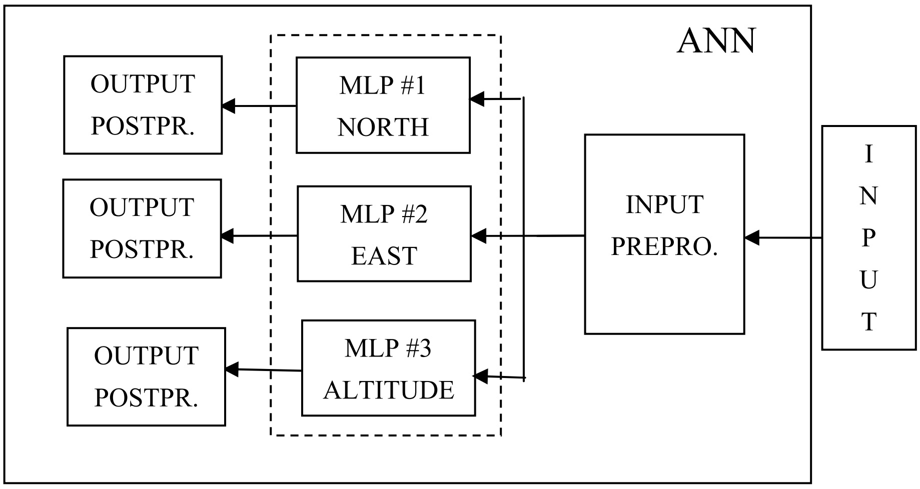

Figure 3-5 ANN based intelligent estimator

The neural network is made more efficient, reliable and stable when preprocessing is applied on the network inputs. Inputs are scaled so as to normalize the mean and standard deviation of the training set to render them zero mean and unity standard deviation. Normalizing values are obtained by calculating the mean and standard deviation of every input in the training set and used to normalize the inputs that are applied to the network in the prediction phase.

In the second layer of the structure, there exist three separate multilayer perceptron, each of which is employed to predict position difference in orthogonal directions. This architecture is exploited instead of three output single MLP architecture in order to avoid the degradation of one output while the others improve. This approach also increases the speed of convergence of the overall system by decreasing the number of neurons in each MLP so, in short, instead of one MLP with three outputs and high number of hidden layer neurons, the proposed ANN structure exploits three MLPs with very small number of neurons, each.

MLP estimated outputs are then sent to the last layer of the architecture where they are postprocessed. In the last layer, network outputs are scaled in order to normalize the mean and standard deviation of the training set such that the outputs will have zero mean and unity standard deviation during training phase.

The scaled outputs are then rescaled in order to provide $\Delta\mathbf{p}_k$ in the postprocessing layer. Since, the postprocessing and preprocessing arrange the network inputs and outputs so as to have zero mean and unity standard deviation, vehicles with different velocity profiles and dynamics would supply similar inputs and outputs to MLPs and this would help the ANN to adapt to different types of vehicles successfully.

3.3.1 Neural Network Topology

Each multilayer perceptron in the ANN architecture of the proposed ANN-aided GPS/INS system is a feedforward network with one or more layers between its input and output layers which is known to be a universal approximator. The number of layers and the number of nodes existing in each layer mainly depends on the complexity of the target function. If insufficient number of neurons is assigned to each layer, the neural network may fail to express the input/output relationship accurately. On the other hand, neural network with excessive number of neurons may show instability and tends to memorize the training set instead of learning the input/output relation.

We conduct a topology study that seeks for an optimum number of hidden layers and an optimum number of neurons in each layer. Candidate MLPs are chosen considering state space order of the system that aids to ANN. Candidate MLPs are each, trained with the data from the first 360 seconds of the land test and the convergence rate of each candidate is recorded. Moreover, each candidate MLP runs to estimate the vehicle altitude for the remaining 240 seconds of the test and the resulting altitude errors are compared in order to choose the optimum topology. Structure selected for height estimation is also accepted to be optimum for north and east position MLPs since the inputs and outputs are same in all dimensions.

Candidate MLPs run in training phase until their convergence rates fall below 2.5x10^{-2}. Their resulting performance characteristics in the training phase are given in Table 3.1.

Table 3.1 Performance of candidate networks

Topology	Transfer functions	Time (s)	MSE (m)
12x3x1	tansig-linear	1.220	0.00826
12x6x1	tansig-linear	1.107	0.00536
12x9x1	tansig-linear	1.300	0.00510
12x6x3x1	logsig-tansig-linear	1.197	0.00544
12x6x6x1	logsig-tansig-linear	1.729	0.00378
12x6x9x1	logsig-tansig-linear	2.537	0.00291

Based on the training performance information, the best candidate is the 12x6x3x1 topology based on its mean square error and time of convergence. In order to reach a final decision, each MLP runs to estimate the vehicle altitude using a 240 seconds data set different from the training set. The error resulting from each MLP is compared in Figure 3-6. Each network runs five times in order to eliminate the random effects stemming from network initialization and the average of five runs is given in the figure.

As seen in Figure 3-6, the architecture of 12x6x3x1 structure is also the best estimator among all candidates with the lowest error as time goes by. These results lead to the selection of this topology for the ANN architecture within the ANN-aided GPS/INS approach. This same architecture is also applied unchanged for the north and east position estimator MLPs given in Figure 3-5.

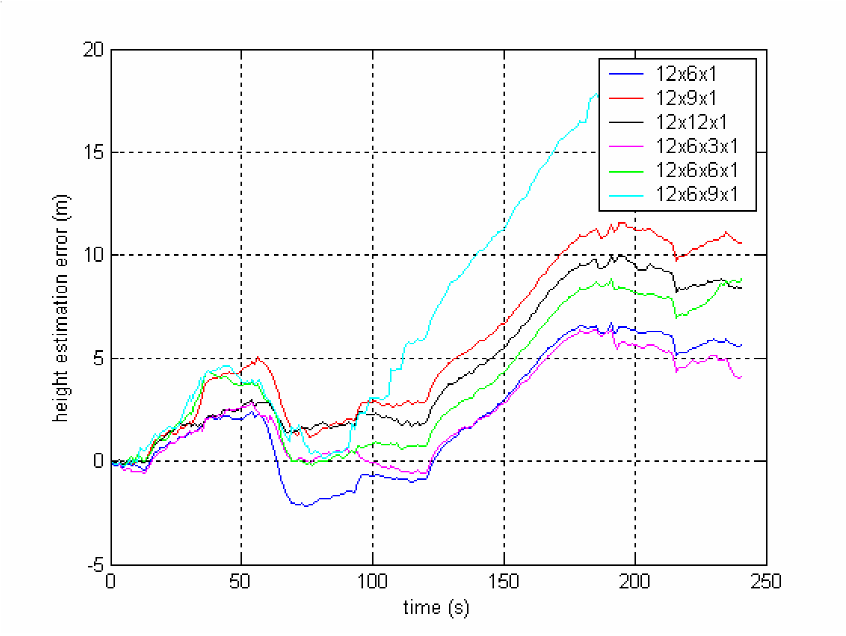

Figure 3-6 Altitude errors for different networks

3.3.2 Training Algorithm

The learning process applied to feedforward neural networks gives the network an ability of adapting to the changing environment and estimating the mathematical model of any dynamic process.

A critical and effective issue in determining how efficient would the proposed network be is the learning algorithm employed in the training process. In order to find out the most effective algorithm for the current concept, we will compare the results of different algorithms and choose the most efficient one for integrating it into the learning phase of our proposed network operation.

The comparison is done with the input data set constructed using the first 360 seconds of the land test given in Chapter 2, and the desired output is assigned to be height difference given in Figure 3-7. The neural network used for comparison is a 2-hidden layer network with 12x6x3x1 structure.

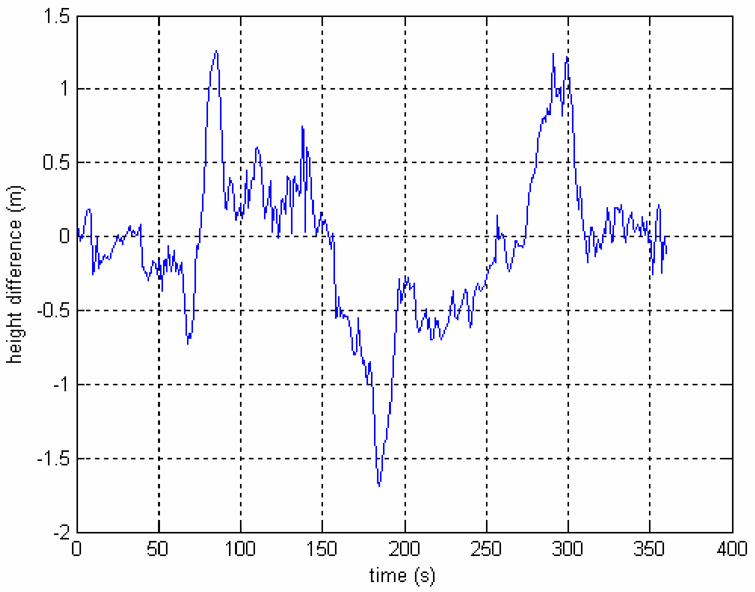

Figure 3-7 Desired output used for comparison

There are various learning algorithms with different efficiency and computational load. The difficulty in choosing a learning methodology for a defined problem stems from the fact that one algorithm is not universally optimum for every problem. In this study, speed of convergence of different learning algorithms is compared as a selection criterion. The algorithms compared are

- ordinary gradient descent,
- gradient descent with momentum parameter,
- gradient descent with adaptive step size,
- conjugate gradient,
- Levenberg-Marquardt

backpropagation algorithms. These algorithms run until the output mean squareerror reaches 0.01 meters. The run time for each algorithm is given in Table 3.2. As seen in Table 3.2, Levenberg-Marquardt is the most efficient and reliable learning algorithm among all. The convergence rate of the three fastest algorithm is given in Figure 3-8. As a result, Levenberg-Marquardt learning algorithm is implemented in our intelligent navigation system.

Table 3.2 Comparison of learning algorithms

Algorithm	Time (s)	MSE (m)
ordinary gd	20.748	0.0163
gd with momentum	17.319	0.0156
gd with adaptive step	1.473	0.00996
conjugate gradient	0.437	0.00999
Levenberg-Marquardt	0.204	0.00987

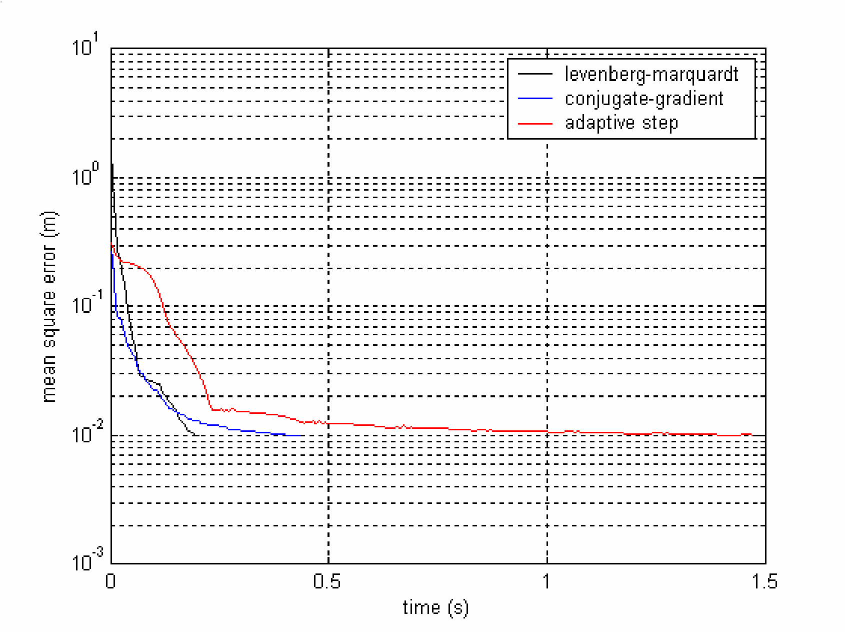

Figure 3-8 Convergence comparisons between 3 learning methods

3.4 ANN Predicted Position Aiding to GPS/INS

Intelligent navigation system shown in Figure 3-2 and Figure 3-3 aims to prevent the navigation system to drift when the GPS is off. As seen in Figure 3-3, ANN-predicted position measurements aid the navigation system by providing intelligent support to the INS in cases of GPS signal loss through a Kalman filter.

Measurements feeding into the Kalman filter also require a-priori information on measurement characteristics in order to model the measurement and associated noise in the Kalman filter. Kalman filter cannot function properly without such information and suitable filter design. Therefore, ANN predicted position characteristics and its impact on the Kalman filter are examined in the following subsections.

3.4.1 ANN Prediction Noise Characteristics

ANN employed in the intelligent navigation system estimates every second the north east and height position differences. As a result, these estimates are added to the last GPS/INS position before the GPS was lost and a position prediction of the current time is obtained as a cumulative sum of these estimates as:

$$\mathbf{p}_k = \mathbf{p}_l + \sum_{i=l}^{k} \Delta \mathbf{p}_i \tag{3.3}$$

where l is the last time when the GPS signals were received. After l, GPS is silent and ANN starts to predict the position. This prediction is then blended with the INS solutions using Kalman estimators. We will see in experimental analyses that the intelligent navigation system reaches then a more reliable and accurate solution.

In order to blend ANN predictions with the INS solutions in a Kalman filter, the uncertainty of the predictions should be determined. As seen in (3.3), position estimates may be examined in two main parts. One is $\mathbf{p}_l$, last GPS/INS position solution before GPS is lost and the initial position estimate for ANN. The other is,

$\sum_{i=l}^{k} \Delta \mathbf{p}_i$, the cumulative sum of the ANN predicted position differences. As a result, the true prediction takes the following form,

$$\mathbf{p}_k = \mathbf{p}_l + \sum_{i=l}^{k} \Delta \mathbf{p}_i \tag{3.4}$$

However, the computed prediction $\tilde{\mathbf{p}}_k$, carries its own uncertainty and differs from true prediction. $\mathbf{p}_l$ and $\sum_{i=l}^{k} \Delta \mathbf{p}_i$ contributes their individual errors of the added terms to the prediction and causes an uncertainty on the position estimation given in (3.4). These errors are the bias error on initial $\mathbf{p}_l$ and random walk error on $\sum_{i=l}^{k} \Delta \mathbf{p}_i$.

The uncertainty on $\mathbf{p}_l$ is stemming from the uncertainty on the last GPS/INS position. The level of the uncertainty on $\mathbf{p}_l$ is stored in the Kalman state covariance matrix **P** where matrix elements store and update the uncertainty of the GPS/INS solutions as the navigation continues. The first three diagonal elements of this 15x15 matrix are the position solution uncertainty that is, $\mathbf{P}_{11}$, $\mathbf{P}_{22}$ and $\mathbf{P}_{33}$ which are the north, east and height position uncertainties of the GPS/INS system. Consequently, the values of these elements in the last time step where GPS is active determines the uncertainty on $\mathbf{p}_l$. This uncertainty on the ANN estimation generates random bias characteristics on the ANN estimation.

The other source of uncertainty is the ANN prediction, $\sum_{i=l}^{k} \Delta \mathbf{p}_i$. In order to model this uncertainty, the noise characteristics of $\Delta \mathbf{p}_i$ should be examined. Noise on $\Delta \mathbf{p}_i$ will occur in different magnitudes in two different modes of the ANN system. It will take low values in the training phase, while it will be relatively high in the prediction phase. In order to model this uncertainty in the Kalman filter, one should know the noise level of the prediction phase. However, the only information in hand about the noise level is in the training phase.

In order to circumvent this problem, one should relate the levels of training and prediction phases and estimate the prediction phase uncertainty level. The land test data has been used to derive this relation. The neural network is trained with the data of the first 360 seconds of the land test and each network also runs to predict the three position differences for the remaining 240 seconds of the test. As a result, the noise levels shown in Figure 3-9, Figure 3-10 and Figure 3-11 are found. As seen in the figures, the training phase (first 360 seconds) noise power is less than the prediction phase (last 240 seconds) noise power in all directions. Comparing noise power at training and prediction phases, their interrelationship is determined. In training phase, the standard deviations of the noise on position differences in three dimensions are found to be $\sigma_n = 0.1975\ \text{m}$, $\sigma_e = 0.1002\ \text{m}$ and $\sigma_h = 0.0782\ \text{m}$ for north, east and height, respectively. In prediction phase, standard deviations increased to $\sigma_n = 0.3318\ \text{m}$, $\sigma_e = 0.3582\ \text{m}$ and $\sigma_h = 0.1406\ \text{m}$. From these values, standard deviations of prediction phase may be approximated as nearly three times of training standard deviations.

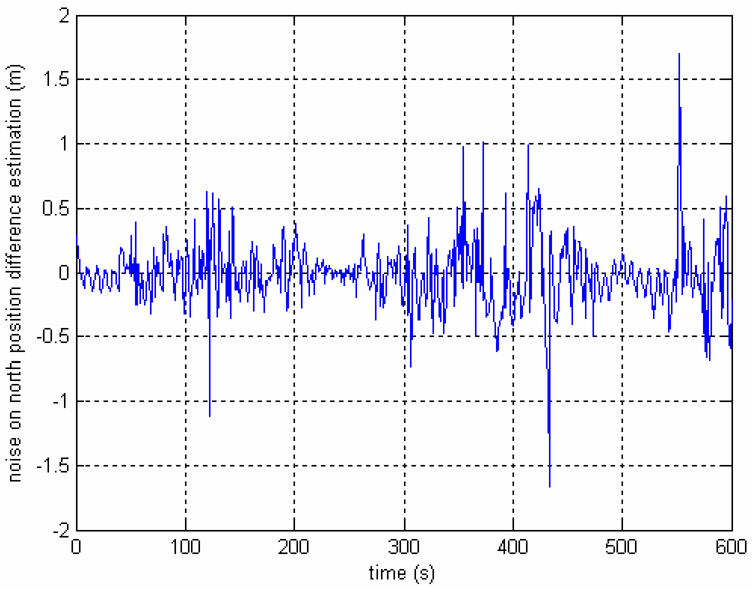

Figure 3-9 ANN north channel noise level in land test

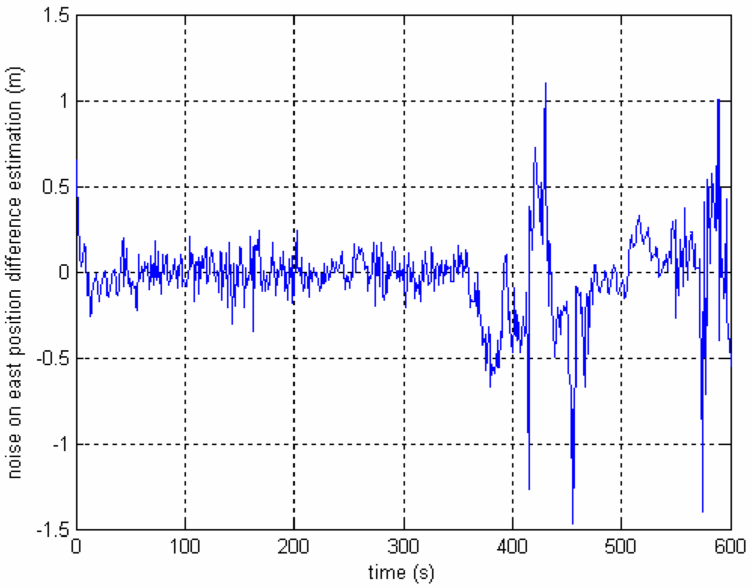

Figure 3-10 ANN east channel noise level in land test

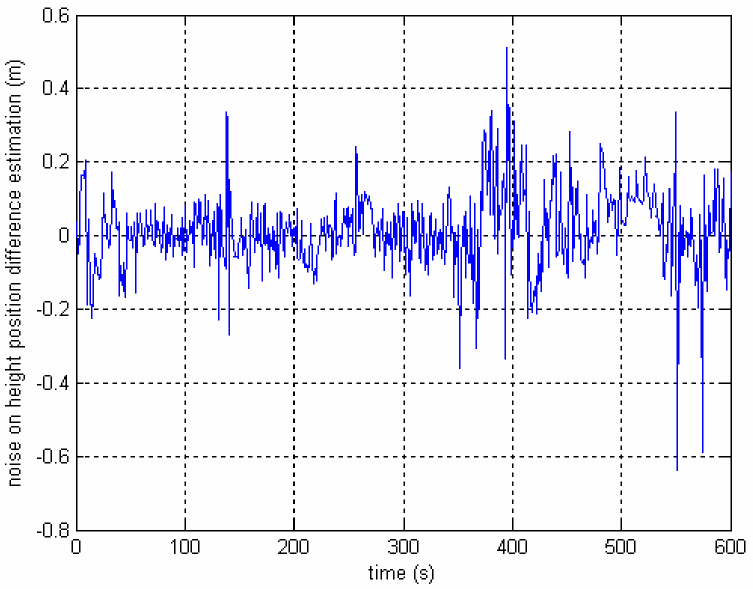

Figure 3-11 ANN height channel noise level in land test

Determining the uncertainty levels on $\mathbf{p}_l$ and $\sum_{i=l}^{k} \Delta\mathbf{p}_i$, computed ANN prediction $\tilde{\mathbf{p}}_k$, can be rewritten as

$$\tilde{\mathbf{p}}_k = \mathbf{p}_l + \mathbf{b}_l + \sum_{i=l}^{k} \Delta\mathbf{p}_i + \sum_{i=l}^{k} \Delta\mathbf{w}_i \tag{3.5}$$

where $\mathbf{b}_l$ is the bias with an uncertainty given in **P** matrix and $\sum_{i=l}^{k} \Delta\mathbf{w}_i$ is a random walk whose uncertainty can be estimated using the training phase standard deviations.

3.4.2 Addition of Measurement States to Kalman Filter

In the absence of GPS measurements, the inputs to the navigation Kalman filter are formed by taking the difference between the ANN predicted position given in (3.5) and the INS calculated position (Figure 3-3). To realize the Kalman filter, it is required to relate these error inputs to the filter states. Employing (3.5), the measurements are

$$\mathbf{z}_k = \delta\mathbf{p}_k = \mathbf{p}_I - \mathbf{p}_N = \mathbf{p}_I - \mathbf{p}_N - \sum_{i=l}^{k} \Delta\mathbf{p}_i + \mathbf{v}_k + \mathbf{w}_k \tag{3.6}$$

and

$$\mathbf{v}_k = \sum_{i=l}^{k} \Delta\mathbf{w}_i + \mathbf{b}_l \tag{3.7}$$

where I stands for INS, N stands for ANN, $\mathbf{p}_N$ and $\mathbf{v}_k$ are transformed from north, east, height into latitude, longitude, height and $\mathbf{w}_k$ is the noise on ANN predictions stemming from ANN input uncertainty. As seen in (3.6), the position error includes uncertainties on the ANN predictions. The uncertainty given in (3.7) compares to three states modeled in Kalman filter for each position measurement error. Therefore, Kalman filter in ANN prediction phase has a 18-state structure, where 15 of these states are same as in the 15-state GPS/INS filter and 3 states are added to model measurement error given in (3.7). These new states can be modeled as random walk with an initial bias as:

$$\begin{aligned} \dot{x}_{16} &= w_L \quad , \quad \mathbf{P}_{16,16}(0) = \mathbf{P}_{11}(l) \\ \dot{x}_{17} &= w_\lambda \quad , \quad \mathbf{P}_{17,17}(0) = \mathbf{P}_{22}(l) \\ \dot{x}_{18} &= w_h \quad , \quad \mathbf{P}_{18,18}(0) = \mathbf{P}_{33}(l) \end{aligned} \tag{3.8}$$

where standart deviations of input noises are three times of the training standart deviations as stated in Section 3.4.1 and initial uncertainty on each state is the uncertainty on the last GPS/INS position before the system enters the prediction phase. New 18-state **Q** matrix is formed by addition of corresponding input variances for each new state and **P** matrix is expanded to 18-state structure by adding initial uncertainties of these states. The ANN measurements may be related to the filter states by the following measurement matrix

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{0}_{3x12} & \mathbf{I}_{3x3} \end{bmatrix} \quad (3.9)$$

since every measurement is composed of position and the corresponding error state given in (3.8). The uncertainty effect on the measurement that results from the uncertainty $\mathbf{w}_k$ of the ANN inputs, is modeled in measurement noise covariance matrix, $\mathbf{R}_{3x3}$. In order to see the effect of noise on the position errors, two error sources are defined on velocity and attitude. It should be noted that noise on the other inputs has minor effects on ANN solutions. Effects of these error sources are investigated individually.

For a tactical GPS/INS system, the velocity error has a standard deviation less than 0.5 m/s and attitude has a standard deviation less than 3 mrad in all directions. Therefore, velocity error is assigned standard deviation values of 0.25 m/s, 0.5 m/s and 1 m/s while attitude error is adjusted to 1mrad, 1.5 mrad, 3 mrad and 5mrad.

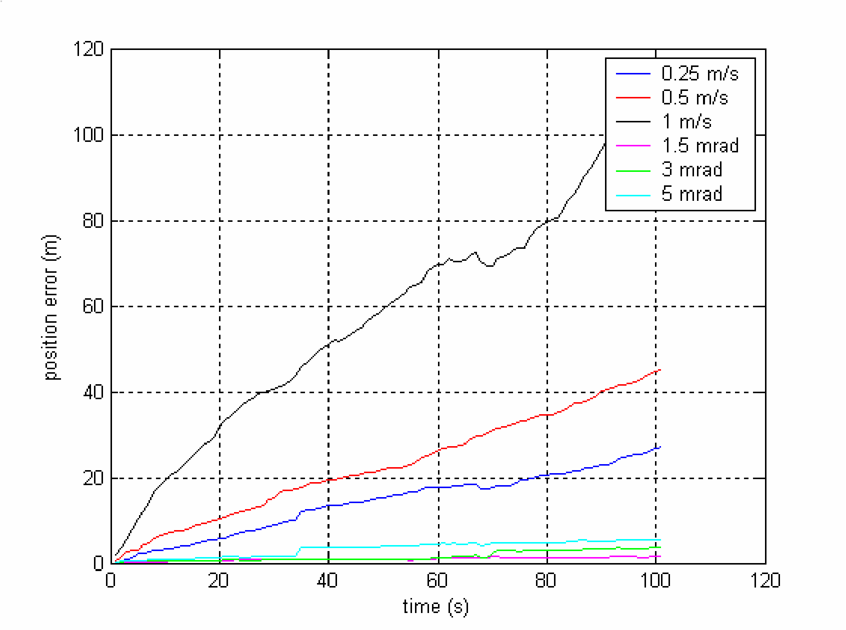

Figure 3-12 Effect of input errors on position error

As seen in Figure 3-12, ANN is less sensitive to noise added to attitude inputs. In fact, the main cause of the ANN uncertainty is the noise on the velocity inputs. The effect of the velocity error on the position error grows proportionally with time as seen in the figure. The relation between ANN input velocity noise and the measurement noise, which is the noise on the ANN position estimates is determined using data given in Figure 3-12 as

$$\mathbf{w}_k \quad with \quad \boldsymbol{\sigma}_{\mathbf{w}_K} \cong \sum_{i=l}^{k} \frac{1}{\sqrt{3}} \boldsymbol{\sigma}_{\mathbf{V}_i} \Delta t \tag{3.10}$$

Thus, measurement noise covariance matrix, $\mathbf{R}_{3x3}$ is determined using velocity uncertainties at the ANN input given in the Kalman state covariance matrix, **P** as

$$\mathbf{R} = \begin{bmatrix} \left(\frac{\Delta t}{\sqrt{3}(R_N + h)} \sum_{i=l}^{k} \sqrt{\mathbf{P}_{44}} \right)^2 & 0 & 0 \\ 0 & \left(\frac{\Delta t}{\sqrt{3}\left(R_E + h\right)\cos L} \sum_{i=l}^{k} \sqrt{\mathbf{P}_{55}} \right)^2 & 0 \\ 0 & 0 & \left(\frac{\Delta t}{\sqrt{3}} \sum_{i=l}^{k} \sqrt{\mathbf{P}_{66}} \right)^2 \end{bmatrix} \tag{3.11}$$

where $\mathbf{P}_{44}, \mathbf{P}_{55}$ and $\mathbf{P}_{66}$ uncertainties on the north, east and down velocity components, respectively.

3.5 ANN Predicted Position Difference Aiding to GPS/INS

Intelligent navigation system shown Figure 3-4 is an alternative architecture intending to aid the navigation system in cases of GPS signal loss. This architecture provides ANN predicted position differences to the Kalman filter for an intelligent support when the GPS is off.

In this architecture, the inputs to the Kalman filter are formed by taking the difference between the position difference estimates of ANN and that of the INS calculated ones. In order to realize the Kalman algorithm, these error inputs should be related to the filter states through an observation matrix.

The measurement residual formed by differencing ANN and INS solutions are

$$\mathbf{z}_k = \delta\Delta\mathbf{p}_k = \Delta\mathbf{p}_{I,k} - \Delta\mathbf{p}_{N,k} = \mathbf{p}_{I,k} - \mathbf{p}_{I,k-1} - (\mathbf{p}_{N,k} - \mathbf{p}_{N,k-1}) + \mathbf{w}_k \qquad (3.12)$$

where I stands for INS, N stands for ANN, and $\mathbf{w}_\mathrm{k}$ is the noise on ANN predictions. (3.12) may be rearranged as

$$\begin{aligned} \mathbf{z}_k = \delta\Delta\mathbf{p}_k &= \mathbf{p}_{I,k} - \mathbf{p}_{N,k} - (\mathbf{p}_{I,k-1} - \mathbf{p}_{N,k-1}) + \mathbf{w}_k \\ &= \delta\mathbf{p}_k - \delta\mathbf{p}_{k-1} + \mathbf{w}_k \end{aligned} \qquad (3.13)$$

and

$$\delta\mathbf{p}_{k-1} = \delta\mathbf{p}_k - \Delta t\delta\mathbf{v}_k^n + \frac{1}{2}\Delta t^2\mathbf{C}_b^n\delta\mathbf{f}_k^b \qquad (3.14)$$

Substituting (3.14) into (3.13) and taking measurement interval $\Delta t = 1$ second, Kalman measurement is expressed as

$$\mathbf{z}_k = \delta\Delta\mathbf{p}_k = \delta\mathbf{v}_k^n - \frac{1}{2}\mathbf{C}_b^n\delta\mathbf{f}_b^k + \mathbf{w}_k \qquad (3.15)$$

Using (3.15), the Kalman measurement matrix **H** takes the following form

$$\mathbf{H} = \begin{bmatrix} \mathbf{0}_{3x3} & \mathbf{I}_{3x3} & \mathbf{0}_{3x3} & -\frac{1}{2}\mathbf{C}_b^n & \mathbf{0}_{3x3} \end{bmatrix} \qquad (3.16)$$

where the 15 state Kalman filter state vector is

$$\begin{bmatrix} \delta L & \delta\lambda & \delta h & \delta v_n & \delta v_e & \delta v_d & \varepsilon_x & \varepsilon_y & \varepsilon_z & \delta f_x & \delta f_y & \delta f_z & \delta\omega_x & \delta\omega_y & \delta\omega_z \end{bmatrix}^T.$$

The noise analysis given in Section 3.4.1 shows that in prediction phase the standard deviations of the noise on position differences in three dimensions are

$\sigma_n = 0.3318$ m, $\sigma_e = 0.3582$ m and $\sigma_h = 0.1406$ m for north, east and height, respectively. We model the uncertainty in measurement noise covariance matrix as 5 times of the found values of $\sigma_{h,n,e} \cong 0.3$ m in order to decrease the dependency of the system on ANN and to stand in the safe side in case of wrong ANN predictions. This yields to an $\mathbf{R}_{3x3}$ which may be expressed as

$$\mathrm{R} = 2.25\mathbf{I}_{3x3} \tag{3.17}$$

3.6 Sensitivity Analysis

3.6.1 Sensitivity to Neural Network Weights

When backpropagation learning is applied for training a neural network, the connection weights may converge to different values depending on the learning algorithm, learning rate, initial weights and input pattern. Thus, in order to select weights generating lower sensitivity and to estimate of the output pertubations due to weight uncertainty, it is important to analyze the sensitivity of the neural network (Choi (1992), Lamy (1996)).

Statistical sensitivity to weight perturbations for weights ***W*** and an input pattern is defined as:

$$\mathbf{S} = \lim_{\sigma \to 0} \frac{\sqrt{\mathrm{var}(\Delta \mathbf{x}_{3,out})}}{\sigma} \tag{3.18}$$

where $\Delta \mathbf{x}_{3,out}$ is the output error, var is the variance and σ is the standart deviation on the nominal values of each weight.

In a multi-output network as it is the case in this study, each output has its own sensitivity. For sufficiently small perturbations on the weights, the error and also the sensitivity of the network can be approximated using first order Taylor series expansion. However, for high amount of perturbations, sensitivity should be determined with numerical analyses.

In order to find the sensitivity, the following equation, which gives the change in neural netwrok output due to the changes in previous layer weights, is employed. The notation can be found in (Saad (1998)).

$$\Delta \mathbf{x}_{out,3} = \left[\mathbf{D}^1\mathbf{G}(\mathbf{v}^1)\Delta\mathbf{W}^1 + \mathbf{D}^2\mathbf{G}(\mathbf{v}^2)\Delta\mathbf{W}^2 + \mathbf{D}^2\mathbf{G}(\mathbf{v}^2)\Delta\mathbf{W}^2 \right]\mathbf{x}_{out,2} \tag{3.19}$$

Using the given relation and also making use of numerical analyses, the sensitivity of the intelligent navigation system is found and plotted as in Figure 3-13. In the figure, lines are the results of the analytical approach and symbols ◊, * and ○ stand for numerical results. The graphics display the standard deviation of the 3 outputs with respect to standard deviation of the weight uncertainty.

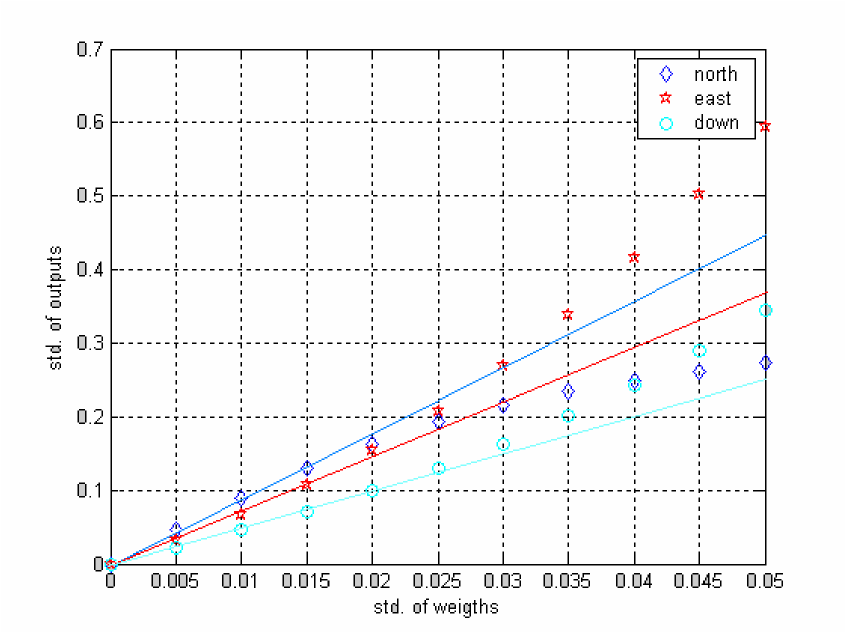

Figure 3-13 Standard deviation of the output for test

Numerical and analytical analyses match for small perturbations on the weights and differ for larger pertubations since the error and also the sensitivity of the network is approximated using first order Taylor series expansion for numerical analyses. Moreover, all MLPs show similiar sensitivity to weight uncertainty since they own same architecture.

After training the ANN with the data from the first 360 seconds of the land test for many times it is seen that the standard deviation of the weights is less than 0.005. Therefore, with a sensitivity less than 10 in all directions (Figure 3-13), the ANN position difference estimations would not deviate significantly. So ANN system used in our intelligent navigation system is not sensitive to weight uncertainties.

Having found that the system is not sensitive to weight changes, we concentrate on the ANN sensitivity to the training set in the next section.

3.6.2 Sensitivity to Learning Set

The neural network performance in the prediction phase is highly dependent on the training pattern. If training pattern does not include enough information on the vehicle dynamics and environment, the neural network predicted positions may be far from being precise and the accuracy of the overall intelligent system may degrade when the system is exposed to conditions different from the ones learned in the training phase. Thus, having insufficient training information about possible conditions that may occur, degrades the navigation solutions.

Sensitivity of the proposed system to the learning phase is investigated employing a flight scenario for an airborne vehicle. The main impetus behind the use of an airborne vehicle instead of a land vehicle for the analysis is its capability to move at very different dynamics. Training the network in some dynamic environment and running it for prediction in a very different dynamic is only possible with a high dynamic vehicle such as a subsonic aircraft.

In our study, we make use of a subsonic aircraft data taking off from an airport and climbing to 12000 meters in first 500 seconds with smooth heading changes. After reaching the 12000 meter altitude, the aircraft goes through an 180˚ head change with a low-g maneuver in 200 seconds and then begins its high-g maneuvers for the rest of flight.

The system runs in the learning phase for 500 seconds of flight (traced in blue) and runs in prediction phase for two consecutive 200 seconds of flight traced in red and green in Figure 3-14. Each phase is termed “training” for the trained portion of the trajectory, “prediction I” for the first sequence of prediction and “prediction II” trajectory for the last sequence of prediction.

As seen from Figure 3-14, the “training trajectory” and the “prediction I trajectory” consist of low-g maneuvers and share low dynamics of flight. On the other hand, “prediction II trajectory” consists of high-g maneuvers different from the “training trajectory”. Thus, “prediction I” is expected to be more accurate than “prediction II” since the training set is composed of low-g maneuver data of “training trajectory”.

Comparison of “INS only” and ANN aided GPS/INS errors in 3D are given in Figure 3-15 and Figure 3-16 for two different phases of the trajectory. These figures show that the ANN aided GPS/INS provides better navigation solutions in low dynamic environment of “prediction I trajectory” than high dynamic environment of “prediction II trajectory” and ANN aided GPS/INS system is sensitive to the training data.

As a result, it is seen that richness of the information in the training set is critical for the accuracy of intelligent navigation system and the solution degrades when the system is exposed to dynamics very different than the ones included in the training phase. Thus, for an accurate prediction phase, the system should experience the dynamical environment, it may undergo during the GPS signal loss, during the training phase.

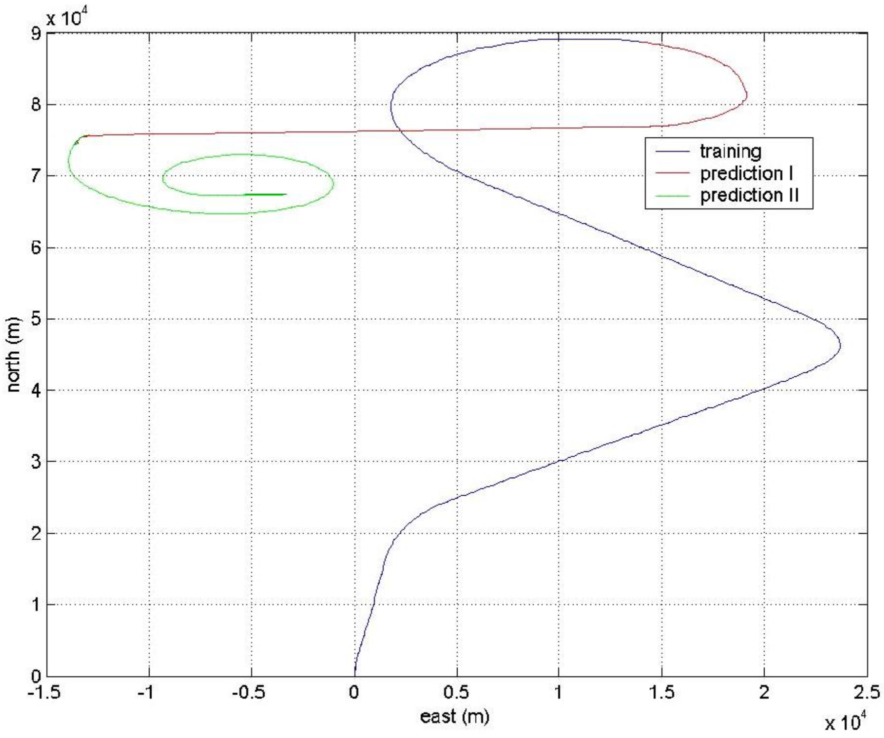

Figure 3-14 Trajectory employed for sensitivity analysis

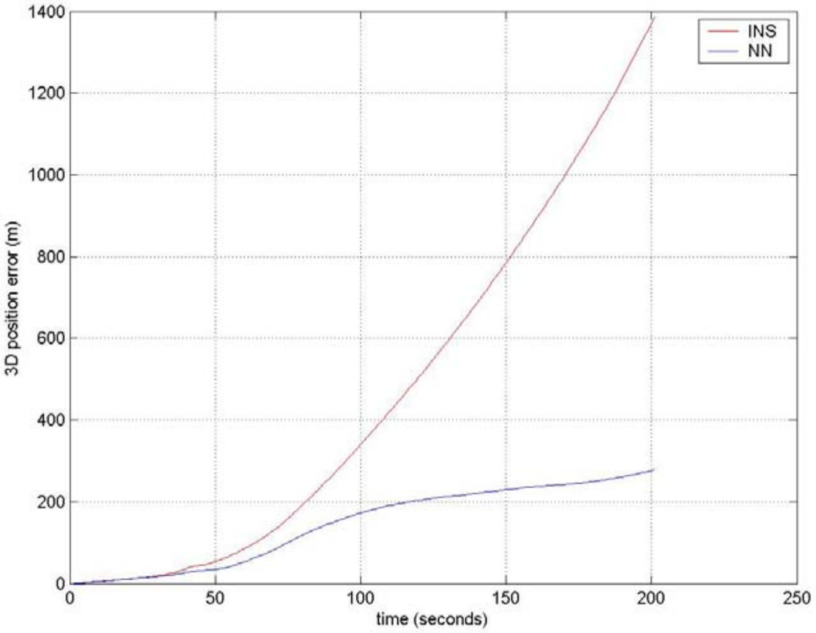

Figure 3-15 Comparison of 3D position errors for 'prediction I' trajectory

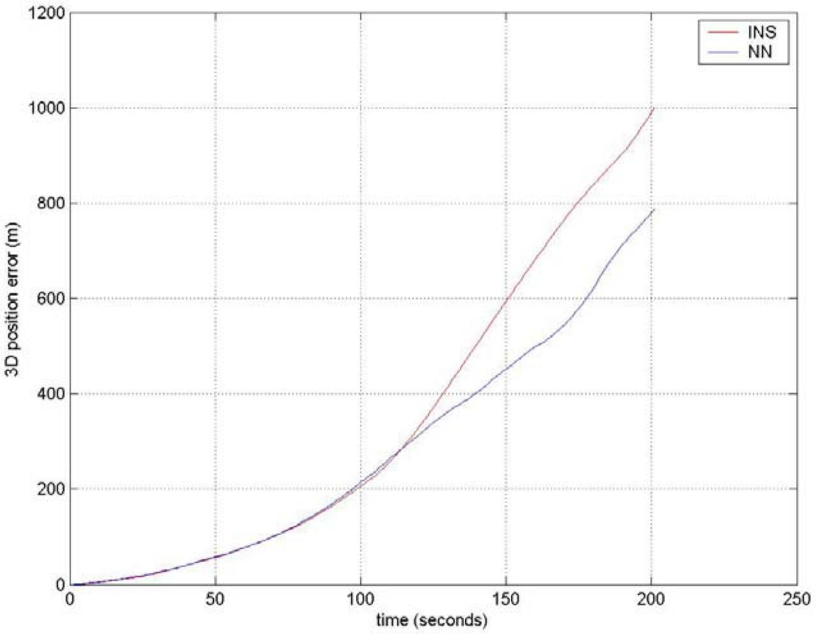

Figure 3-16 Comparison of 3D error for ‘prediction II’ trajectory

3.7 Summary

GPS/INS systems yield high performance navigation solutions provided that the GPS receiver operates. However, when it fails to receive satellite signals GPS/INS would yield degraded navigation solutions. In order to handle the problem in a cost effective manner, a mechanism aiding the integrated GPS/INS system with an artificial neural network is proposed in this chapter.

In the proposed system, the ANN is trained with the position difference of the last second when GPS is on and ANN leaves its training phase and estimates every second the position difference when GPS signal is absent. Two different mechanisms, namely “position aiding” and “position difference aiding” are proposed in order to integrate position difference predicting ANN to the GPS/INS system. Both mechanisms use the ANN provided predicted position differences, to provide information to the integration filter and share the same ANN architecture.

This chapter also investigates the neural network structure composed of preprocessing, hidden (3 MLPs for each direction) and postprocessing performance analyses comparing different training algorithms and topologies concluded, 12x6x3x1 (logsig-tansig-linear) MLPs with Levenberg-Marquardt training algorithm are most suitable to be used due to lower time of convergence.

Towards this decision, a detailed noise analysis of the ANN is conducted in order to integrate both aiding mechanisms to the Kalman filter. A 18-state Kalman filter is realized for “position aiding”, while the GPS/INS filter is used for “position difference aiding”. Measurement noise levels and characteristics are generated and analyzed.

Having constructed the ANN aided GPS/INS system, sensitivity analysis is conducted on the neural network and change of the output pertubations due to the weight uncertainties are investigated. As a result, it is found that the system is not sensitive to weight changes.

Another sensitivity analysis is run in order to investigate how the neural network performance in the prediction phase is dependent on the training pattern. For the analysis, flight scenario for an airborne vehicle is utilized and it is seen that richness of the information in the learning set is critical for the accuracy of intelligent navigation system. Thus, the system should experience all possible dynamics during the training phase for an accurate ANN-aided GPS/INS system operation.

In the next chapter, real time test results showing the performance of the proposed navigation systems are presented.

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 Introduction

This chapter demonstrates the performance of two intelligent navigation systems, namely ANN predicted position aided navigation and ANN predicted position difference aided navigation, with two land tests: a) cluttered field where the route has severe attitude changes and populated with buildings and trees, b) open field where the route is smooth. Both tests are conducted using the van introduced in Section 2.6.1 as the test vehicle. In this chapter, GPS/INS computed navigation solution is employed as reference and the intelligent navigation system and inertial system is compared to this reference in order to find the performances.

The experimental setup employed in order to carry out land tests was described in Section 2.6.1. The cluttered and open field trajectory, test vehicle and test hardware used during the tests were also introduced in the same section. The reader is referred to that section for recalling the test equipment namely the ALV and the test path.

This chapter presents the navigation results of a van traveling in the cluttered field and open field test routes, using intelligent aiding in order to correct the inertial errors occurring during the GPS signal loss. The following section also conduct a comparative analysis based on the test results.

More specifically, in sections 4.2 and 4.3, GPS/INS computed navigation solution is compared to the proposed intelligent navigation system in terms of performance evaluation. The proposed intelligent navigation system adopts the two different architectures presented in Chapter 3: namely the ANN predicted position aided: called the ANN-I/INS system and the ANN predicted position difference aided: called the ANN-II/INS system in the following sections.

The results of cluttered field test are comparatively analyzed in Section 4.2 while the results of open field tests are presented in Section 4.3.

4.2 Cluttered Field Campus Test

This test is conducted in Middle East Technical University campus for 500 seconds where raw IMU data, GPS data and GPS/INS data are collected throughout the test navigation. INS only and ANN aided navigation data is obtained by postprocessing the raw IMU data. ANN is trained using the test data collected within the first 230 seconds of the route; afterwards it begins to estimate the position difference and position for the remaining 270 seconds of the test navigation, based on its prior learning.

Figure 4-1, Figure 4-2 and Figure 4-3 presents the north position, east position and height solutions of GPS/INS, INS only, ANN-I/INS and ANN-II/INS structures.

The main portion of the test path is aligned with the north direction. Thus, the north channel of the inertial navigation system is observable during the test navigation. As a result, INS only north position error (800 m) given in Figure 4-1 is less than the INS only east position error (3000 m) given in Figure 4-2. As the test vehicle navigates in north-south direction, it excites the errors in this direction and allows the Kalman filter to observe and correct these errors. This is not the case for east-west channel of the navigation system.

Vertical channel error is about 120 m after 270 seconds of INS only navigation. This is due to the fact that the z-axis accelerometer bias converges to the real value as a result of high observability in this channel. This channel is highly observable since it senses 1-g gravitational force during the test, which is not the case in other accelerometers sensing horizontal force components. As a result, INS only navigation gives the best results in vertical channel.

Observability in GPS/INS system is directly related to the learning set of ANN system. All the error sources should be excited in a GPS/INS system to provide high observability over the system states. The system should experience all possible dynamics during “GPS on” navigation to achieve observability. This may be translated into ANN learning as the availability of a rich learning set covering all possible dynamics that the vehicle may face during navigation that would improve the ANN performance during prediction phase. From this discussion, one can conclude that ANN aided navigation results should provide better position performance in the channels where INS only position performances are better.

If we compare the position performances in north, east and vertical channels for both ANN aided navigation architectures, we see that the vertical position error is around 20 m, while north position error is 100 m and east position error is 150 m. This is what we expect from the above discussion.

Moreover, predicted position aided and predicted position difference aided intelligent navigation systems have similar performances such that both intelligent systems drifts only 200 meters during 270 seconds of GPS signal loss (Figure 4-4). As they use the same neural network architecture and they share the same source of information, it is natural to have same errors for these two approaches. It should be noted that the only real difference between these two architectures is the aiding mechanism and this does not change the uncertainty characteristics of the information fed into the Kalman filter. Thus, both mechanisms reveal same performance characteristics.

As seen in Figure 4-1, Figure 4-2 and Figure 4-3, the ANN aided structure position solutions outperform INS only position solutions in all directions. While INS only navigation drifts nearly 3000 meters, intelligent aiding bind the errors at nearly 200 meters (Figure 4-4) during the 270 seconds of GPS signal loss. A similar trend may be observed in velocity solutions given in Figure 4-5. ANN aided system binds the velocity error under 7 m/s which exceeds the value 20 m/s in the case of INS only navigation. Thus, the intelligent navigation system suppresses successfully the navigation errors using the generated navigation data within the first 230 seconds of the test route. Our proposed system has been found to outperform the non-aided GPS/INS system in cluttered field land test.

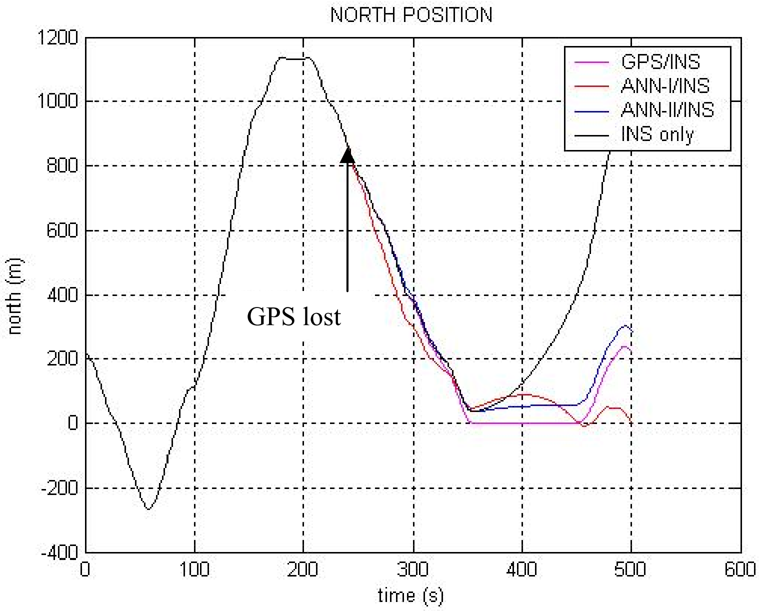

Figure 4-1 The north position solutions of GPS/INS, ANN-I/INS, ANN-II/INS and INS only structures.

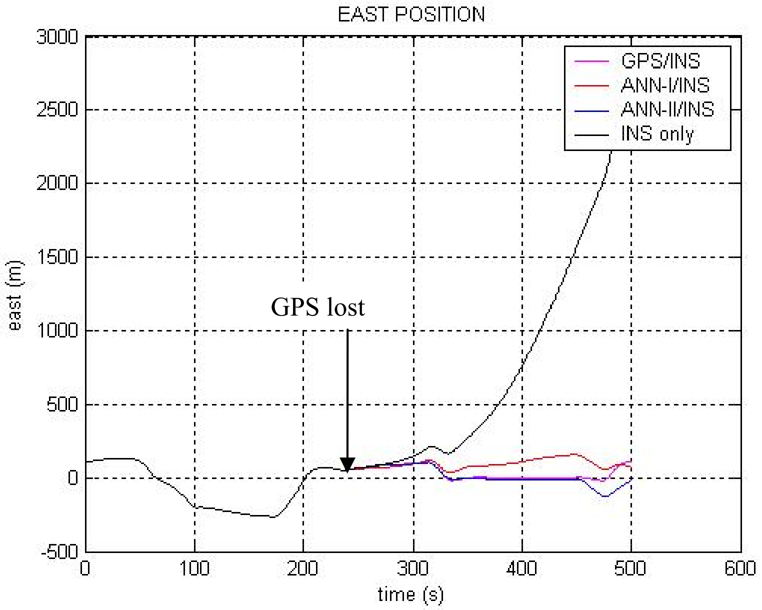

Figure 4-2 The east position solution of different structures. Both of the ANN aided structures suppress INS only structure errors as proposed.

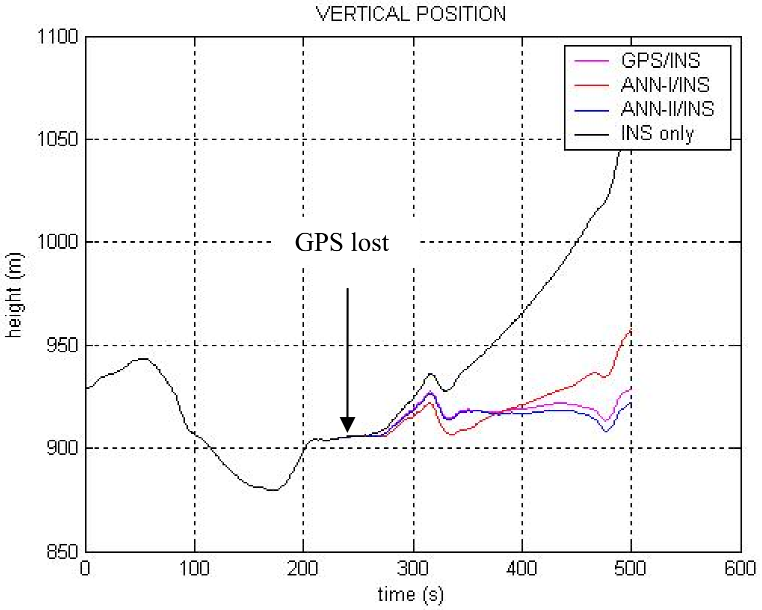

Figure 4-3 INS only solution drifts very fast while ANN aided solutions outperform the INS only solution.

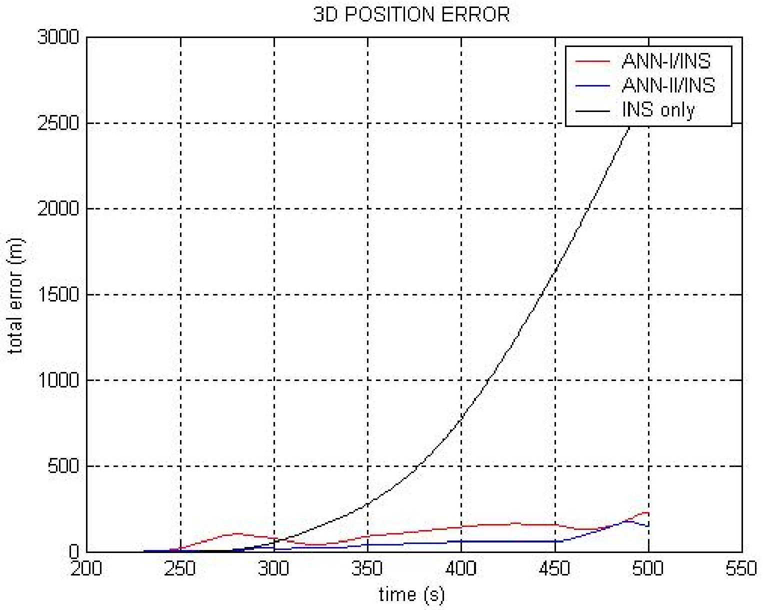

Figure 4-4 Total position error after 270 seconds of GPS loss

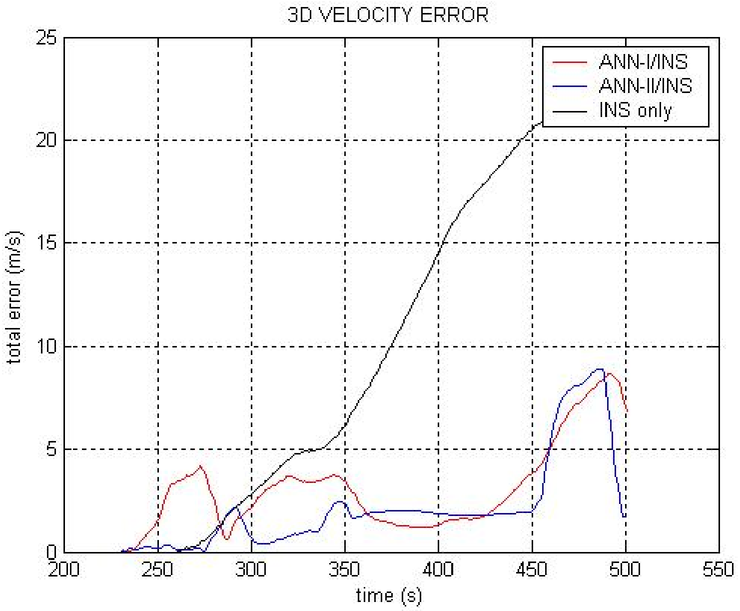

Figure 4-5 Intelligent structures prevent the velocity to drift

4.3 Open Field Test

The test is conducted in the open field for 250 seconds. Raw IMU data, GPS data and GPS/INS data are stored in the laptop computer throughout the test. INS only and ANN aided navigation data is obtained by postprocessing the raw IMU data as in the cluttered field campus test. ANN is trained using the test data collected within the first 200 seconds of the route; afterwards it begins to estimate the position difference and position for the remaining 50 seconds of the test run.

Intelligent navigation system and INS only position solutions does not deviate from the true track as in cluttered field test since the test run is short. However, intelligent navigation systems still provide more accurate solutions than the INS only system.

Figure 4-6 and Figure 4-7 present north and east position solutions provided by all navigation systems under test. As seen in Figure 4-6, INS only system shows a higher drift trend compared to the performances of the intelligent systems. However, the error does not grow because of the short test time. Yet, the intelligent systems have still less error and achieve to follow the reference solution provided by GPS/INS system. In east direction, all of the systems generate position errors comparable to each other (Figure 4-7).

Examining both Figure 4-6 and Figure 4-7, it is seen that the errors in both directions are similar in magnitude. This is one of the main differences between open field and cluttered field tests. As mentioned earlier, the test vehicle navigates in north-south direction and excites the errors in this direction. Thus, it allows the Kalman filter to observe and correct these errors in cluttered field test. However, open field test route is aligned with the north-east direction, which makes both directions equally observable. Thus, unlike the cluttered field test open field test generates comparable errors of 7 m in north and 5 m in east direction.

Vertical channel error of intelligent systems is around 4 m, while it exceeds the 12 m value in INS only system (Figure 4-8). This is due to the learning capability of the intelligent system during "GPS on" period of the test. In vertical channel, it is also seen that the intelligent navigation systems follow the GPS/INS solution while the only-INS system deviates into the wrong direction.

As a result, ANN aiding achieves to decrease the position error of the INS only system from 32 m to 11 m (Figure 4-9). In cluttered field test, this decrease was from 3000 m to 200 m. This difference between two tests is mainly due to the short test time of the open field run. As seen from the error growth trends and the velocity errors (Figure 4-10) of intelligent navigation and the INS only system, the error would probably explode if the run was longer.

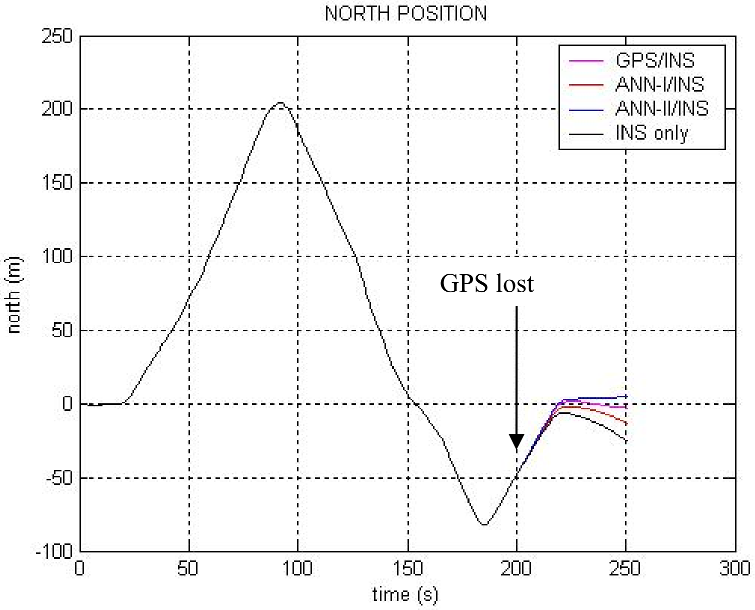

Figure 4-6 The north position solutions of GPS/INS, ANN-I/INS, ANN-II/INS and INS only structures for the short navigation test in which GPS signal is lost for last 1 minute.

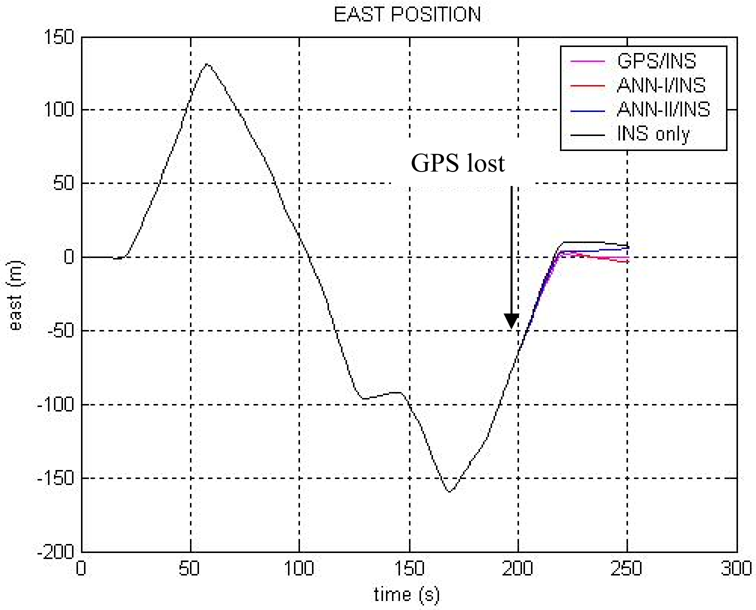

Figure 4-7 The east position solutions for the test.

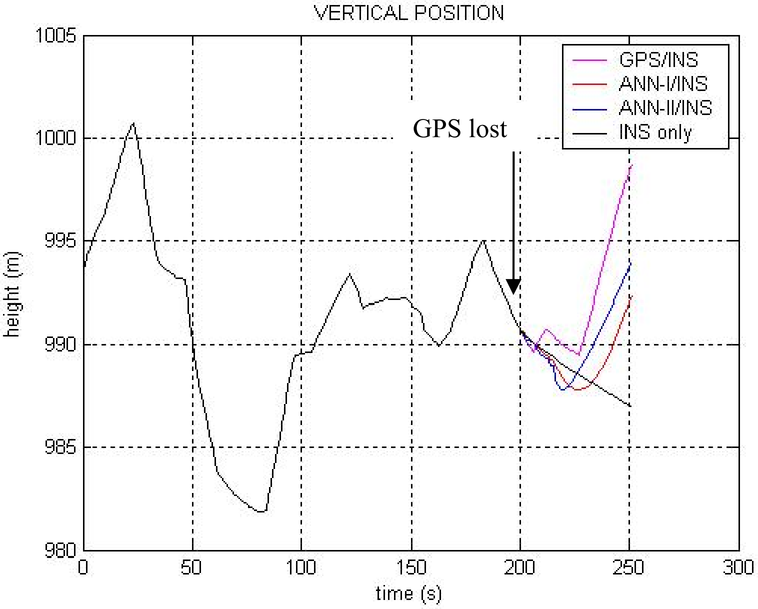

Figure 4-8 ANN aided navigation achieves to follow GPS/INS solutions while INS drifts off.

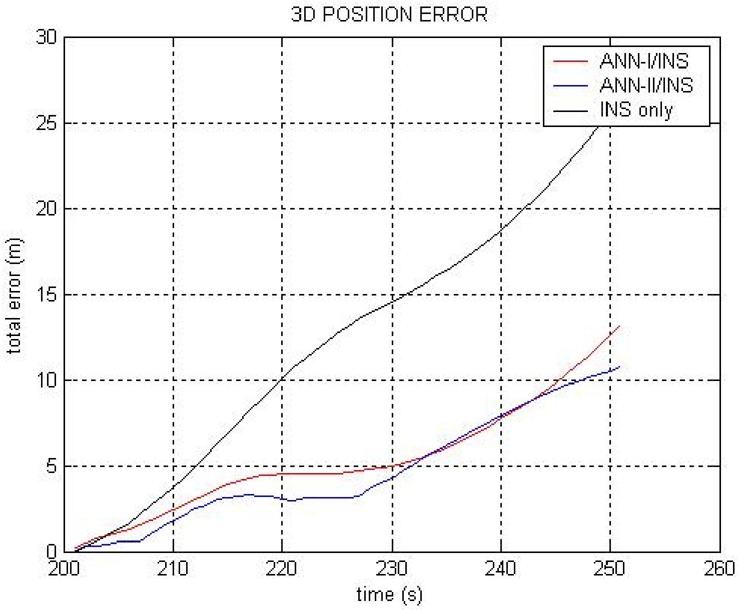

Figure 4-9 ANN structures produce only the 1/3 of INS position error at the end of the test.

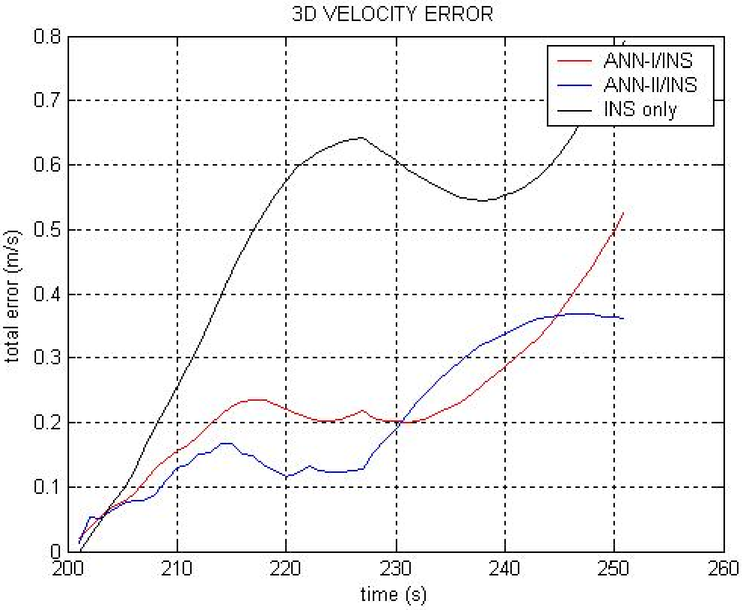

Figure 4-10 ANN achieves to diminish the INS velocity error throughout the navigation.

4.4 Conclusion

This chapter demonstrated the performance of the ANN predicted position aided navigation and the ANN predicted position difference aided navigation with a cluttered field and an open field land test.

The proposed structure is found to perform extremely well in ALV tests. The main reason is that GPS/INS+ANN system learns navigation behavior patterns from past experience which aid substantially navigation accuracy.

The proposed system shows that with enough training, it can decrease a 3 km error to 200 meters as in the cluttered field test (Figure 4-4). It achieves to provide accurate position solutions beyond a GPS signal loss of 270 seconds. Moreover, the proposed system can provide accuracy of nearly 10 meters during 1 minute of GPS loss as seen in Figure 4-9 of open field test.

Especially “10 meters accuracy during 1 minute of GPS loss” performance achieved at open field test is very critical for vehicles that enter enclosed areas for short periods of time.

As a result, the intelligent navigation system meets the demand for high performance integrated navigation systems with high reliability and accuracy even in the absence of GPS signal. It outperforms INS only navigation in all directions, in both tests.

CHAPTER 5

VIBRATIONAL EFFECTS IN AUTONOMOUS LAND VEHICLE DYNAMICS

5.1 Introduction

Autonomous Land Vehicles have received considerable attention after their introduction into the military and commercial applications such as monitoring, agriculture, exploration and survillance. In the previous sections, we have developed an intelligent, reliable and accurate navigation system which is critical to provide the autonomy in ALVs. Although an accurate and reliable navigation system is one of the requirements for autonomy, another equally crucial requirement is to have a robust planner.

Autonomy in the autonomous land vehicles (ALVs) depend heavily on a controller which is robust under unexpectedly changing environments, therefore has to be intelligent enough to know by experience what to expect from the dynamic behaviors in state space. ALV controllers still stand as a challenging research topic due to their high operational and dynamical complexity. One of the main problems arising in the robustness of ALV controllers, is the dynamic interaction of the vehicle with irregular roads which causes vibratory motion on the chassis and may prevent the vehicle and its cargo function properly (Mays (2000)). As stated in Chapter 2, vehicle vibration has a harmful effect on the performance of navigation systems since inertial sensors have a limited bandwidth and reject the motion vibration at high frequencies. This vibration contributed navigation system errors are eliminated using mechanical filters known as vibration absorbers and “coning compensation” and “sculling compensation” algorithms in the navigation subsystem. Thus, the remaining vibration has no adverse effect on the computations.

However, the vibration still affects the other electronic and mechanical components during the operation of the ALV. This vibration may damage ALV cargo (e.g. tracking camera) and may prevent it to function properly during the mission. An intelligent control that suppresses the vibration is then critical for the health of cargo and mission. However, such a control mechanism requires the precise investigation of the states related to the oscillatory motion, in order to decrease the unwanted effects originating from these oscillations.

Due to complex problems within ALVs, their control still stands as a challenging research topic where the main problem lies in mathematical modeling and global analyses of the dynamics of ALVs in different environmental conditions. The global analysis of the state space is of great importance, especially in this control modality towards the stability of vehicle/road nonlinear system. One can adopt the widely used technique of generating domain of attractions using cell to cell mapping (Hsu (1980a, b)).

This chapter mainly deals with

- modelling of the vehicle/road nonlinear system,
- the analysis of the global stability not only with cell to cell mapping but together with Lyapunov exponents,
- and the analysis of vehicle response to different road conditions.

The first step of our approach is the problem of modelling the system. Detailed studies conducted in the literature are given in Section 5.2 and the model used in this thesis is introduced in Section 5.3.

For the global analysis of system behavior the well-known method of cell to cell mapping, proposed by Hsu is used (Section 5.4). This method is based on partitioning an N-dimensional state space into a large number of cells and determining the stability characteristics of these cells. Domain of attraction is then formed as sets of cells in the cellular state space after a predefined number of integration performed by taking the center of every new cell as the initial point of the next iteration (Hsu (1980a, b)). More than just processing cells by this well known methodology of cell to cell mapping, we hereby expand the methodology by also examining the system behavior in control space through the Lyapunov exponents of the mapping in order to classify cells as regular and chaotic. This classification will enable control system designer to determine the regular space of the system through its chaotic boundaries and will provide the maximum operational state space to the controller.

This will lead in Section 5.5 to the frequency response of the vehicle in terms of changes in its state space to different road conditions using cell to cell mapping. The results of the analyses conducted in this chapter will lead us to the novel approach on how to generate a control less sensitive to vibration for the ALV we use, which will be the topic of Chapter 6.

5.2 The Existing Works: Vehicle/Road Interaction

Analysis on vehicle/road/cargo and/or occupant systems have been done intensively since early 70’s, exploring linear or nonlinear models assuming random or nonrandom road irregularities (İleri (1990)).

Patil et al. (1978), investigated the multi-degree of freedom model of tractor/occupant over sinusoidal road irregularity. Numerical responses for two types of suspensions are found and analysis are carried out in order to obtain the suspension parameters minimizing occupant's vibration.

Dodds and Robson (1973) described road irregularities as a stationary random process and argued that the response of a vehicle on a given road is determined by the theory of random vibrations. Kamash and Robson (1978) stated that the precise road description is not a necessity but that the statistical description of a road can be admissible.

Czerny and Popp (1987), formulate the vehicle as a linear time-invariant multi-axle system over irregular road modelled as colored noise. LeBlanc et al. (1987), has studied the effects of vehicle suspension nonlinearities on the irregular guideways experimentally. They showed that the nonlinearities mainly depend on the tires, suspension spring and damping coefficients.

Misoi and Carson (1989) indroduced the results showing that a vehicle's suspension acts as a low-pass filter and by driving faster than the velocity at which the sinusoidal road irregularity matches the suspension resonant frequency, one can reduce the vibrations.

While all the above analyses use classical techniques of nonlinear dynamics such as linearization, approximate methods to determine periodic, quasi-periodic constraints and assumption of well-behaved state regions, Ileri has introduced a novel numerical analysis of the nonlinear vehicle models (İleri (1990)) and has shown that the state space of a land vehicle over irregular road is composed of islands of regular and periodic motions among a multitude of chaotic trajectories.

Recently, Mays and Faybishenko (2000), analysed the irregularity as a complex dynamic system resulting spatial chaos. Hence, they have shown that chaotic motion of the vehicle is typical where periodic motion of accumulation states is slightly disturbed by any uncertainty inherent in the states of the system dynamics.

This thesis work also uses modern analysis methods in the investigation of vehicle response on irregular roads to contribute to the existing literature by providing the operational state space to the controller for different road conditions (Chapter 5) and decontaminating it as much as possible from any uncertainty (Chapter 6).

5.3 Description of the ALV Model

Our GPS/INS guided autonomous land vehicle (ALV) equipped with tracking camera was introduced in Section 2.6 as a vehicle designed to function on irregular terrains as well as smooth ones (Figure 2-14). We model our vehicle as composed of a chassis body and lumped tires represented by nonlinear springs in parallel with velocity dependent dampers (Figure 5-1) where the ALV and terrain model parameters are given in Table 5.1. These model parameters are acquired from (İleri (1990)) where a detailed modelling work has been conducted using experimental data. Equations of motion are derived based on the following assumptions:

-the road profile is approximated by a sinusoidal shape,

-a phase shift is introduced between the sinusoidal shape of the road portions in order to realize irregularity,

-the vehicle is considered in the longitudinal plane only,

-forces and couplings due to wheel rotations are neglected,

-translational and angular displacements are assumed to be small enough for tire and spring motions so that nonlinear springs remain within their elastic range,

-the vehicle is kept in cruise control with constant velocity.

ALVs generally cruise with a predetermined constant velocity. Road irregularities are generally washboards in unpaved roads with a sinusoidal shape and amplitude of nearly 50 mm (Mays (2000)), and such a road profile introduces a phase shift between the forces on front and back tires. The forces on tires stemming from low amplitude road irregularities cause small translational and angular displacements enough for the nonlinear springs to remain within their elastic range. Hence, the assumptions given above for our ALV/road interaction are general and enough for our model to be as close as possible to the physical system.

The chassis displacement follows a nonlinear behavior that we modeled as

$$\begin{aligned} & m\ddot{y}+\sum_{i=1}^{3}k_{i1}\left(y-a\theta-\xi_1\right)^i+\sum_{i=1}^{3}k_{i2}\left(y+b\theta-\xi_2\right)^i \\ & +\sum_{i=1}^{3}c_{i1}\left(\dot{y}-a\dot{\theta}-\dot{\xi}_1\right)^i+\sum_{i=1}^{3}c_{i1}\left(\dot{y}+b\dot{\theta}-\dot{\xi}_2\right)^i=0 \end{aligned} \tag{5.1}$$

based on the mentioned assumptions.

The pitch vibration on the other hand obeys the following equation

$$\begin{aligned} & m\rho^2\ddot{\theta}-\sum_{i=1}^{3}ak_{i1}\left(y-a\theta-\xi_1\right)^i+\sum_{i=1}^{3}bk_{i2}\left(y+b\theta-\xi_2\right)^i \\ & -\sum_{i=1}^{3}ac_{i1}\left(\dot{y}-a\dot{\theta}-\dot{\xi}_1\right)^i+\sum_{i=1}^{3}bc_{i1}\left(\dot{y}+b\dot{\theta}-\dot{\xi}_2\right)^i=0 \end{aligned} \tag{5.2}$$

Here, θ is the body pitch angle, $\dot{\theta}$ the angular rate, y the vertical displacement and $\dot{y}$ the vertical velocity. As seen from (5.1) and (5.2), the given spring-mass-damper system consists of third order nonlinear damping and stiffness coefficients.

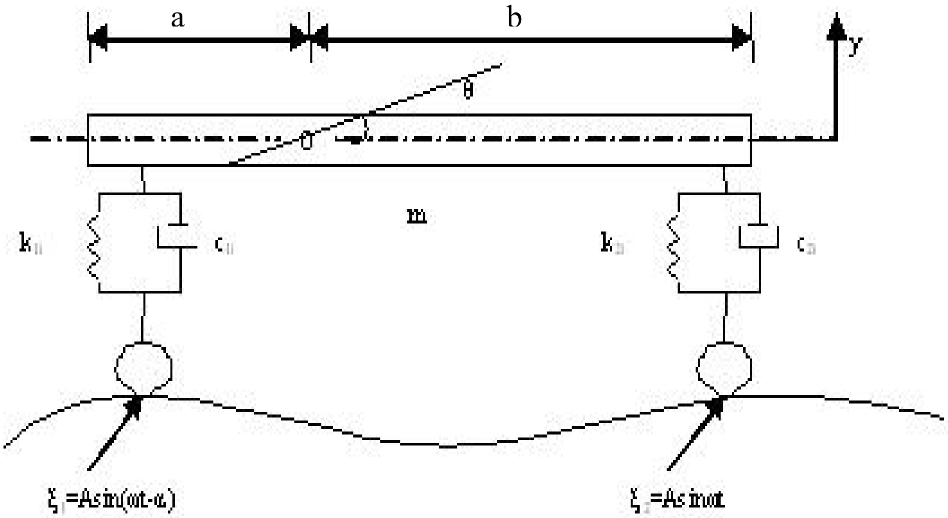

Figure 5-1. Schematic diagram of the model

Table 5.1 System Parameter Values

k_{i1}	Spring constant of back tire	496.38 kgf/cm^{i}
k_{i2}	Spring constant of front tire	553.28 kgf/cm^{i}
c_{i1}	Damping constant of back tire	4.434 kgf/cm^{i}/sec^{i}
c_{i2}	Damping constant of front tire	2.374 kgf/cm^{i}/sec^{i}
m	Vehicle mass	271.889 kgf/cm/sec^{2}
a	Distance of back tire from C.G.	84.7 cm
b	Distance of front tire from C.G.	118.5 cm

ρ	Radius of gyration	102.24 cm
L	Length between tires	203.2 cm
A	Amplitude of impressed vibration	5 cm
α	Phase angle between front and back tires	160°
ω	Forcing frequency	1 rad/sec

Under certain conditions, such a system with two degrees of freedom may exhibit chaotic fluctuations (Moon, (1987)). In Section 5.4, we will attempt to capture this behavior by the global stability analysis conducted on our ALV system.

5.4 Global stability analysis of ALV using cell to cell mapping

In order to determine the system characteristics and capture a global picture of stability, cell to cell mapping technique is applied to generate the state space of our ALV dynamics given in Table 5.1. over irregular roads. y_1 and y_2 of the states vector $[y_1, y_2, y_3, y_4] = [y, \dot{y}, \theta, \dot{\theta}]$ are forced to take different initial conditions such that global analysis is conducted on a projection subspace where y_3 and y_4 are taken constant and the 2D space is directly affected by the vertical translational motion of the vehicle. The space is divided into 2D cells of dimension *hxh* such that a point y_i belongs to cell Z_i if

$$(Z_i - \tfrac{1}{2})h \le y_i < (Z_i + \tfrac{1}{2})h \tag{5.3}$$

where interval size for both dimension is taken as h=0.096, in our case.

As stated before, vibration absorbers with resonance frequency of 35 Hz are introduced to the physical environment and the effective vibrational amplitudes is guaranteed to be less than 35 Hz. Thus, a sampling frequency greater than 70 Hz is enough to capture the dynamical behavior from Nyquist theorem. So the system of differential equations is discretized using the Euler integration rule with a time step of $1\text{x}10^{-2}$ seconds (100 Hz). At each step, starting from cell $\boldsymbol{Z}(\text{n})$, the centre point $\boldsymbol{y}(\text{n})$ of $\boldsymbol{Z}(\text{n})$ is first taken in the discrete process as the initial state and the point $\boldsymbol{y}(\text{n}+1)$ is generated as output of the Euler integration such that $\boldsymbol{Z}(\text{n}+1)$, the image cell of $\boldsymbol{Z}(\text{n})$, is obtained as the cell within which the calculated point $\boldsymbol{y}(\text{n}+1)$ lies. The next iteration is performed by taking the centre point of $\boldsymbol{Z}(\text{n}+1)$ as the initial point and this discrete cell to cell mapping is continued for 1000 steps.

At every step, we also find the Jacobian matrix in order to calculate cell Lyapunov exponents what we incorporate into cell to cell computations. Lyapunov exponents are used to quantify the expansion and contraction occurring in a dynamical system. They are used to determine the stability of any type of steady state behaviour. What distinguishes a strange attractor from the other types of attractor is the existence of at least one positive Lyapunov exponent. Incorporating Lyapunov exponents into cell to cell mapping, one can analyze the global phase plane and determine the chaotic (if there exists) and stable (regular) regions of the dynamical systems which is in our case the ALV dynamics on irregular roads.

Vibration sensitive and insensitive stability regions of our ALV system are obtained in Figure 5-2 for a sinusoidal terrain roughness with amplitude of 5 cm. over a state space partitioned into 101x101 cells and using vehicle parameters given in Table 5.1. The interval lengths h_1 and h_2 defining the sizes of each side of a cell are selected as 0.096 and 0.096 for y_1 and y_2, respectively. The analysis in state space is conducted over a ±5 cm displacement by 101x101 cells with each having 0.096 cm cell size since irregularities on unpaved roads impress vibration of peak amplitude up to 5 cm as mentioned in (Mays (2000)).

The regular part of the domain of attraction is depicted with dark cells while the other cells are chaotic according to the Lyapunov exponents. For every regular (dark) cell given in the figure, the maximum Lyapunov exponent is found to be negative. Other cells have positive maximum Lyapunov exponents. The regular region is the domain where chassis controller design can be achieved in order to totally suppress the road induced vibrations.

Consequently, the analysis using cell to cell mapping equipped with Lyapunov exponents shows us that the chassis of the vehicle travelling on an irregular road can produce a chaotic response depending on the initial condition.

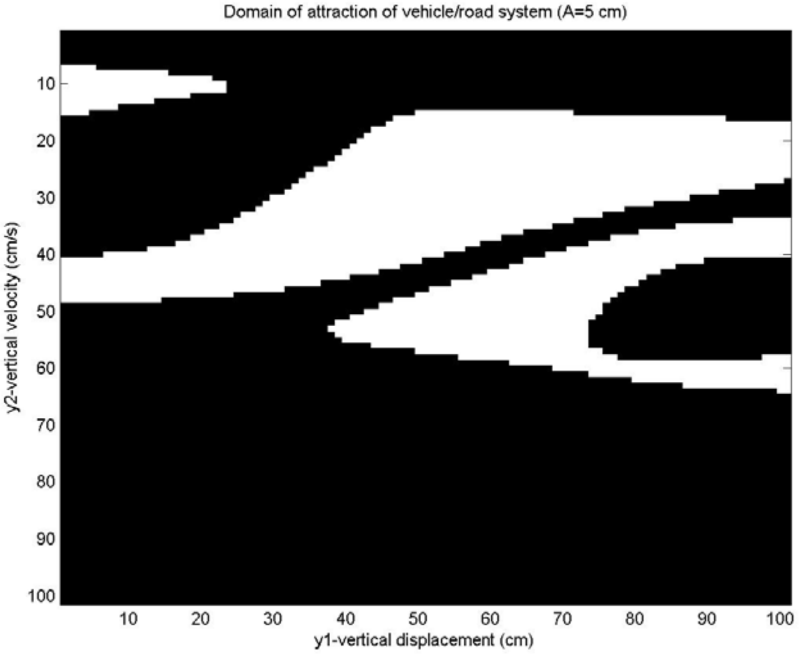

Figure 5-2 y_1-y_2 phase plane for road vibration of 5 cm

The dimension of the attractor shown in Figure 5-2 is indicative of chaos and should have a fractal value. The dimension of the system can be found using the Lyapunov dimension definition.

The Lyapunov exponents of the chaotic attractor seen in Figure 5-2 are found to be

$$\begin{aligned} \lambda_1 &= 80.1 \\ \lambda_2 &= 73.9 \\ \lambda_3 &= -67.2 \\ \lambda_4 &= -76.5 \end{aligned} \tag{5.4}$$

Then the Lyapunov dimension of the attractor is found as follows:

$$d_{lyp} = 2 + \frac{80.1 + 73.9}{|-67.2|} = 4.28 \tag{5.5}$$

which is fractal.

As a result, capturing chaotic fluctuations and determining the regular regions for chassis controller design is achieved through the phase plane for road irregularities as in Figure 5-2.

Another important issue arising at this step of analyses is the effect of different road conditions on the stability conditions of our ALV system. We analyze this issue in Section 5.5.

5.5 Change in stability due to road condition changes

As given before, irregularities on the unpaved roads impress vibrations which can be approximated by a sinusoidal shape. Different road conditions induce sinusoidal vibrations of different frequencies and peak amplitudes. As a result, one should investigate the changing road conditions and its effect on the vehicle dynamics in order to capture the overall picture of the stability/chaos and control of the chassis.

The effect of road-induced frequency changes can be directly investigated by frequency analysis of the vehicle/road model given in (5.1) and (5.2). We use the time-integration method for analyzing the frequency response of the system from the initial state [0, 0, 0, 0] by applying sinusoidal inputs with different frequencies until the system reaches its steady state for each frequency. The analysis is conducted below 5 Hz since this is the peak frequency where the vibrational amplitudes have severe effects on the system (Patil (1988)). The resulting frequency response is given in Figure 5-3-Figure 5-6. As seen in the figures, the vehicle suspension acts as a low-pass filter and road induced vibrations over 1 Hz are damped. Consequently, vibrations below 1 Hz are investigated to find out the effect of changing frequency on the overall picture of the stability/chaos of the chassis. The phase plane at 0.5 Hz is given in Figure 5-7 while the one for 0.16 Hz is given in Figure 5-2. As seen in the figures, the chaotic area decreases with the increase in frequency.

Consequently, it is seen in the analysis that as the vibration frequency increases the chaotic field would vanish. Chaotic vibrations with strong high frequency components cannot exist in the ALV system due to the fact that the system behaves like a low pass filter (Figure 5-3-Figure 5-6). If road induced vibrations are over 1 Hz, they are filtered by the system and cannot lead to chaotic oscillations. However, vibrations below 1 Hz pass through the system and excite the chassis in a manner that strong chaotic vibrations occur.

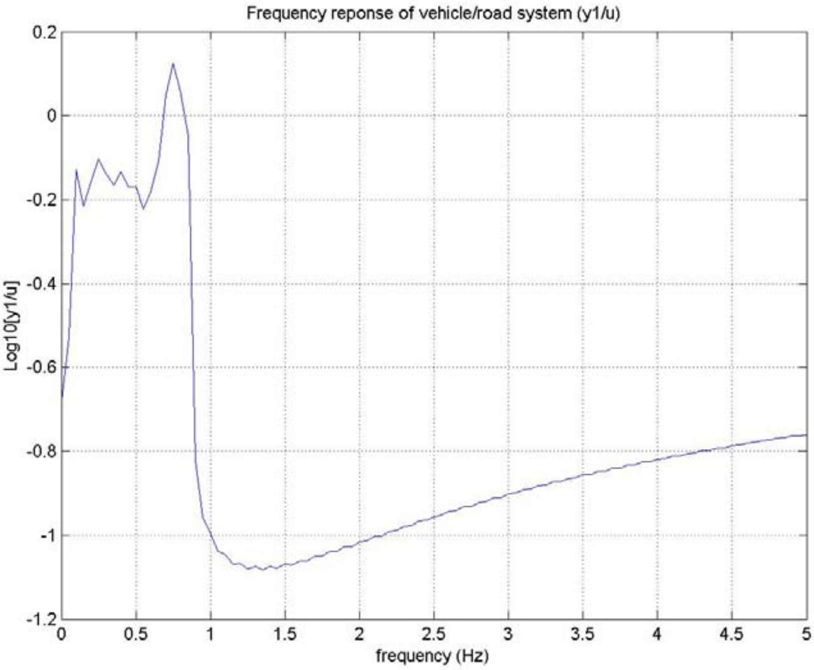

Figure 5-3 Frequency response of vertical displacement

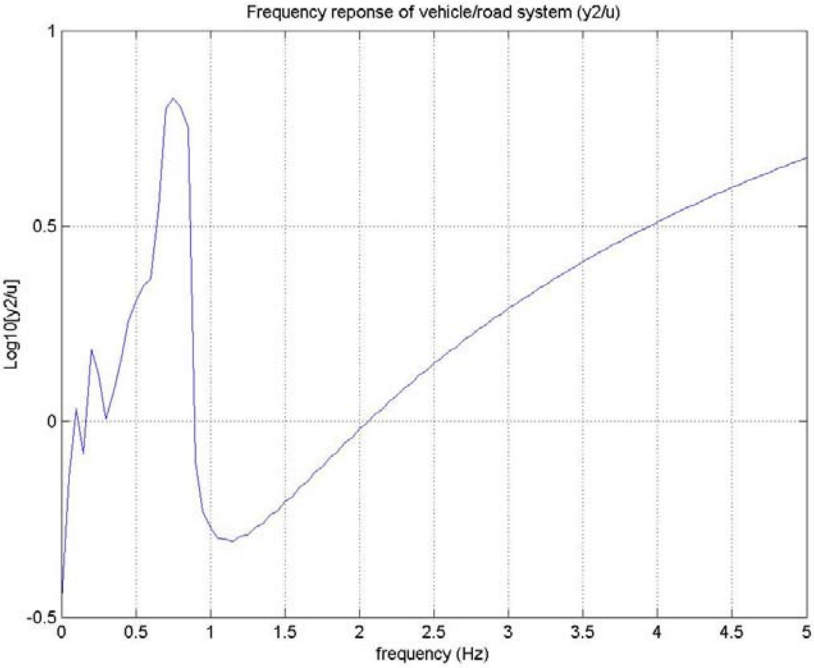

Figure 5-4 Frequency response of vertical velocity

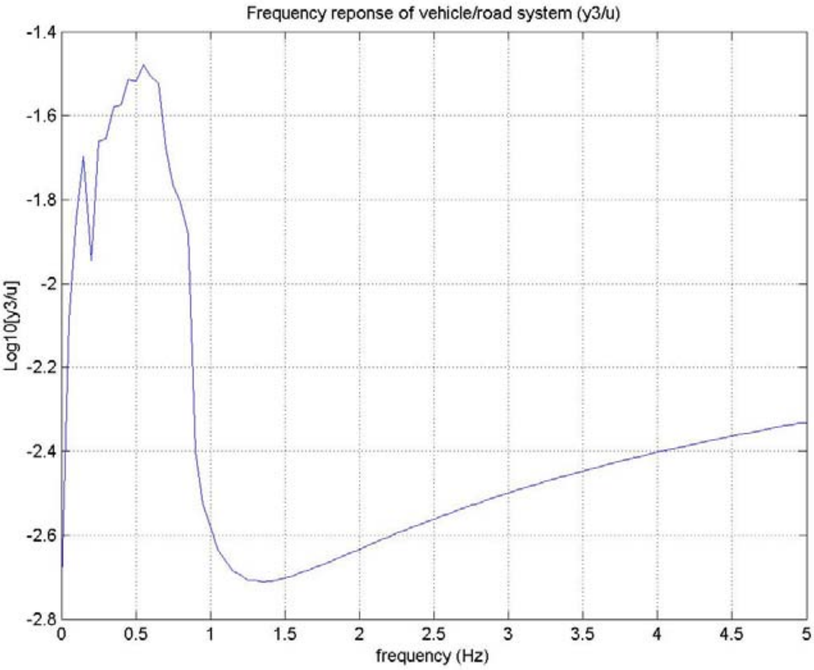

Figure 5-5 Frequency response of pitch angle

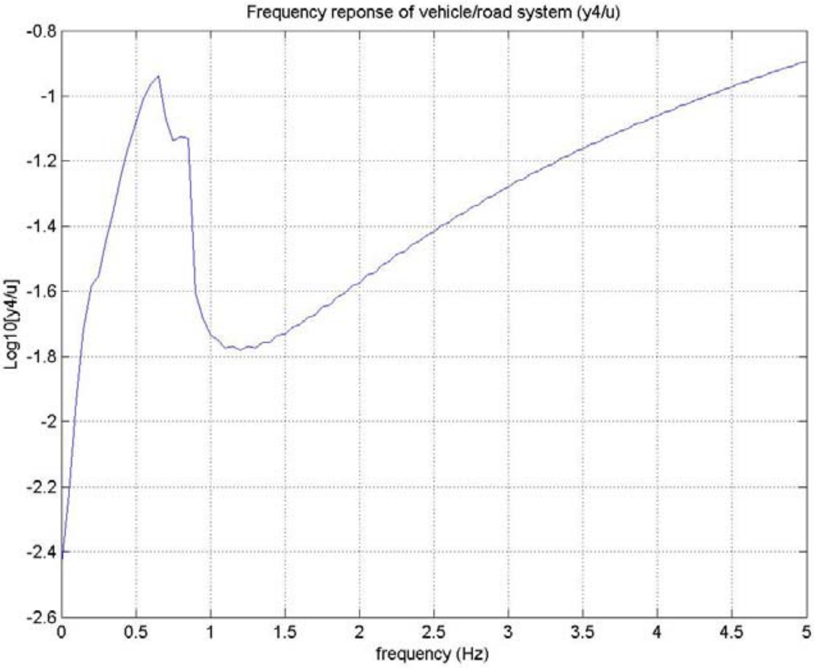

Figure 5-6 Frequency response of pitch rate

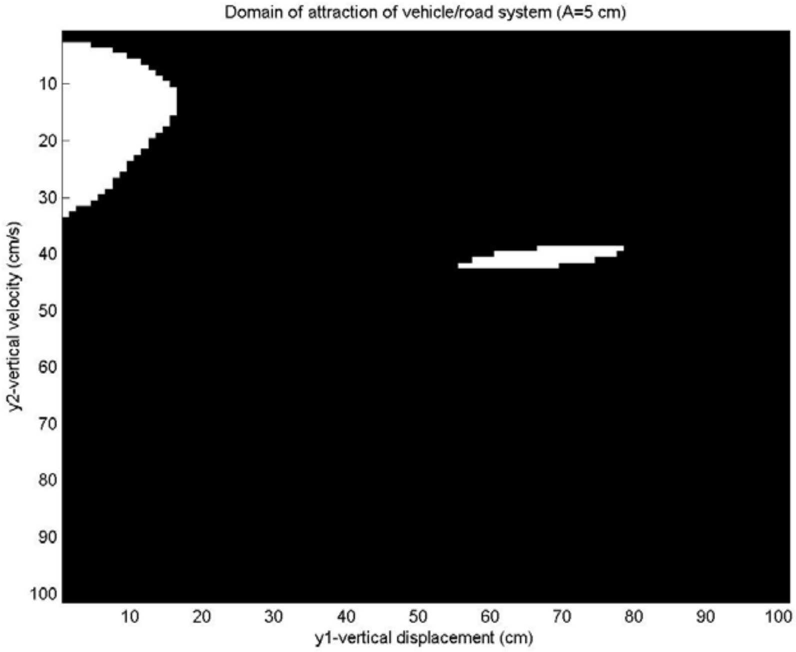

Figure 5-7 y_1-y_2 phase plane for road vibration of 5 cm

Different road conditions may also induce different peak amplitudes. As mentioned in (Mays (2000)), irregularities on unpaved roads impress vibration of peak amplitude up to 5 cm. In order to investigate the changes of road conditions affecting the domain of attraction for the current system, the impressed vibration amplitude, A is varied with different values such as 5, 3 and 1 cm with the forcing frequency of 1 rad/sec given in Table 5.1. The domain of attractions for changing vibration amplitudes is given in Figure 5-2, Figure 5-8 and Figure 5-9. As seen from the figures, as the vibration amplitude increases, the chaotic region in the attraction domain decays in its area coverage. It is found that chaotic region decreases 15% and 24%, respectively when vibration amplitude changes from 1 cm to 3 cm and to 5 cm.

Change in road conditions and thus vibration has a serious effect on regular regions in state space, in particular, the controllable regular region increases with the increasing vibration amplitude. Therefore closely repeated irregularities on

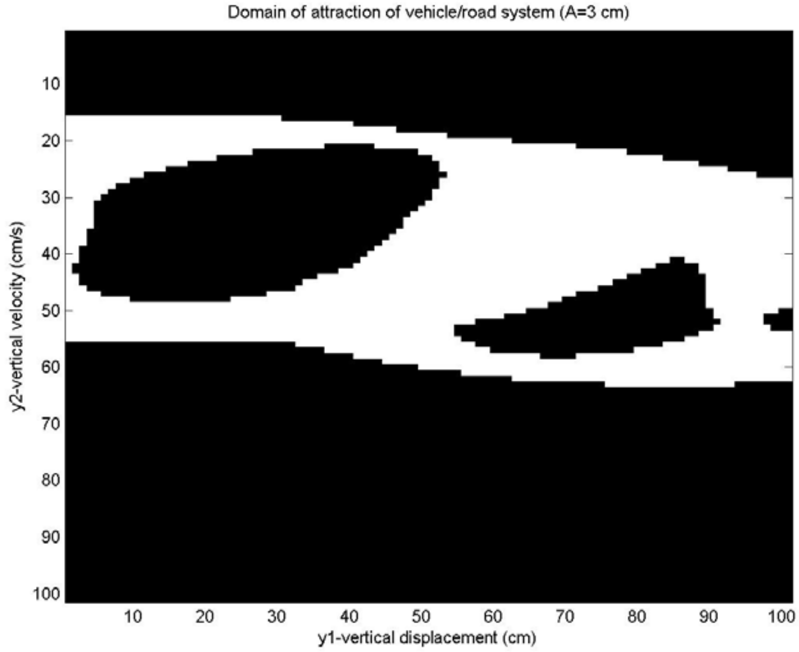

Figure 5-8 y_1-y_2 phase plane for road vibration of 3 cm

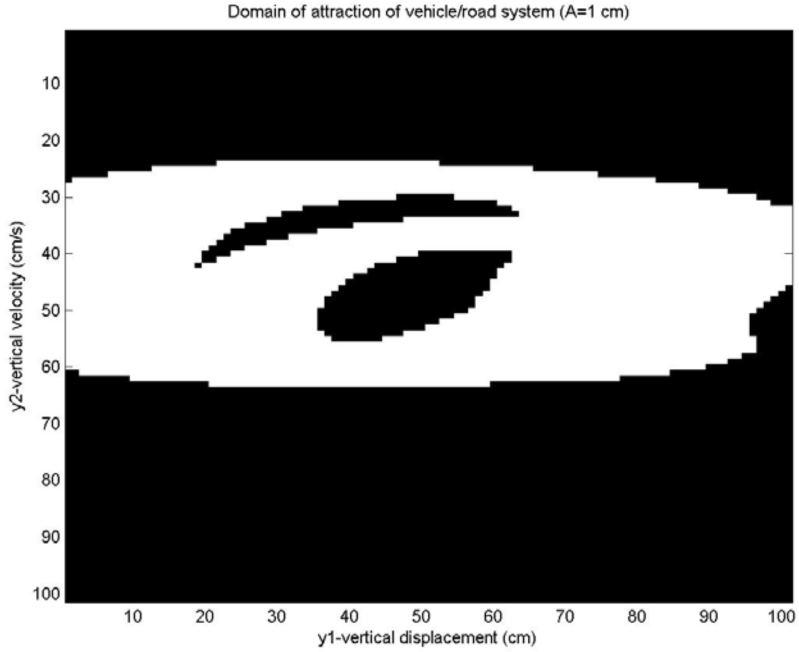

Figure 5-9 y_1-y_2 phase plane for road vibration of 1 cm roads (vibration of sharp small magnitudes-1 Hz), which is the frequent nature of country roads, will decrease the controllable regular regions in state space.

After determining the regular region of the system, the challenge remains in determining the boundary region, since regions are sensitive to changes on their boundaries. So if we want to increase robustness of the vehicle to uncertainty and vibrations created by irregularity of roads, we have to not only determine the boundary of the regular regions but also smooth its roughness enlarging the regular region thus the regular controllable region.

Analyses are focused on the 5 cm vibration case from now on without loss of generality. Chapter 6 deals basically with the identification of the regular/chaotic states that make up the boundary region of the domain. This chapter employs the fractal dimensions and the rough set theory to construct a more detailed picture of the regular region and its boundary by modelling the inherent uncertainty.

CHAPTER 6

DETECTING ONSET OF CHAOS AND INTELLIGENT OPERATIONAL REGION ENLARGEMENT

6.1 Introduction

In analyzing nonlinear dynamical systems, one is often interested in the local behavior of the system around its equilibrium states. The local system behavior information around equilibrium point and the determination of the global regular domain are critical in order to conclude on the control strategies and to assign the relevant parameters for the system controller.

In some cases such as vehicle/road interaction, the regularity of the boundary region of a domain of operation in the state space of a dynamical system is frequently fractal and the precise identification of the regular region imbedded in that domain is nearly impossible. This identification uncertainty renders the use of classical approaches very inefficient.

In order to overcome the problem of fractal boundaries a probabilistic approach instead of a precise prediction is suggested in (Hsu (1980a, b)) where domains of attractions are generated using cell to cell mapping in examining the regularity of nonlinear systems. However, such domains of attraction based on cell to cell mapping technique are far from being precise on the boundary of the region: having possible chaotic elements, the boundary region of the domain of attraction introduces an uncertainty into the regularity region. Comparing Figure 5-2 and

Figure 6-1, one can easily see the precision problem of cell to cell mapping technique. Some of the regions, which are classified as regular in Figure 5-2 are in fact chaotic in Figure 6-1 and vice versa. As a result, the boundary region contains an uncertainty in the identification.

The extraction of regular regions in state space using cell to cell mapping technique renders the precision of the region mainly dependent on the cell size. Figure 5-2 is generated for cell size 0.096, while Figure 6-1 is generated for the same system for cell size 0.024. To have a true picture of stability for a dynamical system, the cells should be infinitely small and the state space should be divided into infinitely many cells. This means every point in state space will then be treated as a cell and cell to cell mapping technique becomes a point to point mapping process, which takes an infinite, infeasible, amount of time.

At this point, the only approach is seen as adjusting the cell size according to design purposes and taking this discretized but uncertain stability region as the global stability information (Thompson (1990)). However, until now, no study has been conducted to model the uncertainty in this discrete possibly stable region and to form a control strategy suitable for that model.

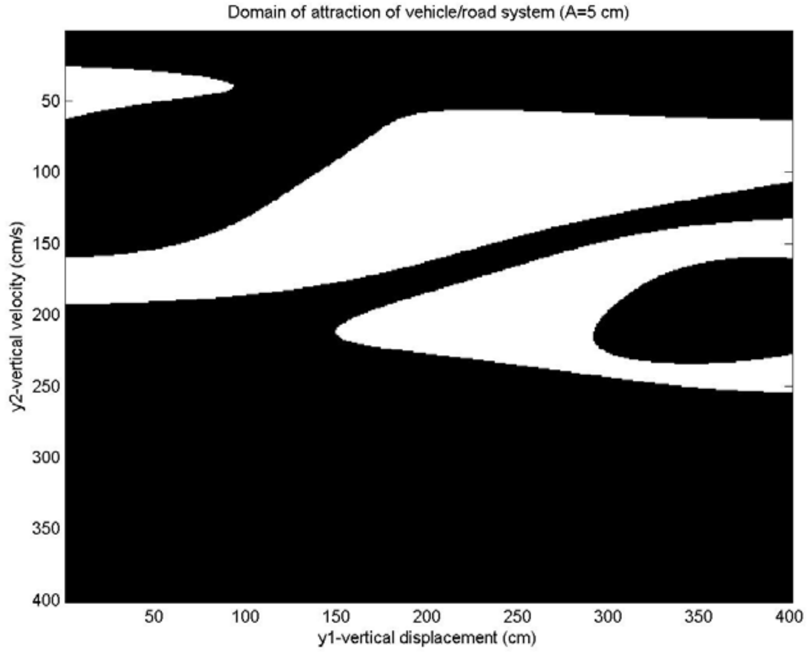

Figure 6-1 y_1-y_2 phase plane for road vibration of 5 cm (400x400 cells)

We propose a novel approach to provide a solution in this virgin field by modelling and smoothing the imprecision stemming from cell to cell mapping using an intelligent technique that integrates several phases:

- modelling the dynamic system and determining the states which cause uncertainty in the regular region,
- analyzing global stability not only with cell to cell mapping but together with Lyapunov exponents,
- determining the boundary region of the attraction domain and modelling the uncertainty in stability of this domain using rough set theory and the fractal dimension concept,
- smoothing uncertainty in this boundary region via a reinforcement learning algorithm.

In the previous chapter, we have completed the first two phases of our approach and reached an uncertain global stability picture of the ALV system on irregular roads. The last two steps of the approach directly attack the problem of uncertainty stemming from cell to cell mapping in an intelligent manner.

In our new approach (Kaygisiz and Erkmen (2001, 2002)), the fractal dimension of each cell in the regular region is analyzed and used as an attribute in a Rough Set model in order to represent metrically the uncertainty, inherent in the regular region, as a roughness texture especially of its boundary region (step 3). This boundary region is the main source of uncertainty of the regular domain generated using cell to cell mapping. Rough set theory, we use in our approach to model the roughness of the uncertain region in the domain of regular attractors/repellers, is a concept proposed by Zdzislaw Pawlak for modelling uncertainty and vagueness that we overviewed in Appendix G. We extend this theory by assigning a fractal dimension as an attribute of roughness to rough sets. The novelty of our work is this extension of rough sets with fractal dimension for their use in nonlinear system control. Smoothing of such a roughness in the regular region of the control space is achieved by reinforcement learning based on the chaos history of the system dynamics of the ALV on irregular roads so that its control becomes more efficient (step 4).

Section 6.2 deals basically with the third step of our approach, which is to identify the regular/chaotic states that make up the boundary region of the domain. This section focuses on the fractal dimensions and the rough set theory to construct a more detailed picture of the regular region by modelling the inherent uncertainty. The modelling technique is applied to our system, which is the ALV on irregular roads.

In Section 6.3, a reinforcement learning algorithm is employed which smoothes the roughness in the boundary region. This smoothing yields a recovery of cells by their decontamination from uncertainty that expands the lower approximation of the rough set. This expansion is generated by including the recovered cells of the boundary region into set of regular cells. Recovery is achieved through reinforcement learning according to the stability history of the cells on a cell trajectory of the dynamical system (ALV on irregular roads in our example) generated by cell to cell mapping.

6.2 Modelling Uncertainty in Stability: Boundary Region Roughness

In uncertainty modelling, we exploit rough set theory for categorizing state space regions with different stability features according to their degrees of uncertainty. Rough set as proposed by Zdzislaw Pawlak is a mathematical approach for modelling vagueness in uncertainty and was overwieved in Appendix G. Rough set theory associates a degree of uncertainty to any object of the universe so that any object without uncertainty (roughness) becomes indiscernible within a class.

A set containing indiscernible objects forms the elementary set, any union of which determines a crisp set. All other set includes uncertainty and forms the rough set. Each rough set has 1) a boundary region where vagueness resides, with set elements that cannot be classified either as members of that set or of the complementary set; 2) a lower approximation consisting of the elements that are surely the members of the set and 3) an upper approximation consisting of the possible members of the set.

We found that rough set theory would readily apply to the problem of modelling chaos uncertainty among the regularity regions of a system where cells have to be taken as objects of our rough set. The lower approximation and the boundary region of the set consist respectively of surely regular cells and possibly regular cells, while the other cells are chaotic and not members of the regularity region. Regular cells are the ones where all points in them are regular, while chaotic cells contain only chaotic points. Possibly regular cells, which form the boundary region of the regular domain, are classified either as regular or chaotic at the end of the cell-to-cell mapping but these are cells that contain both regular and chaotic points coexisting within the same cell.

To differentiate the cells into lower, upper and boundary regions, we extract and use homogeneity or nonhomogeneity information contained in the cells generated by the cell to cell mapping. The fractal dimension of each cell is this measure-based information that can be extracted and is suitable for the needed rough set classification. Therefore, we choose to assign the fractal dimension as an attribute of the rough set as a novel expansion of the theory. We use the definition of capacity (Hausdorf dimension) introduced in Appendix F, to compute the fractal dimension. Calculation of capacity dimension requires division of region of interest into a grid of cubes of size ε, which is done during the cell to cell mapping process. Thus, the capacity dimension naturally applies to the computation of the irregularity of a cell as follows:

$$d_c = \lim_{\varepsilon \to 0} \frac{\log N(\varepsilon)}{\log(1/\varepsilon)} \tag{6.1}$$

To compute the fractal dimension d_c of a cell using capacity definition, the cell is divided into square subcells of size ε as in Figure 6-2 .

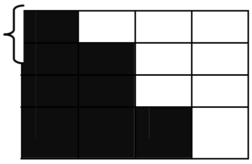

Figure 6-2 A cell divided up into 16 subcells

N (ε) is the number of subcells found to be nonchaotic by cell to cell mapping technique. The process is repeated by decreasing the value of subcell size, and *N (ε)* is recorded for each case. When log *N (ε)* is plotted versus log *(1/ε)* for every cell, the slope of the linear plot is the fractal dimension as ε goes to zero in the limit. An example plot for a two legged robot system investigated in (Kaygisiz and Erkmen, (2004)) is given in Figure 6-3. The phase plane of a two legged robot is composed of islands of regular and periodic motions among a multitude of chaotic, unpredictable trajectories Figure 6-4. Hence, chaotic motion of the robot is typical where periodic motion of accumulation states is slightly disturbed by any uncertainty or noise inherent in the states of the system dynamics. Consequently, fractal dimension of every cell is determined dividing each into subcells to differentiate chaotic and periodic motions of the walking robot. Figure 6-3 shows the log *N (ε)* versus log *(1/ε)* plot generated for two of such cells.

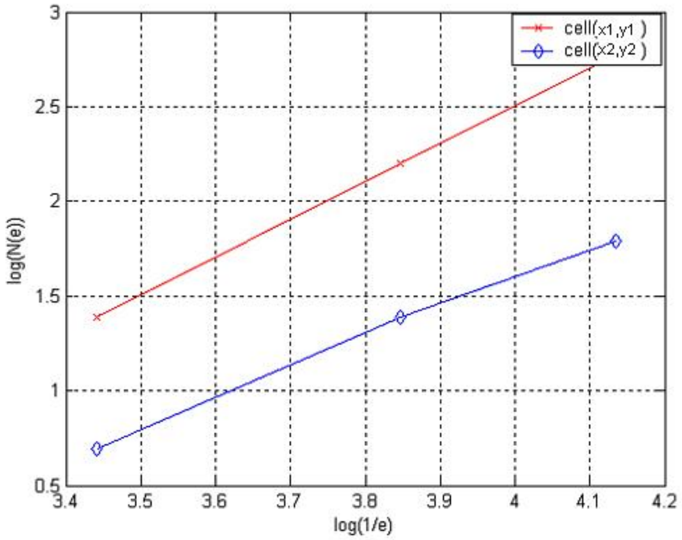

Figure 6-3 Log-log plot to estimate the dimension of a cell

At this point, one can realize that another attribute can be assigned to cells for their classification into rough sets, which is a fuzzy measure of possibility for the regularity of cells in a set. We form this attribute as the ratio of number of regular subcells to total number of subcells within a cell where regular subcells are the ones that are found to be regular using the cell to cell mapping technique and Lyapunov exponents. Thus,

$$R_n = \frac{\#\ \ of\ \ \ regular\ \ \ subcells}{\#\ \ of\ \ \ subcells} \tag{6.2}$$

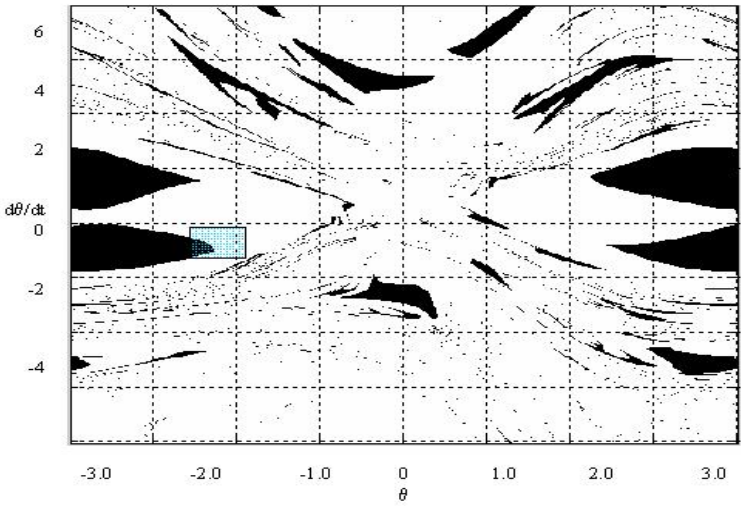

Figure 6-4 Domain of attraction for two legged walking robot

When the subcell size ε tends to zero, this ratio R_n becomes equal to the probability of a point belonging to nonchaotic, regular region. We call this attribute the *regularity number (R_n)*. It follows naturally from our definition that a regular cell has an R_n of one and the R_n of a chaotic cell is zero. The regularity number complements the fractal dimension by providing a coarse overall measure of uncertainty while fractal dimension gives a measure of the roughness characteristics of uncertainty, of its texture. At the limit, d_c and R_n become equivalent by tending towards equivalent information: if the fractal dimension d_c of a cell is an integer number (smooth texture), then the coarse overall measure of the cell R_n does not change with the change of number of subcells in a cell. In our previous studies (Kaygisiz and Erkmen (2001, 2002)), it is observed that R_n is either 0 or 1 when the fractal dimension is an integer number. Thus, in our approach, a rough set contains regular cells of R_n=1 and integer fractal dimension, while uncertain cells are those identified as possibly regular, having a fractional value as fractal dimension and an R_n number $\neq$0 and $\neq$1. If a cell is chaotic it does not contain any regular subcells and consequently its fractal dimension is undefined and R_n=0. However in the general case of fractal d_c, the overall measure

R_n shows a different variability and they begin to complement each other for uncertainty measurements. This concept is given in detail in Section 6.2.1.

6.2.1 Correlation of Regularity Number and Fractal Dimension

Examining the rough set approach, it is seen that the lower approximation for the regular cells consists of all the cells that have integer dimension. That is an important clue bringing us to a point that we can say all the integer dimensional cells or set of subcells are stable. The relation between the number of regular cells and the cell size can be stated as follows:

$$\frac{\log N(\varepsilon_1)-\log N(\varepsilon_2)}{\log\varepsilon_2-\log\varepsilon_1}=\frac{\log N(\varepsilon)-\log N(\varepsilon_1)}{\log\varepsilon_1-\log\varepsilon} \tag{6.3}$$

where ε's are the subcell sizes and $N(\varepsilon)$ is the number of the subcells that are regular in a cell. Rearranging the above equation gives

$$N(\varepsilon)=N(\varepsilon_1)\left(\frac{\varepsilon_1}{\varepsilon}\right)^{\log_{\frac{\varepsilon_2}{\varepsilon_1}}\frac{N(\varepsilon_1)}{N(\varepsilon_2)}} \tag{6.4}$$

Substituting the capacity dimension d_c given in (6.1) into the above equation, we obtain the following relation between the cell dimension and regularity number after rearranging

$$N(\varepsilon)=N(\varepsilon_1)\left(\frac{\varepsilon_1}{\varepsilon}\right)^{d_c} \tag{6.5}$$

into an expression using regularity number (R_n):

$$R(\varepsilon)\frac{c}{\varepsilon^n}=R(\varepsilon_1)\frac{c}{\varepsilon_1^n}\left(\frac{\varepsilon_1}{\varepsilon}\right)^{d_c} \tag{6.6}$$

where c is the size of a region of interest, n is the number of states of the nonlinear system, ε's are the cell sizes. Change in the regularity number with the change of cell size can be defined as:

$$\frac{R(\varepsilon)}{R(\varepsilon_1)} = \left(\frac{\varepsilon}{\varepsilon_1} \right)^{n-d_c} \tag{6.7}$$

From the above relation, it is seen that if the fractal dimension of a cell (or region) is integer then the regularity number does not change when the subcell (or cell) size is changed, hence there exist no uncertainty in regularity. Thus it can be stated that if the dimension is integer in a cell, finer and finer resolutions in the state trajectory will not change the regularity region in this cell and this region can be included in the lower approximation of the rough set. Using this fact, the lower approximation of the rough set can be expanded. The expansion of the lower approximation, which is achieved by smoothing the boundary region using reinforcement learning, is based on the relation given in (6.7). The method for smoothing and its results are given in the following sections.

(6.7) thus gives the correlation between the regularity number and fractal dimension, which are the attributes of the rough set defined for state space classification. Calculating the fractal dimension of the regular region given in Figure 5-2, using the capacity dimension definition it is found that

$$d_c = \frac{\log(118856)\text{-}\log(7555)}{\log(0.096) - \log(0.024)} = 1.9878 \tag{6.8}$$

Employing (6.7), it is found that regularity in the region increases by 1.7% decreasing the cell size four times as in the case of Figure 6-1. This is theoretical maximum expansion of the regular region given in Figure 5-2.

6.2.2 Uncertainty Modelling of ALV/Road Dynamical System

For the ALV system under investigation, the control state space regular region contains 10201 cells generated by cell to cell mapping. The state space has been transformed into rough sets using fractal dimension and the regularity number R_n. In this rough set representation of the vehicle on the irregular terrain of our campus, 7364 cells with R_n=1 and d_c=2 are found to define the fully regular cells for the vehicle (map of Figure 5-2 with cells of Figure 6-5 removed). 2411 cells with R_n=0 and d_c undefined, form the set of fully chaotic cells where vehicle chassis vibration control is not feasible. 426 partially or possibly regular cells with $0<R_n<1$ and $d_c \neq 2$ but fractional valued (Figure 6-5) are also found. These cells are not usable as is for control due to their uncertainty and vibration sensitivity. Therefore they represent lost regions for control. If we can recover them by reducing the uncertainty, they would lead to the enrichment of the region usable for control.

In the following section, a reinforcement learning algorithm is employed which smoothes the roughness in the boundary region. This smoothing yields a recovery of cells that expands the lower approximation of the rough set. This expansion includes the recovered cells of the boundary region and is achieved through reinforcement learning according to the stability history in the state cell trajectory of the ALV chassis vibration control on the irregular roads.

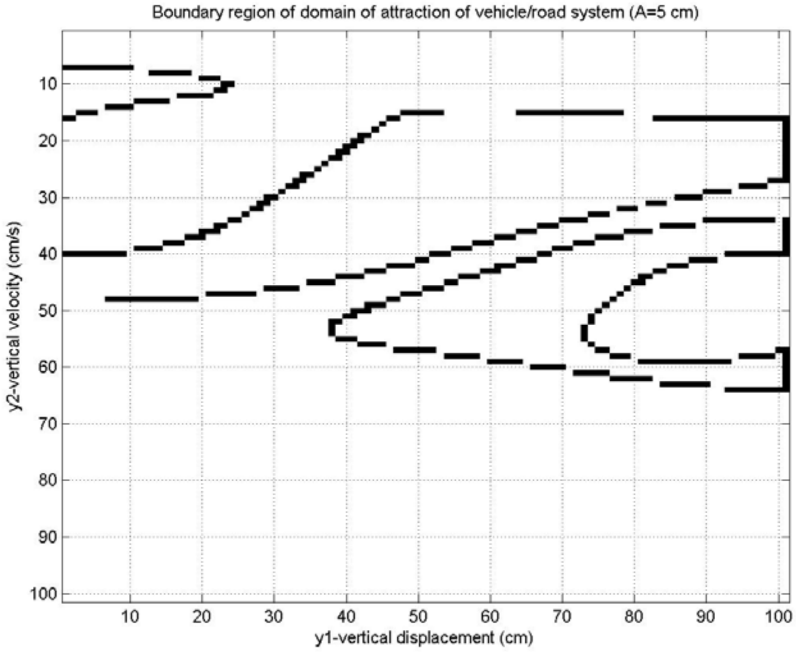

Figure 6-5 Boundary region of the rough set (partially regular areas)

6.3 Smoothing the Chaos Roughness in Regularity

This section aims at decreasing the uncertainty in the boundary region of the rough set and thus smoothing the chaos roughness in that region. The applied method is the *linear-reward inaction* type of reinforcement learning which is based on finite-state automata that adjust the probability of taking different actions according to the previous successes and failures where we consider regularity as success and chaoticity as failure. In our implementations, the cells in the boundary region are rewarded by a history of nonchaotic behavior (regularity) and the instant probability of being regular, (R_n). Reward is decremented when an event other than regularity occurs for the point in cell “i”. Taking numerous random points from each cell and finding out if they are regular or not, R_n of each subcell is iterated by the following reward formulas:

When the point in the cell “i” is regular, the regularity measure is reinforced

$$R_n^i(k+1) = R_n^i(k) + \alpha\left(1 - R_n^i(k)\right) \tag{6.9}$$

When the point in the cell “i” is chaotic, the regularity is penalized by this chaotic behaviour and decreased.

$$R_n^i(k+1) = R_n^i(k) - \alpha R_n^i(k) \tag{6.10}$$

We will illustrate our cell recovery procedure that performs smoothing of the chaos roughness in the boundary region of the rough set, on the vehicle chassis control implementation. In the application example, every cell is divided into 4 and 16 subcells in order to find the fractal dimension and 16-subcell structure is used to compute the regularity number. In this example, the reinforced or penalized regularity measure R_n of each subcell is found using the above formulas, iterated with 100 points chosen randomly within the subcell considered and this process is repeated for every subcell of the boundary region. To provide a slow convergence rate and make the algorithm fully convergent α is taken as a small value: 0.05.

As a result, the R_n values of the subcells considered, either converged to 1 (meaning that uncetainty is fully eliminated and their fractal dimension has become integer making these cells fully regular) or to 0. The subcells with R_n values approximately 1 at the end of the reinforcement learning process are added to the lower approximation. This helps to increase the level of knowledge about the characteristics of the regular and chaotic region. Using this method, the regular region is enlarged by highly probable regular cells according to their past history of nonchaotic behavior. Thus the boundary region of the rough set in the state space is diminished enriching the region of cells to be used for control.

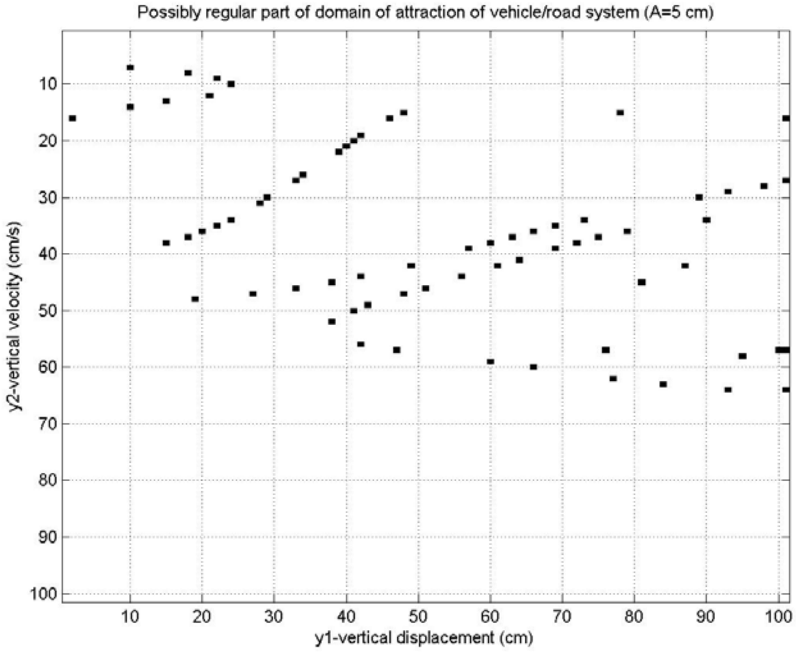

Figure 6-6 Possibly regular part of the boundary region

Including subcells with R_n>0.95, 15.96% of the boundary region (68 cells) has been added through the algorithm to the regular region and the region is expanded by 1% while the rest of the region is converged to the chaotic area for our ALV implementation. The boundary region added to the lower approximation is given in Figure 6-6 and represents the usable portion of the boundary region for control in the motion of the robot on irregular roads. When Figure 6-6 is compared to that of Figure 6-5 we can measure the degree of smoothness achieved after reinforcement learning.

It is easy to see that the smoothing algorithm using a 16-subcell structure has achieved a 1% expansion which is less than the theoretical maximum 1.7% as expected.

6.4 Performance Analysis

In order to check the performance of our approach in expanding the regular region and its indirect effect to control, two analyses are conducted on the system.

First, the system is initiated at a regular point and observed if it enters to one of the 68 cells of the boundary region that has been added through the algorithm to the regular region. After entering one of these added cells, the system is observed in order to find out if it is still regular or jumps to the chaotic region. This analysis helps us to see if we have added the correct cells to the regular region.

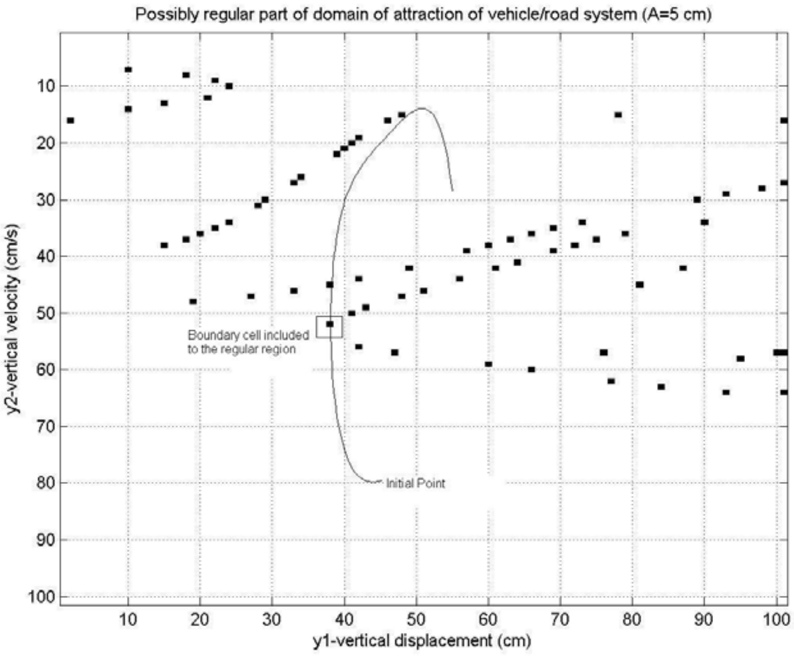

Figure 6-7 Vehicle/road system initiated at regular region

As seen in the figure, the system is started from the regular initial state [0.3994, -1.9233, 0.0128, -0.0148] and flied through boundary cell [-1.248, 0.144, 0, 0] boundary cell (Figure 6-7). It should be noted that Figure 6-7 shows the 2D projection of the system state space where y_3 and y_4 are kept zero. Thus, in order

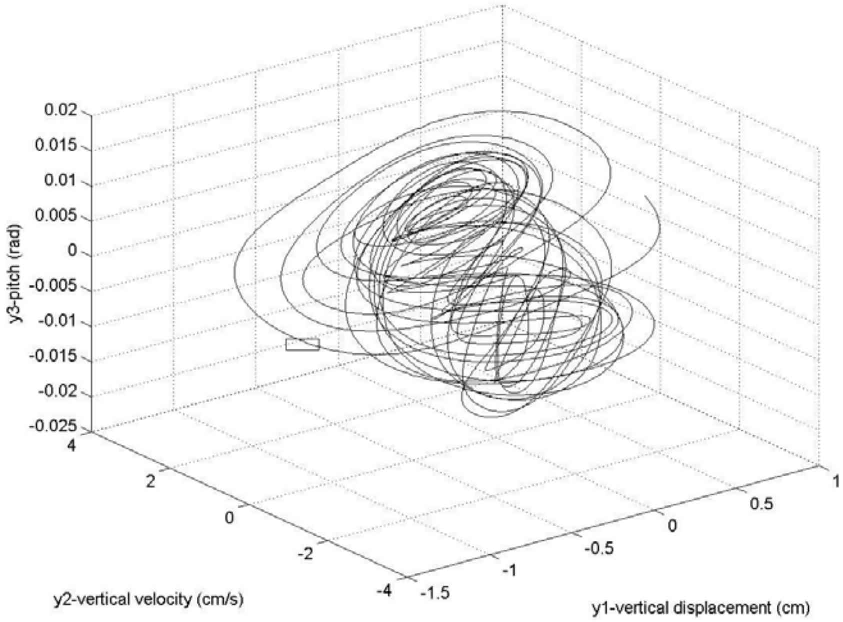

Figure 6-8 3D phase plane of the system (pitch angle is added)

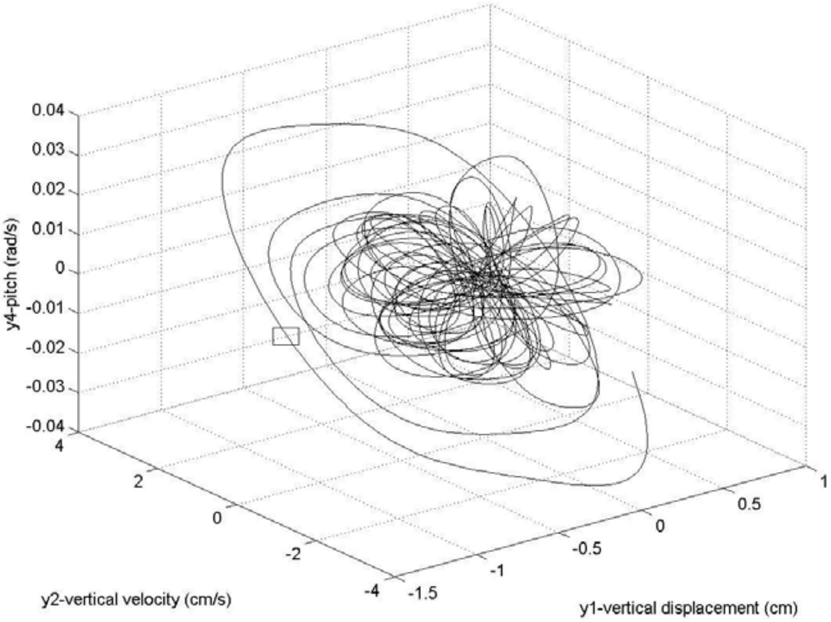

Figure 6-9 3D phase plane of the system (pitch rate added) to gain more insight to the system and understand how it passes through the boundary cell, 3D phase planes are generated. These phase plane plots are given in Figure 6-8 and Figure 6-9. As seen from the figures, the system enters to the y_3=0 and y_4=0 plane, hence enters to our boundary cell which is included to the regular region by smoothing process, and immediately leaves this y_3=0 and y_4=0 plane where the smoothing process was conducted.

The critical issue in this analysis is if the system stayed in regular region after leaving the smoothed cell or not. In order to see this time plot of the system is generated (Figure 6-10). As seen from the figure, the system left the boundary cell at t=0.65 seconds and continued to travel in through the state space. We applied small disturbances to system states during its in order to deviate it from its given trajectory and see if it is chaotic. However, the trajectory has not changed for these small disturbances and it is concluded that the system has stayed in the regular region. Thus, the boundary cell included to the regular region is also regular.

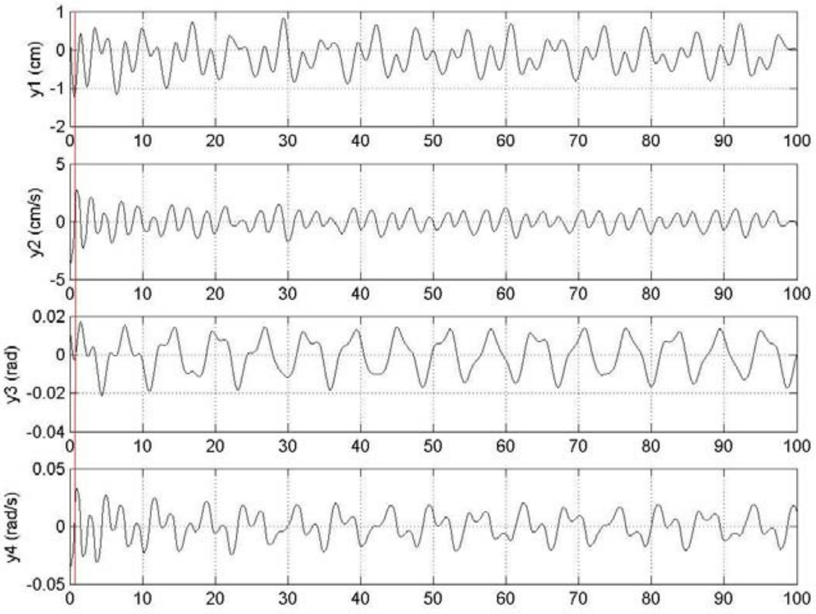

Figure 6-10 Time domain plot of the system

Another test is conducted by choosing cells from the upper approximation of the rough set, hence the chaotic region to see if we have excluded the right cells during the smoothing process. Trajectories of such a cell [0, 2, 0, 0] and its ε neighborhood [-0.001, 2.001, -0.01, 0.001] is given in Figure 6-11.

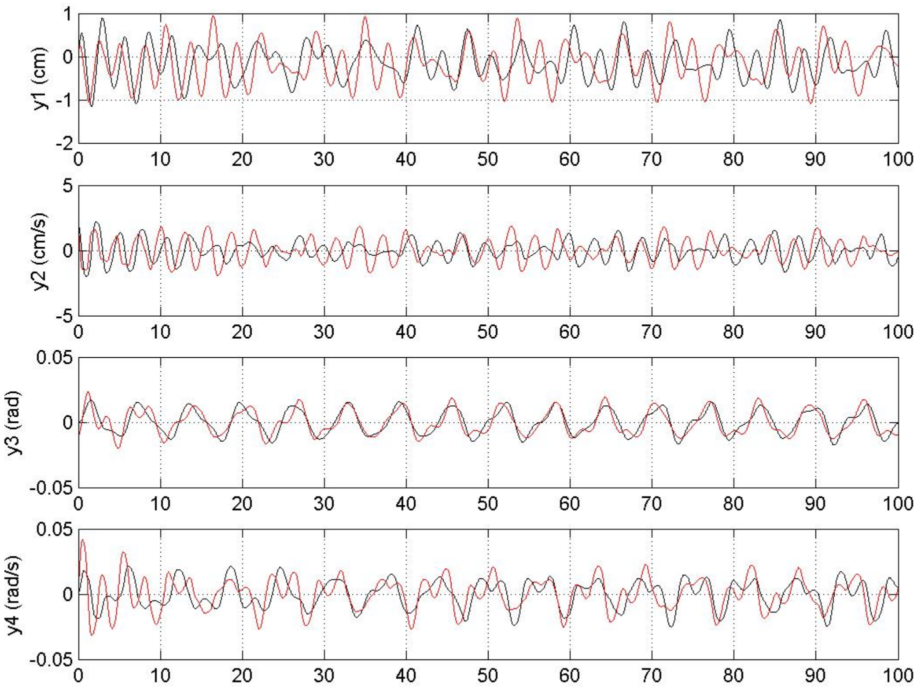

Figure 6-11 Chaotic trajectory

As seen a small uncertainty of ε makes the system behavior unpredictable as time goes by. This means our example cell is really chaotic and the decision of excluding it from the regular region was correct.

Above analysis gives us an insight of what is achieved by fractal/rough set modelling and smoothing. A second analysis is applied to see directly the metric performance measures of the approach. We applied the following procedure in order to reach a metric measure. The procedure is

a. Pick a random initial condition from regular region of the state space,

b. Run the system for 1000 iterations with a time step of $1x10^{-2}$ seconds,

c. Observe the entire trajectory. If it lies within the regular region, record the trial as regular, otherwise as chaotic,

d. Repeat a, b and c steps for 10000 randomly chosen initial points,

Given procedure is repeated before and after smoothing in order to determine the effect of smoothing algorithm to the total uncertainty of the region. As a result of the procedure 9961 points are found to be regular before smoothing and 9959 points are found to be stable after smoothing.

The percentage of regular points in the region defined as regular region has decreased from 99.61 % to 99.59 % after smoothing. Besides this small increase in the uncertainty the regular region is expanded nearly 1 % due to the smoothing process. The applied test procedure shows that the proposed approach is promising.

6.5 Discussion

In the novel approach introduced in this chapter, we increase the richness of the information in the regular region of a nonlinear system using fractal/rough set representation of that region. The fractal/rough set model developed is a new approach brought to the area of the uncertainty modelling in chaotic systems. The approach introduces the fractal dimension of the elements in the rough set as a measure of stability roughness. The uncertainty represented as roughness is then minimized using reinforcement learning. Minimization of uncertainty in the regular region is done as smoothing of the irregularity of system stability in the boundary region.

In this chapter without loss of generality, the regular region for 5 cm road vibration case is investigated in order to decrease the uncertainty in cells which are in the boundary region and are not usable for control due to their uncertainty.

Besides the high nonlinearity of the system that easily can lose its regularity in dynamics, the smoothing performance in this loss of regularity recovers about 15.96 % of boundary. This performance can vary due to the dynamical complexity of the system. The smoothing performance for a two legged walking robot system has been found to be 6.19 % of boundary in (Kaygisiz and Erkmen, (2004)).

Smoothing performance in the current system is investigated by tests and the approach has provided promising results of 1% expansion in the regular space with only 0.02% increase in uncertainty.

CHAPTER 7

CONCLUSION

7.1 Concluding Remarks

Autonomous land vehicles (ALVs) have been employed in miscellaneous operations succesfully in recent years. The main feature of these vehicles is the ‘autonomy’ that relies on robust planners and high performance navigation systems. Most outdoor applications utilizes GPS/INS integrated systems in order to yield high performance navigation solutions. However, there arises two critical problems at the navigation of ALVs during the operations, which are GPS signal loss and dynamic effects of the irregular roads preventing the navigation system to function properly. In real time applications, an autonomous vehicle entering into an environment shadowing the GPS signal becomes incapable of receiving the GPS signal and depends on low cost inertial sensors for navigation solutions. INS providing accurate navigation solutions with the aid of external measurements drifts very fast when GPS signal is lost and the system relies only on the low cost inertial sensors. On the other hand, road irregularities causing random oscillations may be a source of navigation system failure. In this thesis study, these two problems are investigated and intelligent solutions are presented.

INS providing accurate navigation solutions with the aid of GPS measurements drifts very fast when GPS signal is lost and the system relies only on the low cost inertial sensors. Consequently, we have focused on the development and implementation of a GPS/INS system for land navigation possessing the capability of circumventing the GPS signal loss problem without causing the system to drift.

In order to handle such a problem in a cost effective manner, predictions are performed to estimate the position from learned GPS/INS behavior patterns and provide an effective intelligent support to the INS in cases of GPS signal loss. Multilayer perceptrons are integrated to the navigation system for providing the intelligent support to INS.

Two different system architectures and related algorithms, in order to integrate the neural network to the GPS/INS system, are developed and the optimum neural network structure in terms of memory, time and estimation performance is realized. A detailed noise analysis of the ANN system is conducted and ANN outputs are integrated to the Kalman filter for proposed architectures. Consequently, two different intelligent systems aiding to GPS/INS with position or position difference predictions and their algorithms are developed and implemented.

The intelligent structures employing the artificial neural network predictions achieved to diminish the navigation errors and allow the inertial system to navigate more accurately without any external aid. Comparisons between ANN-aided GPS/INS and GPS/INS systems have shown that the intelligent aid is capable of decreasing the system position error to less than 1/10th of GPS/INS error in the case of GPS signals loss.

Another problem in ALV operations is road irregularities causing random oscillations which may be a source of electronic and mechanical system failure. We have focused on the modelling and global stability analysis of the vehicle chassis on irregular roads in order to give a rough picture of the random oscillation problem. The vehicle is analysed for changing road conditions and frequency response of the system is determined.

Making the rough picture of the stability domain more precise and smoothing the uncertainty is critical for control purposes. We increase the richness of the information in the regular region of a nonlinear system using fractal/rough set representation of that region. The fractal/rough set model developed is a new approach brought to the area of the uncertainty modelling in chaotic systems. The approach introduces the fractal dimension of the elements in the rough set as a measure of stability roughness. The uncertainty represented as roughness is then minimized using reinforcement learning. Minimization of uncertainty in the regular region is done as smoothing of the irregularity of system stability in the boundary region.

We apply, in this thesis work, our novel technique to a vehicle on irregular roads system. Besides the high nonlinearity of the system that easily can lose its regularity in dynamics, the smoothing performance in this loss of regularity recovers about 15.96 % of boundary. This performance can vary due to the dynamical complexity of the system.

The main contributions of this thesis are:

- Development of the inertial navigation system aided by GPS and the understanding of the associated practical issues.
- Implementation of a real time inertial navigation system aided by GPS. This system can provide accurate navigation solutions for autonomous vehicles.
- Development of an inertial navigation system aided by artificial neural network structure providing position/position difference estimates. The structure requires no external measurement and yields a low cost accurate navigation solution in the absence of GPS signal.
- Analysis of vehicle/road interaction for irregular roads.

- A new method for increasing the richness of the information in the regular region of a nonlinear system using fractal/rough set representation of that region. The fractal/rough set model developed is a new approach brought to the area of the uncertainty modelling in chaotic systems.

7.2 Suggestions for Future Work

Based on this thesis work, several extensions, related to inertial navigation system aiding and nonlinear system stability uncertainty smoothing might be done.

In this thesis work, a loosely coupled tactical grade system comprises of a 15-state Kalman filter using the external measurements provided from GPS is used in order to demonstrate our intelligent aiding process for the ALV navigation where the GPS receiver can only receive less than 4 satellite signals and cannot provide position/velocity solutions.

However, a tightly coupled GPS/INS system will allow GPS pseudorange aiding to the navigation system even if one satellite is visible to the GPS antenna. Implementing the intelligent structure based on ANN to a tightly coupled system, the performance of the overall navigation system might be improved in 3 or less visible satellites case. Receiving pseudorange aid from 3 or less visible satellites, a tightly coupled approach slows down the drift of navigation system. Thus, applying our intelligent approach to a tightly coupled approach, which means 3 more position observations in addition to visible satellite observations, this degradation may be even stopped in most of the navigation applications.

Furthermore, the intelligent navigation system, which is used in a land vehicle in the thesis, may be applied to the highly dynamical airborne vehicle.

Another possible extension is related to nonlinear system stability uncertainty smoothing. In this thesis work, the smoothing algorithm is dealt with the extension of the regular region. It detects and adds highly probable regular cells to the lower approximation. This smoothing approach may be extended by also detecting highly probable chaotic cells to add them to the upper approximation. Such an approach will enable 2-way smoothing of the boundary region.

Moreover, our uncertainty modeling and roughness smoothing method may be implemented with an intelligent control algorithm that detects transition to chaos to obtain a human like controller that may be applied to highly nonlinear systems.

REFERENCES

[1] Antsaklis P.J., “Neural networks for control systems”, IEEE Transactions on Neural Networks, vol. 1, no. 2, June 1990, pp. 242-244

[2] Barbour N., Schmidt G. “Inertial sensor technology trends”, Proc. of 1998 Workshop on Autonomous Underwater Vehicles, August 1998, pp. 55-62

[3] Barto A.G., Sutton R.S., Anderson C.W., “Neuronlike adaptive elements that can solve difficult learning control problems”, IEEE Transactions on Systems, Man, and Cybernetics, vol. 13, no. 5, 1983, pp. 834-846

[4] Bessen A.S., Levine J., “Strapped-down navigation”, Data Systems Engineering, vol. 17, 1964, pp.6-14

[5] Bortz J.E., “A new mathematical formulation for strapdown inertial formulation”, IEEE Transactions on Aerospace and Electronic Systems, vol. 7, no. 1, Jan. 1971, pp. 61-66

[6] Brown G.B., Hwang P.Y.C., Introduction to Random Signals and Applied Kalman Filtering, 1992, John Wiley & Sons Inc.

[7] Chen G., Dong X., “Control of chaos-a survey”, Proceedings of the 32nd Conference on Decision and Control, December 1993, pp. 469-474

[8] Chiang K.W., Nouraldin A., El-Sheimy N., “Multisensor integration using neuron computing for land-vehicle navigation”, vol 6., GPS Solutions, 2003, pp.209-218

[9] Choi J.Y., Choi C.H., "Sensitivity analysis of multilayer perceptron with differentiable activation functions", IEEE Transactions on Neural Networks, vol. 3, no. 1, 1992, pp. 101-107

[10] Cichocki A., Unbehauen R., Neural Network for Optimization and Signal Processing, 1992, John Wiley Publishers

[11] Czerny L., Popp K., "Nonstationary random vibration of vehicles on irregular guideways", The Dynamics of Vehicles on Roads and Tracks, vol. 17, 1987, pp.90-100

[12] Depatment of Defense World Geodetic System 1984-Its Definition and Relationship with Local Geodetic Systems, 2000, National Imagery and Mapping Agency

[13] Dodds C.J., Robson J.D., "The description of road surface roughness", Journal of Sound and Vibration, vol. 31, no.2, 1973, pp.175-183

[14] Feigenbaum M.J., J. Stat. Phys. 19, 25, 1978

[15] Gamble A.E., Jenkins P.N., "Low cost guidance for the multiple launch rocket system (MLRS) artillery rocket", IEEE AES Systems Magazine, January 2001, pp. 33-39

[16] Garg S.C., Morrow L.D., Mamen R., "Strapdown navigation technology: A literature survey", Journal of Guidance and Control, vol. 1, no. 3, 1978, pp. 161-170

[17] Gelb A., Applied Optimal Estimation, 1974, MIT Press

[18] Greenspan R.L., "GPS/Inertial integration overview", AGARD AG-331, June 1995, pp. 281-294

[19] Gregobi C., Ott E., Yorke J. ‘ Crises, Sudden Changes in Chaotic Attractors, and Transient Chaos’, Physica 7D, 181-200, 1983

[20] Hagan M.T., Menhaj M. “Training feedforward networks with Marquardt algorithm”, IEEE Transactions on Neural Networks, vol. 5, no. 6, 1994, pp. 989–93.

[21] Haykin, S., Neural Networks, 1997, Prentice Hall Inc.

[22] Hilgard E.R., Bower G.H., Theories of Learning (1975), Prentice Hall

[23] Hornik K., Stinchcombe M., White H., “Multilayer feedforward networks are universal approximators”, Neural Networks, vol. 2, 1989, pp.359-366

[24] Hsu C.S., “A theory of cell-to-cell mapping dynamical systems”, ASME 47, 1980, pp. 931-939

[25] Hsu C.S., “An unravelling algorithm for global analysis of dynamical systems: An Application of Cell-to-Cell Mappings”, ASME 47, 1980, pp. 940-948

[26] Ignagni M.B., “Optimal strapdown attitude integration algorithms”, Journal of Guidance, Control, and Dynamics, vol. 13, no. 2, March-April 1990, pp. 363-369

[27] Ignagni M.B., “Efficient class of optimized coning compensation algorithms”, Journal of Guidance, Control, and Dynamics, vol. 19, no. 2, March-April 1996, pp. 424-429

[28] Ignagni M.B., “Duality of optimal strapdown sculling and coning compensation algorithms”, Journal of the Institute of Navigation, vol. 4, no. 2, summer 1998, pp. 85-95

[29] İleri L., Chaotic Dynamic Analysis of a Vehicle Model Over Road Irregularities, M.Sc. Thesis, Middle East Technical University, 1990

[30] Jiang Y.F., Lin Y.P., "Improved strapdown coning algorithms", IEEE Transactions on Aerospace and Electronic Systems, vol. 28, no. 2, April 1992, pp. 484-490

[31] Kalman R. E., "A new approach to linear filtering and prediction problems", Journal of Basic Engineering, vol. 82, March 1960, pp. 35-45

[32] Kalman R. E., Bucy R.S., "New results in linear filtering and prediction theory", Journal of Basic Engineering, vol. 83, December 1961, pp. 95-107

[33] Kamash K.M.A, Robson J.D., "The application of isotropy in road surface modelling", Journal of Sound and Vibration, vol. 57, no. 1, 1978, pp.89-100

[34] Kaplan E.D., Understanding GPS: Principles and Applications, 1996, Artech House

[35] Kay S.M., Fundamentals of Statistical Signal Processing-Estimation Theory, 1993, Prentice Hall Inc.

[36] Kaygısız B., "GPS simülasyonu ve hamveri kullanımı", Kayseri III. Havacılık Sempozyumu Kitapçığı, May 2000, pp. 123-126

[37] Kaygısız B., Erkmen A.M., Erkmen I., "Smoothing stability roughness of a robot arm under dynamic load using reinforcement learning", in Proc. of 2001 IROS, 2001, Maui, Hawaii

[38] Kaygısız B., Erkmen A.M., Erkmen I., "Smoothing stability roughness of fractal boundaries using reinforcement learning", in Proc. of IFAC 15th World Congress, 2002, Barcelona, Spain

[39] Kaygısız B., Gökpınar S., “GPS/INS guided field artillery munitions”, AFCEA Proceedings, May 2003

[40] Kaygısız B., Erkmen A.M., Erkmen İ., “GPS/INS enhancement using neural networks for autonomous ground vehicle applications”, IEEE/RJS International Conference on Intelligent Robots and Systems, Las Vegas, 2003

[41] Kaygısız B., Erkmen A.M., Erkmen İ., “Intelligent compensation of chaos roughness in regularity of walk for a two legged robot”, Journal of Robotic Systems, submitted in 2004

[42] Kumar P.R., Varaiya P.P., Stochastic Systems: Estimation Identification and Adaptive Control, 1986, Prentice Hall

[43] Lamy D., “Modeling and sensitivity analysis of neural networks”, Mathematics and Computers in Simulation 40, 1996, pp. 535-548

[44] LeBlanc P.A., Woodrooffe J.H.F., Yuan B., Ploeg H.L., “Effect of heavy vehicle suspension nonlinearities on pavement loading”, The Dynamics of Vhicles on Roads and Tracks, vol. 17, 1987, pp.223-238

[45] Lefevre H., The Fiber-Optic Gyroscope, 1993, Artech House

[46] Lewantowicz Z.H., Paschall R.N., ‘Deep Integration of GPS, INS, SAR and other Sensor Information’, AGARD AG-331, 1995, pp. 231-264

[47] LITEF, B-290 Triad Technical Specifications and User’s Manual, 2000, LITEF

[48] LITEF, µfors-6 Technical Specifications and User’s Manual, 2000, LITEF

[49] Mahadevan S., Connell J., "Automatic programming of behavior-based robots using reinforcement learning", in Proc. of 9th National Conference on Artificial Intelligence, Anaheim, CA., 1991

[50] Maybeck P., Stochastic Models, Estimation and Control, volume 1 and 2, 1982, Academic Press

[51] Mays D., Faybishenko B.A., 'Washboards in Unpaved Highways as a Complex Dynamic System', Complexity, 2000, pp. 51-60

[52] Miller R.B., "A new strapdown attitude algorithm", Journal of Guidance, Control, and Dynamics, vol. 6, no. 4, July-August 1983, pp. 287-291

[53] Misoi G.K., Carson R.M., Proceedings of the Institute of Mechanical Engineers 203, 1989, p.205

[54] Moon F.C., Chaotic Vibrations: An Introduction for Applied Scientist and Engineers, 1987, Wiley

[55] Moore A.W., Atkeson C.G., "Prioritized sweeping: reinforcement learning with less data and less real time", Machine Learning, vol. 13, 1993

[56] Narendra K.S., Parthasarathy K., "Identification and control of dynamical systems using neural networks", IEEE Transactions on Neural Networks, vol. 1, no. 1, 1990, pp. 4-27

[57] NOVATEL, GPSCard OEM Series Installation and Operating Manual, Novatel Communications Ltd., 1995

[58] Parkinson B.W., Spilker J.J., Axelrad P., Global Positioning System: Theory and Applications, volume 1 and 2, 1995, American Institute of Aeronautics and Astronautics

[59] Papoulis A., Probability Random Variables and Sthocastic Processes, 1987, McGraw Hill Inc.

[60] Parker T.S., Chua L.O., "Chaos: A tutorial for engineers", Proceedings of IEEE, vol. 75, no. 8, 1987, pp. 982-1001

[61] Patil M.K., Palanichamy M.S., "A mathematical model of tractor-occupant systems with a new seat suspension for minimization of vibration response", Applied Mathematical Modelling, vol. 12, 1988, pp.63-71

[62] Pawlak Z., Rough sets: Algebraic and topological approach, Institute of Computer Science Polish Academy of Sciences, 1982

[63] Pawlak Z., Rough Sets, Rough functions and rough calculus, Institute of Theoretical and Applied Informatics Polish Academy of Sciences 1995

[64] Pawlak Z., Rough set approach to knowledge-based decision, European Journal of Operational Research 99, 1997, pp. 48-57

[65] Philips R., Schmidt G.T., "GPS/INS integration", AGARD LS-207, July 1996, pp. 9.1-9.18

[66] Pomeau Y., Manneville P., 'Intermittent Transition to the Turbulence in Dissipative Dynamical Systems', Commun. Math. Phys. 74, 189-197, 1980

[67] Powell J., "The present and future roles of strapped-down inertial navigation systems", Transactions of 8th Symposium on Ballistic and Space Technology, San Diego, 1963

[68] Ruelle D., Takens F., 'On the Nature of Turbulence', Commun. Math. Phys. 20, 167-192, 1971

[69] Quasius G.R., "Strapdown inertial guidance", Space/Aeronoutics, vol. 40, 1963, pp. 89-94

[70] Saad D., On-line Learning in Neural Networks, 1998, Cambridge University Press

[71] Salomon, R., Hemmen L., “Accelerating Backpropogation through Dynamic Self Adaptation”, Neural Networks, vol. 9, no. 4, 1996, pp. 589-601

[72] Salychev O., Inertial Systems in Navigation and Geophysics, 1998, Baumann MSTU Press

[73] Savage P. “Strapdown system algorithms”, AGARD LS-133, May. 1984, pp. 3.1-3.27

[74] Scarselli F., Tsoi A.C., “Universal approximation using feedforward neural networks: A survey of some existing methods and some new results”, Neural Networks, vol. 11, no. 1, 1998, pp. 15-37

[75] Schaal S., Atkeson C., “Robot juggling: an implementation of memory-based learning”, Control Systems Magazine 14, 1994

[76] Schwartz A., “A reinforcement learning method for maximizing undiscounted rewards”, Proc. of the 10th International Conference on Machine Learning, 1993, pp. 298-305

[77] Shepherd A.J., Second-order Methods for Neural Networks, Fast and Reliable Training Methods for Multi-layer Perceptrons, 1996, Springer

[78] Simpson P.K., Artificial Neural Systems, 1990, Academic Press

[79] Siouris G.M., Aerospace Avionics Systems-A Modern Sysnthesis, 1993, Academic Press

[80] Sutton R.S., “Planning by incremental dynamic programming”, Proc. of the 8th International Conference on Machine Learning, 1991, pp. 353-357

[81] Tazartes D.A., “Coning compensation in strapdown inertial navigation systems, 1997, US Patent # 5828980

[82] Tekinalp O., Özemre M., “Artificial neural networks for transfer alignment and calibration of inertial navigation systems”, Proceedings of the AIAA Guidance, Navigation and Control Conference, 2001

[83] Titterton D.H. and Weston J.L., “Strapdown Inertial Navigation Technology”, Peter Pergrinus Ltd., London, 1997

[84] Thompson J.M.T.,Soliman M.S., “Fractal control boundaries of driven oscillators and their relevance to safe engineering design”, Proc. R. Soc. Lond. A 428, 1990, pp. 1-13

[85] Vik B., Fossen T.I., “A nonlinear observer for GPS and INS integration”, Proceedings of 40th IEEE Conference on Decision and Control, December 2001, pp. 2956-2961

[86] Watkins, C.J.C.H., Dayan P., “Q-learning, Machine Learning”, vol. 8, no. 3, “992, pp. 279-292

[87] Widrow B., Lehr M.A., 1990, “30 years of adaptive neural networks: perceptron, madaline and backpropagation”, Proceedings of the IEEE, vol. 78, no. 9, September 1990, pp. 1415-1442

[88] Yu X.H., Chen G.A., “Efficient backpropagation learning using optimal learning rate and momentum”, Neural Networks, vol. 10, no. 3, 1997, pp. 517-527

[89] Yu X.H., Chen G.A., Cheng S.X., “Dynamic learning rate optimization of the backpropagation algortihm”, IEEE Transactions on Neural Networks, vol. 6, no. 3, 1995, pp. 669-677

APPENDIX A

ATTITUDE UPDATE ALGORITHMS

Strapdown inertial navigation systems pass accelerometer measurements given in body frame through the transformation matrix γ and integrate transformed acceleration into position and velocity. This transformation matrix has to be accurate since misalignment causes the acceleration vector to drift from true vehicle acceleration. Even small misalignments cause large navigation errors with accumulation in time.

The transformation matrix has to be updated continuously to reflect body rotation with respect to navigation frame. The update process is based on the measurements obtained from gyroscopes and subject to any error stemming from these sensors.

There exist a number of attitude update algorithms available. All algorithms provide identical results regardless of their implementation differences. The choice of algorithm is related to pros and cons of each approach in different tradeoff areas. The basic three algorithms and their properties can be assessed as follows:

The Euler angle algorithm directly updates Euler angles (roll-φ, pitch-θ and yaw-ψ). In strapdown applications, it is not generally utilized due to roll/yaw singularity of corresponding update algorithm existing when pitch angle, θ is $90°$. Although this does not pose a problem for land and marine applications, this type of representation is generally not preferred even for ground applications. In the updating process, only Euler angles are updated, thus the algorithm does not employ orthonormalization, unlike the Direction Cosine Matrix and quaternion representations. However, the update algorithm contains trigonometric functions and has the greatest computational expense.

The transformation matrix is constructed using Euler angles as:

$$\mathbf{C}_n^b = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & \cos\theta\sin\phi \\ \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi & \cos\theta\cos\phi \end{bmatrix} \quad (A.1)$$

The update of the Euler angles between the body and navigation frame can be achieved by

$$\begin{aligned} \dot{\psi} &= (1/\cos\theta)(\omega_y \sin\phi + \omega_z \cos\phi) \\ \dot{\theta} &= \omega_y \cos\phi - \omega_z \sin\phi \\ \dot{\phi} &= \omega_x + \tan\theta(\omega_y \sin\phi + \omega_z \cos\phi) \end{aligned} \quad (A.2)$$

where $\omega_x, \omega_y, \omega_z$ are the body rotations with respect to navigation frame expressed in body frame.

The Direction Cosine Matrix (DCM) algorithm updates the transformation matrix $\boldsymbol{\gamma}$ and does not possess roll/yaw singularity. It is widely used in aerospace and military applications. Although, nine direction cosines of 3x3 transformation matrix are updated at every step, updating two rows (six elements) of the matrix is enough, since third row of the matrix can be determined from other two rows by a cross product operation. The update algorithm does employ orthogonalization and normalization.

The direction cosine matrix update is a method to generate transformation matrix between each axis of one frame and every axis of another one. The matrix is created using the vector dot products which leads to the cosine of the angle between the two vectors involved that is

$$1_i \bullet 1_j = |1_i||1_j|\cos\alpha_{ij} = \cos\alpha_{ij} \tag{A.3}$$

Thus, dot producting each axis of navigation coordinate system with every axis of body coordinate system as in (A.3), a transformation matrix between two frames results as an array of nine direction cosines as follows:

$$\mathbf{C}_b^n = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \tag{A.4}$$

As seen above, the method is trivial and has low computational load. Another issue in applying the method is how to update the transformation matrix. In order to update the matrix, the following relation is used.

$$\dot{\mathbf{C}}_b^n = \mathbf{C}_b^n \mathbf{\Omega}_{nb}^b = -\mathbf{C}_b^n \mathbf{\Omega}_{bn}^b \tag{A.5}$$

where $\mathbf{\Omega}_{nb}^b$ is the skew-symmetric matrix of the angular velocity of the body frame relative to navigation frame represented in body frame.

The Quaternion algorithm updates a 4x1 vector **q** (q_0, q_1, q_2, q_3) known as quaternion vector. In the approach, the rotation from one frame to another is achieved by a single rotation about quaternion vector **q**. In the updating process, four quaternions are updated at every step, and a normalization done. Update is achieved as:

$$
\begin{aligned}
\dot{q}_0 &= \frac{1}{2}\left(-q_2\omega_y - q_3\omega_z - q_1\omega_x\right)\\
\dot{q}_1 &= \frac{1}{2}\left(q_0\omega_x - q_3\omega_y + q_2\omega_z\right)\\
\dot{q}_2 &= \frac{1}{2}\left(q_3\omega_x + q_0\omega_y - q_1\omega_z\right)\\
\dot{q}_3 &= \frac{1}{2}\left(-q_2\omega_x + q_1\omega_y + q_0\omega_z\right)
\end{aligned}
\tag{A.6}
$$

and the transformation matrix is represented in terms of quaternions as follows:

$$
\mathbf{C}_b^n = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_0q_2+q_1q_3) \\ 2(q_0q_3+q_1q_2) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_0q_1+q_2q_3) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \tag{A.7}
$$

As seen from the above discussion, Euler angle approach is not feasible because of the singularity problem within the update algorithm and its computational complexity. It is difficult to draw conclusion in the favor of DCM versus quaternion approach. Quantitative comparisons based on computer sizing and computer loading have led to inconclusive results (Savage, (1984)). However, DCM representation has advantage over quaternion in terms of physical interpretation. Hence, this thesis makes use of direction cosine matrix for attitude representation and attitude update is achieved at 100 Hz.

In order to update direction cosine matrix in a digital computer environment, the update procedure is discretized.

Inertial measurement unit provides rotation rate information $\boldsymbol{\omega}_{ib}^b$ at 100 Hz and this rotation is compensated with navigation frame's rotation with respect to inertial frame in order to obtain body to navigation rotation rate as:

$$
\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - \mathbf{C}_n^b(\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \tag{A.8}
$$

Integrating the resulting body to navigation rotation rate, angle increments are determined as ϕ_x, ϕ_y, ϕ_z. Using these angle increments, Taylor series coefficients required for discretization are found as

$$\begin{aligned} \alpha &= \frac{\sin\phi}{\phi} \\ \beta &= \frac{1-\cos\phi}{\phi^2} \end{aligned} \tag{A.9}$$

where

$$\phi = \sqrt{\phi_x^2 + \phi_y^2 + \phi_z^2} \tag{A.10}$$

Using (A.9) and (A.10) direction cosine matrix update, $\dot{\mathbf{C}}_b^n = \mathbf{C}_b^n \mathbf{\Omega}_{nb}^b$ takes the following discrete form:

$$\mathbf{C}_b^n(k+1) = \mathbf{C}_b^n(k)\left[\mathbf{I}_{3x3} + \alpha\overline{\boldsymbol{\varphi}} + \beta\overline{\boldsymbol{\varphi}}^2\right] \tag{A.11}$$

where the angular skew symmetric matrix is

$$\overline{\boldsymbol{\varphi}} = \begin{bmatrix} 0 & -\phi_z & \phi_y \\ \phi_z & 0 & -\phi_x \\ -\phi_y & \phi_x & 0 \end{bmatrix} \tag{A.12}$$

Applying (A.11) at 100 Hz attitude is updated in the strapdown system in this work. The obtained transformation matrix is then used to evaluate the acceleration in the navigation frame which is provided to navigation computer for determining position and velocity.

APPENDIX B

VELOCITY AND POSITION ALGORITHMS

Achieving the attitude algorithm and its update, the strapdown computer attempts to construct the velocity and position solutions. The velocity propagation in the navigation frame is given by:

$$\dot{\mathbf{v}}_e^n = \mathbf{C}_b^n \mathbf{f}^b - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) x \mathbf{v}_e^n + \mathbf{g}^n \tag{B.1}$$

In order to accomplish the update process in a digital environment, this relation is discretized. Converting (B.1) into discrete form, velocity vector (velocity of body with respect to earth expressed in navigation, north-east-down, frame) is updated as:

$$\mathbf{v}(k+1) = \mathbf{v}(k) + \mathbf{C}_b^n \mathbf{w}(k) - (2\boldsymbol{\Omega}_{ie}^n \Delta t + \boldsymbol{\Omega}_{en}^n \Delta t)\mathbf{v}(k) + \mathbf{g}^n \Delta t \tag{B.2}$$

In (B.2), $\mathbf{w}_k$ is the compensated velocity increment vector resolved in the body frame and available from accelerometers at 100 Hz. $\mathbf{g}^n$ is the gravity in the vehicle location expressed in navigation frame and $\Delta t = 0.01$ seconds. $\boldsymbol{\Omega}_{ie}^n$ and $\boldsymbol{\Omega}_{en}^n$ are the skew-symmetric matrices of earth rate and craft rate vectors.

Accuracy of velocity is dependent on sensor errors, initial errors and gravity errors as seen in (B.2). In order to achieve good navigation performance, we should employ a good approximation for the true gravity vector in navigation frame. For the navigation system we use, the following approximation is exploited

$$\gamma_h = \gamma\left[1 - \frac{2}{a}(1 + f + m - 2f\sin^2\phi)h + \frac{3}{a^2}h^2\right] \tag{B.3}$$

where γ is the normal gravity on the surface of the ellipsoid given by the closed formula of Somigliana (DoD (2000)). The resultant gravity vector is

$$\mathbf{g}^n = \begin{bmatrix} 0 & 0 & \gamma_h \end{bmatrix} \tag{B.4}$$

Using the updated velocity vector, position on the earth in terms of latitude, longitude and height are updated as:

$$\begin{bmatrix} L(k+1) \\ \lambda(k+1) \\ h(k+1) \end{bmatrix} = \begin{bmatrix} L(k) \\ \lambda(k) \\ h(k) \end{bmatrix} + \begin{bmatrix} \dfrac{v_n(k)}{R_N + h} \\ \dfrac{v_e(k)}{(R_E + h)\cos L} \\ -v_d(k) \end{bmatrix} \Delta t \tag{B.5}$$

where the velocities are north, east and down velocities, R's stand for north and east radius of the earth at the vehicle location.

APPENDIX C

CONING AND SCULLING COMPENSATION

The rotation vector behavior can be described as

$$\dot{\boldsymbol{\varphi}} = \boldsymbol{\omega} + \frac{1}{2}\boldsymbol{\varphi} x \boldsymbol{\omega} + \frac{1}{12}\left(\boldsymbol{\varphi} x (\boldsymbol{\varphi} x \boldsymbol{\omega})\right) \tag{C.1}$$

and for the small angle, $\boldsymbol{\varphi}$ is approximated as

$$\dot{\boldsymbol{\varphi}} = \boldsymbol{\omega} + \frac{1}{2}\boldsymbol{\varphi} x \boldsymbol{\omega} \tag{C.2}$$

To find the incremental rotation vector, (C.2) is integrated within the sampling interval

$$\begin{aligned} \Delta\boldsymbol{\varphi}_k &= \int_{t_{k-1}}^{t_k} \boldsymbol{\omega} dt + \Delta\boldsymbol{\varphi}_c(k) \\ \Delta\boldsymbol{\varphi}_c(k) &= \tfrac{1}{2} \int_{t_{k-1}}^{t_k} \Delta\boldsymbol{\varphi} x \boldsymbol{\omega} dt \end{aligned} \tag{C.3}$$

The second term in (C.3) is the coning term and has to be calculated. The first part is the gyroscope provided angle increment vector which is

$$\boldsymbol{\alpha}_k(m) = \int_{t_{m-1}}^{t_m} \boldsymbol{\omega} dt \tag{C.4}$$

In the above equations *m* is the gyro output rate of 400 Hz and *k* is the moderate update rate of 100 Hz. The order of coning compensation algorithm is *m/k* and there exist various algorithms for different *m/k* values of 2, 3, 4 and so forth. The derivations of the algorithms can be found in (Miller (1983), Jiang (1992), Tazartes (1997), Ignagni (1990,1998), (1996), Salychev (1998)).

In our system, a 4th order algorithm of the following form

$$\begin{aligned}\Delta\boldsymbol{\varphi}_c(k) = \frac{2}{3}\boldsymbol{\alpha}_k(1)x\boldsymbol{\alpha}_k(2) + \frac{2}{3}\boldsymbol{\alpha}_k(3)x\boldsymbol{\alpha}_k(4) + \frac{8}{15}\boldsymbol{\alpha}_k(1)x\boldsymbol{\alpha}_k(3) \\ + \frac{8}{15}\boldsymbol{\alpha}_k(2)x\boldsymbol{\alpha}_k(4) + \frac{7}{15}\boldsymbol{\alpha}_k(1)x\boldsymbol{\alpha}_k(4) + \frac{7}{15}\boldsymbol{\alpha}_k(2)x\boldsymbol{\alpha}_k(3)\end{aligned} \tag{C.5}$$

is employed. The derivation of the algorithm is in (Salychev (1998)).

Compensating the coning effect, the angle increment data is ready for attitude computations. Coning compensated angle increment data is given as

$$\Delta\boldsymbol{\varphi}_k = \sum_{m=1}^{4}\boldsymbol{\alpha}_k(m) + \Delta\boldsymbol{\varphi}_c(k) \tag{C.6}$$

where 4 is the number of gyro samples in the interval of one attitude update.

The absolute acceleration behavior can be described as

$$\frac{d}{dt}\mathbf{v}\,|_b = \frac{d}{dt}\mathbf{v}\,|_i - \boldsymbol{\omega}_{ib}x\mathbf{v} \tag{C.7}$$

where $\mathbf{v}$ is the absolute velocity vector of the vehicle with respect to inertial frame. Integrating the above equation over one update period, the velocity change of the vehicle with respect to inertial frame which is provided to the navigation computer, is calculated.

As stated before, gyroscopes provide angle increments at 400Hz as

$$\boldsymbol{\alpha}_{x,y,z} = \int_{t}^{t+\Delta t_m} \boldsymbol{\omega}_{x,y,z} dt \tag{C.8}$$

Likewise, accelerometers provide velocity increments as

$$\Delta \mathbf{w}_{x,y,z} = \int_{t}^{t+\Delta t_m} (\frac{d}{dt}\mathbf{v}\,|_i)_{x,y,z} dt \tag{C.9}$$

If integration of (C.7) is proceed as a 4 step recurrent discrete process using sensor outputs given in (C.8) and (C.9), the sculling compensated velocity increments at 100 Hz are calculated as

$$\begin{aligned}
w_{x,k}(m) &= w_{x,k}(m-1) + w_{y,k}(m-1)\alpha_{z,k}(m) - w_{z,k}(m-1)\alpha_{y,k}(m) + \Delta w_{x,k}(m) \\
w_{y,k}(m) &= w_{y,k}(m-1) + w_{z,k}(m-1)\alpha_{x,k}(m) - w_{x,k}(m-1)\alpha_{z,k}(m) + \Delta w_{y,k}(m) \\
w_{z,k}(m) &= w_{z,k}(m-1) + w_{x,k}(m-1)\alpha_{y,k}(m) - w_{y,k}(m-1)\alpha_{x,k}(m) + \Delta w_{z,k}(m) \\
w_{x,k}(m) &= w_{x,k}(m-1) + w_{y,k}(m)\alpha_{z,k}(m) - w_{z,k}(m)\alpha_{y,k}(m) + \Delta w_{x,k}(m) \\
w_{y,k}(m) &= w_{y,k}(m-1) + w_{z,k}(m)\alpha_{x,k}(m) - w_{x,k}(m)\alpha_{z,k}(m) + \Delta w_{y,k}(m) \\
w_{z,k}(m) &= w_{z,k}(m-1) + w_{x,k}(m)\alpha_{y,k}(m) - w_{y,k}(m)\alpha_{x,k}(m) + \Delta w_{z,k}(m)
\end{aligned} \tag{C.10}$$

with initial conditions $w_x = w_y = w_z = 0$. Repeating this process for every new sensor measurement, the resulting body velocity increments with respect to inertial frame are used in the velocity update. The derivation of the algorithm is in (Salychev (1998)).

APPENDIX D

THE KALMAN FILTER

Kalman filtering is a method used for optimal recursive estimation of current system states under the assumption that system model is linear contaminated by white Gaussian system and measurement noises. Discrete time theory of the filter is introduced by R.E. Kalman in (Kalman (1960)), while continuous time theory appears a year later in (Kalman (1961)).

In a Kalman filter, state estimates are based on an optimal combination of predicted states and actual measurements. Kalman filter is optimal such that it generates the best estimate of the quantities from data provided in a noisy environment. Kalman filter estimates are unbiased in a minimum variance sense provided that the required assumptions on the system and measurement noises are met. Detailed analysis on the derivations and theory of the filter can be found in (Papoulis (1987), Gelb (1974), Maybeck (1982)).

D.1 Linear Kalman Filter

A system can be represented as a set of linear differential equations as follows,

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{D.1}$$

where

$\mathbf{x}$ = state vector (n dimensional)

$\mathbf{F}$ = state dynamics matrix (nxn)

B = control gain matrix (nxr)

u = control input vector (r dimensional)

G = noise gain marix (nxs)

w = white Gaussian driving noise vector (s dimensional)

Whenever the deterministic control term $\mathbf{B}(t)\mathbf{u}(t) = 0$ as in navigation applications where there is no control input to the system, the system representation becomes

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}(t) \tag{D.2}$$

In the above representation, the expected value of the driving noise **w**(*t*) is

$$E\{\mathbf{w}(t)\} = \mathbf{0} \tag{D.3}$$

and the covariance of the noise can be given as

$$E\{\mathbf{w}(t)\mathbf{w}(t+\tau)\} = \mathbf{Q}(t)\delta(\tau) \tag{D.4}$$

where $\delta(\;)$ is the Dirac delta function.

The Kalman filter integrates the discrete measurements of external devices in a way that can be described by the following linear measurement equation:

$$\mathbf{z}(t_k) = \mathbf{H}(t_k)\mathbf{x}(t_k) + \mathbf{v}(t_k) \tag{D.5}$$

where **H** is the observation matrix, **v** is the zero mean white Gaussian measurement noise vector with the following covariance:

$$E\left\{\mathbf{v}(t_k)\mathbf{v}(t_j)\right\} = \begin{cases} \mathbf{R}(t) & t_k = t_j \\ \mathbf{0} & t_k \neq t_j \end{cases} \tag{D.6}$$

In real time applications, discrete time system representation is required and is given as

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k \mathbf{x}_k + \mathbf{w}_k \tag{D.7}$$

where discrete time driving noise of the system becomes zero mean, uncorrelated random sequence with covariance $\mathbf{Q}_k$, which can be approximated using the continuous time covariance as:

$$\mathbf{Q}_k = \mathbf{G}(t_k)\mathbf{Q}\mathbf{G}^T(t_k)\Delta t \tag{D.8}$$

provided that Δt is small. $\mathbf{x}_\mathrm{k}$ is the discrete time representation of the state at time t_k and $\mathbf{\Phi}_\mathrm{k}$ is the state transition matrix given as:

$$\mathbf{\Phi}_k = e^{\int_{t_{k-1}}^{t_k} \mathbf{F}(\tau)d\tau} = \mathbf{I} + \Delta t\mathbf{F}(t_k) + \frac{\left(\Delta t\mathbf{F}(t_k)\right)^2}{2!} + \ldots \tag{D.9}$$

Discrete time measurement equation of the system is

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{D.10}$$

where noise covariance becomes

$$\mathbf{R}_k = \mathbf{R}(t_k)/\Delta t \tag{D.11}$$

Kalman Filter propagates the current state estimation $\hat{\mathbf{x}}_{k-1}(+)$ and covariance $\mathbf{P}_{k-1}(+)$ to the next measurement cycle in the propagation state. The state vector is predicted by extrapolating the current state vector using the state transition matrix.

$$\hat{\mathbf{x}}_k(-) = \mathbf{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1}(+) \tag{D.12}$$

Uncertainty of the propagated state given by $\mathbf{P}_k(-)$ is computed using the state transition matrix, previous uncertainty level $\mathbf{P}_{k-1}(+)$ and the driving noise covariance as follows:

$$\mathbf{P}_k(-) = \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}(+)\mathbf{\Phi}^T{}_{k-1} + \mathbf{Q}_{k-1} \tag{D.13}$$

After propagation, the predicted state vector and covariance matrix are updated employing external measurements. Combining the predicted covariance matrix $\mathbf{P}_k(-)$, measurement matrix $\mathbf{H}_k$ and measurement noise covariance matrix $\mathbf{R}_k$, Kalman gain $\mathbf{K}_k$ is calculated as

$$\mathbf{K}_k = \mathbf{P}_k(-)\mathbf{H}_k{}^T[\mathbf{H}_k\mathbf{P}_k(-)\mathbf{H}_k{}^T + \mathbf{R}_k]^{-1} \tag{D.14}$$

Kalman gain matrix assigns weights to the measurements, which are the difference between the actual external measurements and the Kalman prediction of the measurements, based on the measurement noise statistics, $\mathbf{R}_k$ and the prediction uncertainty, $\mathbf{P}_k(-)$. When a measurement is obtained, an estimate of the current state is given by,

$$\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + \mathbf{K}_k[\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k(-)] \tag{D.15}$$

and the error covariance matrix of the estimation is

$$\mathbf{P}_k(+) = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{P}_k(-) \tag{D.16}$$

D.2 Extended Kalman Filter

In real systems with nonlinear dynamics and/or nonlinear measurements, linear Kalman filter approach described above cannot be implemented. An approximate solution is seeked by linearizing the system around its nominal trajectory and apply Kalman filter to this linear system (Kay (1993)).

A nonlinear system with deterministic control term $\mathbf{u}(t)=0$ can be described by the following differential equation:

$$\dot{\mathbf{x}}(t)=\mathbf{f}\left(\mathbf{x}(t),t\right)+\mathbf{G}(t)\mathbf{w}(t) \tag{D.17}$$

where $\mathbf{f}$ is nonlinear time dependent function of $\mathbf{x}$ and $\mathbf{w}(t)$ is the white Gaussian noise driving the system linearly. Moreover, the measurement equation may also be a nonlinear function such as:

$$\mathbf{z}\left(t_k\right)=\mathbf{h}\left(\mathbf{x}\left(t_k\right),t_k\right)+\mathbf{v}\left(t_k\right) \tag{D.18}$$

where white Gaussian noise is additive.

In order to apply the linear Kalman filter, the given nonlinear system and measurement equations should be linearized around a nominal state trajectory. Perturbing the system equations around a nominal trajectory of the state $\mathbf{x}_n(t)$, error state equations are obtained as:

$$\dot{\mathbf{x}}(t)-\dot{\mathbf{x}}_n(t)=\mathbf{f}\left(\mathbf{x}(t),t\right)-\mathbf{f}\left(\mathbf{x}_n(t),t\right)+\mathbf{G}(t)\mathbf{w}(t) \tag{D.19}$$

The above equation may be approximated by Taylor series expansion as:

$$\delta\dot{\mathbf{x}}(t)=\mathbf{F}\left(\mathbf{x}_n(t);t\right)\delta\mathbf{x}(t)+\mathbf{G}(t)\mathbf{w}(t) \tag{D.20}$$

where $\delta\mathbf{x}(t)=\mathbf{x}(t)-\mathbf{x}_n(t)$ represents the first order approximation of the state errors, $\mathbf{F}(\mathbf{x}_n(t);t)$ is a matrix of partial derivatives of **f** along the nominal trajectory:

$$\mathbf{F}\left(\mathbf{x}_n(t);t\right)=\left.\frac{\partial\mathbf{f}(x,t)}{\partial x}\right|_{\mathbf{x}=\mathbf{x}_n(t)} \tag{D.21}$$

Likewise, the perturbation measurement equation is

$$\delta\mathbf{z}(t_k)=\mathbf{H}\left(\mathbf{x}_n(t_k);t_k\right)\delta\mathbf{x}(t)+\mathbf{v}(t_k) \tag{D.22}$$

and

$$\mathbf{H}\left(\mathbf{x}_n(t_k);t_k\right)=\left.\frac{\partial\mathbf{h}(x,t_k)}{\partial x}\right|_{\mathbf{x}=\mathbf{x}_n(t_k)} \tag{D.23}$$

These linear error equations are used in order to implement linear Kalman filter theory. Estimate of the system states are obtained using the error states as

$$\hat{\mathbf{x}}(t)=\mathbf{x}_n(t)+\delta\hat{\mathbf{x}}(t) \tag{D.24}$$

If nominal and true trajectories differ by a large amount, the system may produce unacceptable levels of error. Extended Kalman filtering circumvents this problem by allowing the linearization around the current state estimate at each time step. The extended filter can be realized using the given linearized equations in the linear Kalman filter equations, while maintaining the linearization at the nominal states updated by $\hat{\mathbf{x}}(t)=\mathbf{x}_n(t)+\delta\hat{\mathbf{x}}(t)$.

APPENDIX E

ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANNs) are processing structures, composed of simple computing elements interconnected to each other, which are used to perform complex computational tasks in fields such as control, optimization, signal processing, modelling, system identification and pattern recognition (Cichocki (1992), Simpson (1990), Narendra (1990), Antsaklis (1990), Haykin (1997), Widrow (1990)).

ANN can be considered as a simple replica of the neural system of living creatures. As in the biological nerve systems, ANN consists of simple processing elements called neurons, each of which is a nonlinear multi-input/single-output device. Having an interconnected neuron structure, an ANN has the ability of adapting itself to the changing environment, simulating complex tasks, recognition of the objects with different features.

A basic artificial neuron model is given in Figure E-. As seen in the figure, the neuron has N input lines and a single output. Each input multiplied by a weight is summed up with a threshold value to generate a linear combination of all weighted information. The output of the adder is then processed through a nonlinearity called activation function. Thus, a neuron may be described as:

$$u = \sum_{i=1}^{N} w_i x_i - \theta \qquad \text{(E.1)}$$

and

$$y = f(u) \tag{E.2}$$

where x's are the input signals, w's are the synaptic weight, u is the activation of the neuron, θ is the threshold, y is the output of the neuron and $f()$ is the activation function. Taking $w_0 = \theta$ and threshold input as $x_0 = -1$, may be rewritten as

$$y = f\left(\sum_{i=0}^{N} w_i x_i\right) \tag{E.3}$$

Different activation functions are employed according to applications. Three most commonly used functions are hard-limit, linear and log-sigmoid activation functions (Figure E-1).

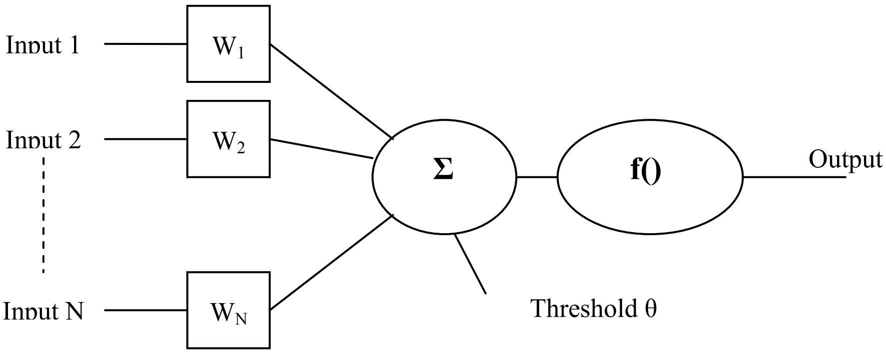

Figure E-1 Neuron Model

The hard-limit function quantizes the output of the neuron to either 0 or 1 values. This function is usually used in neurons acting as classifiers. The linear neurons are used as linear approximators mostly in linear filter design. The sigmoid function may take any input value between plus and minus infinity, and outputs within a range of 0 to 1. It is the most common function in backpropagation neural network applications since it is differentiable, where differentiability is fundamental in the learning process. The logarithmic sigmoid function is

$$f(u) = \frac{1}{1+e^{-u}} \tag{E.4}$$

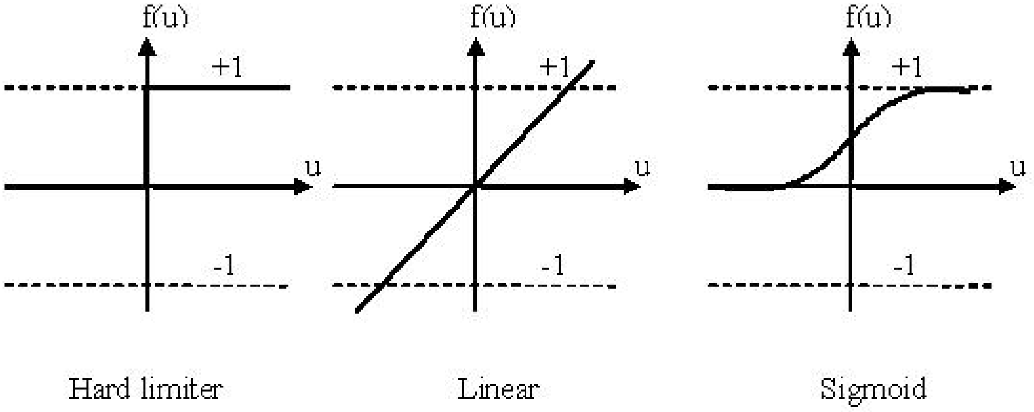

Figure E-1 Common Activation Functions

As stated earlier, ANN uses a fully interconnected neuron structure to be able to achieve complex tasks such as classification, system approximation, optimization and recognition. One large class of neural networks used in the application is the multilayer perceptron (MLP). Multilayer feedforward network is the most popular of all neural network structures and has been the subject of intensive research and diverse applications in recent years. Many well established learning laws are proposed for these systems for their use in modeling, identification and control. It has also been shown that an artificial neural network based on the multilayer feedforward structure with input/output and hidden layers is capable of operating as a universal approximator and approximate generic nonlinear functions (Scarselli (1998), Hornik (1989)).

Multilayer perceptron (MLP) is constructed in order to estimate the given input/output map. Input neurons feed signals to the hidden layer neurons. There may be one or more adjacent hidden layers in an MLP. The last layer is the output layer where the response of the network is produced, by passing the weighted sum of these previous layer outputs through a nonlinear activation function, $f_j^{[s]}$

$$
\begin{aligned}
o_j^{[s]} &= f_j^{[s]}\left(u_j^{[s]}\right) \\
u_j^{[s]} &= \sum_{i=0}^{n_{s-1}} w_{ji}^{[s]} o_i^{[s-1]} \quad ,s = 1,2; j = 1,2,...,n_s
\end{aligned}
\tag{E.5}
$$

where $w_{ji}^{[s]}$ is the synaptic weights by which the jth neuron of the sth layer multiplies ith output from preceding layer $o_i^{[s-1]}$ and n_s is the number of neurons in the sth layer. The synaptic weights $w_{ji}^{[s]}$ are modified by training the network through a learning process.

The learning process applied to feedforward neural networks is known as backpropagation learning. The backpropagation learning process applied to feedforward neural networks equips the network with an ability of capturing complex input-output relationship with a multilevel error minimization feedback.

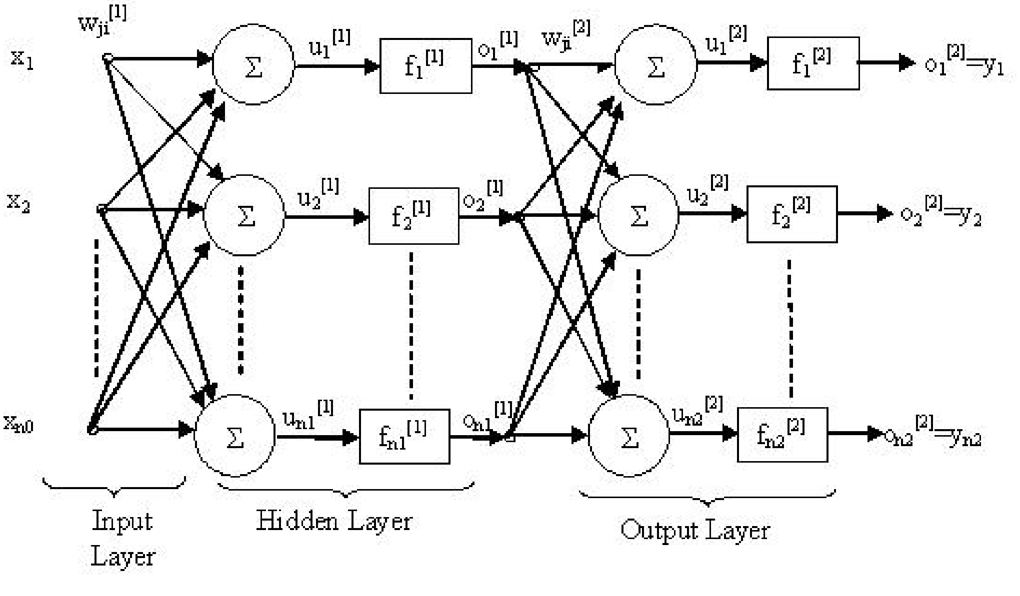

Figure E-2 Multilayer Feedforward Network Architecture

In backpropagation learning cycle known as epoch, an input is propagated forward through the network and the resulting network output is compared to the desired output for generating the output error. This error is backpropagated from the output layer through the input in order to update the connection weights and biases in the network according to a learning rule that minimizes the error. This cycle is repeated for every input-output pair in the training set until the weights and biases converge to their optimal value with minimal error. Training of the network can be performed offline with predetermined input-output pairs, or online with real time generated input-output pairs. Backpropagation algorithm used to train the MLP employs the gradient-descent approach to minimize the mean-squared error function. The error function for the p$^{\text{th}}$ learning sample can be given as

$$E_p = \frac{1}{2}\sum_{j=1}^{n_2}\left(d_{jp} - y_{jp}\right)^2 = \sum_{j=1}^{n_2} e_{jp}^2 \qquad \text{(E.6)}$$

where d_{jp} represents the desired network output and y_{jp} is the actual output of the network. Thus, the global error function over all learning samples is

$$E = \sum_p E_p = \frac{1}{2}\sum_p \sum_{j=0,..,n_2} \left(d_{jp} - y_{jp}\right)^2 \qquad \text{(E.7)}$$

where n_2 is the number of neurons in the output layer.

In backpropagation learning, the synaptic weights $w_{ji}^{[s]}$ are changed by an amount $\Delta w_{ji}^{[s]}$ proportional to the negative gradient of the local error function E_p at every learning cycle. $\Delta w_{ji}^{[s]}$ is

$$\Delta w_{ji}^{[s]} = -\eta \frac{\partial E_p}{\partial w_{ji}^{[s]}},\ \eta > 0 \qquad \text{(E.8)}$$

where η is a small learning parameter.

Change in the weights in the output layer becomes

$$\Delta w_{ji}^{[2]} = \eta \delta_j^{[2]} o_i^{[1]} \tag{E.9}$$

where

$$\delta_j^{[2]} = \left(d_{jp} - y_{jp} \right) \frac{\partial f_{jp}^{[2]}}{\partial u_j^{[2]}} \tag{E.10}$$

For the hidden layer,

$$\Delta w_{ji}^{[1]} = \eta \delta_j^{[1]} o_i^{[0]} \tag{E.11}$$

where

$$\delta_j^{[2]} = \frac{\partial f_{jp}^{[1]}}{\partial u_j^{[1]}} \sum_{k=1}^{n_2} \delta_k^{[2]} w_{kj}^{[2]} \tag{E.12}$$

have been obtained.

The weights are updated as

$$w_{ji}^{[s]}\left(p+1 \right) = w_{ji}^{[s]}\left(p \right) + \Delta w_{ji}^{[s]}\left(p \right) \tag{E.13}$$

using the gradient-descent approach. Although standard gradient-descent approach is simple and easy to understand, it has some significant drawbacks. Learning characteristics of the ordinary gradient descent algorithm is highly dependent on the value of the learning rate η. The algorithm is very slow with small values of the learning parameters, while it becomes oscillatory for large ones, which prevent the algorithm to converge to the desired solution.

One simple approach to improve the algorithm is adding a momentum term to weight changes of the standard backpropagation as

$$\Delta w_{ji}^{[s]}(p) = \eta \delta_j^{[s]} o_i^{[s-1]} + \alpha \Delta w_{ji}^{[s]}(p-1) \tag{E.14}$$

where $0 \le \alpha < 1$. The second term, momentum term, acts like a low pass filter and prevents the network to get stuck in a shallow local minimum. The gradient descent with momentum often provides faster convergence.

In order to prevent the algorithm from generating large weights, which slows down considerably the convergence to an optimal solution in a reasonable time, a term with the decay factor γ is introduced to (E.8) as

$$\Delta w_{ji}^{[s]}(p) = \eta \delta_j^{[s]} o_i^{[s-1]} - \gamma \Delta w_{ji}^{[s]}(p) \tag{E.15}$$

The decay factor improves also the generalization ability of the network. The typical values for γ are 10^{-3} to 10^{-5}. Momentum updating and weight decay may also be used together in learning phase. Other than these simple variations, many new techniques based on gradient-descent approach are proposed such as the ones in (Salomon (1996), Yu (1997), Saad (1998), Hagan (1994), Yu (1995)).

There also exist learning rules based on second order Taylor expansion as conjugate gradient, quasi-Newton and Levenberg-Marquardt optimization methods (Shepherd (1996)). These techniques circumvent the step size problem associated with gradient descent techniques. In this thesis work, variations of gradient descent, conjugate gradient and Levenberg-Marquardt learning algorithms are comparatively used in Chapter 3.

APPENDIX F

NONLINEAR AND CHAOTIC DYNAMICS

Most real world problems confronting the scientists and engineers are neither linear nor even slightly linear. This prevents the analyst to employ linear system tools such as reaching a closed form analytical solution with the knowledge of eigenvalues and eigenvectors. Consequently, with the lack of such analytical tools numerical analyses play a crucial role in analyzing nonlinear phenomena (Parker (1987)).

In recent years, increase in the cost effective computing power let scientists to investigate nonlinear behaviors of dynamical systems. The increased interest in nonlinear systems is also due to the striking discovery of chaotic behavior with the work of Lorenz in 1963. Lorenz has shown that even a simple nonlinear deterministic system may have very complex behavior for certain values of parameters and exhibit chaos.

Though, chaos is accepted to be a state with complete absence of order and loss of control, many scientists have achieved controlling and ordering chaos (Chen (1993)). As a result, a chaotic system can be defined as one in which long term prediction of the system is impossible because of the omnipresent uncertainty in determining its initial state that grows exponentially fast in time.

F.1 Dynamical Systems

There exist two types of dynamical systems with respect to the variables they depend on. These are autonomous and nonautonomous. An autonomous dynamical system is defined with a time implicit following differential equation:

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}\right), \quad \mathbf{x}\left(t_0\right) = \mathbf{x}_0 \tag{F.1}$$

where **f** is nonlinear time independent function of **x** and called vector field. The initial time of the system may always be taken as $t_0 = 0$. The solution of the system with initial condition $\mathbf{x}_0$ is called a trajectory and shown by $\boldsymbol{\varphi}_t\left(\mathbf{x}_0\right)$.

A nonautonomous dynamical system is defined by a time dependent differential equation:

$$\dot{\mathbf{x}} = \mathbf{f}\left(\mathbf{x}, t\right), \quad \mathbf{x}\left(t_0\right) = \mathbf{x}_0 \tag{F.2}$$

f is a function of **x** and **t**, such that the system is time varying. Thus, the initial time of the system cannot be assigned arbitrarily to 0. Trajectory of the system is also time dependent and denoted by $\boldsymbol{\varphi}_t\left(\mathbf{x}_0, t_0\right)$.

Dynamical systems can be classified in terms of their steady state behaviour. Steady state refers to the system behavior as the transients die out. In steady state, four different types of attractors can be defined: point (equilibrium) attractor, periodic (cyclic) attractor, quasi-periodic attractor and chaotic (strange) attractor. Attractors are the regions or set of points generated from initial conditions to $t \to \infty$.

An equilibrium point $\mathbf{x}_e$ of an autonomous system is a solution of $\boldsymbol{\varphi}_t\left(\mathbf{x}_e\right) = \mathbf{x}_e$. At this point, the vector field $\mathbf{f}\left(\mathbf{x}\right)$ vanishes.

In periodic attractors, trajectory of the system passes through the same points within a time period such that

$$\boldsymbol{\varphi}_t\left(\mathbf{x}', t_0\right) = \boldsymbol{\varphi}_{t+T}\left(\mathbf{x}', t_0\right) \tag{F.3}$$

A periodic solution has a Fourier transform with a fundamental frequency component at *1/T* and at its evenly spaced harmonics *k/T*, *k=2,3….*

A quasi-periodic attractor has a solution that can be considered as the sum of periodic solutions having different frequencies such that their ratio is irrational.

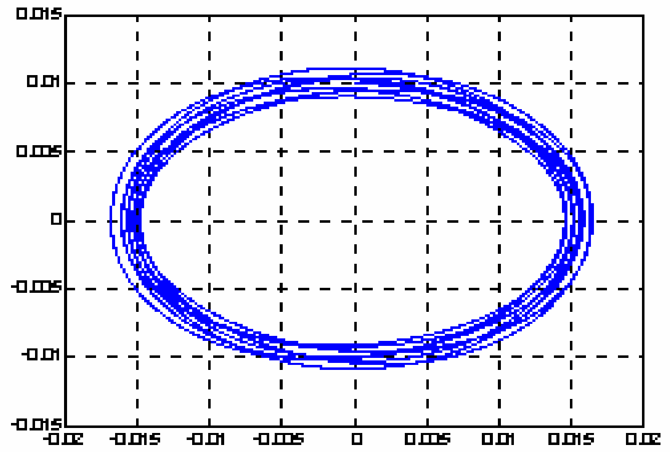

Figure F-1 Periodic Attractor

Chaotic (strange) attractor can be defined as bounded steady state behavior that is not an equilibrium point, not periodic and not quasi-periodic (Figure F-2).

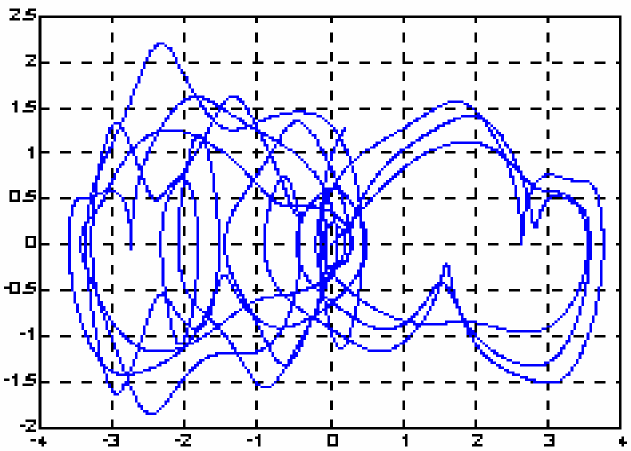

Figure F-2 Chaotic Attractor

F.2 Chaos and Lyapunov Exponents

Chaos can be defined as a bounded steady state behavior that is not an equilibrium point, not periodic and not quasi-periodic. A chaotic spectrum is not composed of discrete frequencies as in periodic and quasi-periodic behavior and has a noise like continuous nature. Dimension of the chaotic attraction domain is not integer, but fractional. The attraction domain always has fractal boundary, thus it possesses fractal dimension.

The most important property of the chaotic systems is their sensitivity to initial conditions. Starting from two different initial conditions arbitrarily close to one another, the trajectories diverge at a rate characteristic of the instability of the system until they become uncorrelated. Thus, the long term behavior of the system cannot be predicted.

On the other hand, instability in a chaotic trajectory is bounded which implies that a chaotic system must contract in some directions and expand in others with contraction outweighing expansion. Lyapunov exponents are used to quantify the expansion and contraction occurring in a dynamical system and are a generalization of the eigenvalues at an equilibrium point and of eigenvectors. A positive exponent corresponds to an expansion in the related eigenvector direction and a negative exponent relates to contraction. They are used to determine the stability of any type of steady state behaviour. If the cumulative of Lyapunov exponents in a system is negative, contraction outweights expansion and the corresponding trajectory is said to be bounded as in chaotic attractors. What distinguishes a strange attractor from the other types of attractor is the existence of at least one positive Lyapunov exponent. Lyapunov exponents are related to the Jacobian of the local linearized model of any nonlinear system, which is a two dimensional error propagation equation:

$$\delta\mathbf{X}_{n+1} = \mathbf{A}_n \delta\mathbf{X}_n \tag{F.4}$$

where

$$\mathbf{A}_n = \begin{pmatrix} \dfrac{df}{dx_n} & \dfrac{df}{dy_n} \\ \dfrac{dg}{dx_n} & \dfrac{dg}{dy_n} \end{pmatrix} \tag{F.5}$$

the Jacobi matrix $\mathbf{J}_n$ relates $\delta\mathbf{X}_n$ to the initial condition:

$$\delta\mathbf{X}_n = \mathbf{A}_{n-1}\mathbf{A}_{n-2}......\mathbf{A}_0\delta\mathbf{X}_0 = \mathbf{J}_n\delta\mathbf{X}_0 \tag{F.6}$$

and is evaluated for each iteration on the exact trajectory.

$\delta\mathbf{X}_n$ can be written in terms of eigenvalues $\mu_i(n)$ and eigenvectors $\mathbf{e}_i(n)$ of $\mathbf{J}_n$ as:

$$\delta\mathbf{X}_n = \mathbf{J}_n\delta\mathbf{X}_0 = \sum_i c_i\mu_i(n)\mathbf{e}_i(n) \tag{F.7}$$

the Lyapunov exponents are then defined as

$$\lambda_i \approx \frac{1}{n} \ln \mu_i(n), \qquad n \to \infty \tag{F.8}$$

$\lambda_i > 0$ corresponds to a local expansion of small areas along e_i while $\lambda_i < 0$ relates to a local contraction of small areas along the corresponding eigenvector. A strange attractor should have at least one positive eigenvalue (Lyapunov exponent) to be chaotic.

Chaos which arises due to the changes in system parameters have found to have 4 basic onsets defining characteristic transition phase triggering different routes to chaos as: intermittency (Pomeau, (1980)), period-doubling (Feigenbaum, (1978)), the quasiperiodic-chaotic route (Ruelle, (1971)), the crisis route (Gregobi, (1983)). The common property of all routes to chaos is the change in the maximum Lyapunov exponents due to the change in system control parameter, η. If η is the control parameter of a system and η_c is the critical value at the transition phase then λ_{max} has a value of nearly $/\eta - \eta_c/^{\alpha}$ where η is in the vicinity of η_c and $\lambda_{max} < 0$ for prechaotic phase while $\lambda_{max} > 0$ at the onset of chaos. As seen, the maximum Lyapunov exponent is the unique measure to determine the "stable to chaotic" transition phase. As the largest exponent approaches zero, the system tends towards chaos. Thus, a strange attractor and transition to chaos is determined using the maximum Lyapunov exponent of a system.

Another important feature of the strange attractor is its fractal dimension. The dimension of an attractor can be defined as a deterministic or probabilistic measure. There are several definitions used to find out the fractal dimension of a strange attractor such as capacity (Hausdorf), Lyapunov, information and correlation dimensions.

The capacity is a deterministic dimension defined as

$$d_c = \lim_{\varepsilon \to 0} \frac{\log N(\varepsilon)}{\log(1/\varepsilon)} \tag{F.9}$$

where N(ε) is the minimum number of cubes (volume elements) of size ε needed to cover the set. The region of state space is divided up into a grid of cubes of size ε, the equations are iterated and the number of cubes in the convergence region are counted. If for very small ε's (as ε goes to 0) the process is repeated, slope of the plot of log *N(ε)* versus log ε gives the fractal dimension of the attractor. In the literature, there exist other dimensions and a detailed analysis of the capacity dimesion and other probabilistic dimensions is given in (Parker (1987)).

Another dimension is the Lyapunov dimension. Let the Lyapunov exponents of a chaotic attractor be $\lambda_1 \geq ... \geq \lambda_n$ and k be the largest integer such that $\lambda_1 + ... + \lambda_k \geq 0$. Then the Lyapunov dimension is given as

$$d_L = k + \frac{\lambda_1 + ... + \lambda_k}{\left|\lambda_{k+1}\right|} \tag{F.10}$$

Information dimension is one of the probabilistic dimensions and utilizes the time behavior of a system such that its calculation is based on the frequency of entrance of the trajectory into a specific volume in the state space. The information dimension is

$$d_i = \lim_{\varepsilon \to 0} \frac{\log S(\varepsilon)}{\log(1/\varepsilon)}, \qquad S(\varepsilon) = -\sum_{k=1}^{N(\varepsilon)} P_k \ln P_k \tag{F.11}$$

where P_k is the frequency with which a trajectory visits k^th^ volume of the state space.

Lastly, the correlation dimension is defined as

$$d_{cor} = \lim_{\varepsilon \to 0} \frac{\ln \sum_{k=1}^{N(\varepsilon)} P_k^2}{\ln \varepsilon} \tag{F.12}$$

and uses again the frequency of entrance to a specific volume of the state space.

Every definition contains its own pros and cons. Although, the capacity definition is simple to understand, it takes the strange attractor as a static object and requires infinite time to estimate the true picture of the attractor, hence its dimension. The Lyapunov dimension is based on Lyapunov exponents and they are easily found using simulation technique. But there is no way to estimate negative exponents in a real time system from time series measurements. As a result, neither capacity nor Lyapunov dimensions are suitable for experimental studies.

Among these four definitions, the correlation and the information dimensions are more suitable for experimental studies. They use the information on the frequency of a trajectory to be in some volume element.

In this thesis study, the land vehicle system is analyzed using simulation tools so capacity and Lyapunov dimensions are utilized in the analysis of Chapter 5 and 6.

APPENDIX G

ROUGH SETS

In the dynamical analysis of the vehicle on irregular roads, we exploit rough set theory for categorizing state space regions with different stability features according to their degrees of uncertainty. Rough set proposed by Zdzislaw Pawlak (Pawlak (1982), (1995), (1997)) is a mathematical approach for modelling vagueness in uncertainty. Rough set theory associates a degree of uncertainty to any object of the universe so that any object without uncertainty (roughness) becomes indiscernible within a class.

A set containing indiscernible objects forms or equivalence class, called the elementary set, any union of which determines a crisp set. All other sets include uncertainty. The rough set is composed of such sets and has 1) a boundary region where vagueness resides, with set elements that cannot be classified either as members of that set or of the complementary set; 2) a lower approximation consisting of the elements that are surely the members of the set and 3) an upper approximation consisting of the possible members of the set.

Approximation Space

Let U be a certain set called universum, and let R be an equivalence relation (indiscernibility relation) on U. The pair $A=(U,R)$ is called an approximation space (Pawlak (1982)) which is formed by the classification of data points into disjoint categories using attributes. The classification represents our knowledge about the data.

Lower approximation of a set

The ***lower approximation*** of a set *X* is described by the objects *x* of the domain (*U*), which are known "with certainty" to belong to the set of interest with respect to the attribute *B*:

$$\underline{B}(X) = \left\{ x \in U : B(x) \subseteq X \right\} \tag{G.1}$$

Upper approximation of a set

The ***upper approximation*** of a set *X* containing objects *x* which "possibly" belong to the set of interest with respect to the attribute *B*:

$$\overline{B}(X) = \left\{ x \in U : B(x) \cap X \neq \phi \right\} \tag{G.2}$$

Boundary of a rough set

The ***boundary region*** of a rough set is a region of uncertainty where the set elements of that region are not known to be inside or outside the set "with certainty" with respect to the attribute *B*:

$$BN_B(X) = \overline{B}(X) - \underline{B}(X) \tag{G.3}$$

APPENDIX H

REINFORCEMENT LEARNING

Reinforcement learning is the technique of training an agent by reward and punishment through an environmental feedback process in order to learn the dynamical characteristics of a system navigating within this environment. Allowing systems to rearrange and improve themselves with their past space, it has been considerably popular in the field of robotics and autonomous systems (Mahadewan (1991), Schaal (1994)).

A critical issue in improvement efficiency is the learning method employed in the training. There exist a number of methods in the literature that can be classified in two main groups as model free and model based learning. While model free learning forces the system to learn a controller without learning a model, model based learning makes the system learn a model in order to derive a controller. The most popular model free learning methods are adaptive heuristic critic introduced by Barto *et.al*. (1983), Q-learning of Watkins (1992) and learning with average reward examined first by Schwartz (1993). The fundamental model based methods are certainty equivalence (Kumar (1986)), Sutton's Dyna architecture (1991) and prioritized sweeping technique proposed by Moore and Atkenson (1993). Many variants of given methods are also employed in different applications in the literature.

Having a simple problem with two different states (chaoticity and regularity), we employed the *linear reward-inaction* algorithm of Hilgard and Bower (1975), which is simpler and more suitable than the complex methods mentioned above, for reinforcement. This easy to apply learning algorithm based on finite-state automata adjusts the probability of taking different actions according to the previous successes and failures. When the success is rewarded, the measure is reinforced as

$$R_n^i(k+1) = R_n^i(k) + \alpha\left(1 - R_n^i(k)\right) \tag{H.1}$$

When a failure occurs, the predetermined measure is penalized by

$$R_n^i(k+1) = R_n^i(k) - \alpha R_n^i(k) \tag{H.2}$$

where R_n is the measure of success/failure and α determines the convergence rate.

VITA

H. Burak Kaygısız received a B.S. degree in electrical engineering from Middle East Technical University, Ankara, TURKEY in 1996, and M.S. degree in electrical engineering from Illinois Institute of Technology, Chicago, IL in 1998. He is presently a research engineer and member of the technical staff of GNC group of TUBİTAK-SAGE. His present areas of interest are guidance, control and inertial navigation and integrated navigation systems.

Publications of the author during his Ph.D. study includes:

- Kaygısız B., "GPS simülasyonu ve hamveri kullanımı", Kayseri III. Havacılık Sempozyumu, May 2000
- Kaygısız B., Erkmen A.M., Erkmen I., "Smoothing stability roughness of a robot arm under dynamic load using reinforcement learning", IEEE/RJS International Conference on Intelligent Robots and Systems, 2001
- Kaygısız B., Erkmen A.M., Erkmen I., "Smoothing stability roughness of fractal boundaries using reinforcement learning", in Proceedings of IFAC 15th World Congress, 2002
- Kaygısız B., Erkmen A.M., Erkmen I., "Detection of transition to chaos during stability roughness smoothing of a robot arm", IEEE/RJS International Conference on Intelligent Robots and Systems, 2002
- Kaygısız B., Gökpınar S., "GPS/INS guided field artillery munitions", AFCEA Proceedings, May 2003
- Kaygısız B., Erkmen A.M., Erkmen İ., "GPS/INS enhancement using neural networks for autonomous ground vehicle applications", IEEE/RJS International Conference on Intelligent Robots and Systems, 2003

- Kaygısız B., Erkmen İ., Erkmen A.M., “GPS/INS enhancement using neural network”, Journal of Navigation, vol.57, no.2, 2004
- Kaygısız B., Erkmen A.M., Erkmen İ., “Detecting onset of chaos and intelligent operational region enlargement (Invited Paper)”, International Journal of Computational Cognition, vol. 3, no.2, 2005
- Kaygısız B., Erkmen A.M., Erkmen İ., “Intelligent compensation of chaos roughness in regularity of walk for a two legged robot”, Journal of Robotic Systems, submitted in 2004