A REAL TIME TEST SETUP DESIGN AND REALIZATION FOR PERFORMANCE VERIFICATION OF CONTROLLER DESIGNS FOR UNMANNED AIR VEHICLES

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

BY

FUNDA KÜREKSİZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

FEBRUARY 2008

Approval of the thesis:

A REAL TIME TEST SETUP DESIGN AND REALIZATION FOR PERFORMANCE VERIFICATION OF CONTROLLER DESIGNS FOR UNMANNED AIR VEHICLES

Submitted by **FUNDA KÜREKSİZ** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

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

Prof. Dr. S. Kemal İder
Head of Department, **Mechanical Engineering Dept., METU** _____

Prof. Dr. Bülent E. Platin
Supervisor, **Mechanical Engineering Dept., METU** _____

Prof. Dr. Tuna Balkan
Co-Supervisor, **Mechanical Engineering Dept., METU** _____

Examining Committee Members

Prof. Dr. Samim Ünlüsoy
Mechanical Engineering Dept., METU _____

Prof. Dr. Bülent E. Platin
Supervisor, Mechanical Engineering Dept., METU _____

Prof. Dr. Tuna Balkan
Co-Supervisor, Mechanical Engineering Dept., METU _____

Asst. Prof. Dr. İlhan Konukseven
Mechanical Engineering Dept., METU _____

Özgür Erinmez
Avionics and Electrical Systems Dept., TAI _____

Date: 08 February 2008

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

Name, Last name: Funda KÜREKSİZ

Signature :

ABSTRACT

A REAL TIME TEST SETUP DESIGN AND REALIZATION FOR PERFORMANCE VERIFICATION OF CONTROLLER DESIGNS FOR UNMANNED AIR VEHICLES

Küreksiz, Funda

M.S., Department of Mechanical Engineering

Supervisor: Prof. Dr. Bülent E. Platin

Co-Supervisor: Prof. Dr. Tuna Balkan

January 2008, 164 pages

In this thesis, a test platform based on real-time facilities and embedded software is designed to verify the performance of a controller model in real time. By the help of this platform, design errors can be detected earlier and possible problems can be solved cost-effectively without interrupting the development process.

An unmanned combat air vehicle (UCAV) model is taken as a plant model due to its importance in current and future military operations. Among several autopilot modes, the altitude hold mode is selected since it is an important pilot-relief mode and widely used in aviation. A discrete PID controller is designed in MATLAB/Simulink environment for using in verification studies. To control the dynamic system in wide range, a gain scheduling is employed where the altitude and velocity are taken as scheduling variables. Codes for plant and controller model are obtained by using real time workshop embedded coder (RTWEC) and downloaded to two separate computers, in which xPC kernel and VxWorks operating system are run, respectively.

A set of flight test scenarios are generated in Simulink environment. They are analyzed, discussed, and then some of them are picked up to verify the platform. These test scenarios are run in the setup and their results are compared with the ones obtained in Simulink environment.

The reusability of the platform is verified by using a commercial aircraft, Boeing 747, and its controller models. The test results obtained in the setup and in Simulink environment are presented and discussed.

Keywords: Unmanned combat air vehicle, discrete PID controller, rapid control prototyping, real-time control application, embedded software application

ÖZ

İNSANSIZ HAVA ARAÇLARININ DENETLEYİCİ SİSTEMLERİNİN TASARIMLARININ PERFORMANS DOĞRULAMASI İÇİN GERÇEK ZAMANLI BİR TEST DÜZENEĞİNİN TASARIMI VE GERÇEKLEŞTİRİLMESİ

Küreksiz, Funda

Yüksek Lisans, Makina Mühendisliği Bölümü

Tez Yöneticisi : Prof. Dr. Bülent E. Platin

Ortak Tez Yöneticisi: Prof. Dr. Tuna Balkan

Şubat 2008, 164 sayfa

Bu tez çalışmasında, denetleyici modelinin performansını doğrulamak için gerçek zamanlılık ve gömülü yazılım özelliklerini temel alan bir düzenek tasarlanmıştır. Bu düzeneğin yardımıyla, tasarım hataları erken fark edilebilecek ve olası sorunlar geliştirme işlemini bölmeden uygun maliyetli olarak çözümlenebilecektir.

Günümüzdeki ve gelecekteki askeri operasyonlardaki öneminden dolayı uçuş modeli olarak bir insansız savaş hava aracı (ISHA) modeli alınmıştır. Birçok otopilot modu arasında pilotlar için önemli bir mod olduğu ve havacılıkta geniş kullanımı olduğu için yükseklik tutma modu seçilmiştir. Doğrulama çalışmalarında kullanılmak üzere bir ayrık zamanlı PID denetleyici modeli MATLAB/Simulink ortamında tasarlanmıştır. Dinamik sistemi geniş aralıkta kontrol etmek için yükseklik ve hızın değişken olarak kullanıldığı bir kazanç tablosu oluşturulmuştur. RTWEC ile elde edilen uçuş modeli kodu xPC kernel'ın koştuğu, denetleyici modelinin kodu da VxWorks işletim sisteminin koştuğu iki ayrı bilgisayara yüklenmiştir.

Simulink ortamında bir dizi uçuş test senaryoları oluşturulmuştur. Bu senaryolar analiz edilip yorumlandıktan sonra bazıları düzeneği doğrulamak için seçilmiştir. Seçilen test senaryoları düzenekte çalıştırılmış ve sonuçları Simulink ortamında elde edilen sonuçlar ile karsılaştırılmıştır.

Düzeneğin tekrar kullanılabilirliği ticari bir uçak olan Boeing 747 ve onun denetleyici modeli kullanılarak doğrulanmıştır. Bu düzenekte ve Simulink ortamında elde edilen test sonuçları sunulmuş ve yorumlanmıştır.

Anahtar Kelimeler: İnsansız savaş hava araçları, ayrık PID denetleyici modeli, hızlı denetleyici prototipleme, gerçek zamanlı kontrol uygulamaları, gömülü yazılım uygulamaları

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor Prof. Dr. Bülent E. Platin and co-supervisor Prof. Dr. Tuna Balkan for their guidance, encouragements, advice, criticism, and insight throughout the research.

I would also like to express my deepest thanks and gratitude to Assoc. Prof. Dr. John Valasek, Texas A&M University, for his useful suggestions and great supports.

I would like to thank to İbrahim Keskiner, Avionics and Electric Systems Manager of TAI Inc., for his tolerance and permission for using the TAI Avionic System Laboratory equipments and I would like to thank my colleagues from TAI, Ahmet Gürbüz and Şeyma Kaplan for their technical assistance and useful discussions.

I would kindly like to thank to Middle East Technical University Mechanical Engineering Department for supplying a Brushless DC Motor with its amplifier and encoder at no cost.

I would specially like to express my thanks to Mehtap Tüysüz and Özge Küreksiz for their useful discussions and patience. In addition, I would like to express my deepest thanks to Dr. Özgür Öner for his encouragements and advice.

This study was supported by Turkish Aerospace Industries, Inc.

To my parents and my sister for their endless love, patience and support

TABLE OF CONTENTS

ABSTRACT iv

ÖZ vi

ACKNOWLEDGMENTS viii

TABLE OF CONTENTS x

LIST OF TABLES xv

LIST OF FIGURES xvii

LIST OF SYMBOLS xxiii

CHAPTERS

1. INTRODUCTION 1

1.1 Introduction 1

1.2 Literature Survey 1

1.2.1 Conclusion of Literature Survey 8

1.3 Objectives of the Thesis 9

1.4 Outline of the Thesis 10

2. DEVELOPED PLATFORM AND ITS COMPONENTS 11

2.1 Test Platform 11

2.1.1 The Need for a Test Platform 11

2.1.2 Platform Facilities and Advantages 12

2.2 A Conceptual Architecture of the Platform..... 14

2.3 Plant..... 17

2.3.1 Interfaces between Actuator and Plant..... 19

2.3.2 Interfaces between Plant and Controller 19

2.3.3 Plant Software Requirements 26

2.4 Controller 26

2.4.1 Interfaces between Controller and Actuator 29

2.4.2 Controller Software Requirements..... 30

2.5 Host PC 33

2.5.1 Interface between Host PC and Plant..... 34

2.5.2 Interfaces between Host PC and Controller 34

2.5.3 Host PC Software Requirements..... 34

2.6 Actuator 35

2.7 Final form of the Platform..... 37

2.8 Concluding Remarks 40

3. UCAV6 CHARACTERISTICS 41

3.1 Brief Information about UCAV 41

3.2 Rationale Behind Choosing UCAV6 as the Plant Model 43

3.3 UCAV6 Linear Model..... 44

3.4 Concluding Remarks 48

4. PLANT MODELING AND CONTROLLER SYSTEM DESIGN 49

4.1 Reference Coordinate Systems 49

4.1.1 Earth-Fixed Coordinate System 49

4.1.2 Body-Fixed Coordinate System 49

4.2 Transfer Functions for Altitude Hold Mode 50

4.2.1 Longitudinal Equations of Motion 50

4.2.2 System Matrices of Longitudinal Motion 51

4.2.3 Transfer Functions of Longitudinal Dynamics 52

4.2.4 Transfer Functions between Altitude and Elevator Deflection 53

4.3 Plant Model Design 55

4.4 Controller Design 63

4.4.1 Gain Scheduling 74

4.5 Concluding Remarks 76

5. PLATFORM TESTS AND RESULTS 77

5.1 Test Scenarios 77

5.1.1 Test Scenarios at 100 ft 78

5.1.2 Test Scenarios at 5,000 ft 79

5.2 Platform Tests 80

5.2.1 Comparative Test Results at 100 ft 82

5.2.2 Comparative Test Results at 5,000 ft 83

5.3 Concluding Remarks 90

6. VERIFICATION OF THE PLATFORM REUSABILITY 91

6.1 Boeing 747 Test in the Platform 94

6.2 Concluding Remarks 97

7. DISCUSSION AND CONCLUSIONS 98

7.1 Summary and Discussion 98

7.2 Conclusions 100

7.3 Recommendation for Future Work 101

REFERENCES 102

APPENDICES

A. COMPONENT SPECIFICATIONS of THE TEST PLATFORM 112

B. INTEGRATION OF THE PLATFORM 115

C. AV-8B HARRIER II SPECIFICATIONS 133

D. STABILITY DERIVATIVES 136

E. UCAV6 PLANT PROPERTIES 138

F. F-4, F-104, LEARJET24 PLANT PROPERTIES 140

G. R/C AIRCRAFT PLANT PROPERTIES 144

H. NORMALIZED ALTITUDE TO ELEVATOR DEFLECTION TRANSFER FUNCTIONS FOR UCAV6 146

I. DIRECT FORM II 148

J. TEST SCENARIOS- DATA STATISTICS 150

K. BOEING 747 PLATFORM TEST RESULTS..... 153

LIST OF TABLES

TABLES

Table 2.1 Alternative models of Sensoray 626 I/O board..... 19

Table 2.2 Real time operating system's advantage/disadvantages 32

Table 2.3 Characteristics of the DC-servomotor..... 36

Table 2.4 Characteristics of the gearbox..... 36

Table 2.5 Specific characteristics of the servo amplifier 37

Table 2.6 Properties of the quadrature encoder 37

Table 3.1 Advantages of future UCAV's over manned aircraft [20] 42

Table 3.2 Future UCAV concerns [20]..... 43

Table 3.3 UCAV6 variables with their descriptions and magnitudes [72] 46

Table 3.4 Description of the flight conditions [72]..... 47

Table 3.5 Description of UCAV6 trim variables [72]..... 47

Table 4.1 Short period damping ratio limits [77]..... 59

Table 4.2 Short period and phugoid mode properties..... 63

Table 4.3 Characteristics of PID controller..... 64

Table 4.4 Continuous PID controller gains at each operating point 67

Table 4.5 Discrete PID controller gains at all operating points 71

Table A.1 Minimum hardware requirements for the host PC [21] 112

Table A.2 Software components on host PC 112

Table A.3 Minimum xPC target hardware requirements [21] 113

Table B.1 Connection diagram between servo amplifier and xPC I/O card..... 126

Table B.2 Connection diagram between servo amplifier and power supply 126

Table C.1 AV-8B Harrier II specifications [75] & [76]..... 133

Table D.1 Meanings, expressions and units of stability derivatives [64] 136

Table E.1 UCAV6 plant properties..... 138

Table F.1 Plant properties of F-104 141

Table F.2 Plant properties of F-4 142

Table F.3 Plant properties of Learjet24..... 143

Table G.1 Plant properties of RC aircraft at different velocity_20m/s 144

Table G.2 Plant properties of RC aircraft at different velocity_23.5m/s 145

Table G.3 Plant properties of RC aircraft at different velocity_27m/s 145

Table J.1 Data statistics of flight plans 150

Table J.2 Data statistics of plant output at 100 ft..... 150

Table J.3 Data statistics of plant output at 5,000 ft with increasing airspeed..... 151

Table J.4 Data statistics of plant output at 5,000 ft with decreasing airspeed 151

Table K.1 Condition of Boeing 747 discrete model 153

LIST OF FIGURES

FIGURES

Figure 1.1 Control setup of NPS FROG UAV [5] 2

Figure 1.2 Design and integration process of R-Max UAV [6] 4

Figure 1.3 $Matrix_x$ product family [7] 5

Figure 1.4 The proposed control system architecture for robotic systems [10] 7

Figure 2.1 Conceptual architecture of the platform 16

Figure 2.2 xPC target view on the platform 17

Figure 2.3 Sensoray 626 I/O board 18

Figure 2.4 A Conceptual model of MIL-STD-1553B bus [87] 22

Figure 2.5 Aitech C772 VME single board computer view 27

Figure 2.6 Aitech C431 A/D, D/A and digital I/O VME board 28

Figure 2.7 The platform view with minimum software packages 38

Figure 2.8 Desired HIL test setup 39

Figure 2.9 Data flow among the controller, plant, and actuator 39

Figure 2.10 Final architecture of the test platform 40

Figure 3.1 Kettering Bug, early UAV [11] 41

Figure 3.2 Atlantic Ocean (Nov. 18, 2004) An AV-8B II Harrier [76] 45

Figure 3.3 UCAV6 mission profile [72] 45

Figure 4.1 Flight path geometry 54

Figure 4.2 Root locus diagram at operating point 8 57

Figure 4.3 The sign convention of the control surface deflection [79] 60

Figure 4.4 UCAV6 plant model in direct II form 62

Figure 4.5 PID gain parameter tuning at operating point 8 66

Figure 4.6 Plant output signal response at operating point 8 67

Figure 4.7 Continuous PID gains for the entire operating points 68

Figure 4.8 Continuous PID gains at 100 ft 68

Figure 4.9 Continuous PID gains at 5,000 ft 69

Figure 4.10 Error message taken when coding continuous PID controller 69

Figure 4.11 Discrete PID gain parameter tuning at operating point 8 72

Figure 4.12 Adjusting solver configuration 72

Figure 4.13 Discrete PID controller gains at entire operating points 73

Figure 4.14 Discrete PID controller gains at 100 ft 73

Figure 4.15 Discrete PID controller gains at 5,000 ft 74

Figure 4.16 UCAV6 discrete PID controller with gain scheduling 76

Figure 5.1 Velocity signals input at 100 ft 78

Figure 5.2 UCAV6 altitude at 100 ft test in Simulink 79

Figure 5.3 Velocity Signals at 5,000 ft 79

Figure 5.4 UCAV6 altitude result at 5,000 ft test in Simulink 80

Figure 5.5 The platform view for UCAV6 controller test 80

Figure 5.6 The Simulink diagram of the controller..... 81

Figure 5.7 The Simulink diagram of the plant 81

Figure 5.8 UCAV6 altitude in both setup and Simulink for 100 ft..... 82

Figure 5.9 The difference between UCAV6 altitude 83

Figure 5.10 UCAV6 altitude in both setup and Simulink tests for 5,000 ft..... 84

Figure 5.11 Attitude of the plant during platform test while airspeed increasing.... 85

Figure 5.12 Attitude of the plant during platform test (Zoomed) 85

Figure 5.13 The difference of the altitude in between Simulink and platform tests. 86

Figure 5.14 Attitude of plant during platform test 1 while airspeed decreasing..... 86

Figure 5.15 Attitude of plant during platform test 2 while airspeed..... 87

Figure 5.16 Difference in altitudes between in Simulink and Platform tests 1..... 87

Figure 5.17 Difference in altitudes between in Simulink and Platform tests 2..... 88

Figure 5.18 VxWorks RS-232 connection 89

Figure 6.1 Boeing 747 entire model sketch 91

Figure 6.2 Boeing 747 plant model..... 92

Figure 6.3 Boeing 747 controller model 92

Figure 6.4 The test platform block diagram for testing Boeing 747 model..... 93

Figure 6.5 Boeing 747 test platform view test 94

Figure 6.6 Altitude during platform and Simulink tests 95

Figure 6.7 Altitude difference in platform and Simulink tests..... 96

Figure B.1 xPC target explorer window 115

Figure B.2 Host PC adjustments 116

Figure B.3 Host PC adjustments – DLM Menu 116

Figure B.4 xPC target bootdisc menu 117

Figure B.5 xPC target communication menu 118

Figure B.6 xPC target settings 119

Figure B.7 xPC target appearance..... 119

Figure B.8 xPC target file system 120

Figure B.9 xPC target PCI devices 121

Figure B.10 xPC target RTW adjustments..... 121

Figure B.11 xPC target options..... 122

Figure B.12 Simulink block: checking Sensoray 626 D/A converter..... 123

Figure B.13 Connection diagram of BLD 5606-SE4P servo amplifier 126

Figure B.14 DC Motor speed monitor test model..... 127

Figure B15 DC motor speed monitor output versus time 128

Figure B.16 Tornado Integrated Development Environment screenshot..... 131

Figure B.17 C772 console on host PC screenshot 132

Figure C.1 3-D view of Harrier II at the same time UCAV6 [72] 135

Figure F.1 3-D view of F-104 140

Figure F.2 3-D view of F-4 141

Figure F.3 3-D view of Learjet24 142

Figure I.1 The block diagram for $\frac{u(z)}{h(z)}$ 148

Figure I.2 The block diagram for $\frac{h(z)}{e(z)}$ 149

Figure I.3 The block diagram for $\frac{u(z)}{e(z)}$ 149

Figure J.1 Obtaining data statistics of the plot 151

Figure J.2 Data statistics window 152

Figure J.3 Saving data statistics results in workspace 152

Figure K.1 Airspeed during platform and Simulink tests 154

Figure K.2 Difference in airspeed between platform and Simulink tests 154

Figure K.3 Angle of attack, α, during platform and Simulink tests 155

Figure K.4 Difference in angle of attack, α, between platform and Simulink tests155

Figure K.5 Sideslip angle, β, during platform and Simulink tests 156

Figure K.6 Difference in sideslip angle, β, between platform and Simulink tests. 156

Figure K.7 Roll rate, p, during platform and Simulink tests 157

Figure K.8 Difference in roll rate, p, between platform and Simulink tests 157

Figure K.9 Pitch rate, q, during platform and Simulink 158

Figure K.10 Difference in pitch rate, p, between platform and Simulink tests..... 158

Figure K.11 Yaw rate, r, during platform and Simulink..... 159

Figure K.12 Difference in yaw rate, r, between platform and Simulink tests..... 159

Figure K.13 Heading angle, ψ, during platform and Simulink tests..... 160

Figure K.14 Difference in heading angle, ψ, between platform and Simulink tests160

Figure K.15 Pitch angle, θ, during platform and Simulink tests..... 161

Figure K.16 Difference in pitch angle, θ, between platform and Simulink tests... 161

Figure K.17 Bank angle, φ, during platform and Simulink tests 162

Figure K.18 Difference in bank angle, φ, between platform and Simulink tests... 162

Figure K.19 Distance along x axes, X, during platform and Simulink..... 163

Figure K.20 Difference in distance, X, between platform and Simulink tests..... 163

Figure K.21 Distance along y axes, Y, during platform and Simulink..... 164

Figure K.22 Difference in distance, Y, between platform and Simulink tests..... 164

LIST OF SYMBOLS

H	Airplane altitude
I_{xx}, I_{yy}, I_{zz}	Airplane mass moments of inertia about X, Y, and Z axes
K_p	Proportional gain
K_i	Integral gain
K_d	Derivative gain
M	Mach number
M_α	Pitch angular acceleration per unit angle of attack
$M_{\acute{\alpha}}$	Pitch angular acceleration per unit rate of change of angle of attack
$M_{T,\alpha}$	Pitch angular acceleration per unit of angle of attack due to thrust
$M_{T,u}$	Pitch angular acceleration per unit change in speed due to thrust
M_u	Pitch angular acceleration per unit change in speed
M_q	Pitch angular acceleration per unit pitch rate
$M_{\delta e}$	Pitch angular acceleration per unit elevator deflection
N_α	Numerator of angle of attack to elevator transfer function
N_θ	Numerator of pitch attitude to elevator transfer function
P, Q, R	Components of angular velocity about X, Y, and Z axes
p, q, r	Perturbed values of P, Q, R

U, V, W	Components of airplane velocity along X, Y, and Z axes
u, v, w	Perturbed values of U, V, W
X, Y, Z	Axes of earth-fixed coordinate system
x, y, z	Axes of body-fixed coordinate system
X_{α}	Forward acceleration per unit angle of attack
$X_{T,u}$	Forward acceleration per unit change in speed due to thrust
X_u	Forward acceleration per unit change in speed
$X_{\delta e}$	Forward acceleration per unit elevator deflection
Z_{α}	Vertical acceleration per unit angle of attack
$Z_{\dot{\alpha}}$	Vertical acceleration per unit rate of change of angle of attack
Z_u	Vertical acceleration per unit change in speed
Z_q	Vertical acceleration per unit pitch rate
$Z_{\delta e}$	Vertical acceleration per unit elevator deflection

Greek Letters

α	Angle of attack
γ	Flight path angle
λ	Longitude
Θ	Pitch angle

θ	Perturbed value of Θ
δ_e	Elevator deflection angle
ρ	Air density

Acronyms

A/C	Aircraft
A/D	Analog to Digital
ADC	Air Data Computer
API	Application Programming Interface
BSP	Board Support Package
D/A	Digital to Analog
DAC	Digital Analog Converter
DOF	Degree of Freedom
EGI	Embedded GPS INS
FMS	Flight Management System
GPS	Global Positioning System
GUI	Graphical User Interface
HIL	Hardware in the Loop
HUD	Head Up Display

I/O	Input/Output
IMU	Inertial Measurement Unit
LRU	Line Replacement Unit
METU	Middle East Technical University
MFD	Multi Function Display
PID	Proportional-Integral-Derivative
PVM	Pulse Width Modulation
RTWEC	Real Time Workshop Embedded Coder
UAV	Unmanned Air Vehicle
UCAV	Unmanned Combat Air Vehicle
VTOL	Vertical Take-off and Landing

CHAPTER 1

INTRODUCTION

1.1 Introduction

Companies in defense industry have to catch up the developing technology to meet the demand on their products. This may bring the need to specialize in different technical areas at the same time. For example, an aerospace company may need to deal with an unmanned aerial vehicle (UAV) project and a spacecraft project during the same period. These are interesting and technically challenging subjects for which the skills and efforts of several teams in a wide range of engineering disciplines have to be aligned in order to establish a successful system design.

To complete their projects rapidly, they need a test platform, which enables to check the simulation models of their dynamic systems by using the actual hardware. Mostly, these platforms are based on real time facilities. By the help of such platforms, design errors can be detected in the earlier stage of the project, reducing the cost of the project.

1.2 Literature Survey

Unfortunately, there exist not too many earlier studies reported in open literature on such test platforms due to the concerns of companies involved in such studies, with regard to competition, intellectual rights, and confidentiality. Therefore, in this section, only a limited number of such studies are covered, which are available in open literature.

A study about designing and rapid control prototyping of flight control system for an UAV was conducted in the Aeronautical and Astronautical Engineering Department

in the Naval Postgraduate School, USA [5]. They designed an autopilot for their aircraft, explored some suitable navigation algorithms, and assembled an onboard computer to perform the computations for flight control and guidance commands.

They used NPS FROG UAV as a test platform, which was a new 6-DOF FROG UAV model developed in Simulink. Two autopilots were designed, one with classical control techniques and the other with modern control. Even though there was an old flight control setup in Aeronautical and Astronautical Engineering Department in the Naval Postgraduate School, a new flight control setup was designed in that study because of the limitations of the old one. The architecture of the new setup is shown in Figure 1.1.

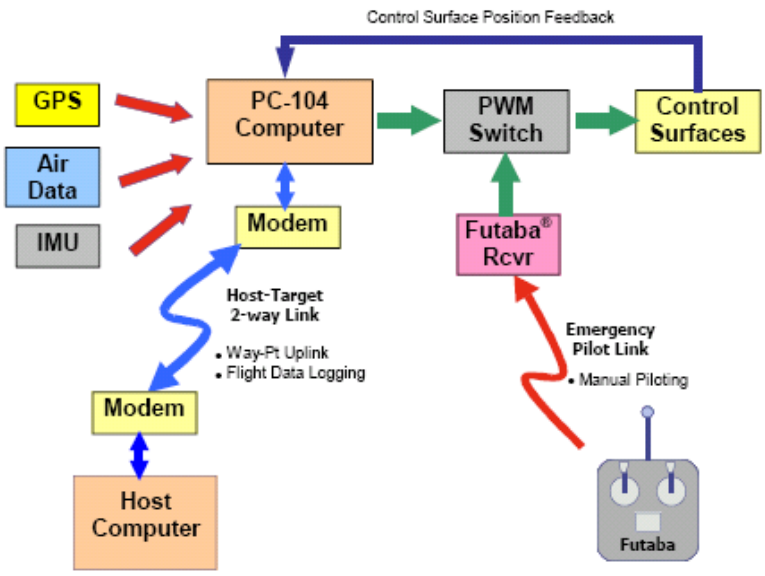

Figure 1.1 Control setup of NPS FROG UAV [5]

The UAV was equipped with an onboard computer that receives commands from the ground station sent over a serial modem and data from the onboard sensors. These sensors were an inertial measurement unit (IMU), a global positioning system (GPS), air data sensors, and control surface potentiometers. The ground station consisted of a computer, a serial modem, and a Futaba® remote control unit. onboard

computer commands can be overridden from ground by a remote control unit via a pulse width modulation (PWM) actuator driver. The guidance, navigation, and control algorithms were developed in MATLAB environment. The onboard computer executes the xPC target kernel that includes the control code to be tested during flight. They specified that the xPC target rapid control prototyping system in their project greatly expedited their development and implementation of the desired control setup.

Several algorithms, such as trajectory tracking, voice control, and integrated infrared/inertial navigation, were flight tested in that study. The results of that study showed that the system was very effective, since it ensured safety and reliability in flight test of the newly developed control algorithms for UAV's.

Another study on the design and implementation of a UAV navigation system was conducted at the School of Aerospace Engineering in Georgia Institute of Technology, USA [6]. In this study, an applied design and integration methodology was used to develop an avionics system as hardware and software packages.

Their test bed was a vertical take-off and landing (VTOL) UAV. R-Max UAV was based on a Yamaha R-Max industrial helicopter, which has six times the avionics payload capability of the X-Cell. Figure 1.2 depicts their realization flow for a complete UAV avionic system.

Different from Yamaha helicopters (R-50 and R-Max), they used multiple sensors to implement an avionics system for redundancy and a significant payload capability for later extensions. To interconnect the components, two different bus interfaces were used. They were a RS-232 serial bus for sensor data and a 100 Mbit ethernet bus for communication between computers.

In that study, the selection of the avionics components and the integration of them into airframe were discussed. ESim was used as a simulation tool in the School of Aerospace Engineering in Georgia Institute of Technology and MATLAB was used to plot every variable in the simulation.

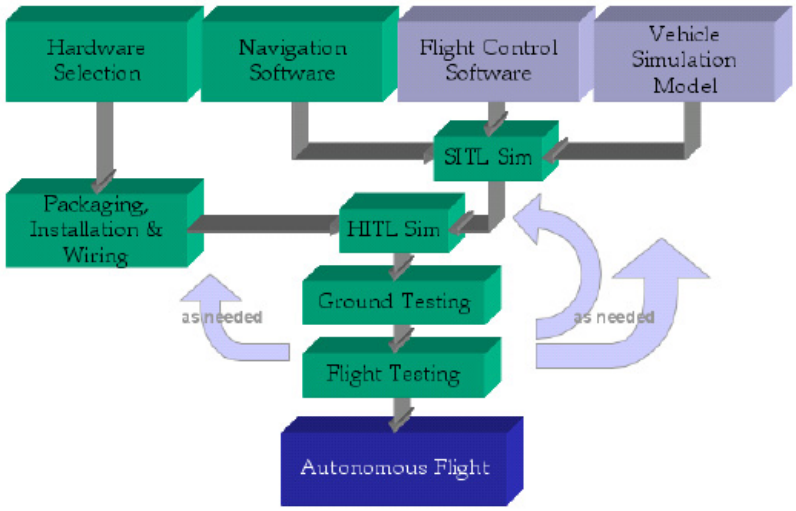

Figure 1.2 Design and integration process of R-Max UAV [6]

The flight code was updated at every other time step at a rate of 50 Hz. The code could be compiled and executed as a stand-alone VxWorks application. Moreover, it could be integrated into the ESim simulation environment. The navigation algorithms were first tested by using software emulated sensor data and then a simulation model of the aircraft. Tests included automatic hover, high-speed maneuvers, and pirouettes, as well as remotely piloted flight (with a human pilot in the simulation loop).

Their navigation system yielded a very good performance and the results were close to those obtained in simulations. At the end of that study, they concluded that by using simulation and sensor data playback, the expensive flight test time to develop the navigation system could be reduced. Their software architecture enabled them some significant time savings during the system development, as well.

Another study, which examined digital controller design process utilizing a rapid control prototyping system, was conducted at Aeronautical Engineering Department in the Naval Postgraduate School, USA [7]. This study was closed to literature at that time.

The purpose of the study was satisfying a background for the "Advance Control of Aerospace Vehicles" course given at Naval Postgraduate School, designing and testing an altitude hold controller for Frog UAV. $Matrix_x$ product family (shown in Figure 1.3) was used for controller design and implementation.

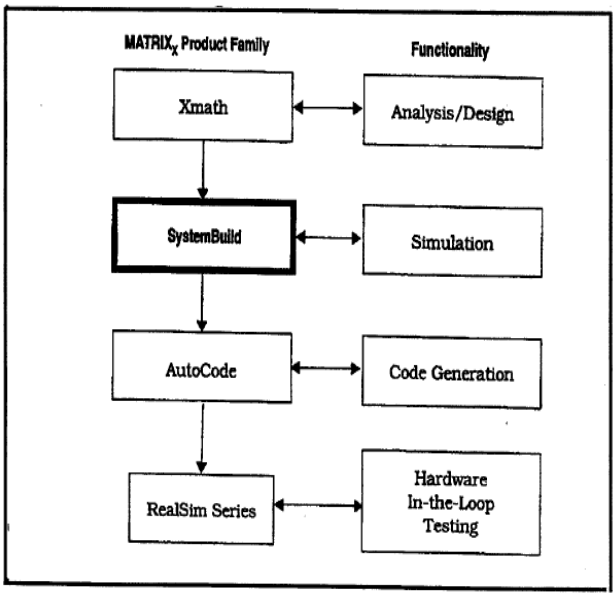

Figure 1.3 $Matrix_x$ product family [7]

The RealSim controller automates the development of the real-time systems by combining graphical modeling software with real time control hardware. The key feature of this product was the AutoCode tool, which automatically programs higher language code such as C or ADA for the designed controller.

The rapid control prototyping architecture of this study consisted of a UNIX workstation and a Windows based host PC. By using a simple speed controller model, $Matrix_x$ production family used in this study was explained as a tutorial. Generally, the controller model was first design in continuous time. After a satisfactory response is achieved, the system was transformed to discrete time and tested again for satisfactory response.

Likewise, in that study, the speed controller was designed in continuous time and analyzed by root locus method. After a satisfactory response was obtained, the system was transformed to discrete time and tested again.

That was a critical study, because the rapid control prototyping system was at its initial stage of development, then. They enabled their students a rapid control prototyping environment to check their control algorithms. In addition to that study [7], Reference 8 gives a digital flight controller design of FROG UAV, the characterization of servo actuator, and integration to the plant model.

Furthermore, another study [9] was conducted also at the Naval Postgraduate School, which was about flight management system (FMS) design, integration, and flight test. The flight management system that was designed in this study was tested by using the test platform designed in the previous study [7]. The study [7] about rapid control prototyping system was complied in 1998 and in 2002; several students completed their theses by using this prototyping system.

Another study [10] closer to this thesis, was published IEEE in 2005. The issue of this study was the development of a real time control system by using xPC target to control a robotic system. It was guided jointly by the Mechanical and Aerospace Engineering Department in the University Singapore and by the Automation and Computer-Aided Engineering Department in Chinese University of Hong Kong.

In that study, they showed the advantages and easiness of developing a robotic control system using xPC target over three case studies, one involved a single DC motor control, the other a robotic hand control, and the third one a teleoperation system control.

Implementing and testing the designed controller were important issues for the robotic systems, in which they tried to present a convenient way to implement a real time robotic control system without requiring any low-level language programming.

Their real-time robotic control system architecture was of hardware in the loop type as shown in Figure 1.4. The host PC required the following software packages: Microsoft Windows 2000 operating system, MATLAB, Simulink, Real-Time Workshop (RTW), xPC target, and C/C++ compiler. Simulink was used to model the dynamic physical systems and controllers.

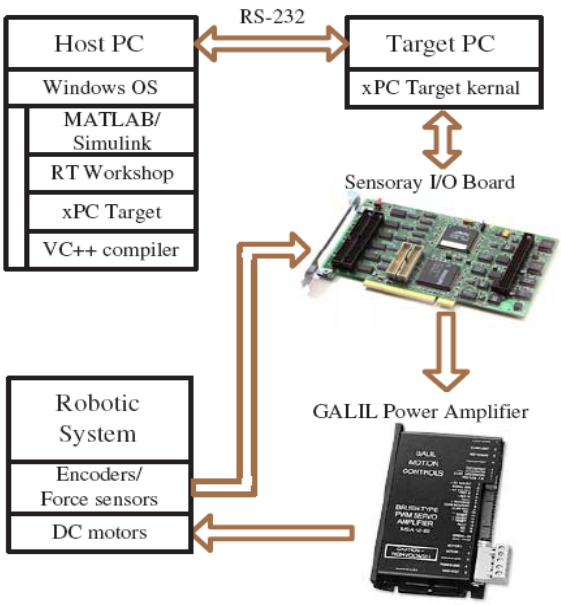

Figure 1.4 The proposed control system architecture for robotic systems [10]

RTW and C/C++ compiler converted Simulink blocks into C code. Then the model was built and downloaded in to the target PC to execute the controller system in real-time. The communication between the host PC and the target PC was accomplished through an RS-232 communication port. The hardware that was used in their proposed architecture included;

- Host PC: Pentium III, CPU 450 MHz, 256 RAM.
- Target PC: Pentium III, CPU 450 MHz, 256 RAM.
- Sensoray I/O board Model 626
- GALIL power amplifier MSA-12-80
- Brush type DC motors and incremental encoders.

Case studies illustrated the applicability of the presented platform. A position feedback PID control, a force feedback loop, and other advanced controllers were designed, tested, and implemented by using the proposed platform. They recommended such a platform to those who wanted to implement a real time robotic control system without needing any low-level language programming.

1.2.1 Conclusion of Literature Survey

In the open literature, several methods and different tools were used for designing, simulating, and implementing controller systems of UAV's. For UAV's Although complex controllers can be designed by using advance control algorithms, the use of PID method is widely preferred by the industry. For that reason, the control systems of UAV’s were designed by PID method in the majority of studies, which were sponsored companies like examples in studies [12] and [13]. They promoted PID method because it had been tested in several flying platforms.

In most studies found in open literature, the avionic system (both software and hardware) was tested for a unique platform during flight tests in the field. Therefore, these studies were not reusable if the characteristics of the platform were changed. For example, in reference [6], they used R-Max UAV as a platform. The avionic systems of R-Max UAV were chosen and implemented on the platform. Then, its navigation system was tested in flight. It was a helpful study, but the experience obtained from that study was not reusable unless the same platform plan was used.

By using rapid control prototyping and real time facilities, a platform was designed as a thesis study in 1998 [7]. In 2002, a group of student at Aeronautical Engineering Department of Naval Postgraduate School, USA, tested their designs by using that platform. The main purpose of that platform was to serve students as hardware in the loop (HIL) test platform without any thoroughness in the computer simulation, code verification, or validation.

The study was an impressive research in 1998; and for that reason, it was closed to literature. However, tools used in the study became out of date. These tools are not used widespread in aerospace industry anymore. Nowadays, MATLAB, Simulink, RTW (or RTWEC), xPC are mostly used instead of Xmath, SystemBuild, AutoCode, RealSim series, respectively. Moreover, in some studies, such as the studies [10] and [5], xPC target was used as a real time environment.

To sum up, the issues of research conducted by universities reflect the requirements of industries involved. Companies not only in aerospace but also in automotive, biotechnology, medical devices, and manufacturing industries want a test bed to analyze their systems and to test their controllers in real-time.

1.3 Objectives of the Thesis

Although, the technological improvement in aerospace industry focuses on rapid control prototyping, real-time operating systems, and embedded software, obtaining a source about these concepts is very difficult because of competition. This thesis aims to fill this vacancy. The object of the thesis is threefold.

The first objective is to design a reusable test platform based on real-time facilities and embedded software. This platform will enable engineers to analyze a system and test its controller in real-time when a project is in the design process. Thus, the design errors can be detected earlier and problems can be solved cost-effectively without interrupting the development process.

The second objective is to check the functionality of the platform. For this reason, a plant and a controller model are needed. An unmanned combat air vehicle (UCAV) model is to be taken as a plant model. A discrete PID controller is to be designed to keep this air vehicle at a desired altitude by using gain scheduling in order to operate over a wide range of flight conditions. A series of test scenarios are to be used for checking the platform functionality.

The final objective of this thesis is to verify the platform reusability by using a different plant and its controller models and present the test results obtained in Simulink environment and in the platform.

1.4 Outline of the Thesis

In Chapter 1 the literature survey on the subject matter and the objective of this thesis are presented.

In Chapter 2, the platform facilities, platform configuration, its conceptual architecture, its components, and the selection criteria of these components are defined and explained.

In Chapter 3, the linear plant model, reasons for selecting a specific plant model and its technical specifications are given.

In Chapter 4, the reference coordinate system, the relation between altitude and elevator deflection of UCAV6 in the form of transfer functions and its plant properties are given. Moreover, continuous time and discrete time PID controllers are designed for the altitude hold mode. An application of gain scheduling technique is also given in this chapter.

In Chapter 5, several suitable platform test scenarios are picked up and selected test scenarios are repeated in the platform. The test results of the platform are discussed and compared with those obtained in Simulink environment.

In Chapter 6, the reusability of the platform is verified by a Boeing 747 model. It is tested in the platform and its test results are discussed and compared with those in simulations.

In Chapter 7, a brief summary on the performed study is given. This chapter also contains some concluding remarks and recommendations for future works.

CHAPTER 2

DEVELOPED PLATFORM AND ITS COMPONENTS

This chapter starts with the need for a platform and its desired facilities. It is followed by a conceptual architecture of the platform. Then, the components of the platform, the selection criteria of them, their rolls, and their alternatives are explained in detail. In the last part of this chapter, the final form of the test platform is summarized.

2.1 Test Platform

2.1.1 The Need for a Test Platform

In the past, companies used to model and simulate their systems in a software environment like MATLAB/Simulink and test them in the field. However, detecting design errors during testing phase in the field is too late and costly to correct them. This approach inflates the size, time, and budget of the project. To speed up the development projects, companies in defense industry as well as in automotive, space, and aviation industries, after modeling and simulating their systems, need to detect design errors in an early stage. Hence, they can solve problems cost-effectively without interrupting the development process. Cost is an important parameter in the industrial sector in order to keep the competitiveness. In addition to that, high technology projects are very expensive due to

- difficulty in finding experienced engineers, especially in rapid control prototyping, real-time operating systems and embedded software areas,
- high cost of the model based design tools, and
- high cost of hardware used.

To reduce the project costs, companies want to analyze and verify their systems under development in real-time during the hardware-in-the-loop test phase of design process. If a reusable test platform is available, this requirement can readily be met. On the other hand, designing such a reusable test platform based on real-time facilities and embedded software is very difficult. Setting such a platform in aerospace industries needs

- a multidisciplinary know-how covering several engineering areas like aerospace, electrical, mechanical, and software,
- some highly expensive design tools,
- a high budget for educating engineers to gain the skills of using engineering tools, which usually come with insufficient documentation, manuals, and technical support,
- an extra effort for understanding codes produced by automatic code generators (These codes are preferred, not only because of their accuracy but also their performance is better than handy codes. Furthermore, automatic code generators produce optimum codes but they use complex templates. It is time consuming to analyze and understand the coding procedure of these generators. Even so, after the coding procedure is understood once, all the codes produced by the same generator can be followed easily.),
- a very expensive hardware to simulate the real environment,
- some costly operating systems to add real-time and embedded software facilities necessary for simulating the real environment depending on particular technological improvement.

2.1.2 Platform Facilities and Advantages

Reusability is one of the necessary and required characteristic of a platform. By the help of reusability,

- the implementation time reduces,
- modifications are made easily,
- the platform becomes adaptable, flexible, fast, and modular which are

essential features needed to build a larger system from smaller parts,

- the operation of the platform does not depend on the plant model on the design method used for the controller (For example, the model of car, train, aircraft, or satellite can be used as a plant model and the controller model can be designed by using methods like PID, fuzzy logic, adaptive, optimum, robust, etc.),
- the know-how obtained from one project can be transferred to the others thus enabling companies to decrease the cost of not only the current but also future projects.

In addition, the platform includes some rapid control prototyping facilities. Listed below are some advantages of rapid control prototyping facilities contributing to the platform. By the help of the platform,

- the product development time and cost are reduced (For example, according to a research [15] about rapid control prototyping at the Ottobrunn Production Centre, (EADS, Space propulsion), Munich, Germany, the cost reduces up to 50% and processing time reduces up to 75% by rapid control prototyping),
- the product gets into the market sooner,
- the communication between the designer and users is enhanced,
- an immediate feedback is provided for a better product,
- the quality of the product can be improved,
- design errors can be detected at the system modeling stage instead of during system integration tests, consequently, problems due to these errors can be solved in the laboratory instead of in the field,
- an opportunity arises for testing the system model part by part and
- the design is verified before coding manually.

The other properties of the platform are as follows.

- The platform includes some real time facilities. Real time means that computer simulates the events at the same speed that they would occur in real life. In this platform, more than one real time operating systems may be used.

these real time computers update information at the same rate as they receive information. They are used for such tasks as navigation, in which the computers must react to a steady flow of new information without any interruption.

- The platform uses the actual hardware. According to the DO-178B standard [2], one should check requirements of a design in a test bed before its implementation. This test bed should be as close to the real environment as possible.
- The platform gives the possibility for testing the tasks of the system in an environment similar to real environment that enables taking the DO-178B certificate more easily.
- The platform supplies an opportunity to check the design by using the hardware of the real product. The advantage of using real hardware reduces the problems that may arise during the system integration.
- The platform enables to focus on detecting and protecting design errors, which are more critical and easily overlooked than application errors.
- The platform provides a rapid simulation for larger number of test scenarios than the ones that can be performed on the actual hardware. Performing such test scenarios in the field tests are very expensive, dangerous, and sometimes even impossible.

2.2 A Conceptual Architecture of the Platform

The main objective of this thesis is to design a test platform, which is based on real time facilities. Moreover, it supplies an opportunity to check the design by using rapid control prototyping facilities. The platform designed within the scope of the thesis consists of some hardware, software, and interfaces components. The components of the platform can be divided into four groups. They are the plant, controller, host PC, and actuator.

The first group of the test platform components is a *plant* which is composed of

- a software used for simulating flight dynamics for which the platform set and
- a hardware where this software runs.

In this study, an xPC target computer is chosen as a plant environment with an ethernet card, a serial communication port, and an I/O card. They play the role of a bridge between the software and hardware environments.

The second group of the test platform components is a *controller*, which is composed of

- a software used for simulating the controller dynamics of the plant and
- a the necessary hardware where this software runs.

In this study, the real-time embedded code of the controller runs in a single board computer. In the setup, its functions are processing the command signal, which is sent from the plant and producing the required control command signal. The controller needs an I/O board to receive the inputs signals and to send the output signal.

The third group of the test platform components is an *host PC*, which consists of

- a software mainly used for modeling, simulating, and analyzing the whole system and
- a required hardware where this software runs.

In this study, the host PC is a desktop PC with an ethernet card and floppy disk. The ethernet card is used for communication with plant and controller. The floppy disk is used for creating a boot disk for xPC target computer. Next to modeling, simulating, and analyzing the whole system at the beginning of the design process, the hosts PC is also used for creating the codes for controller algorithm and plant as well as for downloading these codes to the real-time environments separately before these codes are tested in the setup.

The fourth group the test platform components is an *actuator system*, which is composed of

- a servomotor,
- its driver unit, and
- an encoder.

In this study, the actuator system is added to the setup to realize the behavior of the control surface with hardware instead of applying the control commands via software.

Figure 2.1 shows the conceptual architecture of the platform developed in this study. The I/O boards, ethernet cards, servo amplifier, encoder, and interfaces are not shown in this preliminary sketch of the set-up.

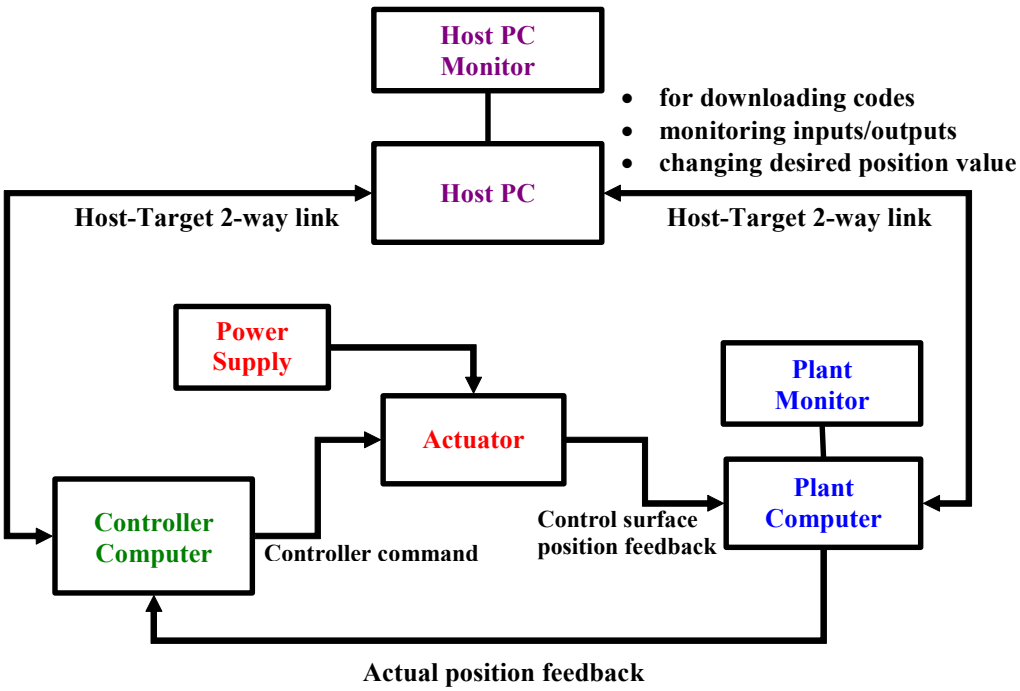

Figure 2.1 Conceptual architecture of the platform

2.3 Plant

In this setup, an xPC target computer is chosen to simulate flight dynamics in real time. By using only a standard PC, the xPC target provides an opportunity for prototyping, testing, and developing a real-time system. It includes some proven capabilities for a rapid control prototyping and hardware-in-the-loop simulation. A picture, which shows the xPC target computer used in the platform is given in Figure 2.2.

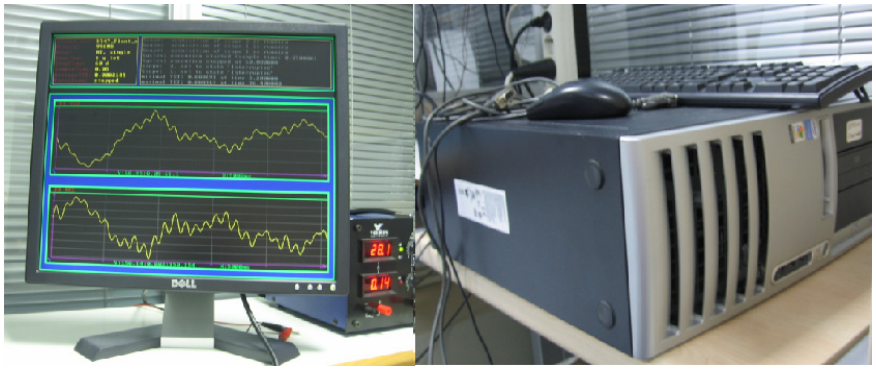

Figure 2.2 xPC target view on the platform

A desktop PC or an industrial PC can be used as an xPC target computer. However, a notebook PC is not advised [22], because of its stable configuration. The target PC can be compatible system with a 32–bit Intel or AMD processor (386 compatible or higher). More information about the minimum system requirements of xPC target computer is given in Appendix A.

The xPC target needs an ethernet card, a serial communication port, and an I/O card to communicate with the host PC, controller, and actuator, respectively. Moreover, if it is an industrial PC, it should be booted by using a boot disk or a flash memory or from its hard disk. The xPC target used in the test platform in this thesis has a floppy disk drive to boot the target computer.

The Ethernet Card of the Plant

The type and series of the ethernet card, which is supported by xPC target computer, change with respect to the target PC processor. If the xPC target has an Intel processor, Pro/100S I82559 ethernet card is advised in the reference [21]. Otherwise, if its processor is AMD, then Lance 79C97x. is offered [21]. Due to xPC target in the thesis test platform has an Intel processor, the communication between the xPC target and host PC is substantiated by the help of a Pro/100S I82559 ethernet card.

I/O Board of the Plant

A Sensoray 626 I/O board is chosen to communicate with the xPC target computer. This I/O board is manufactured by Sensoray embedded electronics. Figure 2.3 shows the view of this board. For this thesis, having xPC target drivers is the most important required property for communicating with the xPC target computer. Otherwise, xPC drivers have to be written.

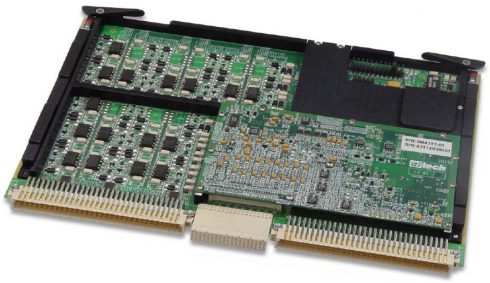

Figure 2.3 Sensoray 626 I/O board

The main reason for using a Sensoray 626 I/O card is its low price in local market with respect to its competitors. Sensoray 626 has also a QNX driver and supports RT-LAB. QNX is a real-time rapid control prototyping environment. RT-LAB is a tool of Opal-RT Technologies Inc., which provides an HIL simulation environment without seeing any codes. The technical specifications of Sensoray 626 are given in Appendix A.

Alternative Models of Sensoray 626 I/O board

The technical properties of Sensoray 626 I/O board alternatives are given in Table 2.1 to compare with Sensoray 626. This table is obtained by the help of references [23] and [24]. The price per quality advantage of Sensoray 626 in local market can be seen clearly in this table.

Table 2.1 Alternative models of Sensoray 626 I/O board

Alternative Models	Advantages	Disadvantages
Humusoft MF624	Good technical support	High price
National Instruments PCI-6602	Well-known company Good technical support	Only reads encoder output Very high price with respect to properties.

2.3.1 Interfaces between Actuator and Plant

In this setup application, a BLD 5606-SE4P servo amplifier is used as the servomotor driver unit. Its selection criterion is explained in Section 2.6. This servo amplifier has an analog signal for the motor speed output. The plant and actuator system communicate by the help of the analog to digital converter of Sensoray 626 I/O board. In this thesis platform, due to the advantage of the motor speed output signal, there is no need to use the encoder interface of Sensoray 626 I/O card.

2.3.2 Interfaces between Plant and Controller

In this thesis study, a single board processor is chosen to simulate controller. Further information about the components and their selection criteria of the controller is given in the following section. In this part of the chapter, a description of the communication alternatives between plant and controller is given. Then, the reasons of choosing RS-232 as a communication procedure are explained. Alternatives of the targets communication of the interface can be summarized as below. They are explained briefly with their advantages and disadvantages for this setup application.

- Avionic Bus
  - MIL-ST-1553 Mux-Bus
  - ARINC 429
- Ethernet
- Analog
- Digital
- Synchro
- Serial communication
  - RS-232
  - RS-422

Analog

An analog signal is a continuous signal. It differs from a digital signal because of meaningful small fluctuations in an analog signal. The primary disadvantage of analog signaling is that any information conveyed by an analog signal is disturbed easily because of noise, random variation etc. As the signal is transmitted over long distances, these random variations become dominant. By shielding, good connections, and several cable types, these losses can be diminished but not completely consumed. Analog communication between targets can be possible because both Sensoray 626 and C431 I/O cards, which is the I/O card of controller, have A/D and D/A ports. However, the test room in which the thesis platform located is very noisy due to other test equipments such as target, computers, PCs, and air-conditioner etc. Hence, a communication via analog signal between targets is not preferred.

Synchro

Synchro signal is used in avionics systems to define three-dimensional movements as electrically. For example, it is used to convey the aircraft attitude information with respect to ground from gyro to related displays. Synchro signal is also an analog signal, so the disadvantage of analog signal is also valid for it.

Avionic Bus

The information traffics between avionic units increases with the development in avionic technology. As the number of avionic units increase, avionic buses are preferred for the communication between these units. Any increase the number of avionic units

- brings extra cable weight,
- reduces the reliability because of too many cable connections,
- increases the number of the platform bug test, and
- increases the modification number of the aircraft.

The communication by using time division can reduce the number of these problems among avionic units. It enables to convey all information rapidly and reliable by using only one line. There are two avionic buses in modern avionic systems; MIL-STD-1553B Mux-Bus and Aeronautical Radio Inc. (ARINC) 429. Military aircrafts use MIL-STD-1553B Mux-Bus and civil aircrafts use ARINC 429 bus.

MIL-STD-1553B Mux-Bus

MIL-STD-1553, which is published by the Department of Defense of the United States, was originally designed for use in military avionics, but has also become commonly used in spacecraft. The most important advantage of MIL-STD-1553B Mux-Bus is that it can have more than one (maximum 31) remote terminals, a bus controller (only one), and one bus monitor. A remote terminal can be used to provide

- an interface between the data bus and an attached subsystem,
- the bridge between a MIL-STD-1553B bus and another bus.

On the other hand, the bus controller

- operates according to a command list stored in its local memory,
- commands the various remote terminals to send or receive messages,
- services any requests that it receives from the remote terminals,
- detects and recovers from errors,
- keeps a history of errors.

Bus monitor role is recording the bus and storing the transactions for analyzing the bus off-line. It cannot transmit messages over the data bus. Figure 2.4 shows the conceptual model of MIL-STD-1553B bus.

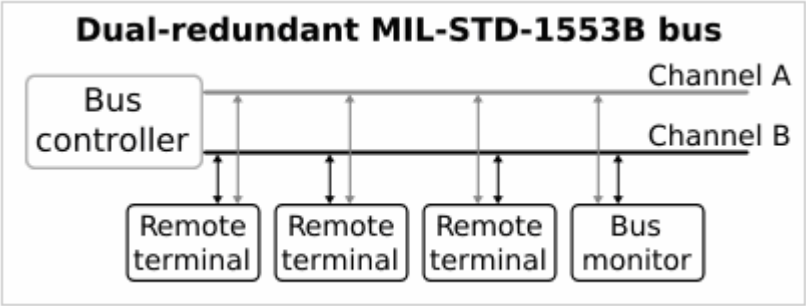

Figure 2.4 A Conceptual model of MIL-STD-1553B bus [87]

Although, MIL-STD-1553B Mux-Bus is used in modern avionic systems and brings several advantages, both I/O boards (Sensoray 626 and Aitech C431) and Aitech C772 board (the single board computer for controller) do not have any 1553 interfaces. The MIL-STD-1553B interface of Aitech C772 is optionally available [30]. Hence, for using this bus protocol, two I/O cards (one for xPC target computer and the other for VxWorks target computer) which are suitable for MIL-STD-1553B Mux-Bus communication, are needed. Since these I/O cards are too expensive, the communication via 1553 is left for implementation in future works. For the thesis application, unfortunately, 1553 Mux-Bus cards are not supported.

Aeronautical Radio Inc. (ARINC) 429

The same problem in MIL-STD-1553B is valid for ARINC 429 bus protocol. Therefore, ARINC 429 bus is not preferred.

Ethernet

Today, ethernet is a widely used technology for local area networks. After developed by Xerox in the 1970s, ethernet became popular with Digital Equipment Corporation (DEC). In 1985, ethernet was officially accepted as IEEE standard 802.3. Full chapters of IEEE 802.3 standard can be downloaded from the reference [80]. Ethernet networks can operate at several speeds up to 10 Gbps. Although ethernet is a very useful alternative due to its speed, writing driver for Aitech C772 card depending on TCP/IP protocol is more difficult than that on RS-232 protocol. Aitech C772 board has two ethernet ports to be activated. However, because of the complexity of writing driver C code for TCP/IP protocol, target's communication is tried via RS-232. If it is found to be insufficient for the thesis setup applications, ethernet will be used as a second communication alternative.

Digital

Similar to TCP/IP protocol, the same situation is valid for the discrete I/O. To communicate via a discrete I/O, C codes should be written by using VxWorks APIs, which are given in Aitech C772 card's library.

Serial communication

- RS-232
- RS-422

Although there are several recommended standards such as RS-170, RS-232, RS-422, RS-423, RS-449, RS-485, RS-530, RS/6000, two common serial communication protocols used in avionic systems are RS-232 and RS-422.

RS-232

RS-232 is a simple and universal communication protocol. For this reason, it is used more commonly. Most personal computers have at least one RS-232 communication port. Moreover, the xPC target supports RS-232 communication port on the PC without needing any extra I/O board.

Serial ports on most computers use RS-232C standard. The 232 is the recommend standard number and C is the latest revision of this standard. The full RS-232C standard specifies a 25-pin "D" connector. Most of these pins are not needed for normal PC communications, and indeed, most new PCs are equipped with male D type connectors having only 9 pins.

RS-232 is an electrical interface and satisfies an asynchronous data transfer. It is a point-to-point communication. That means, only two devices can be on the bus. The maximum cable length limit is 50 feet (approximately 15m). Ignoring the standard, cable can be as long as 10,000 feet (approximately 3km) with a well-shielded cable. However, it is not reliable for aerospace applications. Moreover, the maximum boud rate with this standard is 19,200 bps. The RS-232 standard also defines the voltage levels that correspond to logical one and logical zero levels. Valid signals are plus or minus 3 to 15 Volts. The disadvantages of RS-232 can be listed as below.

- The main disadvantage is that it is not a bus. RS-232 is designed to transfer data between two devices.
- It is not intended for multi protocol or multi point type applications.
- RS-232 is intended for a short distance communication, although the distance depends on data rates and cable type.
- It has unbalanced load. That means the signal travels along in one cable as reference to ground.
- There is less noise immunity.
- There is invalid data due to noise.

The device number and communication distance limits of RS-232 are not important as far as this thesis setup application is concerned. Therefore, RS-232 serial communication is used for transferring the data between xPC and VxWorks targets, which are located near to each other in the laboratory. Approximately 2 meters of RS-232 cable is enough for communication between targets and the length of the cable is well within the limits. There is no need an extra I/O board for the communication between targets, so it is a low cost solution. The xPC target supports RS-232 port on PC and Aitech C772 card has RS-232 interface.

Moreover, the RS-232 drivers of xPC target are supported by The MathWorks. For communication via RS-232 with xPC target, writing an m-file is enough. On the other hand, one has to write RS-232 communication C codes of the VxWorks target depending on Aitech C772 APIs.

The maximum data transfer rate of RS-232 communication is calculated below in order to decide whether it is enough or not for the setup application. The data transfer rate of RS-232 is 115,200 bps for xPC target computer. Assuming the data type in an application is taken as float, which is 32 bit and the values of two variables are aimed to transfer between targets, then the maximum rate of data transfer can be calculated as 1,800 Hz from (115200/ (2*32)). This means that for the transformation of two float variables, the sample time of the system can be $1/1800 = 5.55\text{x}10^{-4} \cong 0.0005$ seconds. This rate is fast enough for the setup application, because several controllers on the actual platform have the sample rate between 0.025 seconds and 0.05 seconds. The calculation can be repeated depending on the data type and the number of the variables.

RS-422

RS-422 is a serial interface standard in which the data is sent in a differential pair (two wires, or twisted pair cable), therefore, it is more reliable than RS-232. Moreover, it allows the use for greater distances and higher data rates than RS-232.

RS-422 data transfer is also asynchronous. Unlike RS-232, one driver can be in a bus with multiple receivers and the maximum number of the receivers on the bus is about ten. The cable length is significantly higher since RS-422 can transfer data up to 4,000 feet (about 1.2 km), and the data transfer rate is about 10 megabits per second. For the thesis application, while using RS-422 recommended standard, there are two needs

- RS-422 I/O board supported by the xPC target (in this case, there is no need to write any driver code because the MathWorks supplies it.)
- RS-422 serial communication drivers suitable for the Aitech C772 board.

The RS-422 protocol is more reliable than RS-232. If an I/O board is bought, MIL - STD-1553B Mux-Bus would provide much more advantages than RS-422.

2.3.3 Plant Software Requirements

The xPC target does not require DOS, Windows, Linux, or any another operating system on the PC. Instead, the PC can be booted with a boot disk including the xPC target kernel. The kernel has no effect on any operating system installed on the PC. The arrangement of boot disk allows using the PC for testing real-time applications. After the application, the PC can be used again as a desktop computer. Creating a target boot disk with a bootable image suitable for the xPC target configuration is explained systematically in Appendix B. Moreover, the BIOS is the only software component that is needed by the xPC target kernel. The BIOS settings of a PC system can affect how the PC works with the xPC target.

2.4 Controller

The real-time embedded code of the controller runs in a single board computer. The controller needs also an I/O board to receive the inputs signals and to send the output signal. These boards are set into a case and communicate with each other via versa modular eurocard (VME) bus.

To perform the controller algorithm during setup application, a VME single board computer is selected, because VME cards are designed for harsh environments. Hence, they are suitable for aircraft platform applications. Such a card, which includes controller codes, can be used on the actual platform after tested in the test platform.

VME bus is a computer data path standard, designed by Motorola, Signetics, Mostek, and Thompson CSF. Firstly, it is developed for the Motorola 68000 line of CPUs in 1981, but later widely used for industrial, commercial, and military applications. Then, it is standardized by the IEC as ANSI/IEEE 1014-1987.

VME bus systems compensate the requirements of real time operating systems. The main requirements of them is to do a task within a certain time limit. Non real-time operating systems do not have such urgency. Moreover, VME bus is used in critical applications because it can withstand shock, vibration, and extended temperatures better than the buses used in desktop computers.

A VME bus system is called as a multiprocessing bus, which means having several master devices. That is, a VME bus system uses a master/slave architecture. In the master/slave architecture, the master controls another device like in the relationship between computer and printer. The computer is the master, and the printer is the slave because the printer cannot control the computer.

Like master/slave configuration, timing, signal voltage levels board dimensions, connector specifications mechanical specifications and electronic specifications are defined by the VME standard, on which the VME bus system is based.

VME Single Board Computer of the Controller

An Aitech C772 PowerPC G3/G4 VME Single Board Computer (SBC) is chosen to verify the controller algorithm in the setup application. It has a high performance processor and extensive I/O capabilities. Figure 2.5 shows the view of the Aitech C772 VME single board computer.

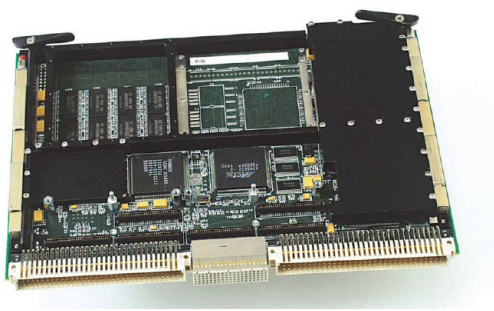

Figure 2.5 Aitech C772 VME single board computer view

In this thesis setup, C codes of controller algorithms run in Aitech C772 PowerPC G3/G4 VME Single Board Computer in VxWorks target hardware. VxWorks target has VME bus and C772 has VME bus connecters. VxWorks Real Time Operating System and Real Time Clock are other important facilities of C772. The technical specifications of Aitech C772 VME single board computer can be found in Appendix A.

Although the definition of Aitech C772 card indicates some extensive I/O capabilities, it does not have any analog to digital (A/D) converter, digital to analog (D/A) converter, or counters for encoders. Therefore, in this setup, in order to communicate with DC motor an extra I/O card is needed, and an Aitech C431 VME I/O board is used for this purpose.

The reason why C772 and C431 cards are chosen is that they are available cards in the avionic laboratory of the company supporting this thesis. They were bought for a project and now they are used only for laboratory applications.

VME I/O Board of the Controller

The Aitech C431 A/D D/A & Digital I/O VME Board is a VME bus slave card providing extensive I/O capabilities including analog to digital converters, digital to analog converters and opto-isolated digital I/O. Like Aitech C772, it is also designed for harsh environment applications. Figure 2.6 shows the view of the Aitech C431 A/D D/A & Digital I/O VME Board.

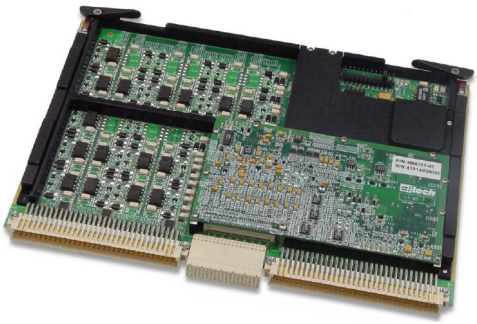

Figure 2.6 Aitech C431 A/D, D/A and digital I/O VME board

The Aitech C431 is a VME Bus slave card including a 16-bit ADC, a 12-bit ADC, a 16-bit DAC, and a 14-bit DAC with high precision data conversion. There are two ADC devices; one is 16-bit and another is 12-bit. The 16-bit device input channels are software configurable to either 32 single-ended input channels or 16 differential channels or a combination of both. The 12-bit ADC controls two completely isolated differential channels. There are two DAC devices, 16-bit and 14-bit. The 14-bit DAC is capable of driving up to 32 single ended completely independent output channels while the 16-bit DAC is capable of driving up to 8 channels.

In addition to ADC and DAC, it includes the digital I/O. This I/O provides high current opto-isolated channels specifically designed for switch activation and high current switching applications. Each of these opto-isolated channels may be individually configured to operate as input or output.

The most important property of the card is VME Bus interface. The VME Bridge provides slave VME capabilities and supports A32/D08/D16/D32 data transfers from/to any standard.

In this thesis setup, the C431 board is planed to use for D/A output channel for analog communication. A C772 card does not satisfy such a communication. The D/A (DAC1) output channel is chosen because it has 16-bit resolution and high resolution is very important to controller setup applications. The user guide of Aitech's C431 board [34] can be followed to become familiar with procedures for the installation and the board's major functional blocks.

2.4.1 Interfaces between Controller and Actuator

The Servo amplifier BLD 5606-SE4P requires analog input between ±5 Volts. Because of this constraint, the connection between VxWorks target computer and servomotor is performed by analog signal by using digital to analog converter on Aitech C431 I/O board.

2.4.2 Controller Software Requirements

In this thesis study, the controller, which has a VME bus system with VME single board computer and I/O board is called as the *VxWorks target computer*, although VxWorks is a real time operating system. The VxWorks operating system is developed and sold by Wind River Systems. It has some similarities to UNIX operating system. It includes a shell, debugging functions, memory management, performance monitoring, and support for multiprocessing. This operating system has a multitasking kernel with pre-emptive scheduling and fast interrupt response, extensive inter-process communications and synchronization facilities. Like xPC target's application, VxWorks applications need a *host* machine. In the setup, only one host PC is enough for both plant and controller environments.

A computer with a VME bus and VxWorks operating system is a common environment for mission critical applications in aerospace. For that reason, a VME single board computer with VxWork operating system is chosen to simulate the controller system but not to simulate flight dynamics. Some notable products designed by using VxWorks operating system are given below. Moreover, the board in which controller algorithms runs can be integrated to the ultimate platform after tested in the platform.

Notable products designed by using VxWorks operating system

- The Spirit and Opportunity Mars Exploration Rovers
- The Mars Reconnaissance Orbiter
- The Deep Impact space probe
- The Boeing 787 airliner (in development)
- The BMW iDrive system
- Linksys WRT54G wireless routers (versions 5.0 and later)
- Xerox Phaser and other Adobe PostScript-based computer printers
- The Experimental Physics and Industrial Control System (EPICS)
- The Apache Longbow attack helicopter
- Hughesnet HN7000 series satellite internet receivers (VSAT)

The main disadvantage of VxWorks operating systems is that an extra payment is needed for each project although the same license is used. On the other hand, the xPC target is a low cost product with a good technical support. There is no need to pay such extra cost for using an xPC target software in each new project. However, the xPC target computer has a PCI bus and this bus is not suitable for harsh environment applications. Hence, it cannot not be used as a flight computer. The competitive companies of VxWorks operating system and xPC target software are listed below.

1.C Executive	5.Embedded Linux	9.Nucleus RTOS	14.ThreadX
2.eCos	6.INTEGRITY	10.OS-9 / 11.OSE	15.TRON
3.ElinOS	7.LynxOS	12.QNX	16.VRTX
4.embOS	8.MontaVista Linux	13.RTEMS/ RTOS	17.Windows CE

Of these, QNX, Embedded Linux, and INTEGRITY are the strongest competitors of VxWorks operating system in the list given above. INTEGRITY is produced by Green Hills Software Inc. It has also VME bus and its user interface is more user-friendly. Like VxWorks, it has a good technical support and it can be put on the actual platform. However, it is not preferred in this thesis application, since the avionic laboratory in which this study is conducted does not have any INTEGRITY license. QNX and Embedded Linux are also common real time operating systems. However, both QNX and Embedded Linux targets have very weak technical supports. As a result of this, designers have to spend time for writing their own interface drivers. Hence, they are not commonly preferred in industry. Table 2.2 summarizes the advantages and disadvantages of real time operating systems.

To download the controller real time codes into VxWorks target computer, an integrated development environment (IDE) is needed. Tornado v5 is used as an IDE between controller and host PC for monitoring the controller application. Alternatively, Workbench v6, which is the new version of Tornado, can be chosen,

but the avionic laboratory in which this study is conducted does not have any Workbench v6 license. Furthermore, for the communication between the plant and controller realized via RS-232, extra C codes should be written by using the API's of Aitech C772 VME single board computer. The serial communication code reaches the communication interface drivers of Aitech C772 on BSP. This code should be changed by user, depending on the number and names of the input and output signals as the controller model is enhanced.

Table 2.2 Real time operating system's advantage/disadvantages

Real Time Operating System	Advantages	Disadvantages
VxWorks	Good technical support Can be used in real systems Supports DO-178B Commonly used in industry VME-bus support	High price Extra payment for each license Extra payment for board support package (BSP)
xPC	Used in most university projects Low cost with respect to properties Supports I/O cards in very wide range Enables I/O drivers No additional payment for each project Commonly used in industry	No VME-bus support (it supports PCI-bus) Depends on a host PC for a starting comment
Integrity	Good technical support Can be used real systems Support DO-178B Commonly used in industry VME Bus	Extra payment for BSP High price
QNX	Low cost with respect to properties Used in most university projects No additional payment for each project	Drivers are not supported by producers Poor technical support No VME-bus support (it supports PCI-bus)
Embedded Linux	Low cost with respect to properties Used in most university projects No additional payment for each project	Drivers are not supported by producers Poor technical support

2.5 Host PC

Any PC that runs a Microsoft Windows platform supporting The MathWorks products can be used as a host PC. It a can be desktop PC or a notebook PC, as long as it contains an ethernet adapter card and a 3.5-inch floppy disk drive. An ethernet card is needed for the communication with the plant and the controller, and a floppy disk is needed for creating a boot disk for the plant. The host PC in this setup is an Intel Pentium 4 with 2.00 GB RAM. The minimum hardware requirements for the host PC can be found in Appendix A.

It was reported that the standard Windows operating systems are not suitable for real time controller applications [78]. On the other hand, for modeling, simulating, analyzing the dynamic system and its controller MATLAB/Simulink environment is used in this thesis, but this environment requires a Windows operating system. Therefore, the host PC should not be used for real time applications. However, it can be used for

- modeling, simulating, and analyzing the whole system at the beginning of the design process (MATLAB/Simulink applications),
- obtaining the dynamic link libraries of both plant and controller models,
- downloading plant algorithm codes to plant,
- downloading controller codes to controller,
- tuning defined variable during hardware in the loop tests,
- controlling the application in plant, such as starting and stopping the target application or changing sample and stop times, etc.,
- renewing some parameter's value; implying that downloading tunable parameters to the plant and changing their values between runs or during a run, and
- checking the performance of the controller application by logging a defined parameters or outputs.

2.5.1 Interface between Host PC and Plant

The communication between the host PC and plant can be established through a serial connection or a network connection [21], since an xPC target computer is chosen to model the plant dynamics. The details of this choice are given in Section 2.4.2. Although the host and target computers may be connected directly with a serial cable using their RS 232 ports, in this thesis, they are connected with a network because the serial port of the xPC target computer is used the connection between the plant and controller.

The network can be a LAN (local area network) or a direct connection using a crossover ethernet cable. Both the host and target computers are connected to the network with ethernet adapter cards using the TCP/IP protocol for communication. When a network connection is preferred, a 10 Mbps, a 100 Mbps, or a 1 Gbps data transfer rate ethernet card is needed. After the communication between host PC and plant is ensured, the plant model is designed in the host PC and it is downloaded to the plant via this interface.

2.5.2 Interfaces between Host PC and Controller

Like the communication between the host PC and plant, the communication between host PC and controller is realized with a network connection.

2.5.3 Host PC Software Requirements

A MATLAB/Simulink environment is required for the host PC to verify the defined tasks explained in the previous section. MATLAB is a software package produced by The MathWorks, Inc. for high-level technical computing and algorithm development. It can be used in wide range of applications by writing some standard commands. These commands can be called by the user from the command line of the MATLAB workspace in an interactive way or by writing MATLAB programs or functions, called as *m-files*. Several MATLAB add-on software libraries, called as

the *toolboxes*, extend the MATLAB environment to solve particular classes of problems. *Simulink* and *Control System Toolbox* are the two common examples of these libraries. They are widely used in control theory applications for simulation, analysis, and design.

In addition to MATLAB/Simulink, Real-Time Workshop (RTW), Real-Time Workshop Embedded Coder (RTWEC), C language compiler, xPC target toolbox, Tornado, and Windows operating system are installed in the host PC. Minimum software components in the host PC and their versions can be found in Appendix A.

The Real-Time Workshop generates optimized, portable, and customizable ANSI C, C++ codes from Simulink models. Generated codes can run on hardware with either a real time or non-real time operating system. The Real-Time Workshop Embedded Coder works with RTW to generate efficient and embeddable source codes. The most important advantage of RTWEC over RTW is the generation of optimum codes, which brings faster speeds in using the memory. Moreover, it enables an interface between the Simulink model and its codes.

The C compiler creates executable code from the C code generated by RTW. Both Microsoft Visual C/C++ version 5.0 or newer and Open Watcom C/C++ version 1.3 C compilers are supported by RTW and RTWEC.

2.6 Actuator

In order to verify the motion of the control surface deflection in this platform, it is decided to integrate an actuator system to the platform. There are several types of rotational actuators such as hydraulic motors, step motors, AC motors, and DC motors. A DC servomotor is used as the actuator unit of the system by taking into account its high performance, easiness of use, cost, and availability,

In this thesis study, an existing DC servomotor with its servo amplifier, gearbox, and encoder in the Control Laboratory of the Mechanical Engineering Department of METU is used. It is suitable for the current application because it has several

advantages like very low friction and long life due to being brushless and has a fast response since it is driven by PWM technique. It is produced by Minimotor/Faulhaber, a firm located in Switzerland and its model number is 3056K012B-K312. The characteristics of the motor are given in Table 2.3. Detailed information about the servomotor can be found in reference [40]. The motor is coupled with a gearbox unit, produced by the same firm. The properties of the gearbox are given in Table 2.4 and can be found in detail in [40].

Table 2.3 Characteristics of the DC-servomotor

Output power (at 22,000 rpm)	48 W
Nominal operating voltage	12 V
No load speed	8,790 rpm
No load current	0.168 A
Stall torque	95 mNm
Direction of rotation	Electronically reversible
Maximum efficiency	73%
Brush type	Brushless
Maximum torque up to	20.7 mNm
Maximum speed up to	28,000 rpm
Maximum current up to	1.94 A

Table 2.4 Characteristics of the gearbox

Reduction	353:1
Shaft diameter	8 mm
Maximum allowable torque	12 Nm
Backlash	15 arcmin max.

The servo amplifier BLD 5606-SE4P is the basic board with an additional module for speed control of the DC servomotors with an additional encoder. Table 2.5 shows its characteristics. In combination possibilities in the servo amplifier data sheet [36], 3056 K series brushless DC servomotors can be combined with BLD 5606-SE4P servo amplifier and HEDS 5500 encoder. An HEDS 5500 encoder is assembled with DC motor and gearbox. Table 2.6 shows the properties of encoder.

Table 2.5 Specific characteristics of the servo amplifier

Format (L × W × H)	119.3 × 90.5 × 37 all in mm
Power Supply	14-56 V DC
Output Voltage	-5 to +5 V DC
Weight	225 g
Switching Frequency	150 kHz
Analog Input Command Voltage Range	±5 V DC
Input Resistance	150 kOhm
Frequency Bandwidth	1 kHz
Operating Temperature	0-70 °C
Logical Input	TTL

Table 2.6 Properties of the quadrature encoder

Supply Voltage	5 V
Output	2 digital signals in quadrature
Number of counts per revolution	512
Maximum count frequency	100 kHz
Operation Temperature	-40 to 100°C
Load Capacitance	100 pF max.
Supply Current	17 mA

2.7 Final form of the Platform

Figure 2.7 summarizes the necessary software packages, hardware components chosen to run them and interfaces between the components of the platform. The output signal range of the Aitech C431 I/O board is from 0 to 10 Volts, but the servo amplifier needs an input between ± 5 Volts. In the data sheet of C431 board, it is defined that the output signal range of the I/O board is either between ± 5 Volts or 0-10 Volts. However, upon setting on the VxWorks target computer, the output voltage range of the board cannot be measured. It depends on the to customer desire. Although the platform is desired as shown in Figure 2.8, this I/O board cannot be used in this platform.

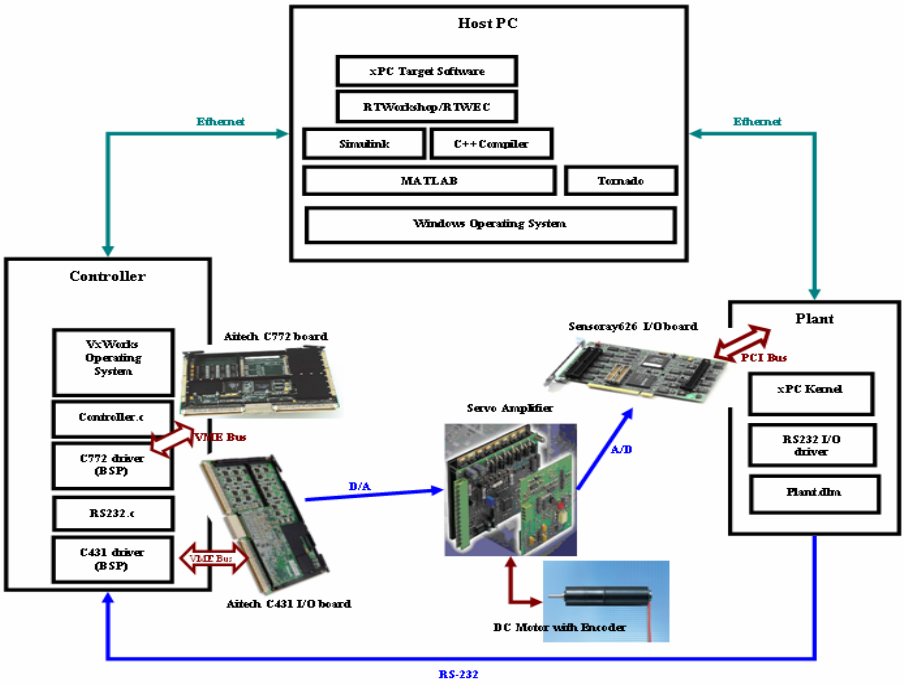

Figure 2.7 The platform view with minimum software packages

There are two alternatives that can be followed. One of them is to design a voltage converter taking an input signal between 0-10 Volts and sending an output signal between ± 5 Volts. This alternative is not preferred, because the communication between controller and actuator is realized via an analog signal therefore it can be easily corrupted. The other alternative is using Sensoray 626 I/O board, which is located on xPC target computer, as an I/O board of controller.

The communication between controller and actuator is realized as shown in Figure 2.9. After the controller is triggered with plant output signal, signal the controller creates controller commands depending on the output. These commands are sent via RS-232 by the C772 board, in which controller algorithms run. The plant takes the controller commands by the help of RS-232 port on the PC. After that, D/A channel of Sensoray 626 I/O board transmits these commands to the actuator. The actuator realizes control surface position and its output signal is taken by A/D channel of the same I/O board. This position signal processed with plant algorithms and the plant output is sent by the help of the same serial port on the PC.

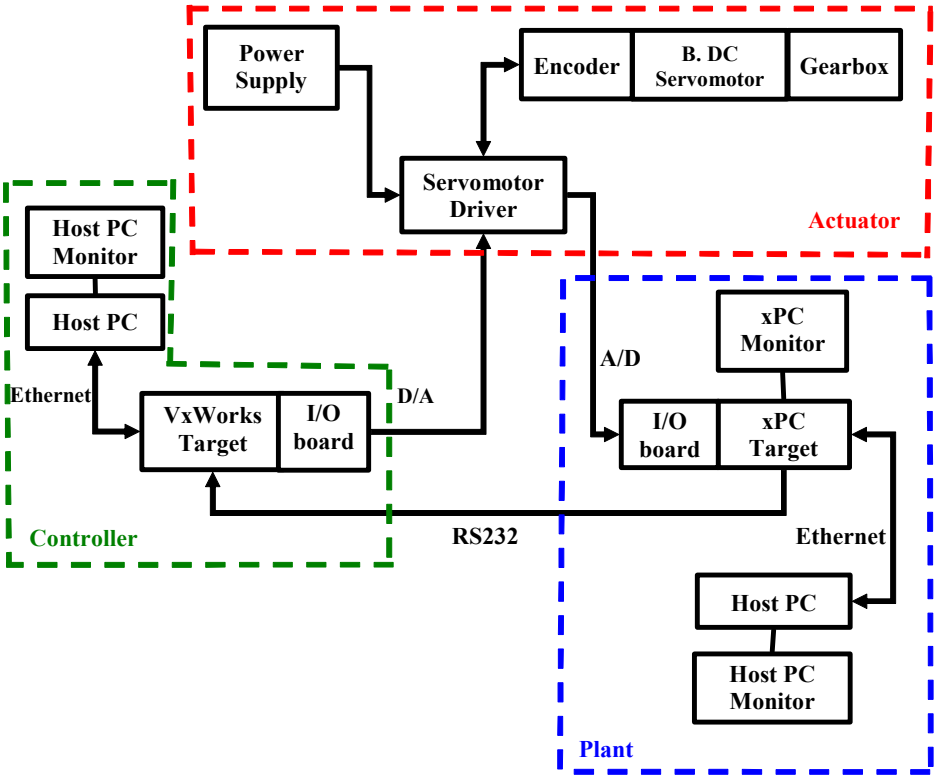

Figure 2.8 Desired HIL test setup

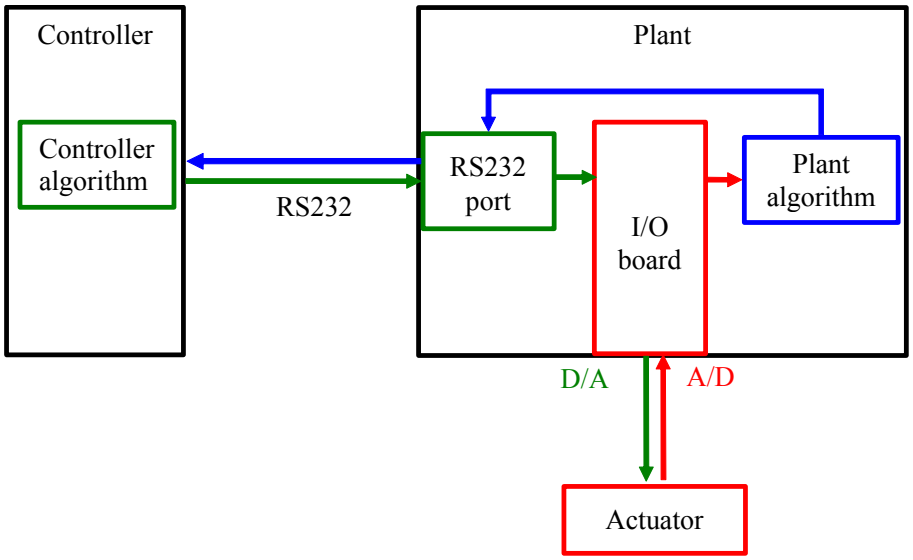

Figure 2.9 Data flow among the controller, plant, and actuator

2.8 Concluding Remarks

In this chapter, necessity and desired platform facilities are defined with their advantages. A conceptual architecture of the platform is given and main components of the platform are explained. The rolls of the platform components, their selection criteria, and their alternatives are explained in detail. Figure 2.10 summarizes the architecture of the test platform designed in this thesis.

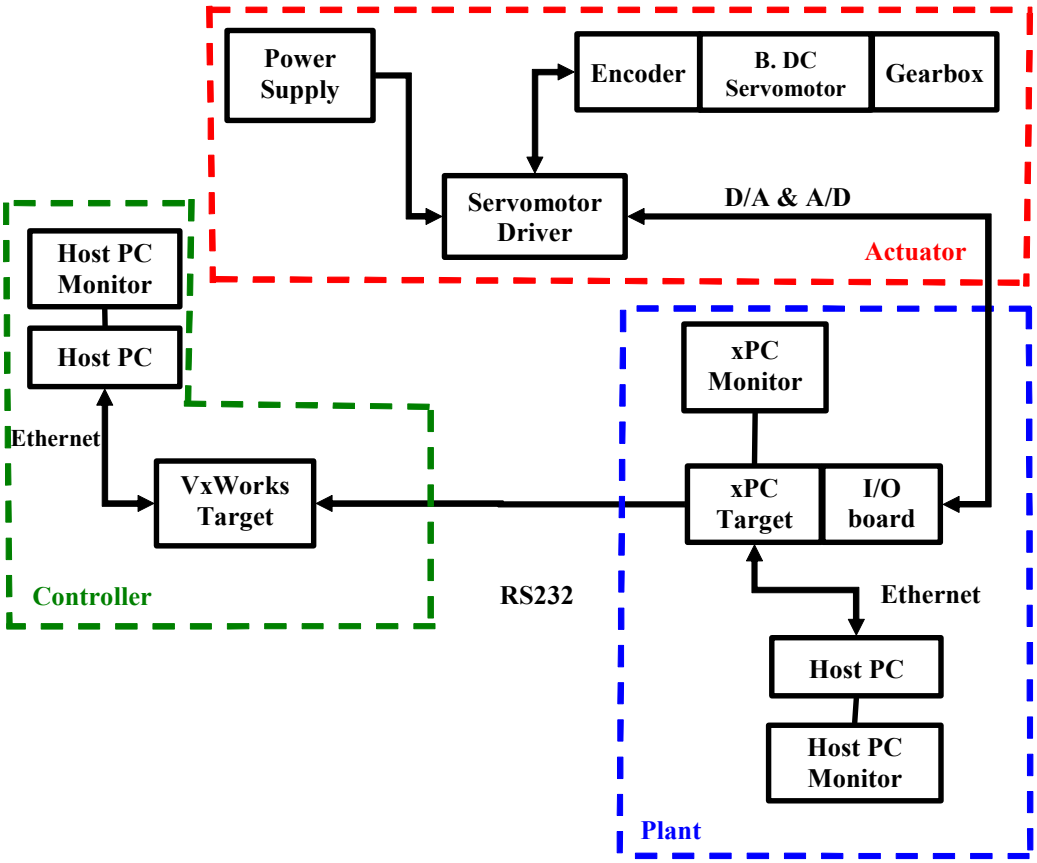

Figure 2.10 Final architecture of the test platform

CHAPTER 3

UCAV6 CHARACTERISTICS

3.1 Brief Information about UCAV

Unmanned aircraft are often thought of as relatively new inventions relying on advanced technology. However, the concept of the UAV was created by Leonardo Da Vinci in 1488. Despite Da Vinci's elaborate concept, the idea was not put into action until the spring of 1918 when the Sperry Company built two unmanned aircrafts [16]. Over twenty years later, during World War II, an improved version of the "Bug" became the first unmanned aircraft to use radio control guidance with a range of over 200 miles [11]. Figure 3.1 shows the earlier version of Kettering Bug UAV.

Figure 3.1 Kettering Bug, early UAV [11]

During the later years of the World War II, Germany used lethal UAVs (V-1's and V-2's); but until the Vietnam War era they were little more than full-sized remote controlled airplanes [17]. As UAVs offered smaller military budget (especially in training and operational cost) without risk to aircrews, their military role grew.

Ordinary UAV's were reformed and the name of the new class is known as Unmanned Combat Air Vehicle or UCAV for short. With their increasing popularity, European countries, including France, Italy, Belgium, Switzerland, Spain, Sweden, and Greece joined to develop an operational UCAV, named "Neuron" [18] and US aerospace companies including Teledyne Ryan, Boeing, Northrop, and Lockheed Martin started preliminary designs on Joint Strike Fighter (JSF) UCAV project [19]. The advantages of UCAV over manned aircraft are summarized in Table 3.1 [20] and future concerns are given in Table 3.2 [20].

Table 3.1 Advantages of future UCAV's over manned aircraft [20]

Vehicle Cost	Cheaper to build since pilot requirements such as cockpit controls and gauges, ejection seat, oxygen system, canopy, and cabin pressurization are unnecessary. Saves about 10% on overall vehicle cost.
Range and Endurance	Longer flight times and ranges due to less drag and better engine placement without the canopy and cockpit. No human limits on flight endurance time.
No Crew Risk	No political risk from casualties or POW's. Can employ non-lethal weapons to put an enemy to sleep such as acoustic or brain wave manipulation. Can operate aircraft in a nuclear, biological or chemical environment with no risk to the pilot.
Survivability	Unmanned design without canopy makes aircraft smaller and lowers radar cross section. No human limits to high 10g+ turns, which helps survival in missile avoidance maneuvers.
Training	Most training for UCAV operators done in simulators. No dependence on weather or maintenance ready aircraft. Periodic major exercise participation such as Red Flags to test unmanned doctrine alone or its interface with manned aircraft.
Training and Support Costs	With only periodic flight training and little to no maintenance on the majority of "stored" UCAV's, there is an order of magnitude reduction in peacetime training, fuel and maintenance support costs.
Personnel	Fewer pilots and support personnel needed. UCAV operators can fly numerous UCAV sorties sequentially or at the same time. With few training flights, less maintenance personnel and equipment is required.

Table 3.2 Future UCAV concerns [20]

Datalink Communications	1. Loss of control due to enemy jamming or signal manipulation 2. Long connectivity lapses due to distance, satellite location, or friendly mutual interference 3. Limited amount of frequency bandwidths to accommodate large numbers of secure links for multiple UCAV operations
Air Refueling	1 Transoceanic deployment distances and communications 2. Risk to KC-135 or KC-10 high value assets 3. Tanker join-up and multi-aircraft air refueling
Operator Situational Awareness	1. Number of aircraft per operator or operator per aircraft 2. ATC and enemy airspace deconfliction from other aircraft 3. Threat reactions, especially visual-only AAA or IR SAMs
Emergencies	1. Aircraft problems due to engine failure 2. No emergency mutual support or visual "battle damage checks" 3. Hung live ordnance procedures and recovery 4. UAV capable alternate airfield recovery due to fuel or weather

Due to the advantages mentioned above and the specialty of the company supporting this thesis study, a UCAV model is chosen as a plant model in this thesis.

3.2 Rationale Behind Choosing UCAV6 as the Plant Model

To check the platform, a controller model is needed in the Simulink environment. On the other hand, in order to design a controller, a plant model is required. However, the flexibility of the platform makes possible to choose any dynamic system model with its controller; for instance, a satellite model or a car model. An UCAV model is preferred as a plant model in this study, for the reasons cited in the previous section. The particular UCAV6 linear model is taken from the reference [88]. It was developed as a part of a basic research project with the Office of Naval Research. The reasons of its preference as the plant model are explained below clearly.

In open literature, the aircraft models are trimmed at one point. This means that the non-dimensional aerodynamic coefficients are given only at one velocity, altitude, and angle of attack value [64]. Moreover, trimming at one point is not enough to show the full performance of the platform, because gain-scheduling applications cannot be integrated in the setup demonstration. Consequently, a UCAV6 model is chosen since it is the most comprehensive model found in literature in this respect.

As explained in Section 1.2, the aerodynamic coefficients of aircrafts are not given in those thesis studies due to confidentiality reasons. Even if such aerodynamic data is given, there always exists a possibility of some intentional blurring made by authors, which makes that data non-trustable. On the other hand, the aerodynamic data provided in academic studies turn out to be more trustable since several of these studies only aim to improve the levels of technical knowledge and skills of students. The UCAV6 model is supported by both Texas University and National Aeronautics and Space Administration (NASA) Ames Research Center.

Alternatively, a generic aerodynamic UCAV model can be created by using the Digital Data Compendium (Datcom) software, which is a computer program to predict fixed-wing aircraft stability and handling properties. It takes an input file containing a geometric description of an aircraft, and outputs its corresponding dimensionless stability derivatives. More information about Digital Datcom can be found The USAF Stability and Control Digital Datcom [67] and [68]. However, this requires some additional work at a serious level, which is beyond the scope of this thesis. The main scope of this thesis is to create a reusable real time test platform for any given controller and plant model in MATLAB/Simulink environment.

3.3 UCAV6 Linear Model

The UCAV6 is a resized form of a AV-8B Harrier II Attack Fighter model with 0.75 scale. At the beginning, the Harrier (AV-8A) was an example of US-UK cooperation and Cold War defense achievements designing by the Boeing Company, British Aerospace, and Rolls-Royce. In 1975, the British Aerospace restarted development

of their second generation Harrier based on the US design. However, the Harrier II (AV-8B) is a much-refined aircraft as compared to its original. The major changes are reduction of weight, increase in engine lift, and the wing enlargement to produce greater lift while increasing fuel capacity and payload [76]. AV-8B Harrier II Specifications are given in Appendix E.

Figure 3.2 Atlantic Ocean (Nov. 18, 2004) An AV-8B II Harrier [76]

The UCAV6 is a representative of subsonic, vertical/short take-off and landing, unmanned combat air vehicle with medium altitude cruise and weapons delivery capabilities. The mission profile for this vehicle is shown in Figure 3.3. Furthermore, UCAV6 variables, their descriptions, and magnitudes are given in Table 3.3.

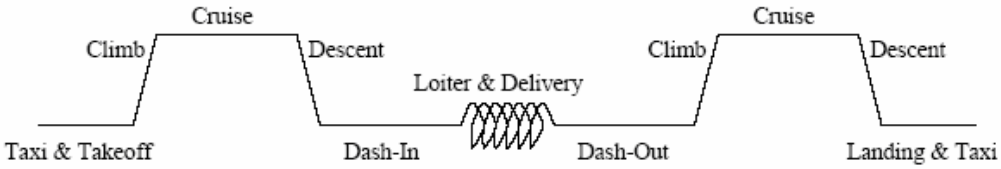

Figure 3.3 UCAV6 mission profile [72]

Table 3.3 UCAV6 variables with their descriptions and magnitudes [72]

Variable Name	**Description**	**Magnitude**
AR	Aspect ratio	4
Alat	3-D array containing all of the lateral/ directional state space A system matrices	4x4x12 matrix
Alon	3-D array containing all of the longitudinal state space A system matrices	4x4x12 matrix
Blat	3-D array containing all of the lateral/ directional state space B input matrices	4x2x12 matrix
Blon	3-D array containing all of the longitudinal state space B input matrices	4x3x12 matrix
I_{xx}	Mass moment of inertia about x-axis	11,435 slug-ft^2
I_{xz}	Mass product of inertia about xz-axes	0
I_{yy}	Mass moment of inertia about y-axis	13,246 slug-ft^2
I_{zz}	Mass moment of inertia about z-axis	29,240 slug-ft^2
S	Wing area	533 ft^2
b	Wing span	46.2 ft
cbar	Mean aerodynamic chord	11.54 ft
m	Aircraft mass	497,606 slugs
rho	Atmospheric densities for each flight condition [slug/ft^3]	1x12 vector
x0	Matrix whose columns contain steady state values for each flight condition.	Given in Table 3.4

In aerodynamics, an aircraft model is called *trimmed* if all moments in pitch, roll, and yaw are equal to zero. For trimming an aircraft model, the values of variables in nonlinear equations are found to satisfy the steady state condition at a given operating point.

The UCAV6 model was trimmed [72] for straight and level flight at 12 different flight conditions. There are 12 linear models corresponding to the flight conditions listed in Table 3.4. From the Table 3.5, linear models are obtained at 100 ft altitude for 8 increasing velocity values and at 5,000 ft for 4 increasing velocity values. Then, system matrices are decoupled for lateral and longitudinal mode. The UCAV6 model has 17 trim variables for both longitudinal and lateral flights. Table 3.5 summarizes the description of trim values.

Table 3.4 Description of the flight conditions [72]

Flight Condition Number	Velocity [ft/s]	Altitude H [ft]	Mach Number	Alfa [degree]	Stick Input [%]	Rudder Deflection [degree]	Nozzle Angle [degree]
1	10	100	0.009	0.701	39.446	89.567	88.8
2	25	100	0.022	0.955	39.492	90.149	86.0
3	50	100	0.045	2.565	37.598	90.977	82.0
4	75	100	0.067	5.786	36.099	90.931	80.0
5	100	100	0.090	6.649	33.886	89.970	80.0
6	150	100	0.134	11.692	28.661	82.867	70.0
7	200	100	0.179	18.515	65.724	56.260	0.0
8	250	100	0.224	12.026	57.600	45.753	0.0
9	300	5,000	0.273	9.728	53.575	49.225	0.0
10	500	5,000	0.446	4.044	47.941	62.651	0.0
11	700	5,000	0.638	1.607	39.182	77.315	0.0
12	900	5,000	0.820	0.525	35.250	88.366	0.0

Table 3.5 Description of UCAV6 trim variables [72]

Trim variable number	**Description**
1	u, velocity along x-axis of body frame [ft/s]
2	w, velocity along z-axis of body frame [ft/s]
3	q, pitch rate about y-axis of body frame [rad/s]
4	θ, Euler pitch attitude angle [rad]
5	H, geometric altitude [ft]
6	v, velocity along y-axis of body frame [ft/s]
7	p, roll rate about x-axis of body frame [rad/s]
8	r, yaw rate about z-axis of body frame [rad/s]
9	φ, Euler bank angle [rad]
10	Ψ, Euler heading angle [rad]
11	δ_s, longitudinal stick input [%]
12	δ_t, throttle input setting [%]
13	δ_n, engine nozzle angle [degree]
14	δ_a, left aileron deflection [degree]
15	δ_r, rudder deflection [degree]
16	δ_e, stabilator deflection [degree]
17	M, mach number

3.4 Concluding Remarks

In this chapter, the rationale behind choosing UCAV model as a plant model is given. Among several aircraft models, the UCAV6 is preferred because it is most comprehensive model found in literature. Alternatively, a generic aerodynamic UCAV model can be created, but that requires some additional work and this work is not in the scope of this thesis study.

The UCAV6 is resized form of a AV-8B Harrier II Attack Fighter model. Although, several aircraft models in open literature are trimmed at one point, UCAV6 model was trimmed at 12 operating points.

CHAPTER 4

PLANT MODELING AND CONTROLLER SYSTEM DESIGN

The general dynamic equations of motion for a rigid airplane are derived in literature, especially in aircraft flight dynamics books, automatic flight control books and in some graduate theses. Therefore, no attempt is made to repeat these derivations in this thesis. Instead, aircraft longitudinal equations of motion are taken directly on page 32 in reference [66].

4.1 Reference Coordinate Systems

In literature, mostly an Earth-fixed coordinate system is used for navigation algorithms, whereas the body-fixed coordinate system is used for aircraft performance equations. In this study, both coordinate systems are used in expressing the airframe dynamics.

4.1.1 Earth-Fixed Coordinate System

In this coordinate system, the origin is fixed to an arbitrary point on the surface of the Earth. The right-handed coordinate system is defined as X pointing North, Y pointing East, and Z pointing the center of the Earth. In this thesis study, the Earth-fixed coordinate system is regarded as an inertial coordinate system and the rotational velocity of the Earth is neglected.

4.1.2 Body-Fixed Coordinate System

In this coordinate system, the origin is located at the center of mass (CM) of the aircraft, x axis points forward out of the nose of the aircraft, y axis points out

through the starboard (right when looked from top) wing, and z axis points down. It is assumed that the aircraft is a rigid body, which means the distance between any two points on the aircraft remains constant during flight. With this assumption, its motion can be considered to have six degrees of freedom (6DOF). However, for the longitudinal motion, the flight is assumed wing level, straight line, and not accelerated.

4.2 Transfer Functions for Altitude Hold Mode

4.2.1 Longitudinal Equations of Motion

The motion of an aircraft in general has 6DOF, which means that the aircraft has the ability to move forward/backward, up/down, left/right, and furthermore rotate about three axes independently in 3-D space. However, in this study, since only the longitudinal motion is considered, its motion has only three degrees of freedom; namely, forward/backward, up/down, and rotation about pitch axis. The following aircraft equations of longitudinal motion are taken directly from reference [66].

$$\dot{u} = X_u u + X_{\dot{u}}\dot{u} + X_w w + X_{\dot{w}}\dot{w} + X_q q + X_{\dot{q}}\dot{q} - W_o q - g\cos\Theta_0\theta + X_{\delta e}\delta_e + X_{\dot{\delta}_e}\dot{\delta}_e \quad (4.1)$$

$$\dot{w} = Z_u u + Z_{\dot{u}}\dot{u} + Z_w w + Z_{\dot{w}}\dot{w} + Z_q q + Z_{\dot{q}}\dot{q} + U_o q - g\sin\Theta_0\theta + Z_{\delta_e}\delta_e + Z_{\dot{\delta}_e}\dot{\delta}_e \quad (4.2)$$

$$\dot{q} = M_u u + M_{\dot{u}}\dot{u} + M_w w + M_{\dot{w}}\dot{w} + M_q q + M_{\dot{q}}\dot{q} + M_{\delta e}\delta_e + M_{\dot{\delta}_e}\dot{\delta}_e \quad (4.3)$$

$$\dot{\theta} = q \quad (4.4)$$

These equations are written in the body-fixed coordinates where dots represent derivative with respect to time, u and w are the components of absolute velocity of body CM along x and z axes, q is the pitch rate defined about y axis, θ is the pitch angle defined as the rotation about the same axis, δ_e is the elevator deflection. The coefficients of the right hand side of these equations are called as the *stability derivatives*. Their meanings are given in Appendix F. From studying the aerodynamic data of a large number of aircraft, it became evident that not every stability derivative was significant and, frequently, a number can be neglected [66].

The stability derivatives, which are insignificant, are given on page 33 in reference [66]. Hence, the following stability derivatives are ignored in this study.

$$X_{\dot{u}}, X_q, X_{\dot{w}}, X_{\delta_e}, Z_{\dot{u}}, Z_{\dot{w}}, M_{\dot{u}}, Z_{\dot{\delta}_e} \text{ and } M_{\dot{\delta}_e} \tag{4.5}$$

After neglected these stability derivatives, the aircraft equations of the perturbed longitudinal motion can be expressed as:

$$\dot{u} = X_u u + X_w w + W_o q - g\cos\Theta_o\theta \tag{4.6}$$

$$\dot{w} = Z_u u + Z_w w + U_o q - g\sin\Theta_o\theta + Z_{\delta_e}\delta_e \tag{4.7}$$

$$\dot{q} = M_u u + M_w w + M_{\dot{w}}\dot{w} + M_q q + M_{\delta e}\delta_e \tag{4.8}$$

$$\dot{\theta} = q \tag{4.9}$$

Each term in Equations (4.6), (4.7), and (4.8) is an acceleration term, but the some stability derivatives appearing in these equations are dimensional, since u and w have units as [m/s], and q as [1/s] [66].

4.2.2 System Matrices of Longitudinal Motion

The state variable representation of the longitudinal motion considering only a single control input, δ_e can be written as

$$\dot{x} = Ax + Bu \tag{4.10}$$

where the state vector is defined as

$$x \equiv \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} \tag{4.11}$$

and the input vector is defined as

$$u \equiv [\delta_e] \tag{4.12}$$

The system matrix, A, and the input matrix, B, are given in the reference [66] as

$$A = \begin{bmatrix} X_u & X_w & 0 & -g \\ Z_u & Z_w & U_o & 0 \\ \overline{M}_u & \overline{M}_w & \overline{M}_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.13}$$

$$B = \begin{bmatrix} X_{\delta_e} \\ Z_{\delta_e} \\ \overline{M}_{\delta_e} \\ 0 \end{bmatrix} \tag{4.14}$$

where

$$\overline{M}_u = (M_u + M_{\dot{w}} Z_u) \tag{4.15}$$

$$\overline{M}_w = (M_w + M_{\dot{w}} Z_w) \tag{4.16}$$

$$\overline{M}_q = (M_q + U_0 M_{\dot{w}}) \tag{4.17}$$

$$\overline{M}_{\delta_e} = (M_{\delta_e} + M_{\dot{w}} Z_{\delta_e}) \tag{4.18}$$

4.2.3 Transfer Functions of Longitudinal Dynamics

The transfer function between the pitch angle attitude θ and the elevator deflection δ_e can then be obtained as

$$\frac{\theta(s)}{\delta_e(s)} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} sI - A \end{bmatrix}^{-1} B \tag{4.19}$$

The expression of this transfer function can be written as [64]

$$\frac{\theta(s)}{\delta_e(s)} = \frac{N_\theta(s)}{D(s)} \tag{4.20}$$

D introduces the denominator of the Equation (4.20) and the notation N_θ introduces the numerator.

$$N_\theta(s) = A_\theta s^2 + B_\theta s + C_\theta \tag{4.21}$$

where the constants A_θ, B_θ and C_θ are

$$A_\theta = Z_{\delta_e} M_{\dot{\alpha}} + M_{\delta_e}(U_1 - Z_{\dot{\alpha}}) \tag{4.22}$$

$$\begin{aligned} B_\theta = {} & X_{\delta_e}\left(Z_u M_{\dot{\alpha}} + (U_1 - Z_{\dot{\alpha}})(M_u + M_{T,u})\right) \\ & + Z_{\delta_e}\left((M_\alpha + M_{T,\alpha}) - M_{\dot{\alpha}}(X_u + X_{T,u})\right) \\ & + M_{\delta_e}\left(-Z_\alpha - (U_1 - Z_{\dot{\alpha}})(X_u + X_{T,u})\right) \end{aligned} \tag{4.23}$$

$$\begin{aligned} C_\theta = {} & X_{\delta_e}\left((M_\alpha + M_{T,\alpha})Z_u - Z_\alpha(M_u + M_{T,u})\right) \\ & + Z_{\delta_e}\left(-(M_\alpha + M_{T,\alpha})(X_u + X_{T,u}) + X_\alpha(M_u + M_{T,u})\right) \\ & + M_{\delta_e}\left(Z_\alpha(X_u + X_{T,u}) - X_\alpha Z_u\right) \end{aligned} \tag{4.24}$$

4.2.4 Transfer Functions between Altitude and Elevator Deflection

One of the purposes of this thesis is to design altitude hold controller to verify the platform functionality. By the help of the elevator deflection, aircraft can be kept at a fix altitude. In this section, the transfer function between elevator deflection and altitude is derived.

Referring to Figure 4.1, the rate of climb, $\dot{h}$, of the airplane which is the velocity in the Z axis of the Earth-fixed coordinate system can be expressed as

$$\dot{h} = U_1 \sin(\gamma) \tag{4.25}$$

where γ is flight path angle and U_1 is the component of the airplane velocity along flight path. Assuming that the flight path angle is small, this expression can be approximated as

$$\dot{h} = U_1 \gamma \tag{4.26}$$

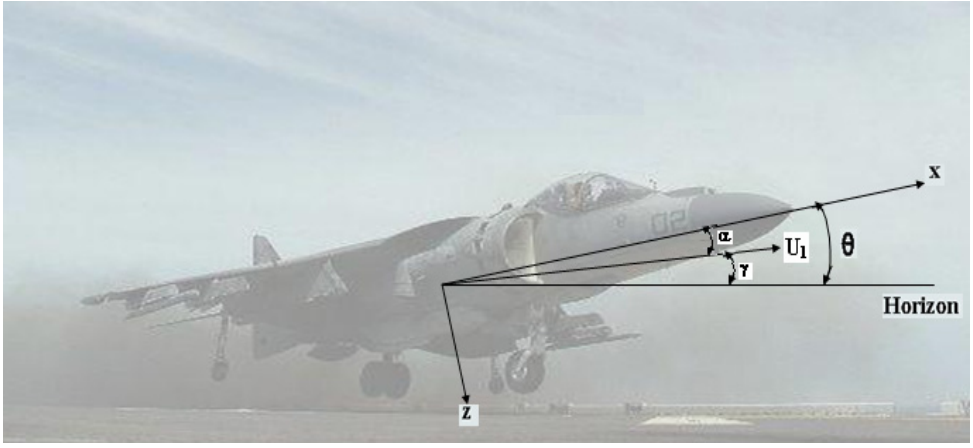

Figure 4.1 Flight path geometry

Taking Laplace transformation of Equation (4.26) with zero initial conditions gives

$$sh(s) = U_1\gamma(s) \tag{4.27}$$

Dividing both sides of Equation (4.27) by s $\delta_e(s)$, the transfer function between the altitude and the elevator deflection is obtained as

$$\frac{h(s)}{\delta_e(s)} = \frac{U_1}{s}\left(\frac{\gamma(s)}{\delta_e(s)}\right) \tag{4.28}$$

It can be seen from Figure 4.1 that, γ can be expressed in terms of θ and α as

$$\gamma = \theta - \alpha \tag{4.29}$$

By substituting Equation (4.29) into Equation (4.28), the following expression is obtained.

$$\frac{h(s)}{\delta_e(s)} = \frac{U_1}{s}\left(\frac{\theta(s) - \alpha(s)}{\delta_e(s)}\right) \tag{4.30}$$

By the help of the relation between angle of attack α and w given on page 45 in reference [66]

$$\alpha = w / U_o \tag{4.31}$$

U_0 denotes the trim or equilibrium value of the airplane forward velocity and w denotes perturbed value of the airplane velocity along z-axes. The following transfer function expression is obtained.

$$\frac{h(s)}{\delta_e(s)} = \frac{U_1}{s}\left(\frac{\theta(s)}{\delta_e(s)} - \frac{w(s)}{U_0\delta_e(s)}\right) \tag{4.32}$$

The $\frac{U_1}{s}$ transfer function brings one more pole to the system, which is located at the origin of the s plane. Therefore, the order of the altitude to elevator deflection transfer function is become five.

4.3 Plant Model Design

The transfer functions of altitude to elevator deflection of UCAV6 at 12 operating points are calculated by using the Equation (4.32) for each operating point and they are given below. The system has five open-loop poles and three open loop zeros. The plant properties including open-loop poles, zeros, gains, velocity error coefficients, damping ratios and undamped natural frequencies for each operating point are given in Appendix C.

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_1} = \frac{0.2\,s^3 + 0.3133\,s^2 - 7.165\,s - 0.1045}{s^5 + 1.794\,s^4 + 2.962\,s^3 + 0.0902\,s^2 + 0.03\,s} \tag{4.33}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_2} = \frac{0.4995\,s^3 + 0.7492\,s^2 - 54.64\,s - 0.3746}{s^5 + 2.3\,s^4 + 3.763\,s^3 + 0.162\,s^2 + 0.066\,s} \tag{4.34}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_3} = \frac{1.809\,s^3 + 1.474\,s^2 - 329.2\,s - 2.393}{s^5 + 2.67\,s^4 + 4.756\,s^3 + 0.2092\,s^2 + 0.0853\,s} \tag{4.35}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_4} = \frac{3.01\,s^3 + 3.651\,s^2 - 935.6\,s - 10.23}{s^5 + 3.263\,s^4 + 6.976\,s^3 + 0.3031\,s^2 + 0.1299\,s} \tag{4.36}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_5} = \frac{5.43\,s^3 + 5.818\,s^2 - 2269\,s - 21.59}{s^5 + 3.652\,s^4 + 8.731\,s^3 + 0.396\,s^2 + 0.171\,s} \tag{4.37}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_6} = \frac{10.07\,s^3 + 5.712\,s^2 - 4455\,s - 45.23}{s^5 + 3.836\,s^4 + 9.54\,s^3 + 0.525\,s^2 + 0.2475\,s} \tag{4.38}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_7} = \frac{14.3\,s^3 + 55.35\,s^2 - 7486\,s - 94.1}{s^5 + 3.92\,s^4 + 9.8\,s^3 + 0.657\,s^2 + 0.3487\,s} \tag{4.39}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_8} = \frac{15.93\,s^3 + 23.69\,s^2 - 1.165e004\,s - 155.7}{s^5 + 4.202\,s^4 + 10.21\,s^3 + 0.8645\,s^2 + 0.4896\,s} \tag{4.40}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_9} = \frac{17.78\,s^3 + 15.54\,s^2 - 1.668e004\,s - 237}{s^5 + 4.96\,s^4 + 11.28\,s^3 + 1.049\,s^2 + 0.565\,s} \tag{4.41}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_10} = \frac{49.75\,s^3 + 181.1\,s^2 - 4.901e004\,s - 724.2}{s^5 + 6.002\,s^4 + 15.69\,s^3 + 1.603\,s^2 + 0.8502\,s} \tag{4.42}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_11} = \frac{107.5\,s^3 + 180.4\,s^2 - 1.154e005\,s - 2586}{s^5 + 7.235\,s^4 + 21.2\,s^3 + 2.28\,s^2 + 1.267\,s} \tag{4.43}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_12} = \frac{199.8\,s^3 + 230.4\,s^2 - 2.241e005\,s - 6683}{s^5 + 8.075\,s^4 + 26.1\,s^3 + 3.804\,s^2 + 2.291\,s} \tag{4.44}$$

Root locus plot shows closed-loop pole locations of the system when a proportional controller is employed. Figure 4.2 give the root locus diagram of UCAV6 at operating point 8 as a sample.

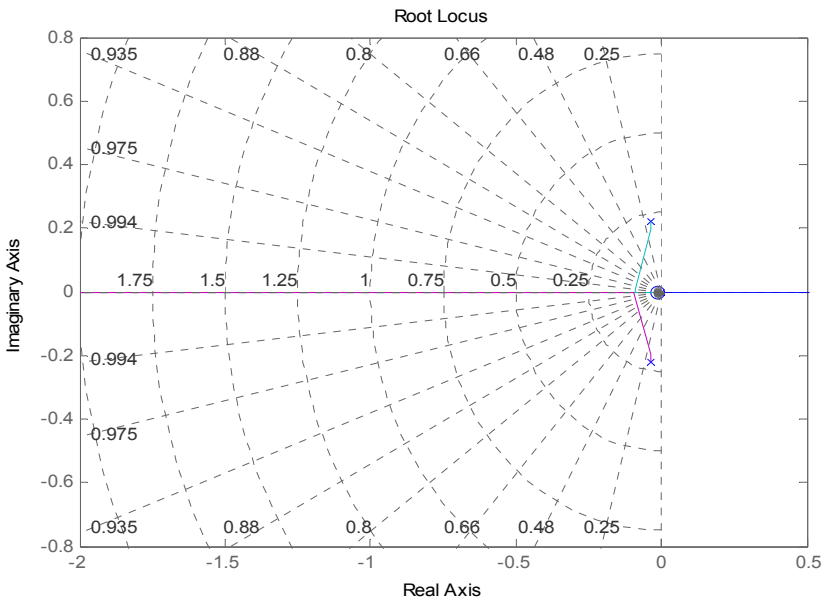

Figure 4.2 Root locus diagram at operating point 8

<u>Normalization of Transfer Functions:</u>

Transfer functions can be normalized in two ways. One way is to make the coefficients of maximum degree of s equal to one, express the numerator and denominator in factored first and/or second order forms, and take the open loop gain outside the expression. Below, the transfer function between the altitude and elevator deflection at operating point 8 is normalized by this method

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_8} = \frac{15.9312\,(s+27.79)\,(s-26.32)\,(s+0.01336)}{s\,(s^2+0.06674s+0.04953)\,(s^2+4.135s+9.884)} \tag{4.45}$$

The other normalization way is that making the coefficients of the smallest powers of s in both numerator and denominator to one. Then the velocity error coefficient can be found as the inverse of the resulting gain, which is found by using the first method. The following transfer function is obtained by using this normalization way for operating point 8.

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_8} = \frac{-318.014\left(-0.1023\,s^3 - 0.1522\,s^2 + 74.84\,s + 1\right)}{s\left(2.042\,s^4 + 8.583\,s^3 + 20.85\,s^2 + 1.766\,s + 1\right)} \tag{4.46}$$

Normalized forms of the transfer function of attitude to elevator deflection are obtained by using the first normalization way.

Discussion and Comparison about h(s)/δe(s) Transfer Functions and Plant Properties

The transfer functions between the altitude and elevator deflection are also calculated for F-104, F-4, Learjet24. These aircraft are preferred because they are all fighter aircraft like Harrier. The transfer functions of these aircraft are obtained by using the aerodynamic data given in reference [64], on pages 563, 570, and 555, respectively. The plant properties and the altitude to elevator deflection transfer function of F-104, F-4, Learjet24 can be found in Appendix F. This aircraft model is chosen because it is trimmed at three operating points. The plant properties of RC aircraft can be monitored in Appendix G, which is taken from the reference [12]. The following results are obtained after examining all plant properties.

- One of the open loop poles is located at the origin of the complex s-plane.
- The remaining four open loop poles are clustered as two pairs of complex conjugate pairs. One pair is named as *phugoid* poles. The other is named as *short period* poles.
- The phugoid pair is located about 85 times for UCAV6 (it changes from operating point to operating point irregularly), 20 times for F-104, 16 times for F-4, 60 times for Learjet24, 65 times for RC aircraft farther away from the imaginary axis of the complex s-plane than the short period pair.
- The phugoid poles go away from the imaginary axis as the airspeed increases.
- For UCAV6, open loop gain values increase with the airspeed exponentially. The gains take values from -3, which is the gain at operating point one, to -2,917, at operating point 12. Also for RC aircraft, open loop gain values raise rapidly depending on velocity.

The equivalent short period damping ratio, ζ_{SP}, shall be within the limits given in Table 4.1 [77]. As it is realized, for level flight 1 the short period damping ratio, ζ_{SP}, shall be between 0.35 and 1.30. The short period damping ratio of UCAV6, ζ_{SP}, increases very slowly as the airspeed increases and they take values between 0.5 and 0.8. Moreover, the short period damping ratio of F-104, F-4 and Learjet24 are 0.31, 1.00, and 0.56 respectively.

Table 4.1 Short period damping ratio limits [77]

Level	Category A and C Flight Phases		Category B Flight Phases	
	Minimum	Maximum	Minimum	Maximum
1	0.35	1.30	0.30	2.00
2	0.25	2.00	0.20	2.00
3	0.15*	-	0.15*	-

* May be reduced at altitudes above 20,000 ft if approved by the procuring activity [77].

According to the reference [77], the damping ratios of phugoid mode should be higher than 0.04 for level one flight, whereas for level two flight, it should be higher than zero. The definition of the flight level one is that the flying qualities are completely adequate for the particular flight phase [66].

For level flight one, the phugoid mode damping ratio shall be at least 0.04. The damping ratios of the phugoid mode, ζ_P, of UCAV6 change between 0.12 and 0.2. They arise moderately as airspeed increase. Similarly, damping ratio of the phugoid mode of F-104, F-4, and Learjet24 are 0.14, 0.22, and 0.06 respectively. Moreover, for RC aircraft, it increases from 0.22 to 0.3 in a slow manner.

If the separation between the frequencies of the phugoid mode and short period mode is small, some problems in handling qualities may arise. If the ratio $\omega_{ph} / \omega_{sp}$ is smaller than the value 0.1, there may be some trouble with the handling qualities [66], which means $10\omega_{ph} < \omega_{sp}$

Except the natural frequency of Learjet24, the natural frequencies of the other A/C (F-4, F-104, RC and UCAV6 at all operating point) satisfy the specified condition. For example, UCAV6 at operating point 8, ω_{sp} = 3.144 rad/s and ω_{ph} = 0.223 rad/s. In addition, $10\omega_{ph}$ = 2.23 rad/s < ω_{sp} =3.144 rad/s. The velocity error coefficient increases as the airspeed increase, which indicates a better steady state tracking. For UCAV6, it begins from -3 and it is -2,917 at operating point 12. Similar ascent can be seen among RC operating point's gain. The velocity error coefficient at operating point 12, which is -2,917, is acceptable when considering its trim condition, because F-4 is trimmed at 35,000 ft with 518 knots and its velocity error coefficient is 1,177.

Except F-4 and Learjet24, the velocity error coefficients sign of F-104, RC airplane and UCAV6 are negative. The sign changes with the design criteria of the A/C. The sign convention used in aerospace for the elevator deflection δ_e, for the rudder deflection δ_r and for the aileron deflection δ_a are shown in Figure 4.3.

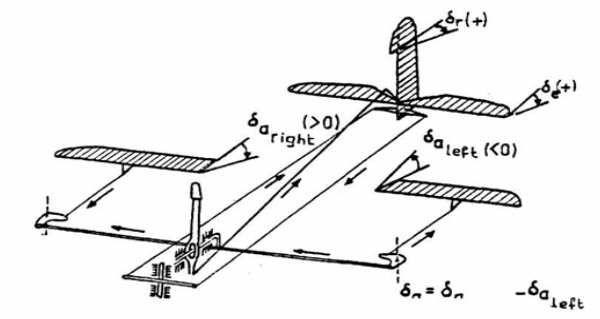

Figure 4.3 The sign convention of the control surface deflection [79]

In literature, the sign convention shown in Figure 4.4 is verified. Unless it is specified otherwise, one should assume the following sign conventions: [79], [69]

- Rudder and/or Vertical Stabilizer Deflection - Trailing edge left is positive
- Flaps Deflection - Trailing edge down is positive
- Aileron Deflection – Left trailing edge down is positive
- Stabilizer Deflection - Trailing edge down is positive
- Elevator Deflection - Trailing edge down is positive

The reason of this assumption is that a negative elevator deflection. It causes drag on tail and that creates positive (a nose-up) pitching moment with respect to center of gravity. Thus, an elevator deflection that causes a nose-up pitching moment on the aircraft is defined to be a negative elevator deflection. It is the reason of negative open loop gains for the UCAV6 and the other aircrafts.

Direct Form II

The flight model, UCAV6, is trimmed at 12 different operating points with respect to altitude and velocity. The coefficients of transfer functions are changed at each operating point as expected. That means, the state space matrices during the flight plan change with time, i.e., A(t), B(t) as these coefficients change with time. To insert these state space matrices in Simulink environment to simulate UCAV6 plant model, "Direct Form II" in other words "C.C.Form" method is used. Further information about Direct Form II Method is given in Appendix I. The altitude to elevator deflection transfer function at operating point 8 is given in Equation (4.40). To convert a transfer function into direct II form, the procedure given below is followed by using the transfer function at operating point 8.

$$\left.\frac{h(s)}{\delta_e(s)}\right)_{OP\_8} = \frac{15.93\,s^3 + 23.69\,s^2 - 1.165e004\,s - 155.7}{s^5 + 4.202\,s^4 + 10.21\,s^3 + 0.8645\,s^2 + 0.4896\,s} \tag{4.40}$$

For UCAV6, the general form of $\frac{h(s)}{\delta_e(s)}$ is,

$$\frac{H(s)}{\delta_e(s)} = \frac{as^3 + bs^2 + cs + d}{s^5 + es^4 + fs^3 + gs^2 + hs} \tag{4.47}$$

As defined in Appendix I, the general form of the transfer function is as in direct form II

$$\frac{u(s)}{e(s)} = \frac{b_0 + b_1 s^{-1} + b_2 s^{-2} + b_3 s^{-3} + b_4 s^{-4} + b_5 s^{-5}}{1 + a_1 s^{-1} + a_2 s^{-2} + a_3 s^{-3} + a_4 s^{-4} + a_5 s^{-5}} \tag{4.48}$$

Multiplying numerator and denominator of Equation (4.48) by $\frac{1}{s^5}$, gives

$$\frac{h(s)}{\delta e(s)} = \frac{1/s^5}{1/s^5} \times \frac{as^3 + bs^2 + cs + d}{s^5 + es^4 + fs^3 + gs^2 + hs} = \frac{as^{-2} + bs^{-3} + cs^{-4} + ds^{-5}}{1 + es^{-1} + fs^{-2} + gs^{-3} + hs^{-4}} \tag{4.49}$$

By comparing Equation (4.48) with Equation (4.49), the following relations are obtained.

$$b_0 = 0, \quad b_1 = 0, \quad b_2 = a, \quad b_3 = b, \quad b_4 = c, \quad b_5 = d \tag{4.50}$$

$$a_1 = e, \quad a_2 = f, \quad a_3 = g, \quad a_4 = h, \quad a_5 = 0 \tag{4.51}$$

The coefficients of the transfer functions obtained by using direct II method are embedded into look-up tables as shown in Figure 4.4. Velocity and altitude values are the scheduling values of these look-up tables.

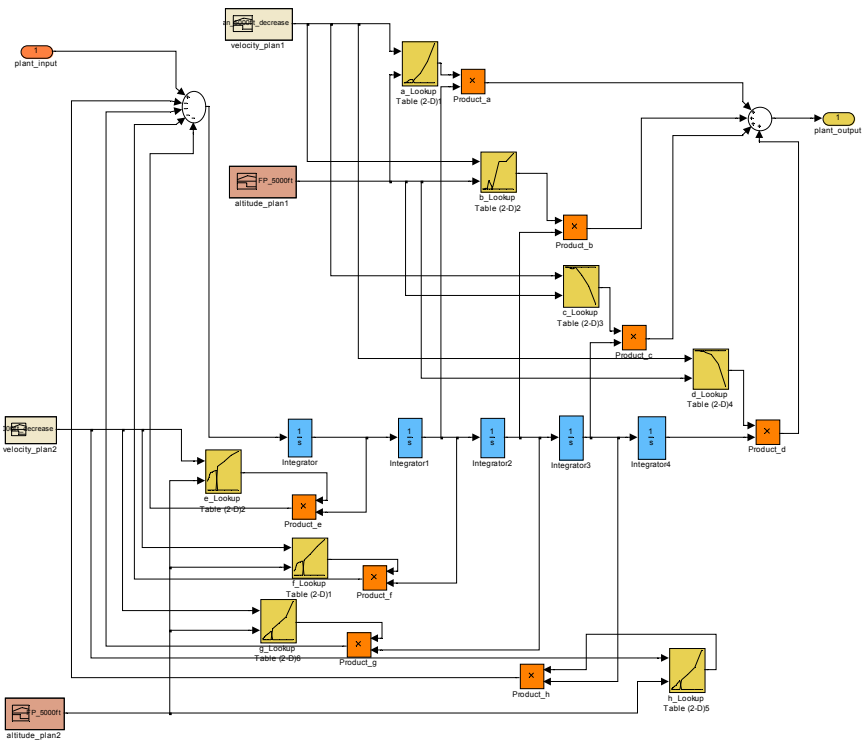

Figure 4.4 UCAV6 plant model in direct II form

4.4 Controller Design

The system matrix A for longitudinal motion is of size 4, indicating that it has 4 eigenvalues. An aircraft is dynamically stable if all its eigenvalues, λ, have negative real parts. However, it becomes dynamically unstable if any of its eigenvalues assumes a positive real part.

For the majority of the aircraft, the characteristic equation can be written as the multiplication of two quadratic factors as.

$$\left(\lambda^2 + 2\zeta_{ph}\omega_{ph}\lambda + \omega_{ph}{}^2\right)\left(\lambda^2 + 2\zeta_{sp}\omega_{sp}\lambda + \omega_{sp}{}^2\right) = 0 \tag{4.52}$$

where;

- ζ_{ph} denotes the damping ratio of long period (phugoid) mode
- ω_{ph} denotes the undamped natural frequency of long period (phugoid) mode
- ζ_{sp} denotes the damping ratio of short period mode
- ω_{sp} denotes the undamped natural frequency of short period mode

These two factors refer to two different oscillatory modes; namely, the short period mode and the long period mode (phugoid mode). The short period mode is an oscillation of the pitch of the aircraft at mainly a constant speed while the phugoid mode is an oscillation of the pitch at mainly a constant angle of attack. Table 4.2 shows the properties of short period and phugoid modes.

Table 4.2 Short period and phugoid mode properties

Mode	Damping ratio	Frequency
phugoid	Very small Sometimes even negative (so, oscillation grows slowly with time, that is unstable)	Very low (long period)
short period	High	High (short period)

The relatively well-damped motion is associated with short period mode. Here, the controller design process is presented for only the flight condition 8. The desired response can be achieved by changing overall system behavior. It is possible with altering the characteristics of the each of proportional (P), the integral (I), and the derivative (D) controls.

PID Control

A proportional-integral-derivative controller (PID controller) is a generic control loop feedback mechanism widely used in industrial control systems [66]. The transfer function of the PID controller can be expressed as

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s} = \frac{K_i (s - z_1)(s - z_2)}{s} \tag{4.53}$$

which indicates that it introduces a pole at the origin and two zeros at z_1 and z_2 by adjusting its parameters. In the time domain, the input signal produced by the PID controller is expressed as

$$u(t) = K_P e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt} \tag{4.54}$$

where the variable e(t) represents the tracking error, as the instantaneous difference between the desired value of the output and its actual (measured) value. Briefly, the typical characteristics of PID controllers can be shown in Table 4.3.

Table 4.3 Characteristics of PID controller

Close loop response	**Rise time**	**Overshoot**	**Settling time**	**Steady state error**
K_p	Decrease	Increase	Increase	Decrease
K_i	Decrease	Increase	Increase	Decrease or Eliminate
K_d	Increase	Decrease	Decrease	No change

Proportional control part is used for reducing the rise time, but it never eliminates the steady-state error, completely. To eliminate the steady-state error, the integral control part is used. However, integral control part causes overshoot and settling time is worse. To improve the relative stability of the system, the derivative control part should be used. It reduces the overshoot, and improves the transient response. Changing one of them change the effect of the other two [84].

The use of a PID controller is preferred in this thesis, because it

- has a simple structure,
- has a robust performance in a wide range of operating conditions,
- is widely used in industrial control systems,
- is widely used in unmanned control processes [12] [13] [82].

Moreover, the aim of the thesis study is not to design a complex controller by using advance control methods. A sufficient controller designed in Simulink environment is enough to verify the platform.

The Simulink Response Optimization block set is used for designing PID controller at each operating point. To assist in design of control systems, a graphical user interface (GUI) is supplied by this blockset. Controller parameters (K_p, K_i, K_d) can be tuned within a nonlinear Simulink model to meet time-domain performance requirements. These requirements are entered in this GUI by either graphically placing constraints within a time-domain window or tracking and closely matching a reference signal.

The Signal Constraint block in the Simulink diagram should be added as shown in Figure 4.5 and connected with a block whose outputs signal wanted to control. Simulink Response Optimization blockset converts time-domain constraints into a constrained optimization problem. Then it solves the problem using optimization routines taken from the Optimization Toolbox. Tuned parameters are changed iteratively with checking and comparing the result of simulation. At the end of all iterations and optimizations, PID controller gains are tuned for each operation points.

During the optimization, the system performance requirements are taken as;

- Overshoot < 10%
- Settling time [5%] < 10 sec
- Rise time [90%] < 5 sec

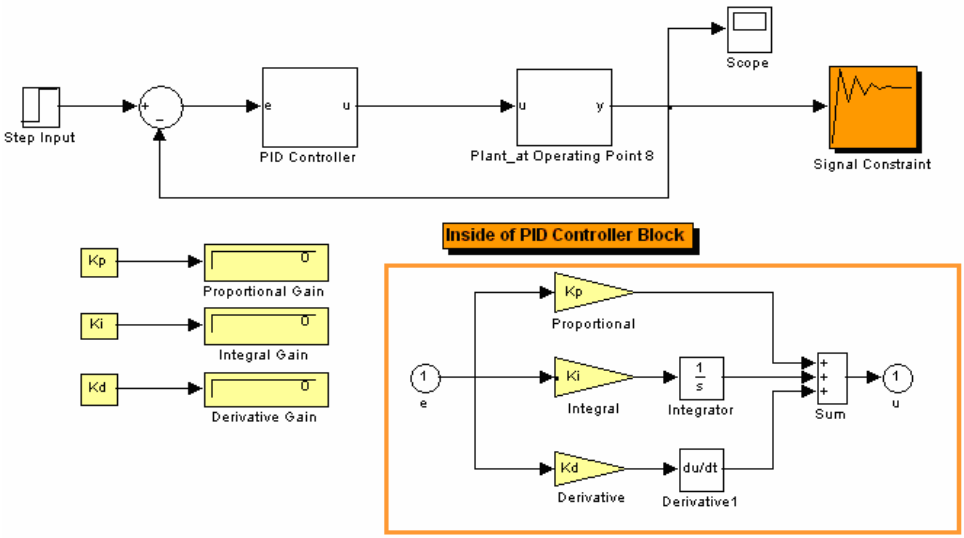

Figure 4.5 PID gain parameter tuning at operating point 8

In this thesis study, all initial values of the gains are taken zero for the first optimization cycle. Then, the optimization is repeated by taking these first optimization results as initial values. Figure 4.6 shows the plant output signals response at operating point 8.

Continuous PID controller gain parameters at the other operating points are obtained by following the same method under the same constraints. These gain parameters are given in Table 4.4 and the following figures show the variation of PID continuous gains. Figure 4.7 shows continuous PID gains at entire operating points and the next two figures (Figure 4.8 and Figure 4.9) show the gains at 100 ft and 5,000 ft separately.

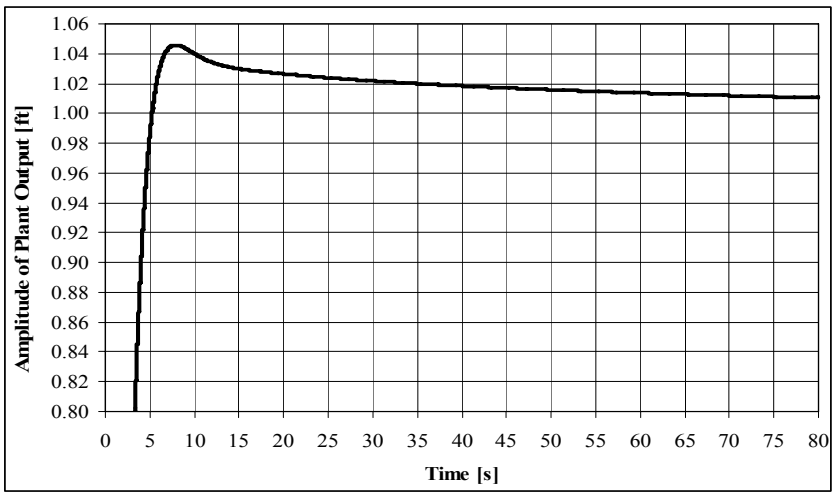

Figure 4.6 Plant output signal response at operating point 8

Table 4.4 Continuous PID controller gains at each operating point

OP#	Velocity [ft/s]	Altitude [ft]	K_p[rad/ft]	K_i[rad/ft.s]	K_d [rad.s/ft]
1	10	100	1.03E-01	2.30E-03	2.89E-01
2	25	100	2.49E-02	3.65E-04	6.36E-02
3	50	100	5.00E-03	1.41E-04	1.13E-02
4	73	100	5.50E-03	1.09E-04	9.70E-03
5	100	100	2.98E-03	4.41E-06	5.17E-03
6	143	100	1.90E-03	7.29E-05	3.40E-03
7	185	100	7.61E-04	7.75E-06	1.30E-03
8	237	5,000	2.55E-04	2.26E-05	7.10E-04
9	288	5,000	3.60E-03	1.84E-05	7.10E-03
10	479	5,000	2.12E-04	4.40E-06	3.51E-04
11	668	5,000	5.11E-05	3.29E-06	1.09E-04
12	854	5,000	3.17E-05	3.97E-06	7.66E-05

After all gain values are obtained at each operating point, they are embedded into lookup tables for generating controller codes. The continuous PID controller model is run in Simulink environment with a fixed step solver. Although, this model can be converted into C codes by using RTWEC for a generic real time target, it cannot converted C codes for VxWorks real time target. The error message taken during coding application for VxWoks is shown in Figure 4.10.

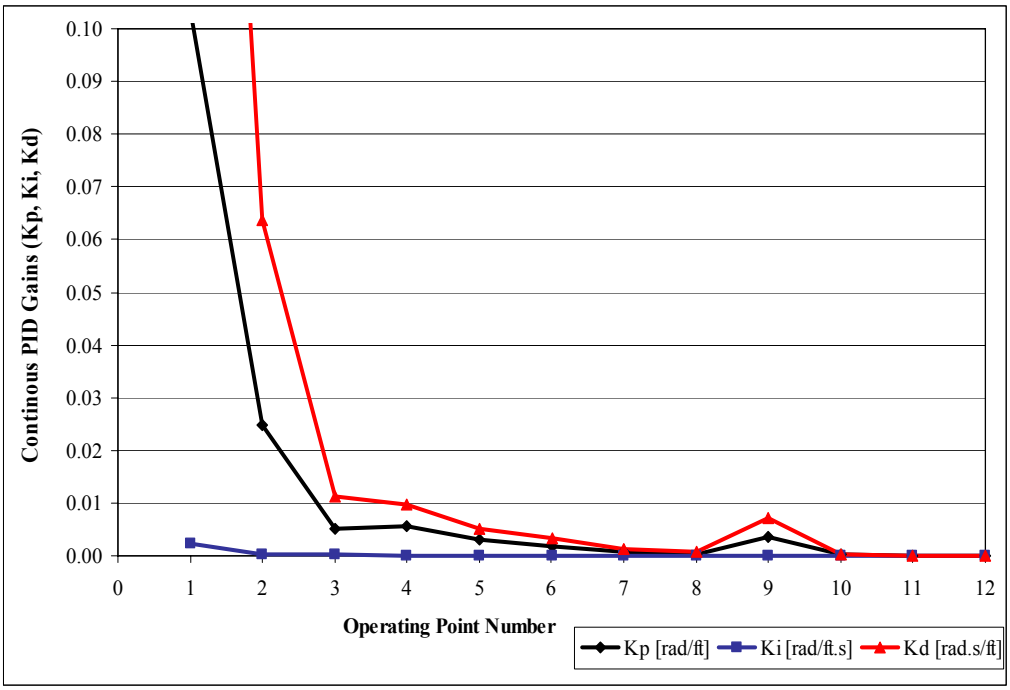

Figure 4.7 Continuous PID gains for the entire operating points

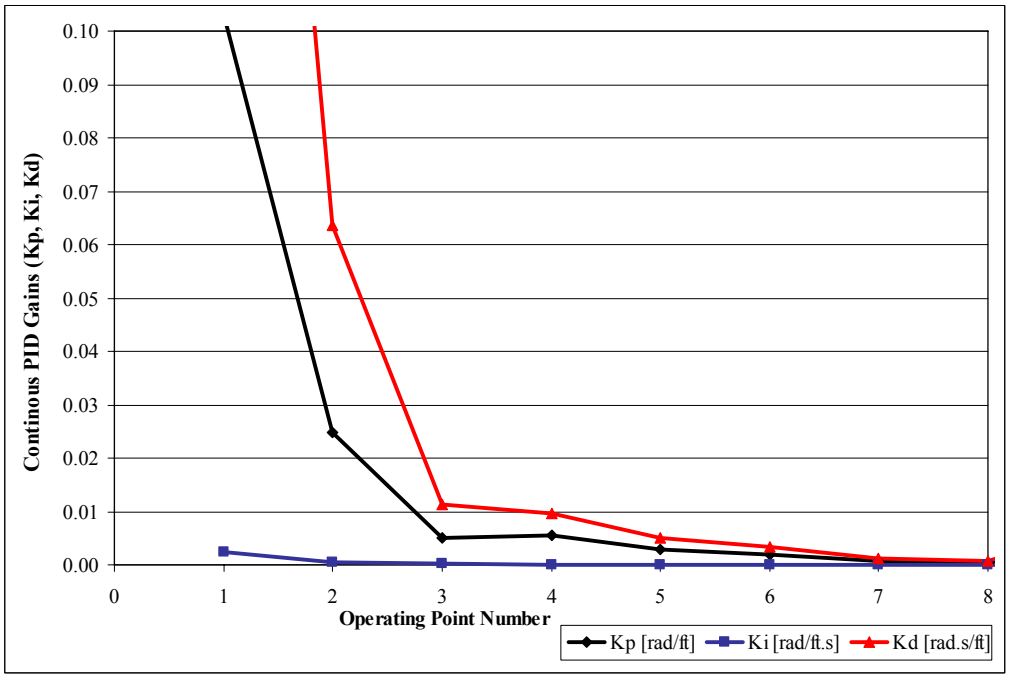

Figure 4.8 Continuous PID gains at 100 ft

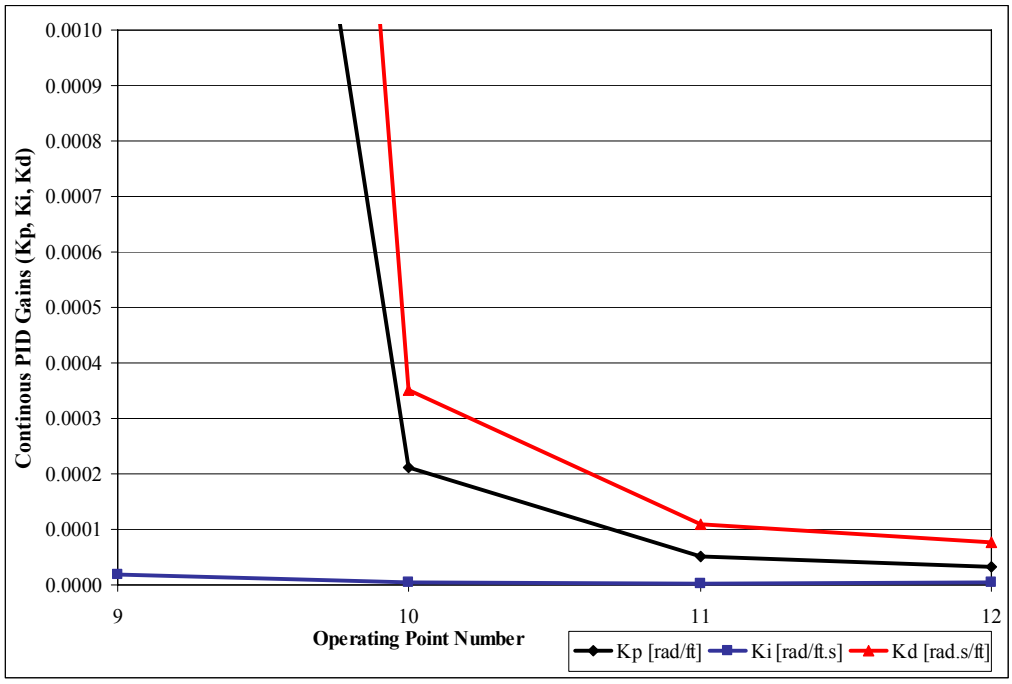

Figure 4.9 Continuous PID gains at 5,000 ft

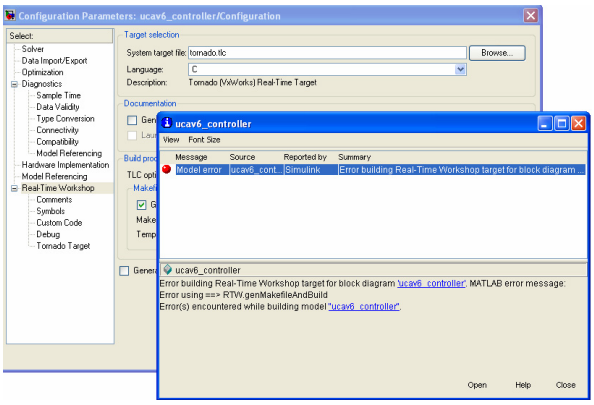

Figure 4.10 Error message taken when coding continuous PID controller

A discrete PID controller should be designed to convert the continuous PID controller model by the help of RTWEC into C codes, which are suitable for VxWorks operating system. The sample time of the discrete PID controller is chosen as 40 ms, because several autopilots work in between 20 ms and 50 ms range.

For example, autopilots of both Sikorsky S-70 Black Hawk Helicopter and Boeing 747 work at 50 ms sample time. To control a dynamic system, a controller system dynamics should be more rapid than plant dynamics. Approximately 5 times more rapid controller is enough to satisfy this requirement. The discrete PID controller with 40 ms sample time is convenient for the UCAV6 plant model, because its maximum undamped natural frequency is about 5 [rad/s]. For a five times more rapid controller system design, the sample time should be minimum 0.04 seconds (T_s = (1/5)*(1/5) = 0.04 second).

Discrete PID Controller

There are several ways for mapping the transfer functions from the s-domain to z-domain such as forward difference approximation, backward difference approximation, and bilinear (Tustin) approximation, etc.. In this thesis study, the bilinear (Tustin) transformation method is applied in order to design the discrete PID controller because it is common use in designing discrete PID controllers. In this method, the transfer functions are converted from s-domain to z-domain by using the following approximation

$$z = e^{sT_1} \approx \frac{1 + sT_s/2}{1 - sT_s/2} \tag{4.55}$$

The discretization $H_d(z)$ of a continuous transfer function H(s) is derived by

$$H_d(z) = H(s') \tag{4.56}$$

where

$$s' = \frac{2}{T_s}\frac{z-1}{z+1} \tag{4.57}$$

where T_s is the sampling time of discrete system In the continuous PID controller design. Using this transformation, the transfer function of the discrete PID controller is obtained as below.

$$G_c(z) = K_p + K_i \frac{Ts}{2}\frac{z+1}{z-1} + K_d \frac{2}{Ts}\frac{z-1}{z+1} \tag{4.58}$$

or

$$G_c(z) = \frac{\left(K_p + \frac{K_i Ts}{2} + \frac{2K_d}{Ts}\right)z^2 + \left(K_i Ts - \frac{4K_d}{Ts}\right)z + \left(-K_p + \frac{K_i Ts}{2} + \frac{2K_d}{Ts}\right)}{z^2 - 1} \tag{4.59}$$

The parameters of this discrete PID controller are tuned by following the same procedure followed in the continuous PID controller design. Similar system performance requirements are used for the discrete PID controller except the settling time, which is taken as lower than 10 sec. Figure 4.11 shows the discrete PID controller gains tuning model for operating point 8. The discrete PID controller gains at all operating points are listed in Table 4.5.

Table 4.5 Discrete PID controller gains at all operating points

OP#	**Velocity [ft/s]**	**Altitude [ft]**	**K_p [rad/ft]**	**K_i [rad/ft.s]**	**K_d [rad.s/ft]**
1	10	100	5.60E-03	1.87E-03	1.99E-01
2	25	100	1.06E-03	7.10E-04	3.88E-02
3	50	100	3.76E-04	2.07E-04	1.09E-02
4	73	100	2.17E-04	9.96E-05	5.62E-03
5	100	100	1.57E-04	6.09E-05	3.43E-03
6	143	100	9.63E-05	4.28E-05	1.68E-03
7	185	100	5.12E-05	3.59E-05	1.03E-03
8	237	5,000	4.55E-05	3.34E-05	6.72E-04
9	288	5,000	5.22E-04	2.65E-04	5.98E-03
10	479	5,000	3.50E-05	2.17E-05	3.94E-04
11	668	5,000	1.92E-05	1.56E-05	2.68E-04
12	854	5,000	2.35E-05	1.50E-05	4.970E-05

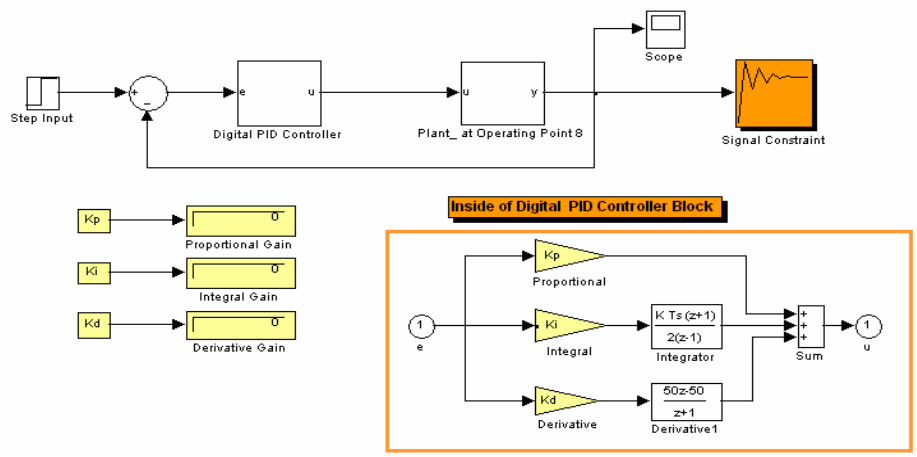

Figure 4.11 Discrete PID gain parameter tuning at operating point 8

The solver configuration in the optimization window in Signal Constrain Block is not changed automatically when the solver configuration of simulation is recovered. Therefore, during the calculation of gains, the solver configuration should be adjusted in not only Solver under Configuration Parameters window but also Optimization under Simulations Options as shown in Figure 4.12. Otherwise, the error message, “Could not find any gain values”, appears despite of finding the gain values are possible under the defined condition. Figure 4.13 shows the discrete PID controller gains at entire operating points. Figure 4.14 and Figure 4.15 show the gains at 100 ft and 5,000 ft respectively.

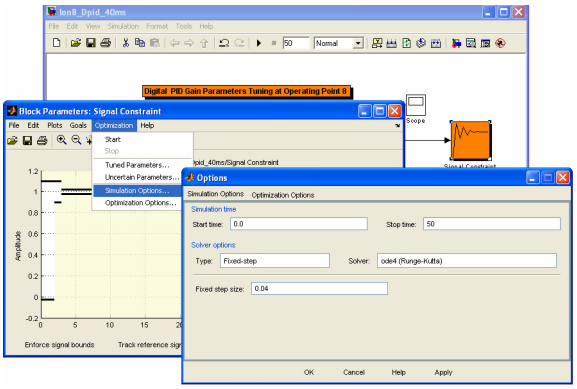

Figure 4.12 Adjusting solver configuration

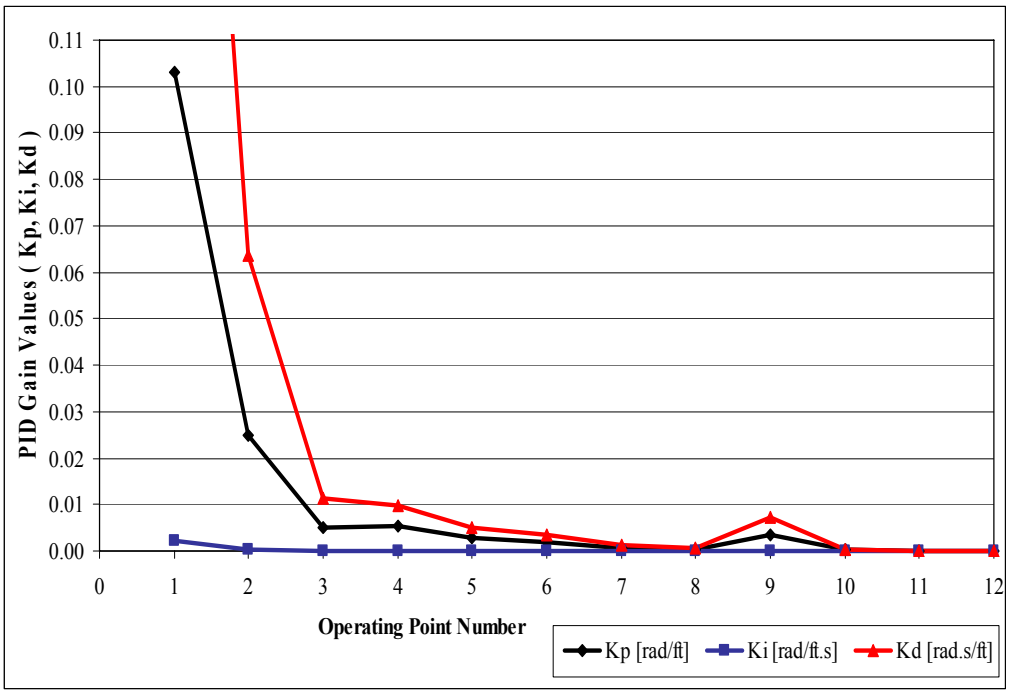

Figure 4.13 Discrete PID controller gains at entire operating points

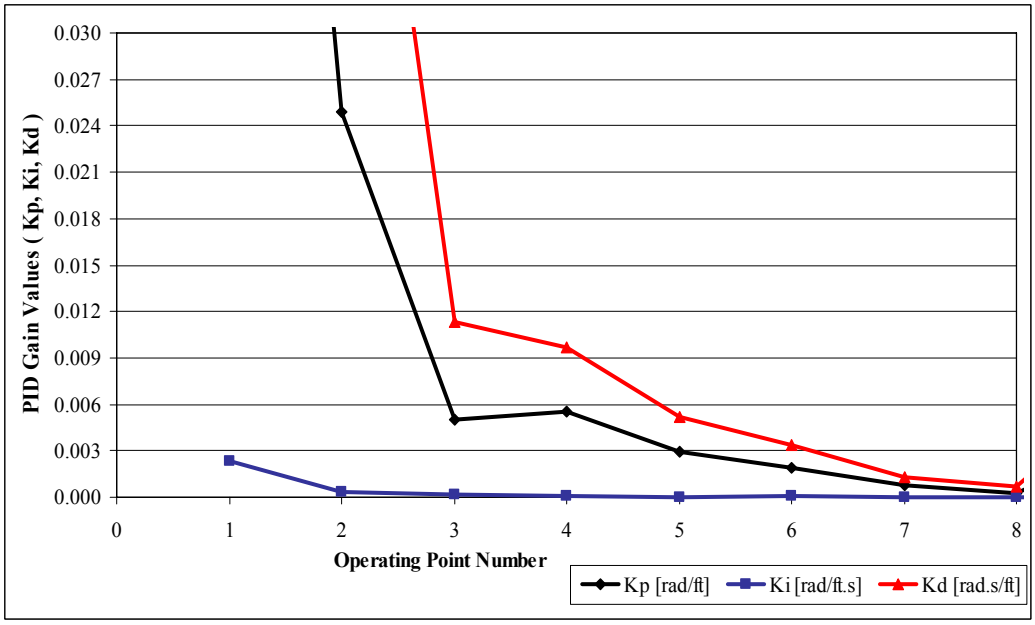

Figure 4.14 Discrete PID controller gains at 100 ft

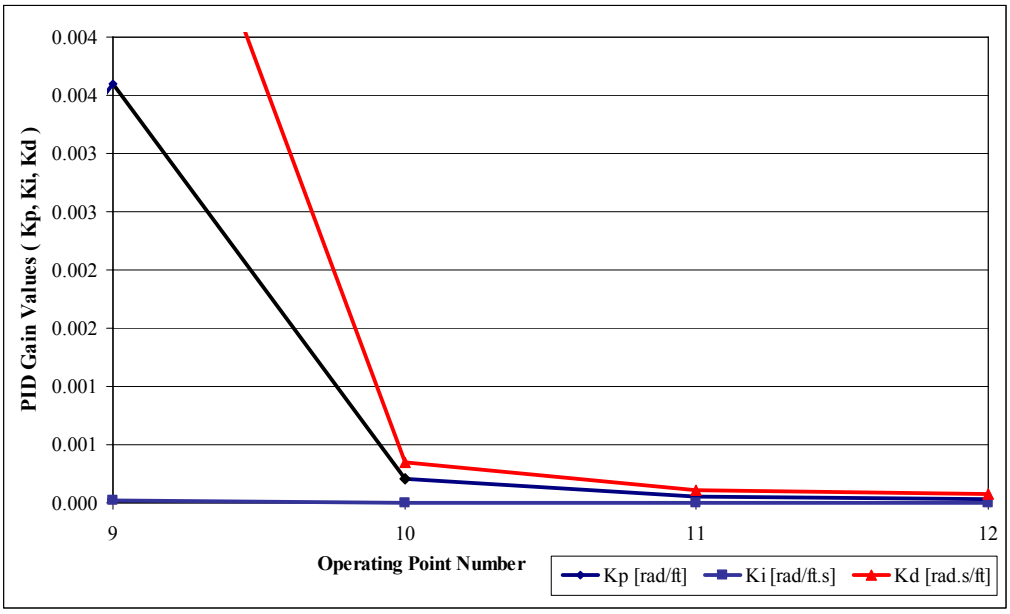

Figure 4.15 Discrete PID controller gains at 5,000 ft

4.4.1 Gain Scheduling

Gain scheduling is one of the most popular approaches to nonlinear control design and is widely applied in fields ranging from aerospace to process control [3]. Aircrafts are nonlinear systems, which operate over a wide range of flight conditions and the gain scheduling method is preferred especially in their flight controls systems [3].

When one uses transfer functions to represent the local linearized dynamics of nonlinear systems, the terms of these transfer functions change from one operating point to another. Therefore, the controller parameters chosen for each operating point change from one such point to another. The operational variables used to define these operating points are called scheduling variables. The scheduling variables are used to determine in what operating region the system is currently and to enable the appropriate linear controller. Mostly, the altitude and Mach number are taken as the scheduling variables in the autopilots. In this thesis, altitude and velocity are taken as the scheduling variables of the UCAV6 controller model, which are dependent on specific velocity and altitude values. Therefore, the

controller gains are determined using the UCAV6 transfer functions obtained at twelve trim points. The gain set obtained is embedded in to the lookup tables to employ the corresponding PID controllers resulting in satisfactory performances. The 12 operating points are used as break points in these tables. Except at break points, gain values are determined by using linear interpolation. Thus, the autopilot maintains its adequate performance over the entire altitude and speed envelope.

Next to its advantages, the gain scheduling method has some difficulties and limitations, which are listed in below [41].

- Due to gain scheduling, robustness, stability, and command response can all change significantly.
- Scheduled gains are not equal to the design gains except at the break points.
- Performance can be poor, because the scheduled gains do not guarantee to be stable. For avoiding performance loss, some additional simulations between breakpoints should be run to increase the number of breakpoints. However, this requires an increase in the number of trim points.
- To improve performance adjusting the scheduled gains causes time consuming.

Figure 4.16 shows the Simulink diagram of the discrete PID controller of UCAV6 with gain scheduling. Since the UCAV6 model is trimmed at the operating points either at 100 ft or at 5,000 ft, the performance of the discrete PID controller is tested at 100 ft and at 5,000 ft separately due to a large difference between these altitude values, hence also in controller gain values.

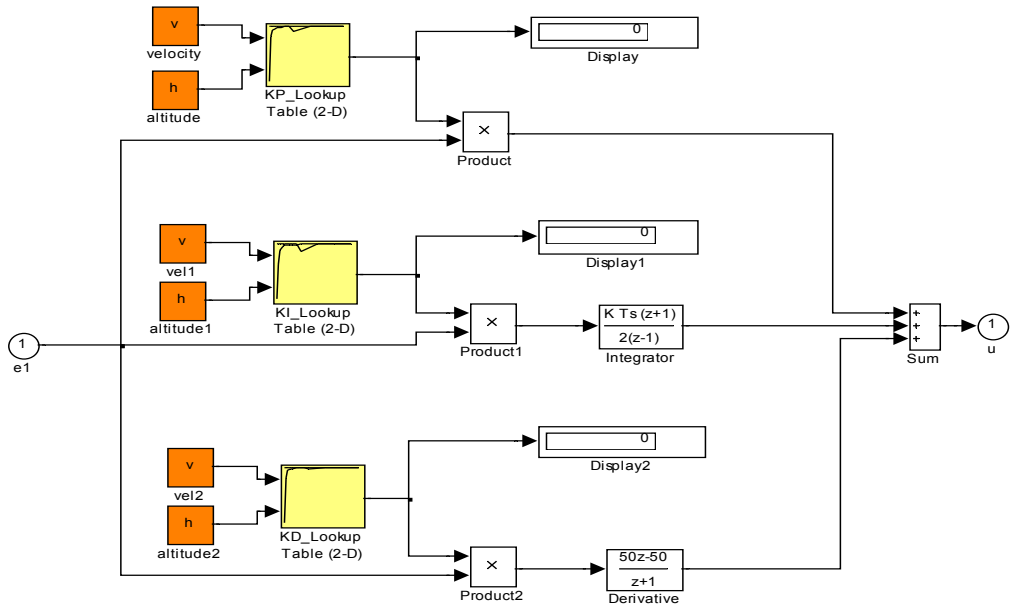

Figure 4.16 UCAV6 discrete PID controller with gain scheduling

4.5 Concluding Remarks

Since the general dynamic equations of motion for a rigid airplane are derived and available in open literature, so their derivations are not repeated in this thesis. Instead, the aircraft longitudinal equations of motion are taken directly from reference [66]. By the help of these equations of motion, the transfer function between altitude and elevator deflection is derived for UCAV6, F-104, F-4, and Learjet24 aircrafts and plant properties of UCAV6 are compared to plant properties of these aircraft. Direct II method is used to insert state space matrices of UCAV6 in Simulink environment.

After plant model is finalized, the PID controller is designed by tuning its gains both in continuous and discrete forms. The discrete PID controller is used in the setup application, because continuous PID controller model cannot be converted to C codes with RTWEC for VxWorks operating system. Finally, a gain scheduling is performed taking the altitude and velocity as the scheduling variables of UCAV6.

CHAPTER 5

PLATFORM TESTS AND RESULTS

The second objective of this thesis study is to verify the functionality of the platform. In addition to plant and controller models of UCAV6, a flight scenario is needed to verify the platform. The controller gains change rapidly while switching from 100 ft to 5,000 ft due to the high difference between altitudes of trim points. This rapid change disturbs the behavior of the whole system. It would have been better, if some additional densely populated trim points were available at the altitudes between 100 ft and 5,000 ft. Therefore, it is preferred to test the controller in the platform for two separate scenarios, one for at 100 ft and the other for 5,000 ft, and discuss their Simulink results separately.

5.1 Test Scenarios

Minimum one flight plan is constructed for 100 ft and for 5,000 ft altitudes to verify the platform functionality. The flight plans are generated simply by changing the velocity input with respect to time because altitude and velocity are set as scheduling variables of the UCAV6 model. Depending on these scheduling variables, the transfer functions coefficients of UCAV6 and controller gain values are recalculated during the simulation.

For a fixed altitude, the only scheduling variable of UCAV6 is its velocity. The change in airspeed is reflected into the simulation by designing a velocity signal block with Signal Builder Block.

5.1.1 Test Scenarios at 100 ft

Before testing the controller performance in the test platform, eight flight scenarios are set in Simulink environment by changing velocity signal input. The velocity signal is chosen because it is the scheduling variables of the lookup tables in discrete PID controller model. These test scenarios are obtained by changing the slope of the signal both at 100 ft and at 5,000 ft altitude separately. Depending on the slope of the velocity signal, the simulation time is decided. These eight flight scenarios are tested in Simulink environment with a simulation, which includes both controller and plant dynamics. Among these flight scenarios, one scenario is selected depending on data statistics of the plant output. The velocity signal input for testing the controller performance at 100 ft is shown in Figure 5.1. The information about data statistics of this flight plan and their obtaining procedure are explained in Appendix J. Figure 5.2 shows the UCAV6 altitude signal in Simulink environment for the selected velocity signal input

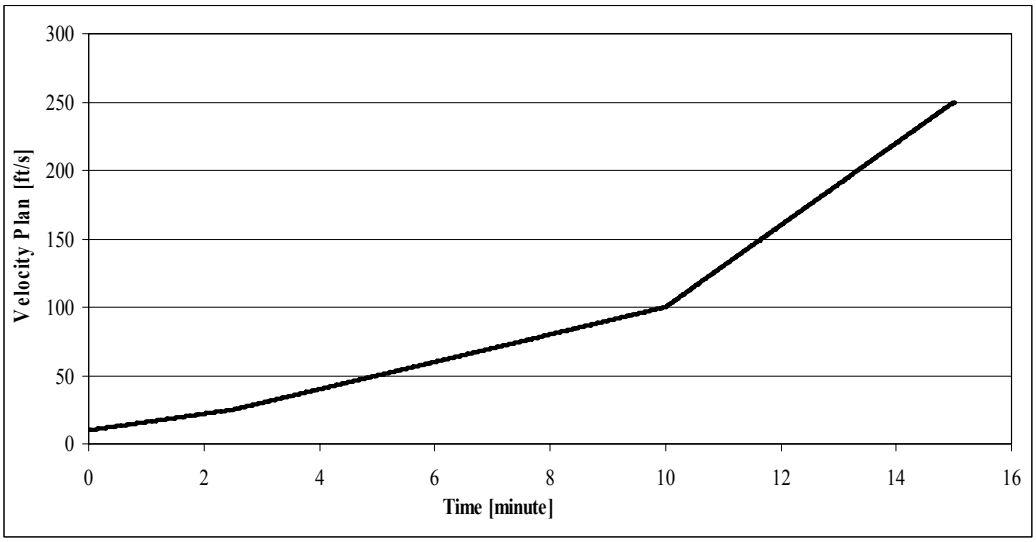

Figure 5.1 Velocity signals input at 100 ft

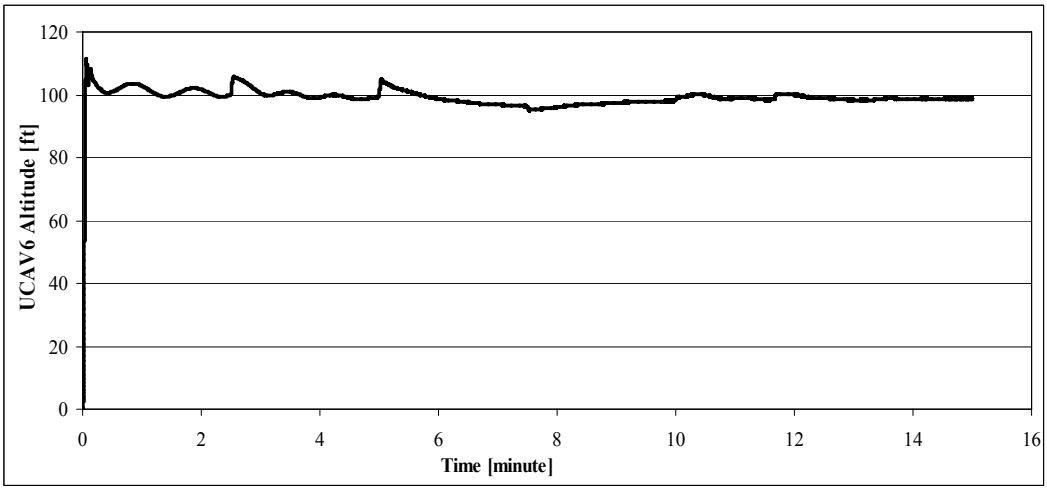

Figure 5.2 UCAV6 altitude at 100 ft test in Simulink

5.1.2 Test Scenarios at 5,000 ft

To check the controller performance at 5,000 ft, nine flight scenarios are established and two of them are chosen. Similar to the test scenarios obtained for the controller test at 100 ft, these scenarios are obtained by changing the slope of the velocity signal input. The simulation time differs with the slope. Two test scenarios are chosen depending on the data statistics results of the plant output. The data statistics results can be found in Appendix J. Figure 5.3 shows the chosen flight scenarios while airspeed increasing and decreasing. Moreover, the UCAV6 altitude values during simulations at 5,000 ft can be monitored in Figure 5.4.

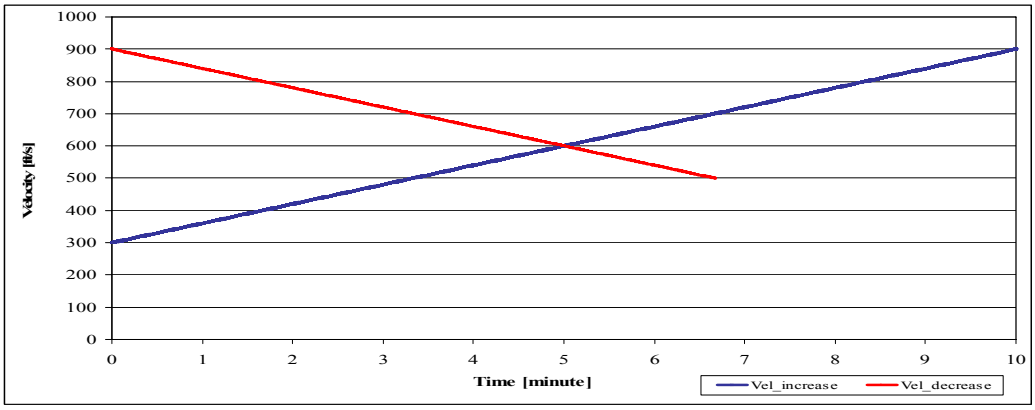

Figure 5.3 Velocity Signals at 5,000 ft

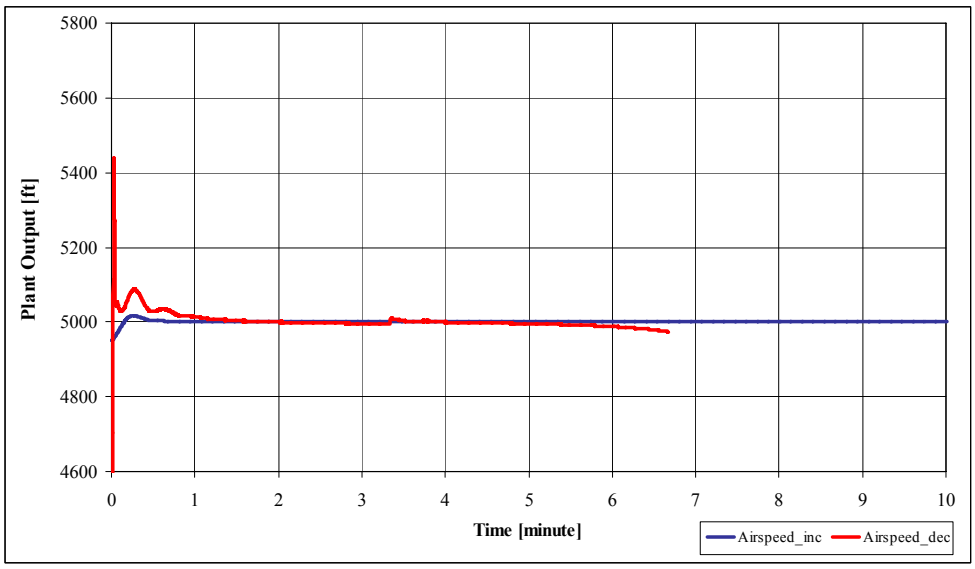

Figure 5.4 UCAV6 altitude result at 5,000 ft test in Simulink

5.2 Platform Tests

Although several controller designers find to satisfy their controller performance in Simulink environment enough, this approach is changing with the emerging rapid control prototyping tools. In this section, the performance of a UCAV6 controller model is tested in the platform. This platform, shown in Figure 5.5 is used for both at 100 ft and at 5,000 ft flight tests.

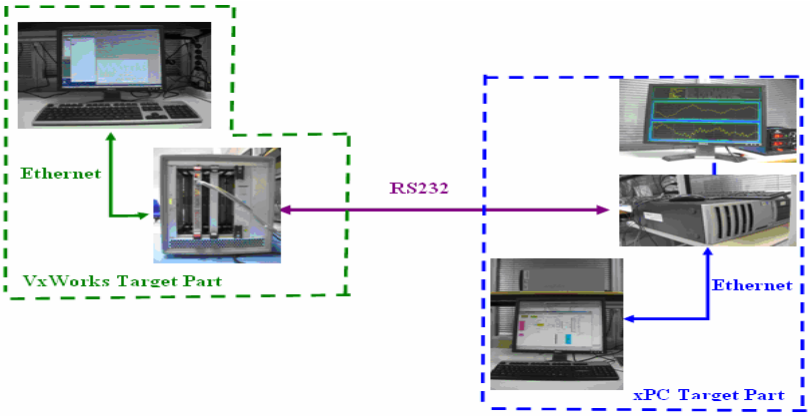

Figure 5.5 The platform view for UCAV6 controller test

Figure 5.6 shows the Simulink diagram of the controller. For the flight test at 100 ft, the codes of the controller model are downloaded in to VxWorks target and plant model codes are downloaded in to xPC target. The Simulink diagram of the plant at 100 ft is shown in Figure 5.7.

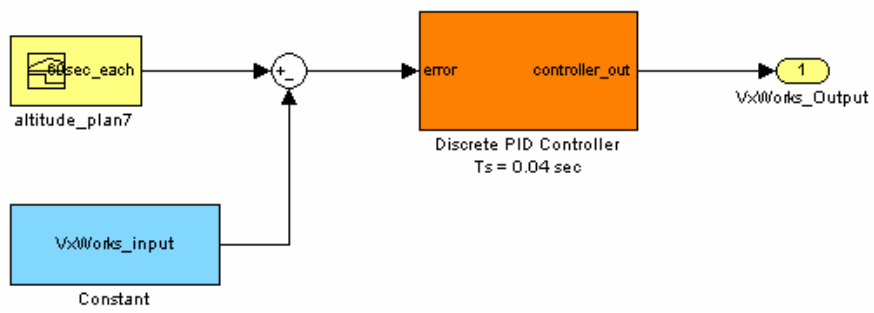

Figure 5.6 The Simulink diagram of the controller

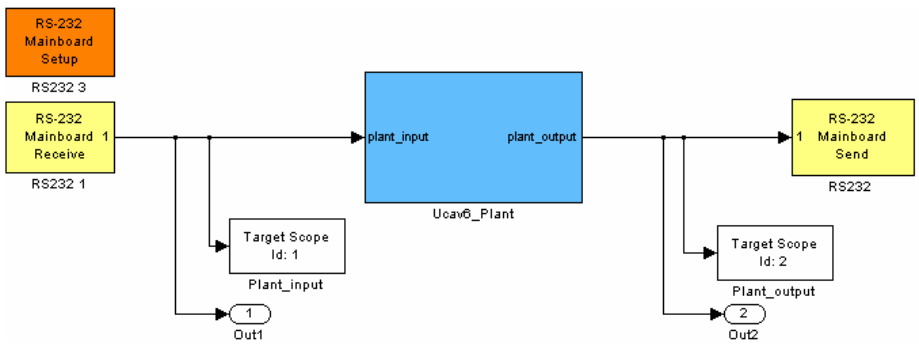

Figure 5.7 The Simulink diagram of the plant

The coding and downloading procedure are explained systematically by handling the entire model of Boeing 747 in Appendix B. The common procedures are followed for all platform test applications.

During the tests in this platform, both controller and plant output signals are logged in the host PC via ethernet connection. The vertical movement of the plant can also be monitored in both the target scope and the host scope. The image of the plant output is not displayed in the real time due to the priority of this scope task in the xPC target application.

5.2.1 Comparative Test Results at 100 ft

Figure 5.8 demonstrates the behavior of the plant during platform tests. Flight test results obtained in the platform gives nearly the same result as those obtained in Simulink. The difference between altitude values for 100 ft simulation tests can be found in Figure 5.9.

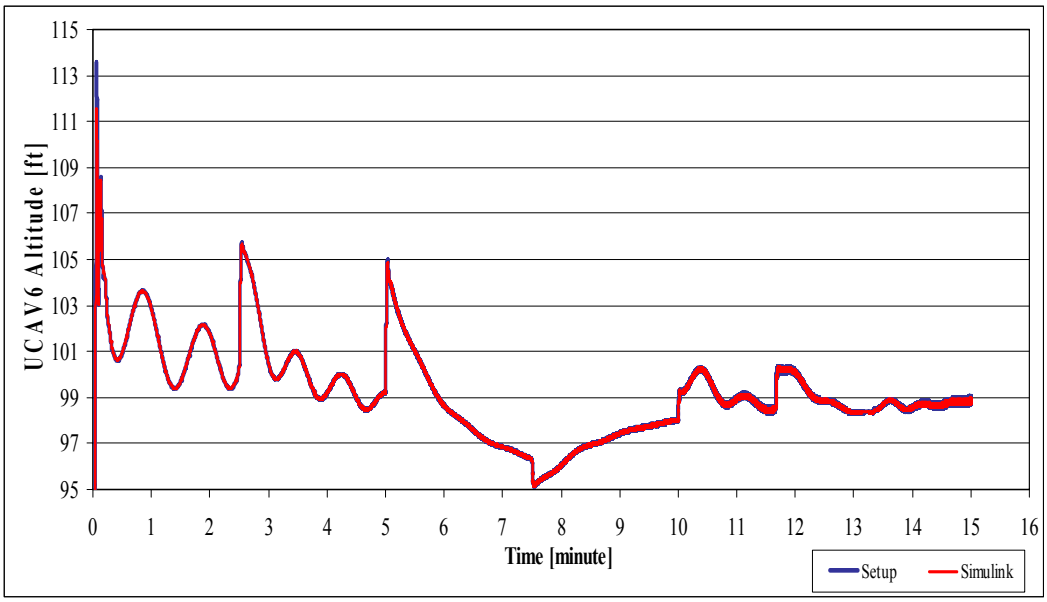

Figure 5.8 UCAV6 altitude in both setup and Simulink for 100 ft

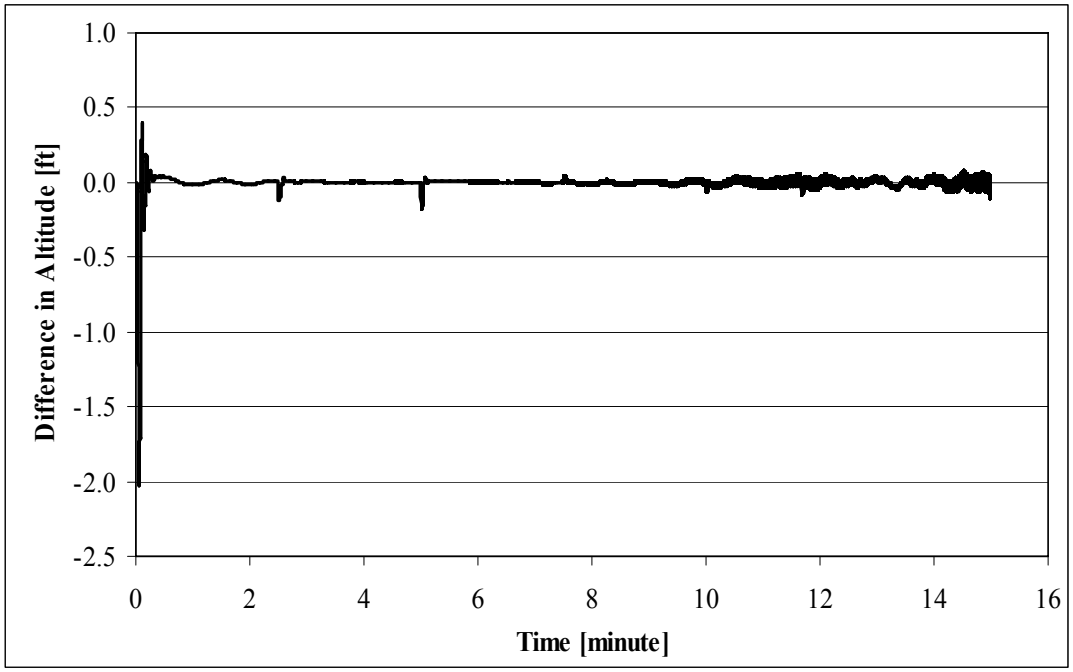

Figure 5.9 The difference between UCAV6 altitude

In this flight plan, the controller tries to keep UCAV6 at 100 ft as the airspeed of UCAV6 increases from 10 ft/s to 250 ft/s. The maximum difference between UCAV6 altitude values is approximately 2 ft and it can be ignored compared to the desired value.

5.2.2 Comparative Test Results at 5,000 ft

For the flight test while airspeed increases from 300 ft/s up to 900 ft/s, the test in the platform is repeated two times. The altitude results, which are obtained from the simulation in both test platform and Simulink environment, are given in Figure 5.10. Moreover, data statistics of the UCAV6 altitude during the flight tests at 5,000 ft can be observed in Appendix J.

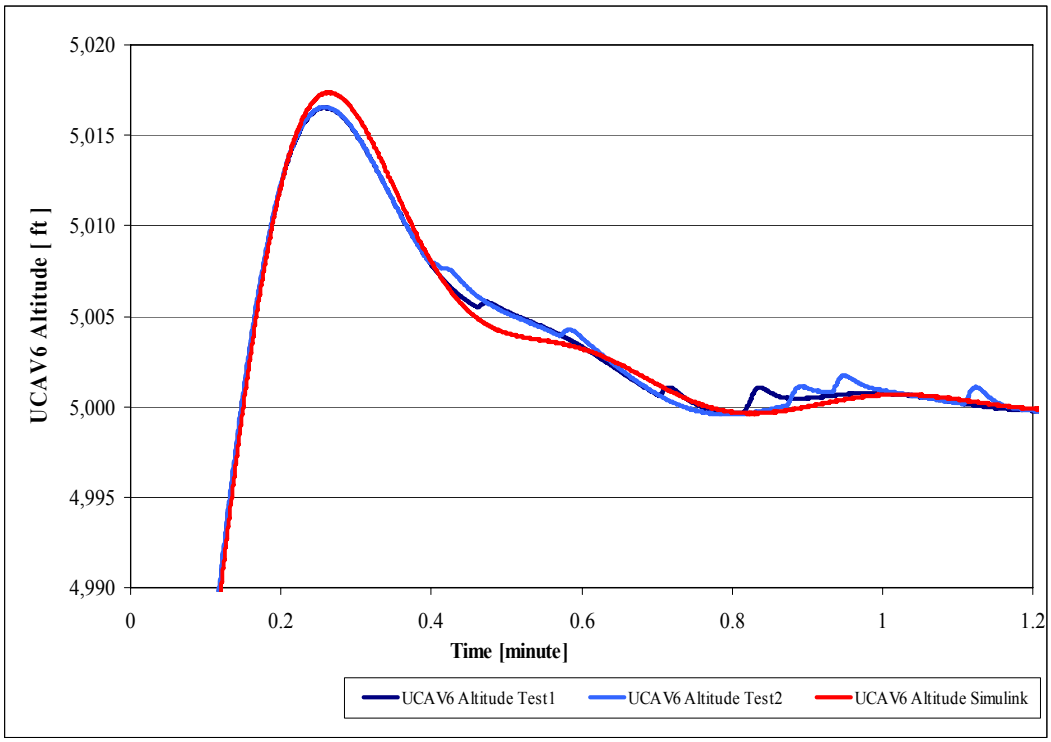

Figure 5.10 UCAV6 altitude in both setup and Simulink tests for 5,000 ft

The altitude values of UCAV6, which are obtained from approximately one-minute simulation in both test platform and Simulink environment, are near to each other. As shown in Figure 5.10, the difference between these simulation results is very small to be realized. However, as the platform test is lasted to 10 minutes, there occurs an unexpected noise. This noise is observed after the UCAV6 altitude signal damped to its desired value. Figure 5.11 shows the noise effect during the test at 5,000 ft while airspeed increasing. Figure 5.12 is obtained by zooming one of tiny spikes in Figure 5.11 when time is around 2.66. Under this spiky noise, the plant behaves as it is subject to an impulse. In about 4 seconds, controller set the altitude to the desired value. Figure 5.13 shows the difference between the UCAV6 altitude results. The maximum difference between these test results (obtained in setup and in Simulink environment) is about 3 ft if the spiky region is considered and it is only about 1 ft if the spiky region is not considered; hence, it is ignorable compared to 5,000 ft elevation.

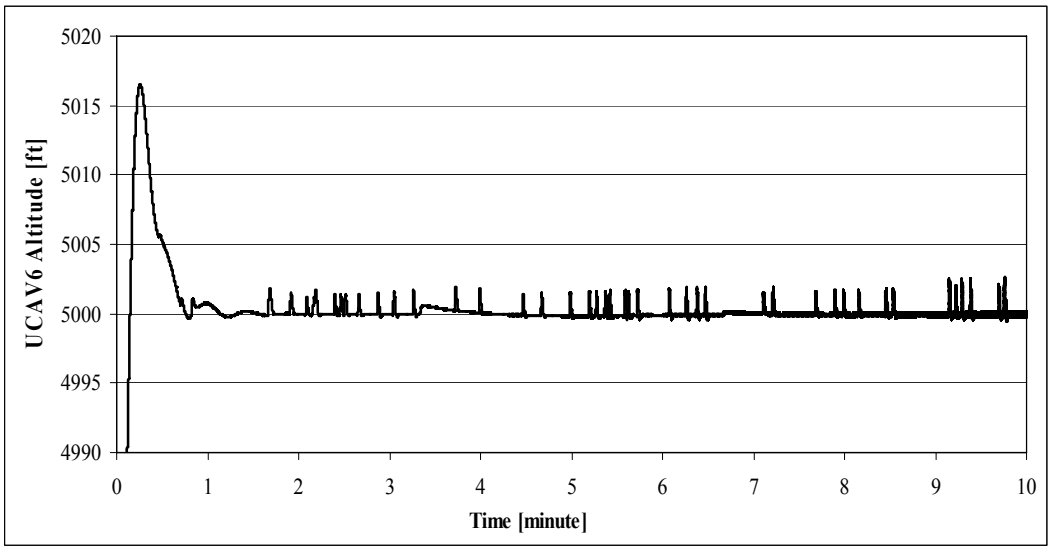

Figure 5.11 Attitude of the plant during platform test while airspeed increasing

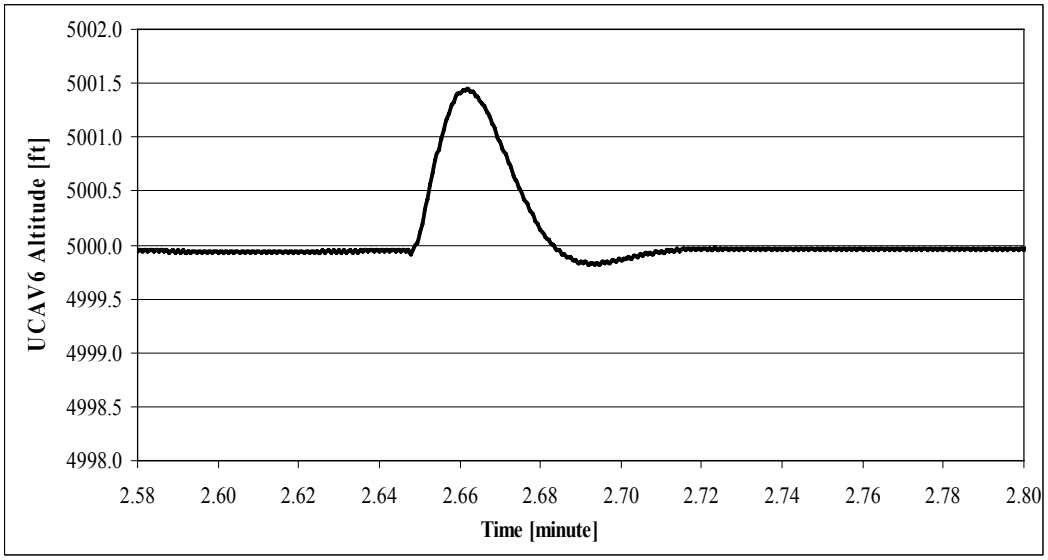

Figure 5.12 Attitude of the plant during platform test (Zoomed)

The controller performance is tested two times as the airspeed decreases from 900 ft/s to 500 ft/s at 5,000 ft altitude. Figure 5.14 and Figure 5.15 show the attitude of the plant during these tests.

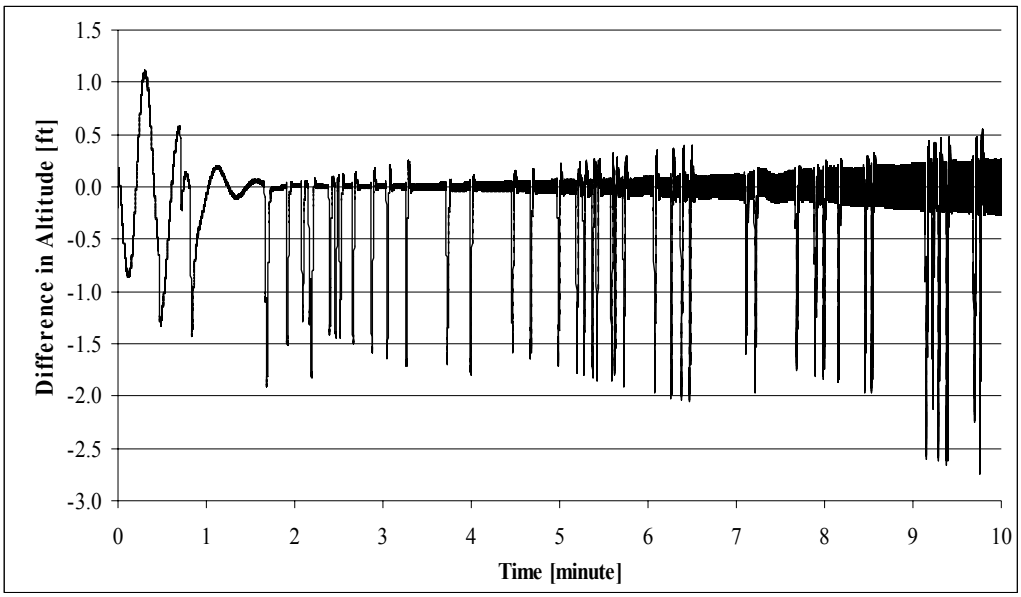

Figure 5.13 The difference of the altitude in between Simulink and platform tests

The test results shows that the number of spikes is less than that of obtained in test as the velocity increasing. The difference between the altitude values during both tests taken in the setup and in Simulink are showed in Figure 5.16 and Figure 5.17.

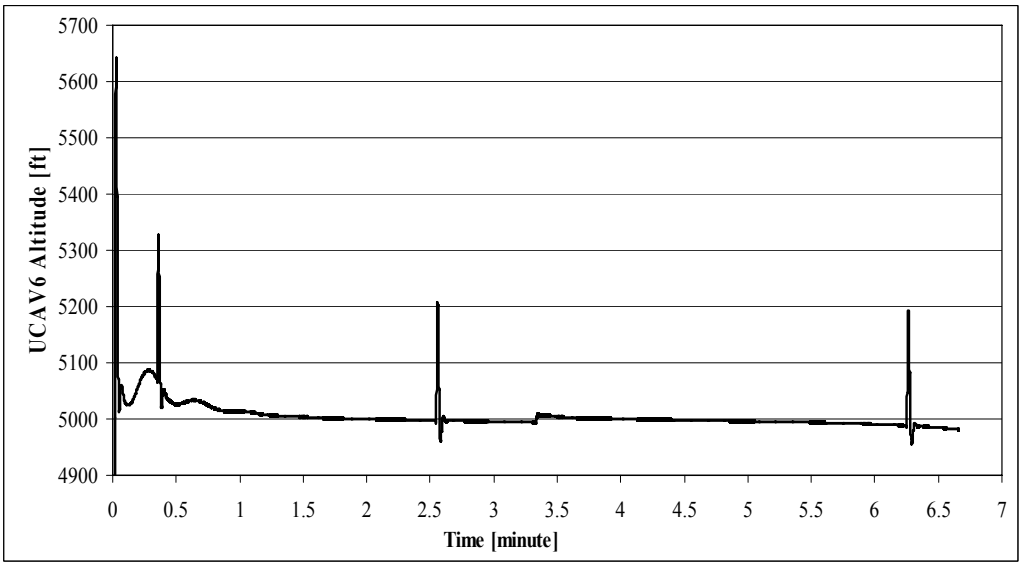

Figure 5.14 Attitude of plant during platform test 1 while airspeed decreasing

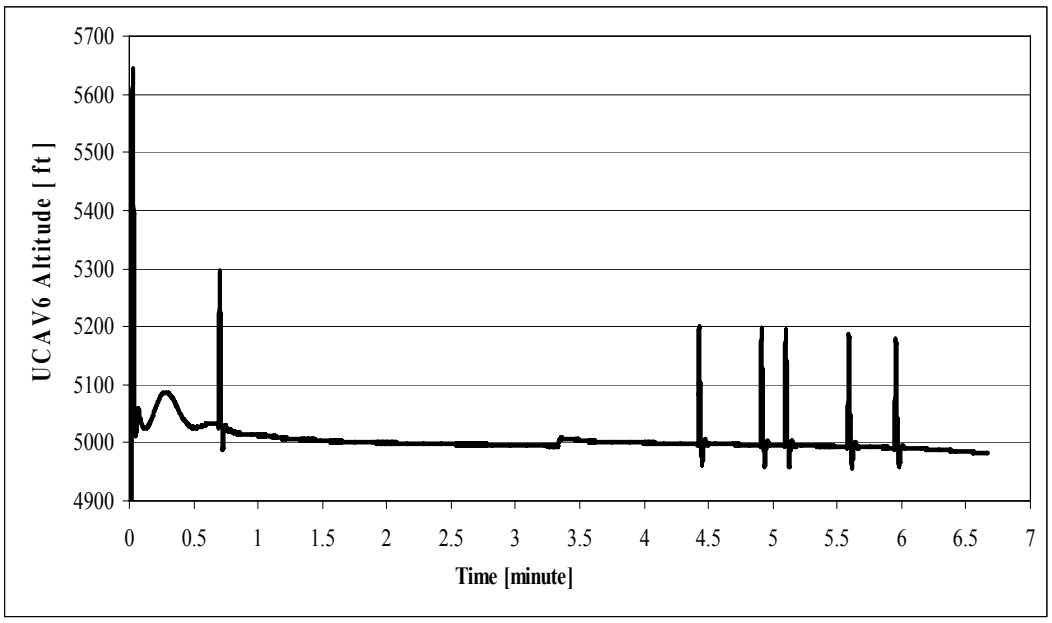

Figure 5.15 Attitude of plant during platform test 2 while airspeed

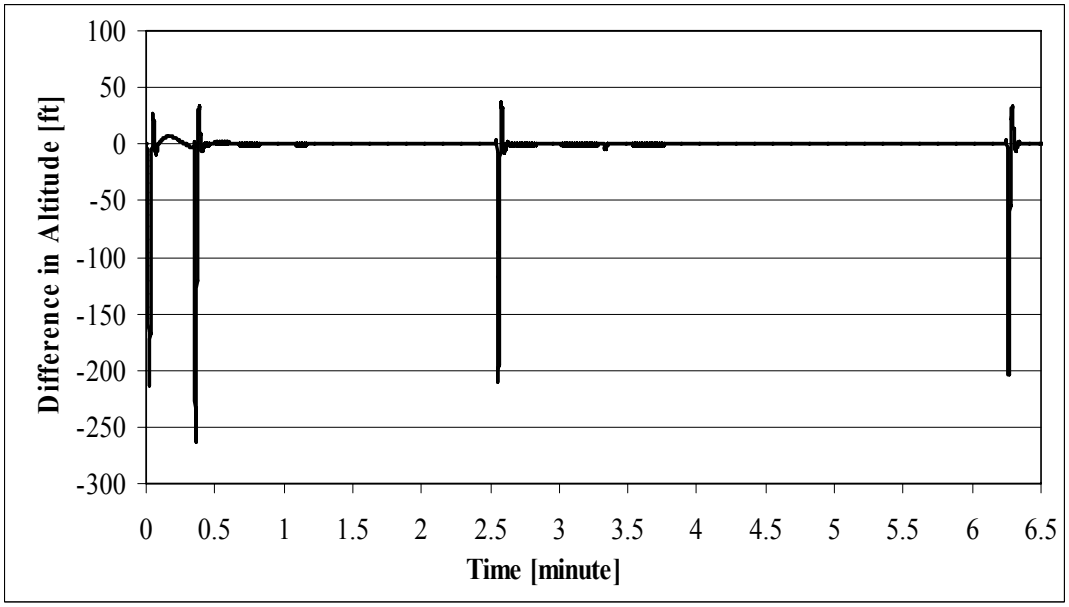

Figure 5.16 Difference in altitudes between in Simulink and Platform tests 1

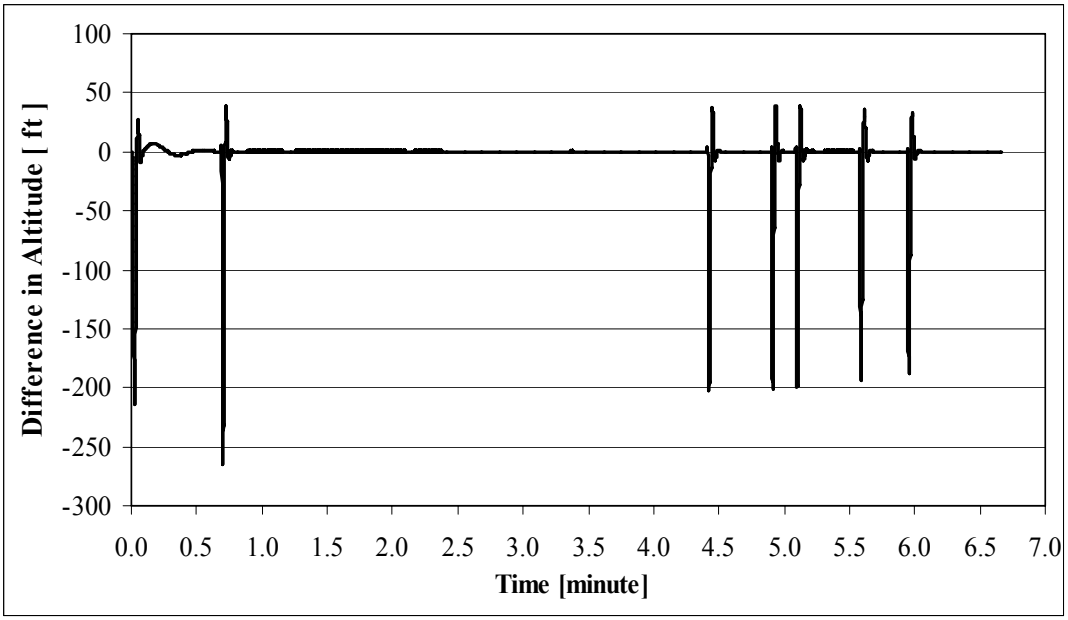

Figure 5.17 Difference in altitudes between in Simulink and Platform tests 2

The difference between altitude values may be due to rapid changes in derivative gain values of the controller at 5,000 ft or due to noisy character of RS-232 communication. The UCAV6 model is trimmed at 5,000 ft for four increasing velocity values. The transfer function coefficients of the plant model change rapidly among these operating points. In addition to this rapid change in plant parameters, the controller gains, especially derivative gains, also change quickly. Operating points are used as the break points for the look up tables in both plant and controller models. Between the break points, the coefficient of the transfer functions and the controller gains are calculated by using linear approximation method. It is the default method for the approximation of the values in the lookup tables. Different method may be used for the result that is more accurate.

The difference between altitude values in Simulink and in setup during the tests may also be due to the noisy communication between plant and controller environment, which is realized via RS-232 communication. Aitech C772 single board does not have any apparatus for serial communication. Thus, the communication cable is connected to this board pin to pin as shown Figure 5.18. Due to this connection

method, it is very easy to catch noise during communication. To ensure about the reliability of the communication, an apparatus for serial port of Aitech C772 board could have been used. But, it is not employed due to its cost and transportation time. Alternatively, another interface could have been chosen for the communication between the plant and controller. In that case, minimum two extra I/O board are needed. For example, MIL-STD-1553B can be chosen for this communication. It is more reliable interface than RS-232 and it is used in most military aircrafts. But the I/O boards suitable for 1553 communication are too expensive, therefore this alternative is not preferred in this study.

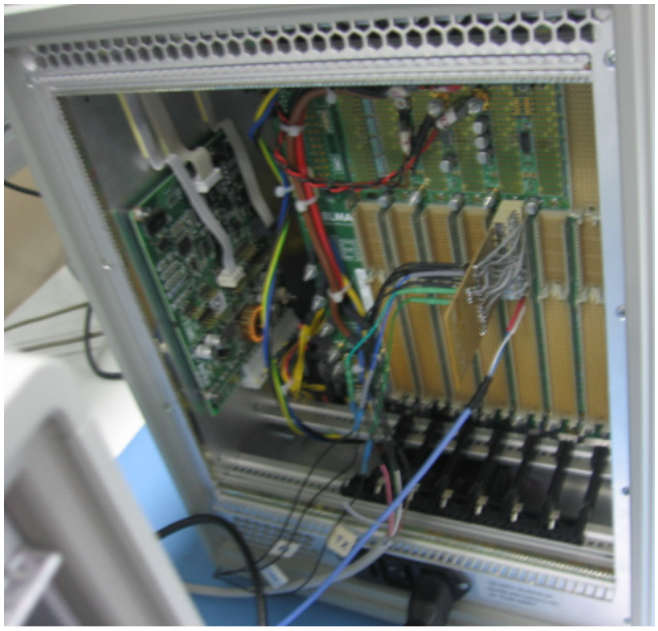

Figure 5.18 VxWorks RS-232 connection

As a soft solution for this problem, a low pass filter could have been designed and used. However, this is not employed either, since this thesis study does not aim to find the reason of this difference or to design a perfect controller.

5.3 Concluding Remarks

In this chapter, test results of some flight plans are presented for 100 ft and 5,000 ft elevations while changing the velocity as a function of time. For 100 ft elevation, eight flight scenarios are set and tested in Simulink environment. The result of one of them is presented. Similarly, nine flight scenarios are generated and tested for 5,000 ft elevation and results of two of them are presented.

During the 5,000 ft tests, an unexpected noise problem is observed. It may be due to the sharp changes in the derivative gains of the controller, the RS-232 communication interface between controller and plant, or something else. This problem may be solved by designing a low pass filter. However, find the exact reason behind this noise problem, to solve it, or to design a perfect controller are not in the scope of this thesis study. Therefore, no attempt is made towards in that direction.

Although some noise affect the behavior of the plant, it is demonstrated that the controller can keep UCAV6 model at the desired altitude both at 100 ft and 5,000 ft flight tests. Therefore, the performance of the controller can be considered acceptable for the operation ranges of flight tests in the platform.

CHAPTER 6

VERIFICATION OF THE PLATFORM REUSABILITY

The final objective of this thesis study is to verify the platform reusability. In choosing such a model suitable for coding with RTWEC, an attention is also paid on bringing not much extra work. A Boeing 747 model, which is taken from [86], is used for this purpose. This model is a closed loop model including both flight dynamics model and its controller taken from reference [86]. Hence, the reusability of the platform is verified with a model taken from an external source. Before testing this model in the platform, the controller and plant models are separated to download to different targets PCs. Figure 6.1 shows Boeing 747 entire model sketch and the following two figures (Figure 6.2 and Figure 6.3) show plant and controller models after being separated. The controller of Boeing 747 is a discrete PID controller for longitudinal and lateral motion.

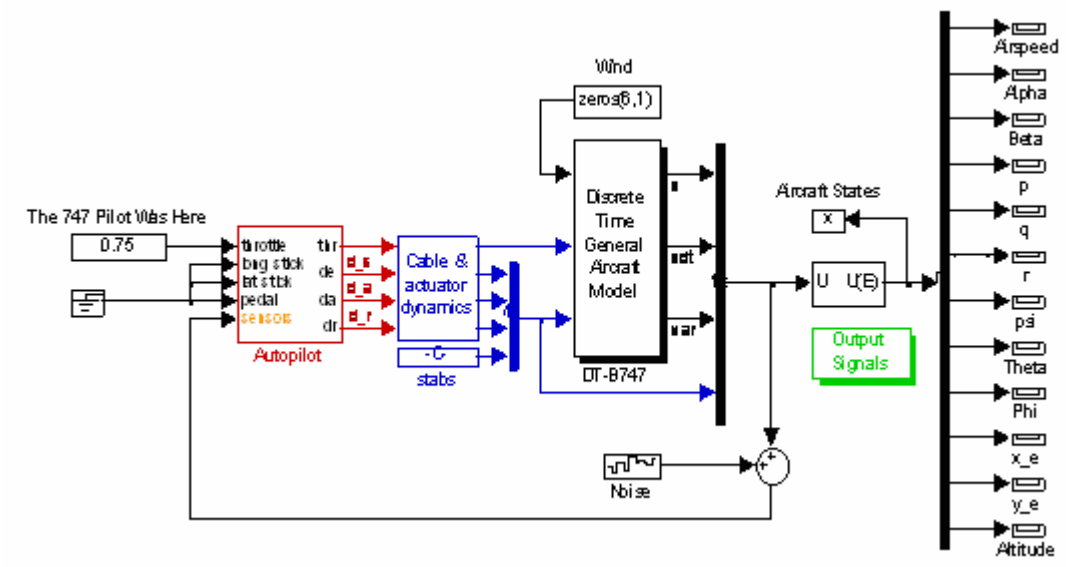

Figure 6.1 Boeing 747 entire model sketch

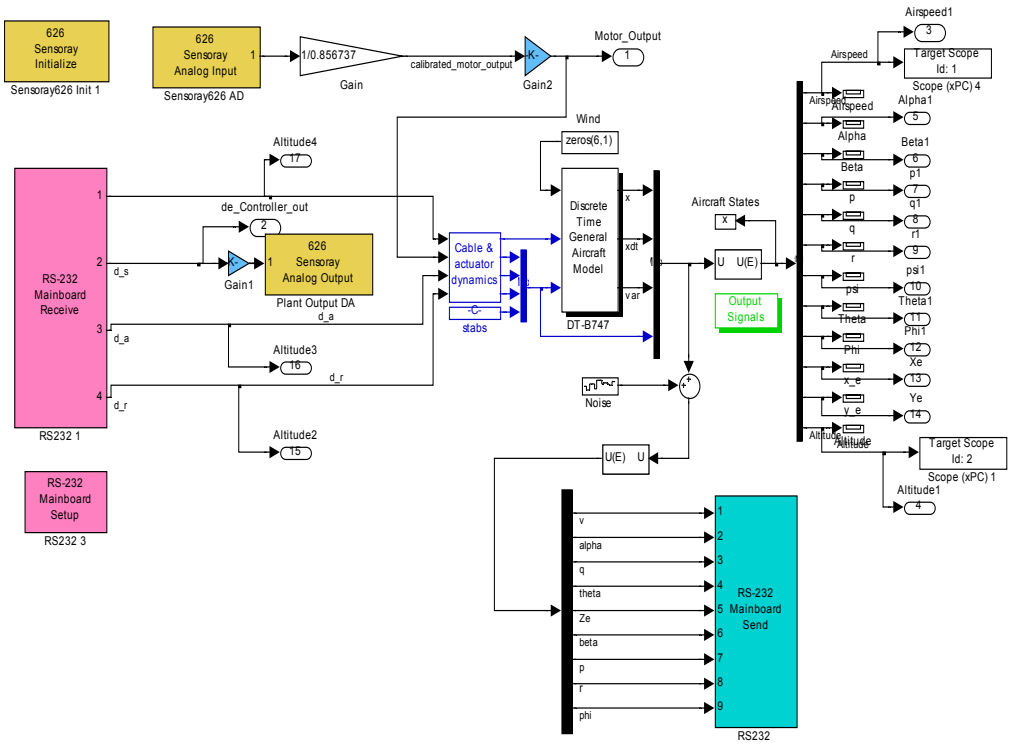

Figure 6.2 Boeing 747 plant model

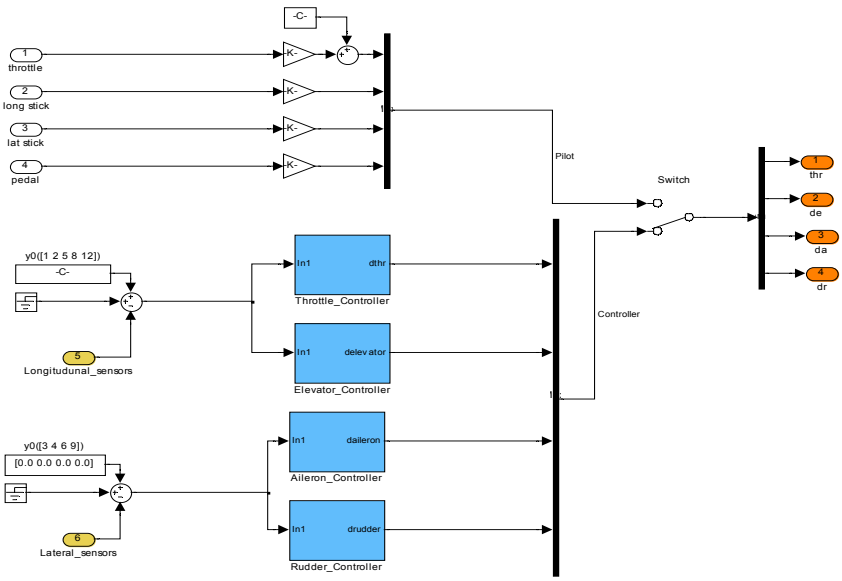

Figure 6.3 Boeing 747 controller model

After downloading process, the plant model runs in the xPC target and the controller model runs in the VxWorks target. To test Boeing 747 model in this platform, an actuator system, which is composed of DC servomotor, its driver unit and an encoder, are added to the platform. The actuator system is added to the platform, because Boeing 747 model includes cable and actuator dynamics block. The integration procedure of actuator system is explained in detail in Appendix B. Figure 6.4 and Figure 6.5 give the platform view that is used for Boeing 747 model test.

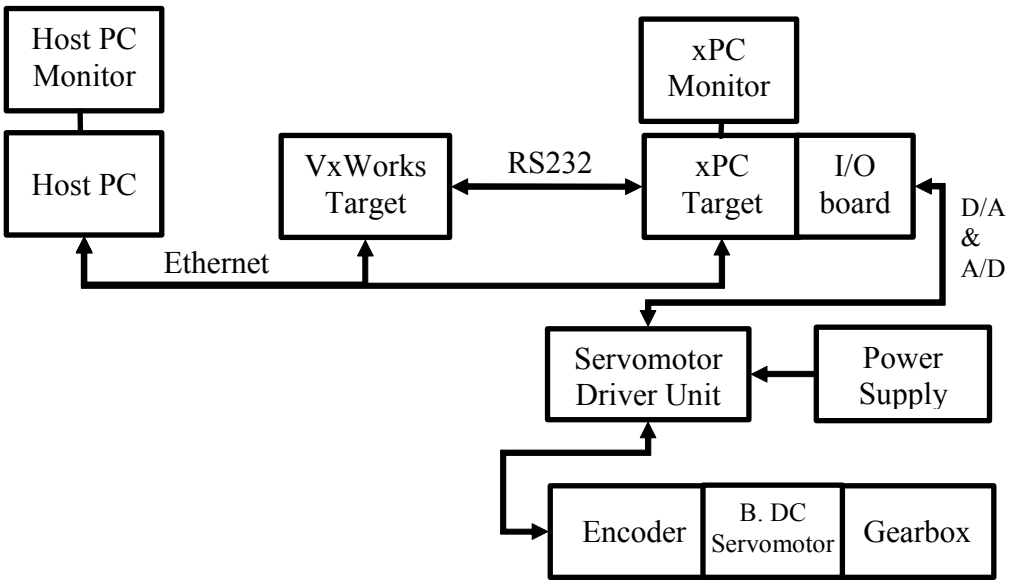

Figure 6.4 The test platform block diagram for testing Boeing 747 model

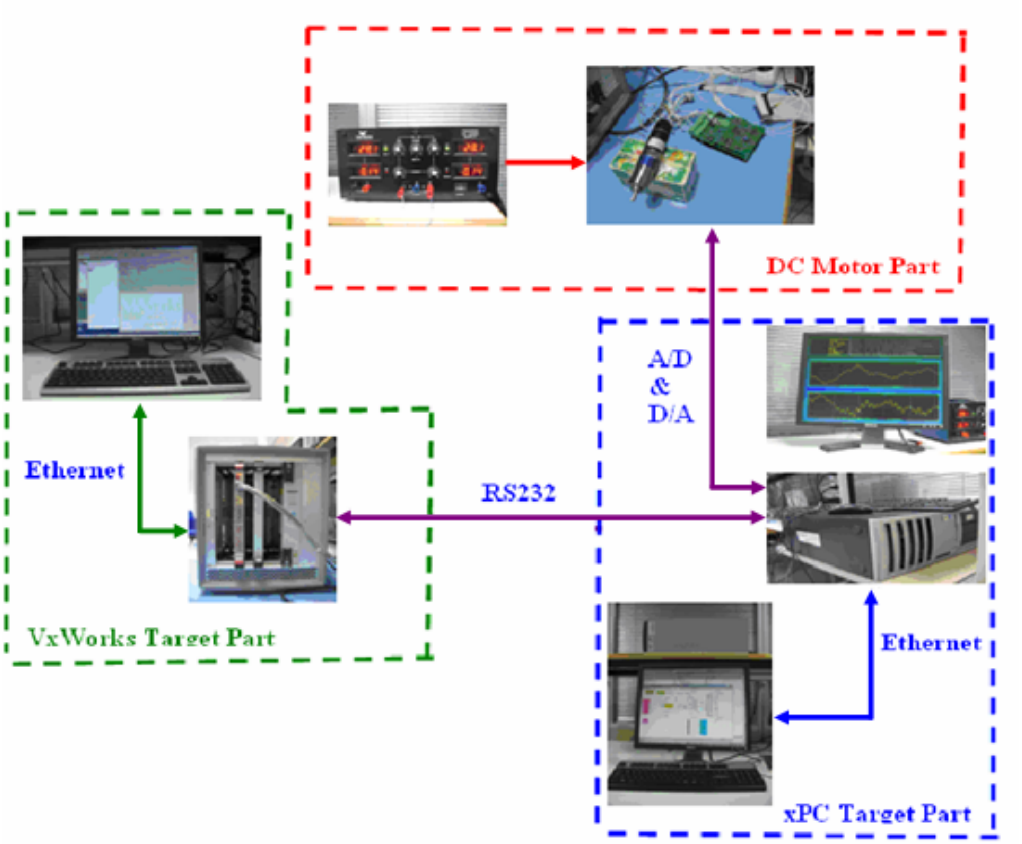

Figure 6.5 Boeing 747 test platform view test

6.1 Boeing 747 Test in the Platform

The Boeing 747 model is trimmed at one point and its controller tries to keep the plant at the trim point. All the trim values of the model can be found in Appendix K. There is no need to use gain scheduling method, because it has only one trim point. The controller of Boeing 747 has two modes, which are longitudinal and lateral. Its longitudinal controller intakes actual airspeed, alpha, q, theta, altitude variables and its lateral controller intakes actual beta, p, r, phi values are required. Desired values of the variables are their trim values. Since the desired values are far away from the trim point, the behavior of the plant changes, so its controller cannot adapt to this change.

Both setup and Simulink model tests run 60 seconds, which is higher than the settling time of this model. During these tests, all control surface deflections (δ_{thr}, δ_e, δ_a, δ_r) are calculated. However, the platform has only one servomotor to simulate the control surface motion. For that reason, only one control surface deflection can be simulated by using the actuator system. Among control surfaces of the aircraft, the elevator is chosen for this simulation, because its one of the necessary input for altitude hold mode. In this thesis study, a discrete PID controller is designed to keep UCAV6 at the desired altitude. Hence, Boeing 747 controller performance is tested in this platform for altitude hold mode. Figure 6.6 shows the altitude values during the test in setup and Simulink. The difference between these values is given in Figure 6.7. Test results of the other plant outputs, which are airspeed, alpha, q, theta, beta, p, r, psi, phi, X, Y are given in Appendix K.

Altitude Test Results:

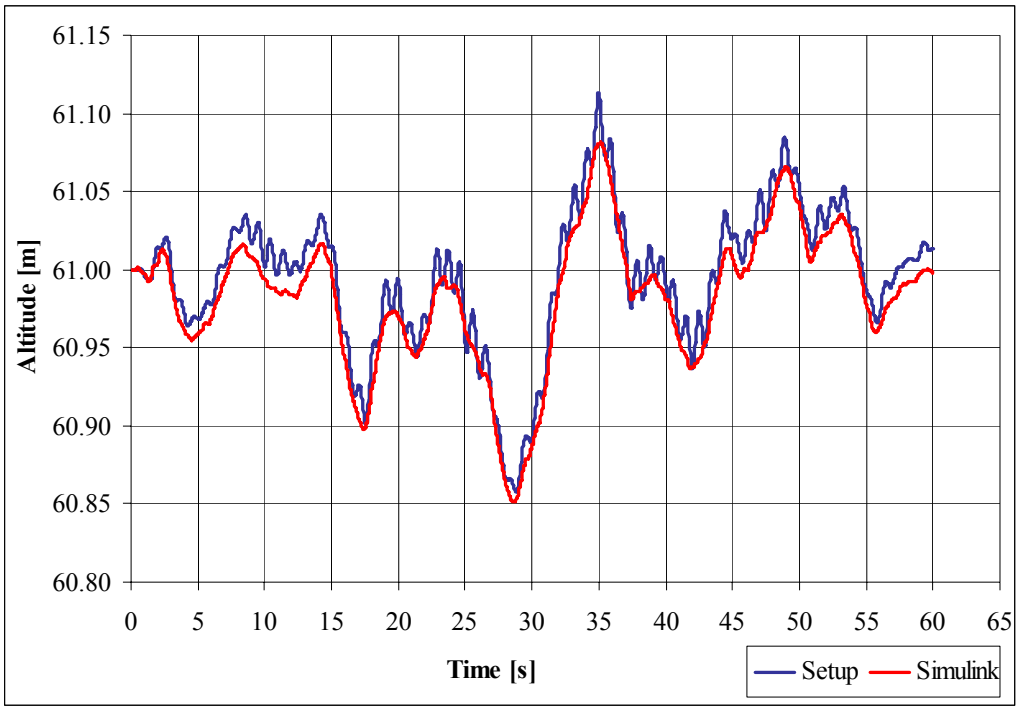

Figure 6.6 Altitude during platform and Simulink tests

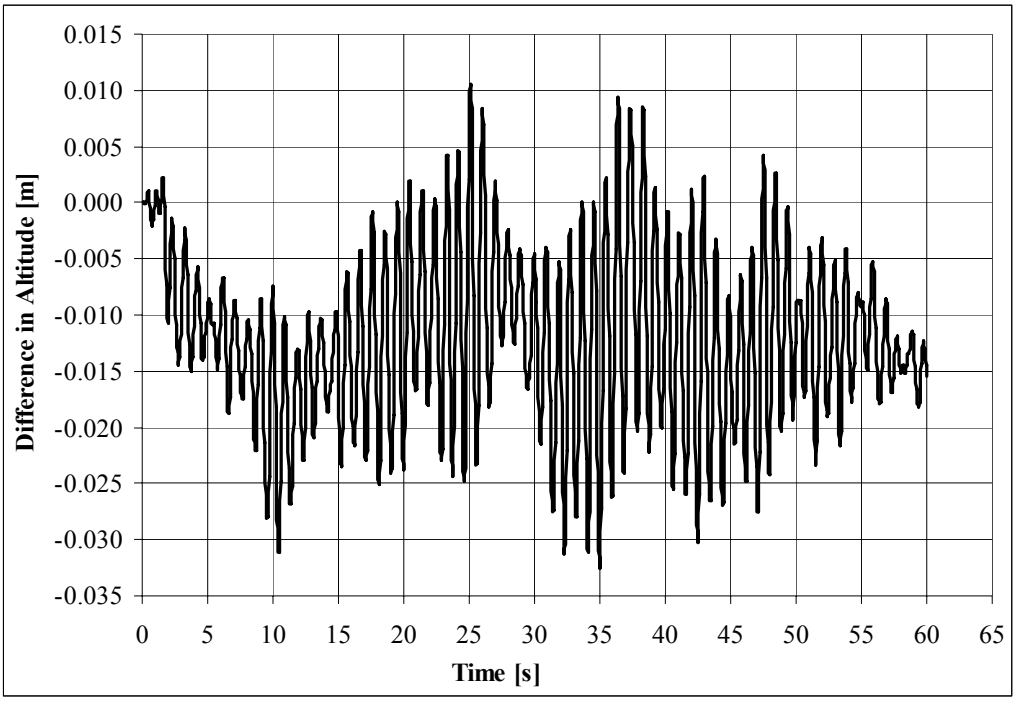

Figure 6.7 Altitude difference in platform and Simulink tests

As monitored from these figures, the differences of the altitude values are limited to a small range (peak to peak 40 cm), hence they can be ignored with respect to the desired altitude. Therefore, it can be said that the behaviors of the plant both in platform and Simulink tests are similar to each other.

For a further application, generating and setting the flight management and navigation algorithms for this mode are possible. For a suitable design, the platform can be enlarged by integrating real LRUs or their emulators.

6.2 Concluding Remarks

The final objective, which is verifying the reusability of the platform, is completed by using Boeing 747 entire model. Flight dynamics of the model is run in xPC target and the codes of its controller are run in VxWorks target. Moreover, to simulate its elevator deflection with hardware, an actuator system is added to the platform.

The airspeed, alpha, q, theta, altitude, beta, p, r, phi values are logged in host PC during the platform test. The results of the tests except the test result of altitude hold mode are given in Appendix K with comparison those in Simulink environment. The differences between these results are very small with respect to the desired values of the plant output.

CHAPTER 7

DISCUSSION AND CONCLUSIONS

7.1 Summary and Discussion

Companies not only in aerospace but also in automotive, biotechnology, medical device, manufacturing, etc. industries need a testbed to check their dynamic system designs in real-time. Therefore, the main aim of this study is to design a test platform based on real-time facilities and embedded software, especially oriented towards aerospace applications.

The platform designed and developed in this study is composed of some software to simulate the dynamic system and its controller and necessary hardware on which these software run. The components of the platform can be divided mainly into four groups as the plant, controller, host PC, and actuator. The platform is developed by the help of some modern software tools, which are used in industry for real time control applications. MATLAB/Simulink, RTW, RTWEC, Visual C compiler, xPC target toolbox, Tornado, VxWorks operating system are some of them.

More than one real time operating system are used in this platform. The xPC target is chosen to simulate the flight dynamics in real time and the VxWorks operating system is selected for implementing the control system. The xPC target is low cost and it has a better technical support than its competitors do. On the other hand, the VxWorks is designed for harsh environmental conditions, and it is tested in several mission critical applications in real life. Therefore, a VME single board computer with VxWorks operating system can be integrated on an actual platform after tested in this platform.

To verify the functionality of the platform, a specific plant and a specific controller model are used. A UCAV6 model is chosen as a plant model, because it is the most comprehensive model found in literature for aerospace applications. This linearized model has 12 operating points, which were obtained at 100 ft altitude for 8 increasing velocity values and at 5,000 ft for 4 increasing velocity values. For this plant model, a discrete PID controller is designed in MATLAB/Simulink environment.

Among several autopilot modes, the altitude hold mode is selected because of its being an important pilot-relief mode and wide use in aviation applications. To control the dynamic system in a wide range, a gain scheduling is employed by taking altitude and velocity variables as scheduling variables. Moreover, at 100 ft, eight flight scenarios are generated by changing velocity signal input slope and these scenarios are tested in Simulink environment. One of them is used to obtain system performance.

Nine flight scenarios are generated at 5,000 ft also by changing velocity signal input and two of them are chosen to check the controller performance at that altitude. During the 5,000 ft tests, an unexpected noise problem is observed. It may be due to the sharp changes in the derivative gains of the controller, the RS-232 communication interface between controller and plant, or something else. Although some noise affects the behavior of the plant, it is demonstrated that the controller can keep UCAV6 model at the desired altitude both at 100 ft and 5,000 ft flight tests. Therefore, the performance of the controller can be considered acceptable for the operation ranges of flight tests in the platform.

The reusability of the platform is verified by using an autopilot controlled Boeing 747 model, in which both plant and controller models are taken from reference [86]. In order to establish the similarity to the UCAV6 model tests, the performance of altitude hold mode of Boeing 747 controller is tested in this platform.

The Boeing 747 model has a cable and actuator dynamics blocks and this model is suitable for adding the simulation of elevator deflection. To simulate elevator deflection with hardware, a brushless DC motor with its encoder and servo amplifier are integrated to the platform. The test results show that the behavior of the plant in setup and in Simulink environment is close to each other, where the difference between them are very small compared to desired altitude value.

7.2 Conclusions

In aerospace industry, obtaining an open source about rapid control prototyping, real-time applications in an experimental setup, and embedded software applications is very difficult because of competition between companies. This thesis study provides a comprehensive background related to these concepts.

In this thesis study, a reusable test platform based on real-time facilities and embedded software are designed and realized. Any plant and its possible controller models designed in Simulink environment can be tested in real time by the help of this test platform**.**

An UCAV6 model is used for the plant and a discrete PID controller is designed to check the functionality of the platform. The performance of the discrete PID controller is tested with some test scenarios at 100 ft and 5,000 ft to verify the functionality of the platform.

A noise problem is encountered while running the tests at 5,000 ft, especially after reaching the desired elevation, which is most probably due to sudden jumps in the derivative gains of the controller, or the interface used between controller and plant, or something else. However, since the aim of the study is limited to the design of a real time, reusable test platform and to the demonstration of the controller performance in this platform by some tests, no further attempt is made to alleviate this problem.

The reusability of the platform is confirmed by using Boeing 747 model. This model includes both the flight dynamics model with its controller algorithms. The plant and controller models are separated from each other and downloading in to real time environments. The differences of the plant outputs in between setup and Simulink tests are very small with respect to desired plant outputs.

To sum up, this thesis study enables a beneficial background for several further applications related to a real time test platform and with modern tools used in industry, hence it can be called as a beneficial, realistic, and reusable study.

7.3 Recommendation for Future Work

Without too tight budget constraint for designing the platform, more reliable interface alternatives can be employed. Hence, a more powerful platform can be designed. For example, for the communication between the plant and controller, MIL-STD-1553B bus can be chosen instead of RS-232.

As a further application, FMS algorithms can be generated and inserted to either on the UCAV6 autopilot model or any other autopilot models. Then by the help of the platform, these FMS algorithms can be confirmed in real time.

The test platform can be enlarged with real LRUs like Embedded GPS INS (EGI), Air Data Computer (ADC), Radar Altimeter, etc, or their emulators in further studies. Hence, the platform enables to check the performance of the real LRUs and the reaction of the controller under normal and unexpected conditions, which are possible to occur during the flight. The outputs of these LRUs, such as airspeed, rate of climb, engine performance, attitude of the aircraft, can be shown on a multi function displays (MFD).

REFERENCES

[1] Pratt, R.W., "Flight Control Systems: Practical issues in design and implementation", Institution of Engineering and Technology, 2000.

[2] DO-178B, "Software Considerations in Airborne Systems and Equipment Certification", RTCA SC-167 and EUROCAE WG-12, 1 December 1992.

[3] Lin, C. L., Su, H.W., Intelligent Control Theory in Guidance and Control System Design, Institute of Automatic Control Engineering, Feng Chia University, Taichung, Taiwan, R.O.C., Vol.24, No.1, 2000, pp.15-30, http://nr.stpi.org.tw/ejournal/proceedinga/v24n1/15-30.pdf, Last accessed on February, 2008

[4] Potter, B., "Using MathWorks Tools to Develop DO-178B Certified Code", Honeywell, Inc., 2004. http://www.mathworks.com/company/newsletters /aero_digest/aug04/Honeywells.pdf, Last accessed on February 2008.

[5] Lim, B.A., "Design and Rapid Prototyping of Flight Control and Navigation System for an Unmanned Aerial Vehicle", M.S. Thesis, Aeronautical and Astronautical Engineering Department, Naval Postgraduate School, Monterey, CA, USA, January 2002.

[6] Dittrich, J.S., "Design and Integration of an Unmanned Aerial Vehicle Navigation System", M.S. Thesis, Aeronautical and Astronautical Engineering Department, School of Aerospace Engineering, Georgia Institute of Technology, Pittsburgh, PA, USA, 2002.

[7] Froncillo, S.J., "Design of Digital Control Algorithms for Unmanned Air Vehicles", M.S. Thesis, Aeronautical and Astronautical Engineering Department, Naval Postgraduate School, Monterey, CA, USA, 1998.

[8] Flood, C.H., **"**Design and Evaluation of a Digital Flight Control System for Frog Unmanned Aerial Vehicle**",** M.S. Thesis, Aeronautical and Astronautical Engineering Department, Naval Postgraduate School, Monterey, CA, USA, 2001.

[9] Rivers, T.C., "Design and Integration of a Flight Management System (FMS) for the Unmanned Air Vehicle FROG", M.S. Thesis, Aeronautical and Astronautical Engineering Department, Naval Postgraduate School, Monterey, CA, USA, 1998.

[10] Low, K.H., Wang H., Wang, M.Y., "On the Development of a Real Time Control System by Using xPC target: Solution to Robotic System Control", Proceedings of the 2005 IEEE International Conference on Automation Science and Engineering, Edmonton, Canada, 1-2 August 2005. pp. 345-350

[11] Thompson, K.E., "F-16 Uninhabited Air Combat Vehicles", A Research Report, Air Command and Staff College, Air University, Maxwell Air Force Base, AL, USA, July 1998.

[12] Ataç, S., "GPS Based Altitude Control Of An Unmanned Air Vehicle Using Digital Terrain Elevation Data", M.S. Thesis, Mechanical Engineering Department, Middle East Technical University, Ankara, Turkey, June 2006.

[13] Nilsson, C., Hall, C., Heinzen, S., Chokani, N., "GPS Auto-Navigation Design For Unmanned Air Vehicles", AIAA Aerospace Sciences Meeting and Exhibit, 40th, Reno, NV, January 2002.

[14] MIL-STD-498 Software Development and Documentation, Superseded by IEEE-12207, 1994.

[15] Ransom C., "Advantages of Rapid Prototyping technology", EADS, Space propulsion, Rapid Prototyping at the Ottobrunn Production Centre, Astrium GmbH 81663, Munich, Germany, 2007.

[16] SRA International, Inc., http://www.uavforum.com, Last accessed on October 2007.

[17] Fahlstrom, P.G., "Introduction to UAV System", UAV Center Co., Ltd., 302-834, South Korea, November 2002.

[18] Moore, R.A., "Unmanned Aerial Vehicles", Aerospace America, February 1989.

[19] Fulghum, D.A., "Groom Lake Tests Stealth", Aviation Week, 5 February 1996. pp.27.

[20] Thompson, M.C., "A Bridge to the Future of Air Combat", Aerospace Power Journal, USAF, Spring 2000, Vol.14, No. 1, pp. 22-36.

[21] xPC target for Use with Real-Time Workshop® User's Guide Version 2, The MathWorks, Inc., MA, USA, October 2004.

[22] xPC target for Use with Real-Time Workshop® Getting Started Version 2, The MathWorks, Inc., MA, USA, October 2004.

[23] MF624 multifunction I/O card data sheet, Humusoft, Inc., Czech Republic, 2007.

[24] PCI-6602 data sheet, National Instruments, Inc., http://sine.ni.com/nips/cds /view/p/lang/en/nid/1123, Last accessed on February 2008.

[25] Wind River System, "Run Time Platforms", http://www.windriver.com/ /platforms/safety_critical/index.html, Last accessed on February 2007.

[26] VxWorks® Programmer's Guide, 5.4, Wind River Systems, Inc., Alameda, CA, USA, 1999.

[27] Collier, J., "An Overview Tutorial of the VxWorks® Real-Time Operating System", Wind River Systems, Inc., Alameda, CA, USA, March 2004.

[28] Tornado BSP Developer's Kit for VxWorks® User's Guide, Wind River Systems, Inc., Alameda, CA, USA, 1999.

[29] Tornado User's Guide. Wind River Systems, Inc., Alameda, CA, USA, 1999.

[30] C772 Power PC G3/G4 VME Single Board Computer Data Sheet, Aitech Defense Systems, Inc., Chatsworth, CA, USA, 1999.

[31] C772 Firmware Programming Instructions Revision 1.2, Aitech Defense Systems, Inc., Chatsworth, CA, USA, 1999.

[32] C772 Power PC G3/G4 VME Single Board Computer User's Guide Revision 2.5, Aitech Defense Systems, Inc., Chatsworth, CA, USA, 1999.

[33] C431 A/D, D/A, and Digital I/O VME Board Data Sheet, Aitech Defense Systems, Inc., Chatsworth, CA, USA, 1999.

[34] C431 A/D, D/A and Digital I/O VME Board User's Guide Revision 1.7, Aitech Defense Systems, Inc., Chatsworth, CA, May 2005.

[35] Sensoray Model 626 PCI Multifunction I/O Board Instruction Manual Revision F, Sensoray Co., Inc., Oregon, USA, 28 January 2004.

[36] Servo Amplifier 4-Quadrant PWM for Brushless DC-Servomotors Operating Instructions Edition 2, Minimotor Sa, 6980 Croglio Switzerland, 20 August 2003.

[37] Optical Encoders, Minimotor Sa, 6980 Croglio, Switzerland, 2003.

[38] Quick Assembly Two and Three Channel Optical Encoders Technical Data Agilent Technologies, Inc., Obsoletes 5965-5875E (11/99) 5988-2579EN 4, May 2001, http://cp.literature.agilent.com/litweb/pdf/5988-3996EN.pdf, Last accessed on February 2008.

[39] Selection Chart, Combination Chart and Ordering Information, Minimotor Sa, 6980 Croglio, Switzerland, 2001.

[40] Brushless DC-Servomotors Electronic Commutation, Drive Electronics Combination Chart, Faulhaber Minimotor Sa, 6980 Croglio, Switzerland, 2003.

[41] Valasek, J., Aero 625 Digital Control of Aerospace Systems Course Notes, Texas A&M University, College Station, TX, USA, 2006.

[42] Wong, H., "An Overview of RS-232 and RS-485 Serial Communications", National Online Seminars, National Semiconductor Corporation, 2003, http://www.national.com/onlineseminar/2003/interface/transcript.html, Last accessed on February 2008.

[43] Getting Started with Real-Time Workshop® 7, The MathWorks, Inc MA, USA, September 2005.

[44] Real-Time Workshop® 7 Target Language Compiler, The MathWorks, Inc., MA, USA, September 2005.

[45] Real-Time Workshop® 7 User's Guide, The MathWorks, Inc., MA, USA, September 2007.

[46] Getting Started with Real-Time Workshop® Embedded Coder 5, The MathWorks, Inc., MA, USA, September 2007.

[47] Real-Time Workshop® Embedded Coder 5 User's Guide, The MathWorks, Inc., MA, USA, September 2007.

[48] Orenstein, D., "QuickStudy: Application Programming Interface (API), Article on Computer World", http://www.computerword.com, Last accessed on 10 January 10 2000.

[49] Manuals of C772 Firmware (FLC 2.1), Aitech Defense Systems, Inc., Chatsworth, CA, USA, 1999.

[50] Aitech C772 MIL-1553 Drivers, Aitech Defense Systems, Inc., Chatsworth, CA, USA, 1999.

[51] BSPPPC.26 (VxWorks Board Support Package for Power PC-Based Boards), Aitech Defense Systems, Inc., Chatsworth, CA, USA, 1999.

[52] C772 Firmware Programming Instructions, Aitech Defense Systems, Inc., Chatsworth, CA, USA, 1999.

[53] C772 VME Driver Application Program Interface Programmer's Guide, Aitech Defense Systems, Inc., Chatsworth, CA, USA, 1999.

[54] VxWorks 1553B Device Driver for C772 Programmer's Guide, Aitech Defense Systems, Inc., Chatsworth, CA, USA, 1999.

[55] Software Development Kit (SDK626), Sensoray Co., Inc., Tigard, OR, USA, 2004.

[56] Sensoray 626 Software Development Kit for Linux (626 Linux SDK), Sensoray Co., Inc., Tigard, OR, USA, 2004.

[57] Active-X of Sensoray 626, Sensoray Co., Inc., Tigard, OR, USA, 2004.

[58] Sensoray 626 QNX driver (626 QNX), Sensoray Co., Inc., Tigard, OR, USA, 2004.

[59] ENEL 541 Course Lab 1 Report, Introduction to the MATLAB Real-Time Workshop and the xPC target System, Electrical and Computer Engineering Department, The University of Calgary, Canada, 2006, http://www.enel.ucalgary.ca/People/Westwick/Courses/ENEL441/Labs/lab1.pdf, Last accessed on February 2008.

[60] ENEL 541 Course Lab 2 Report, Effect of Sampling on the Frequency Response of a DC-Motor, Electrical and Computer Engineering Department, The University of Calgary, Canada, 2006, http://www.enel.ucalgary.ca /People/Westwick/Courses/ENEL441/Labs/lab2.pdf, Last accessed on February 2008.

[61] ENEL 541 Course Lab 3 Report Discrete-Time PD Control of a DC-Motor, Electrical and Computer Engineering Department, The University of Calgary, Canada, 2006, http://www.enel.ucalgary.ca/People/Westwick /Courses/ENEL441/Labs/lab3.pdf, Last accessed on February 2008.

[62] ENEL 541 Course Lab 4 Report State Space Modeling of a Cart and Pendulum: Introduction to State Observers, Electrical and Computer Engineering Department, The University of Calgary, Canada, 2006, http://www.enel.ucalgary.ca/People/Westwick/Courses/ENEL441/Labs/lab4. pdf, Last accessed on February 2008.

[63] ENEL 541 Course Lab 5 Report, State-Feedback Control of the Cart/Pendulum System, Electrical and Computer Engineering Department, The University of Calgary, Canada, 2006, http://www.enel.ucalgary.ca/ People/Westwick/Courses/ENEL441/Labs/lab5.pdf, Last accessed on February 2008.

[64] Roskam, J., Airplane Flight Dynamics and Automatic Flight Controls PartI, Design, Analysis and Research Corporation, Lawrence, KS, USA, 2001.

[65] Roskam, J., Airplane Flight Dynamics and Automatic Flight Controls PartII, Design, Analysis and Research Corporation, Lawrence, KS, USA, 2001.

[66] McLean, D., Automatic Flight Control Systems, Prentice Hall International Ltd., UK, 1990.

[67] The USAF Stability And Control Datcom Users Manual Volume I, Affdl-Tr-79-3032, McDonnell Douglas Astronautics Company, St. Louis Division, St Louis, MO, USA, April 1979.

[68] The USAF Stability And Control Datcom Users Manual Volume I, Updated by Public Domain Aeronautical Software, Santa Cruz, CA, USA, December 1999.

[69] Etkin, B., and Reid, L. D., Dynamics of Flight Stability and Control, John Wiley & Sons, New York, NY, USA, 1996.

[70] Roskam, J., "Airplane Design Part VI: Preliminary Calculation of Aerodynamic, Thrust and Power Characteristics," Roskam Aviation and Engineering Corporation, Lawrence, KS, USA, 1987.

[71] Stevens, B. L., and Lewis F. L., Aircraft Control and Simulation, John Wiley & Sons, New York, NY, USA, 1992.

[72] Valasek, J., UCAV6 Linear Model, Texas A&M University, College Station, TX, USA, 2003.

[73] Valasek, J., Chen, W., "Observer/Kalman Filter Identification for Online System Identification of Aircraft", AIAA Journal of Guidance, Control, and Dynamics, Vol. 26, No. 2, March–April 2003.

[74] Doebbler, J., Valasek, J., Monda, M., Schaub, H., "Boom And Receptacle Autonomous Air Refueling Using A Visual Pressure Snake Optical Sensor", Texas A&M University, College Station, and Virginia Tech, Blacksburg, AIAA Guidance, Navigation and Control Conference Keystone, CO, AIAA 2006-6504, 21-24 August, 2006.

[75] Donald, D. and Lake, J., The Encyclopedia of World Military Aircraft., Barnes & Noble, USA, 2000.

[76] Aircraft Museum, http://www.aerospaceweb.org/aircraft/, Last accessed on 14 January, 2008.

[77] MIL-F-8785C, Military Specification Flying Qualities of Piloted Airplanes, 5 November 1980.

[78] Angstadt, R., Estrada, J., Diehl, H.T., Flaugher, B., Johnson, M., "Microsecond Delays on Non-Real Time Operating Systems", United States Department of Energy Fermi National Accelerator Laboratory, Fermi Research Alliance, LLC DE-AC02-07CH11359, USA, April 2007, http://des-docdb.fnal.gov/0005/000572/001/ieeeRt2007_5.pdf, Last accessed on February 2008.

[79] Test Engineer Guideline, The University of Washington Aeronautical Laboratory (UWAL), 3 January 2005, http://www.uwal.org/download /documents/Testing%20Suggestions.doc, Last accessed on February 2008.

[80] IEEE 802.3-2005/COR, IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks--Specific requirements, http://standards.ieee.org /getieee802/802.3.html, Last accessed on February 2008.

[81] Franklin, G.F., Powell, J.D., Workman, M.L., Digital Control of Dynamic Systems, Addison-Wesley Longman, 2000.

[82] Abusleme, A., Cipriano, A. and Guarini, M., "A Low-Cost Altitude Control System for the Kadet Senior Radio-Controlled Airplane", IEEE Transactions on Education, Vol. 46, No. 1, pp. 50-60, February 2003.

[83] Simulink® Response Optimization User's Guide Version 3, The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA, USA, March 2006.

[84] Shaw, J.A., "The PID Control Algorithm: How it works, how to tune it, and how to use it", 2nd ed., Process Control Solutions, 374 Cromwell Drive Rochester, NY, USA, 2002.

[85] Sarı, I., Design, Construction and Performance Testing of A Ball-Beam Balancing Experimental Set-Up, M.S. Thesis, Mechanical Engineering Department, Middle East Technical University, Ankara, Turkey, September 2002.

[86] Campa, G., Airlib: Aircraft Library, B747 model, July 2006. http://www.mathworks.co.uk/matlabcentral/fileexchange/loadFile.do?objectId=3019&objectType=file, Last accessed on February 2008.

[87] Furgerson, J., Frodyma, P., "Avionics Databus Solutions; MIL-STD-1553 Tutorial v1.3", AIM GmbH Sasbacher Str.2 79111, Freiburg, Germany, December 2002.

[88] Valasek, J., Downing, D.R., Digital Flight Control Systems: Analysis and Design Course Notes, University of Kansas, San Diego, California, USA, September 2006.

APPENDIX A

COMPONENT SPECIFICATIONS OF THE TEST PLATFORM

This Appendix gives the specifications of the test platform components.

Table A.1 Minimum hardware requirements for the host PC [21]

Hardware	**Description**
Communication	One free serial port (COM1 or COM2) with a 9-pin or 25-pin D-sub connector, or an Ethernet card connected to a network
CPU	Pentium, Athlon, or later
Peripherals	Hard disk drive with 60 MB of free space One 3.5-inch floppy disk drive
RAM	128 MB or more

Table A.2 Software components on host PC

Minimum Software Components	**Software Configuration of the setup host PC**
32-bit operating system	Microsoft Windows XP Professional version 2002 Servis Pack 2
MATLAB	MATLAB Version 7.1
Simulink	Simulink Version 6.4
Real-Time Workshop	Real-Time Workshop Version 6.4 Real-Time Workshop Embedded Coder Version 4.4
C language compilers	Microsoft Visual C/C++ Professional Edition Version 6.0
xPC target Software	xPC target Version 2.8
Tornado ide	Tornado Integrated Development Environment version 5

Table A.3 Minimum xPC target hardware requirements [21]

Hardware	Description
Chip set	PC compatible with UART, programmable interrupt controller, keyboard controller, and counter
Communication	One free serial port (COM1 or COM2) with a 9-pin or 25-pin D-sub connector, or an Ethernet card connected to a network. The xPC target software includes a serial null modem cable and an Ethernet card for the target PC.
CPU	Intel 386/486/Pentium or AMD K5/K6/Athlon with or without a floating-point coprocessor
Keyboard and mouse	When one creates stand-alone applications, keyboard and mouse are needed to control the target PC If a keyboard is not connected, the BIOS might display an error message (keyboard failure). With current BIOS, the BIOS setup to skip the keyboard test can be used.
Monitor	The MathWorks recommends using a monitor, but it is not necessary. One can get all the target information using xPC target functions on the host PC.
Peripheral	One 3.5-inch floppy disk drive. A hard disk drive is not required. If the xPC target Embedded Option is installed, files can be copied to a hard disk or flash memory and boot from that device.
RAM	8 MB or more

Technical Specifications of Sensoray 626 I/O Board :

- Six versatile 24-bit counters for encoders/timers with interrupt generation
- Fail-safe Digital I/O
- E-stop Relay
- Battery backup of counters
- Four 14-bit D/A outputs, 20 kHz update rate
- Sixteen 16-bit differential A/D inputs, 15 kHz rate
- Watchdog timer may be used to reset PCI bus
- 48 bi-directional digital signals for solid state relay boards

- QNX driver
- xPC target Real-time rapid control prototyping software available
- Simulink software available through Quanser Consulting Inc.
- RT-LAB available through Opal-RT

Technical Specifications of Aitech C772 VME Single Board Computer

The C772 is based on the Motorola low power, high performance PowerPC™ microprocessor. The C772 supports a variety of processors Motorola MPC750, Motorola MPC755, Motorola MPC7400 and IBM PPC750 with an L2 cache sizing up to 2 MB fast SRAM. Technical Specifications of C772 board is as below.

- Rugged VME Single-Slot SBC (Universe II Advanced PCI to VME Bridge)
- G3/G4 PowerPC®750 @ 500MHz processor
- Dual Redundant MIL-STD-1553B
- 256 MB of DRAM protected by an EDAC mechanism
- 8 MB of User Flash Memory
- Up to 256 MB Flash File Memory
- Up to 2 MB Level 2 Cache
- Ethernet Interface (10/100 Mbps)
- 16 Discrete Buffered I/O Lines
- Four RS-232/422/485 Multi-Protocol High-Speed Serial Ports
- Four RS-232/422 Standard UART Ports
- Four 32 bit programmable Timers and Watchdog Timer
- Real Time Clock
- Two PMC Slots for Flexible I/O Expansion
- Fully VME64x Compliant
- VxWorks Real Time OS
- Conduction or Air Cooled Versions
- Vibration and Shock Resistant
- 32 bit @ 33 MHz PCI Local Bus

APPENDIX B

INTEGRATION OF THE PLATFORM

This appendix serves as a tutorial defining how to use the tools utilized in this setup and how all components of the setup are brought together. Hence, this appendix would be helpful to those who want to generate such a platform, to learn using some tools utilized in this platform application, and to design only one part of the platform for their applications.

B.1 Setting up and using xPC target tool

xPC target Explorer Window: By writing `xpcexplr` command on MATLAB Command Window, *xPC target Explorer* window pops up as in Figure B.1. Using the menu in this window enables to adjust the properties of host PC and target PC, to load target application to the xPC target, and to run, test, and monitor the results with using scopes. It is also possible to control xPC target with MATLAB command line.

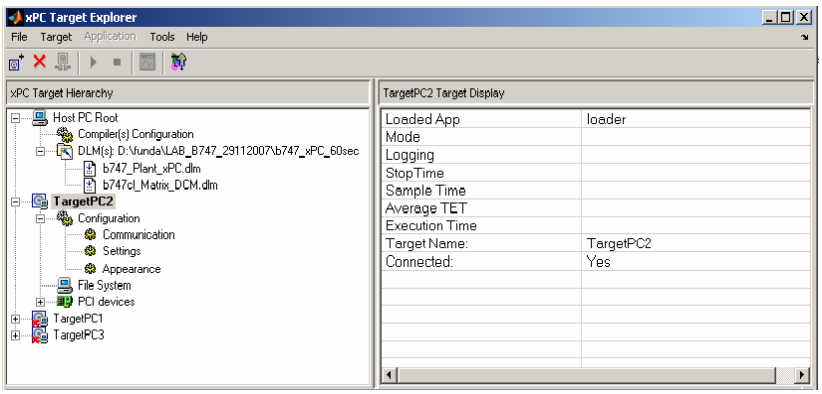

Figure B.1 xPC target explorer window

Host PC Settings: *Compiler Configuration* and *DLM* are two submenus of *host PC* adjustments as can be seen in Figure B.2. *Compiler Configuration* menu is used to define the compiler path in the *host PC*.

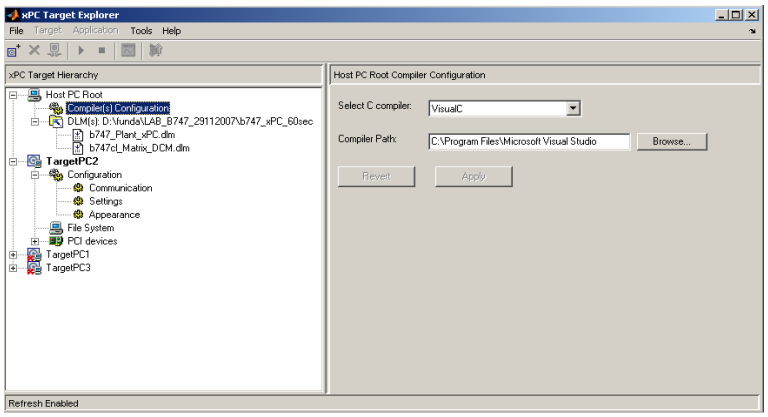

Figure B.2 Host PC adjustments

The way of loading a target application into the target computer is explained in Figure B.3. *b747_Plant_xPC.dlm* and *b747cl_Matrix_DCM.dlm* files are seen as examples in the same figure. Reading "*Simulink Model and Simulating the Model*" topics in reference [21] will be useful in order to set up Simulink models and add target scopes.

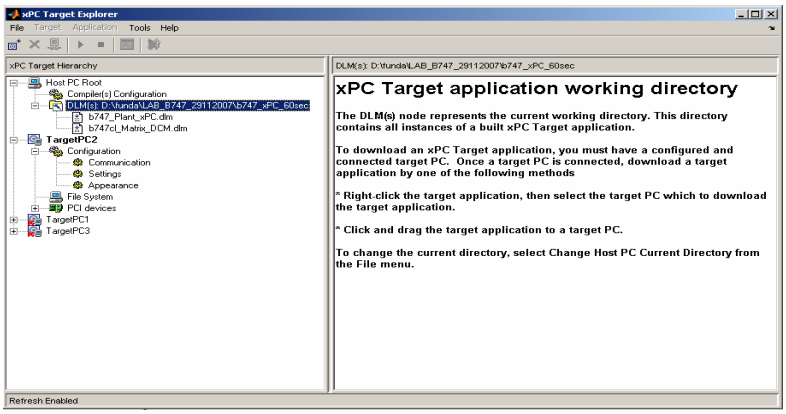

Figure B.3 Host PC adjustments – DLM Menu

Target PC Adjustments

For target *PC* configuration, firstly *Communication*, "*Settings*" and "*Appearance*" adjustments should be made. After these adjustments, a bootdisc (Floppy Disc) is prepared for the target computer. Figure B.4 shows the bootdisc interface of xPC target computer. A detailed information about configuration of a bootdisc is given "*Installation and Configuration of Target Boot Disk*" part in Reference [22]. As pointed out in xPC target configuration in Section 2.3, it is required that the xPC target must have a floppy disc drive except xPC target embedded option be bought. The xPC target tool used in this thesis does not include this option.

It is possible to define more than one target PCs by right-click on "*Host PC Root*" in the *xPC target Explorer* window. Each target PC needs some adjustments. Optionally, more than one target PC configuration can be used to improve the platform. In this thesis, the new target can be defined instead of VxWorks target.

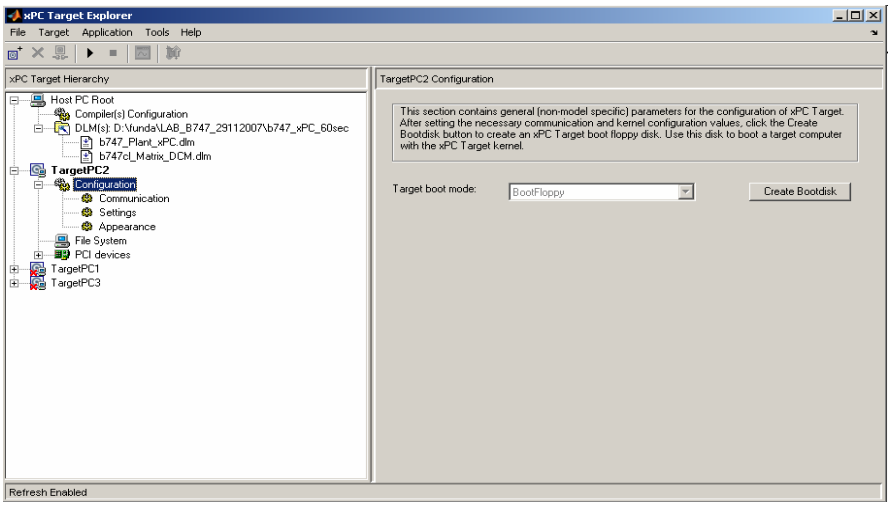

Figure B.4 xPC target bootdisc menu

Communication: TCP/IP is preferred as the communication interface. This interface is recommended by The MathWorks company because of its fast and easy set up. Arrangements are accomplished by the TCP/IP interface. Alternative of this

interface is RS-232. The most important reason of not choosing RS-232 is that the RS-232 port on xPC target hardware is used for communication between target computers. For TCP/IP communication, an ethernet card should be installed on target PC appropriate ethernet card alternatives are explained in Chapter 2.

The adjustment of TCP/IP settings is defined in Reference [22] under "*Environment Properties For Network Configuration*" chapter. As seen in Figure B.5, the TCP/IP configuration is entered depending on the IP configuration of the PC used as xPC target computer on the platform.

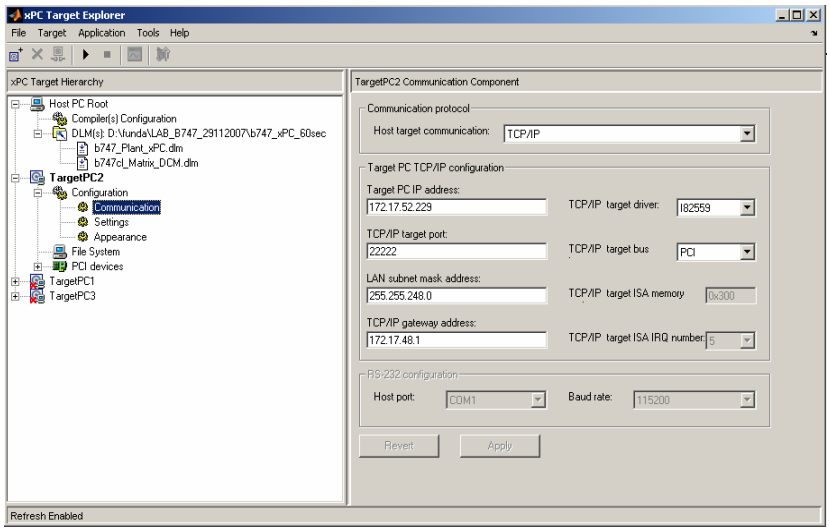

Figure B.5 xPC target communication menu

Settings: In *TargetPC2* (or in *TargetPC1, TargetPC1,* if used*)* settings menu, there are adjustments about *target RAM size* (MB), *maximum model size*, and "enabling" option for *secondary IDE*. The interface of the settings menu is shown in Figure B.6. RAM kernel is used for target applications, data storage, and transferring processes. Target *RAM size* shows the sizes of loaded RAM in the target PC. When "*auto*" option is chosen, target kernel tries to self-assign RAM size automatically. During setup studies, the "*auto*" option is chosen and then target kernel assigns RAM size by itself. No problem is encountered about RAM size. "*Enable Secondary IDE*"

option is not chosen because it is only chosen when it is desired to connect discs to the secondary Integrated Development Environment controller. This kind of use does not stipulate for setup application of this thesis.

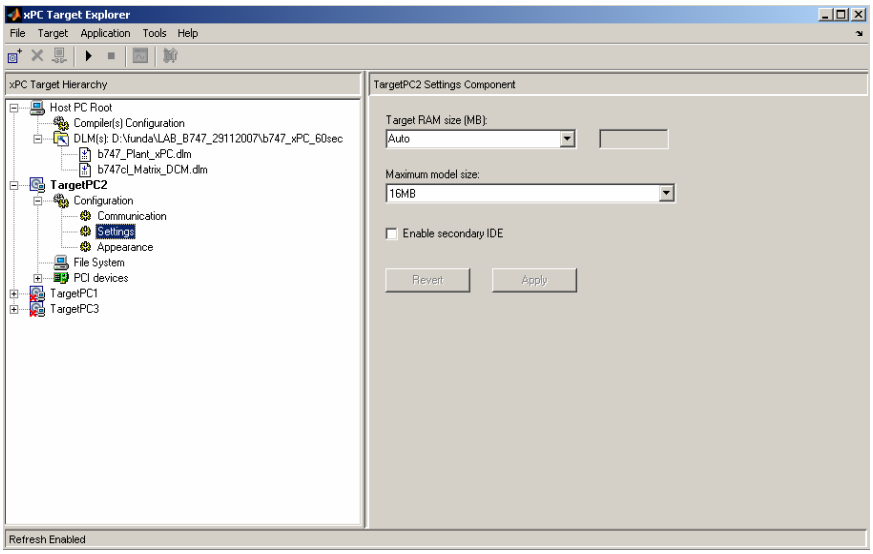

Figure B.6 xPC target settings

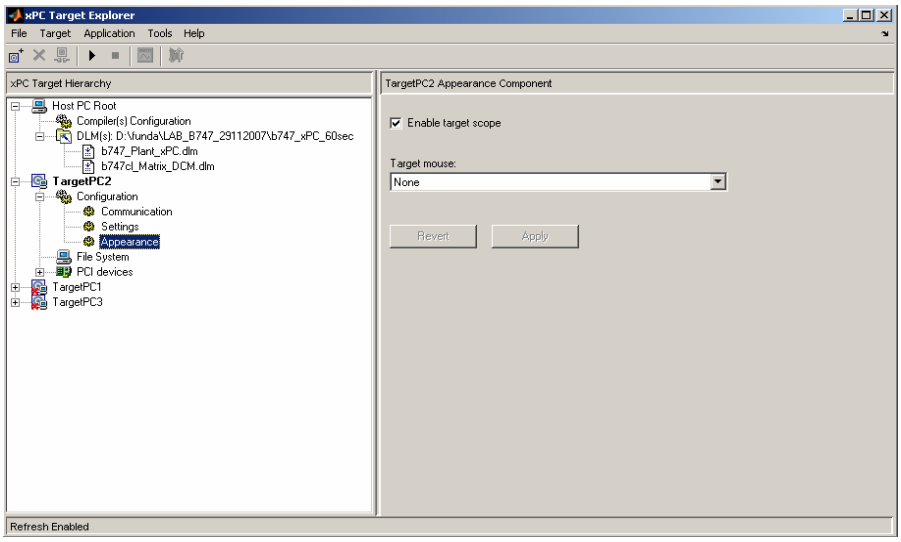

Figure B.7 xPC target appearance

Appearance: In this menu, there are options on enabling a scope on the target display and whether a *Target mouse* is used or not. In setup studies, no target mouse is used but target scopes are employed for monitoring the test results. Figure B.7 shows the interface of the xPC target appearance.

File System: In *File System* submenu under *TargetPC*, as given in Figure B.8, hierarchical structure of drives, folders, and files in the target computer are displayed.

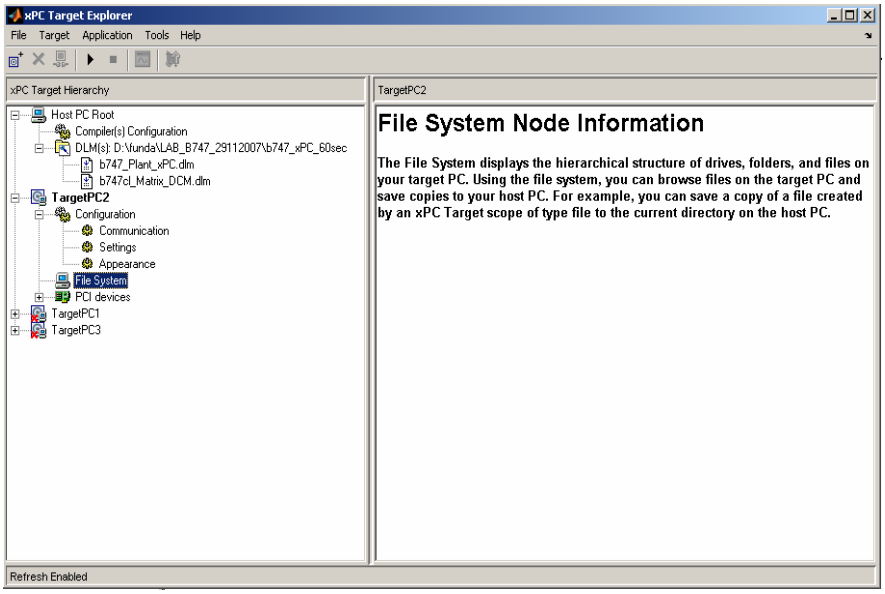

Figure B.8 xPC target file system

PCI Devices: As seen in Figure B.9, devices that are detected in the target computer are listed. The list shows not only supported PCI devices by xPC target but the other PCI devices as well. It is possible to see Sansoray 626 I/O card used in xPC target computer here.

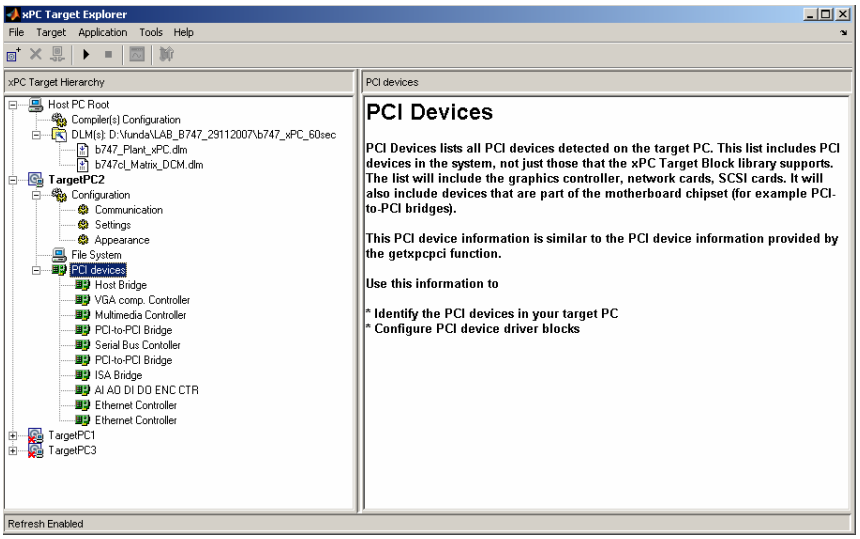

Figure B.9 xPC target PCI devices

RTW Adjustments

In *Configuration Parameters* menu of Simulink model, *system target file* should be chosen as "xpctarget.tlc" and the *language* as C, like shown in Figure B.10. Detailed information about the options can be found in Reference [22] under "*Entering the Real Time Workshop Parameters*" chapter.

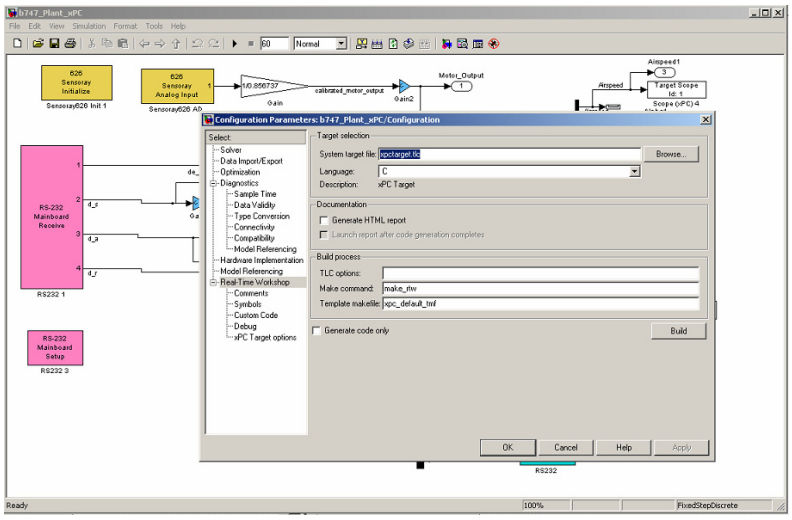

Figure B.10 xPC target RTW adjustments

Building and Downloading Application

In xPC target options page under *Real Time Workshop* menu (in Figure B.11), both *Target Options* should be ticked when one wants to download its application to the default target automatically. "*tg*" is default name of xPC. If desired, it is possible to rename it. The detailed information about the options in this page is located in "*xPC The Target Application*" chapter in Reference [22].

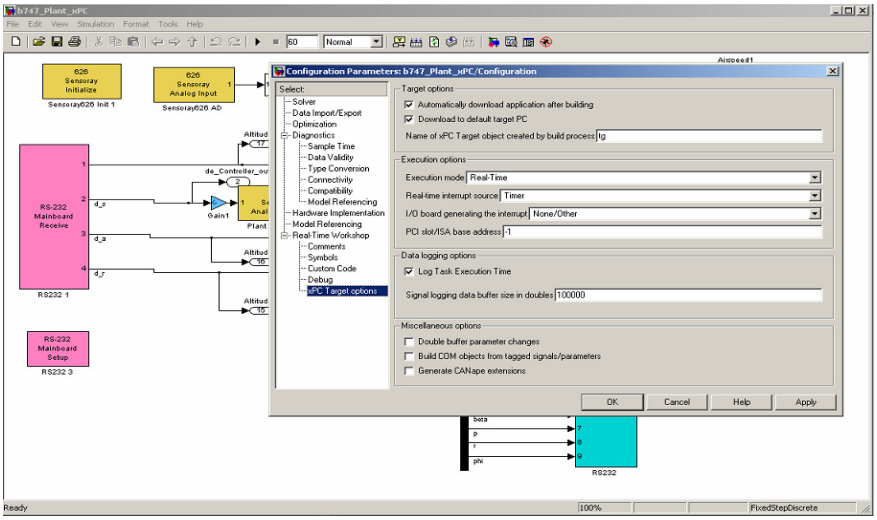

Figure B.11 xPC target options

Test and Problem Solving

Before loading anything to xPC target computer, `xpctest` command should be written on MATLAB command line to verify whether the setup is running correctly or not. The automatic test function of the xPC target is started with this command.

A part of variable values recorded in host PC may not be observed depending on sample time and running time of simulations. This is a very important point although it is not mentioned in reference [22]. Number of recorded logs can be increased by increasing the value of *Signal logging data buffer size in doubles* located in the window in Figure B.11.

At *xPC target Explorer* window, a red cross sign means target is not connected to the host PC. Before running the application, one must check target connection by the help of target PC window by right clicking set up the connection with "*connect*" option.

Checking Sensoray 626 Data Acquisition Card D/A Converter

Sensoray 626 Data Acquisition card includes digital to analog and analog to digital converters. In the applications on this platform, A/D and D/A converters are used to read electrical signal with the xPC, and to send signals from xPC to the real world. To test Sensoray 626 D/A converter, one must open *Simulink Library Browser* and create the model shown in Figure B.12

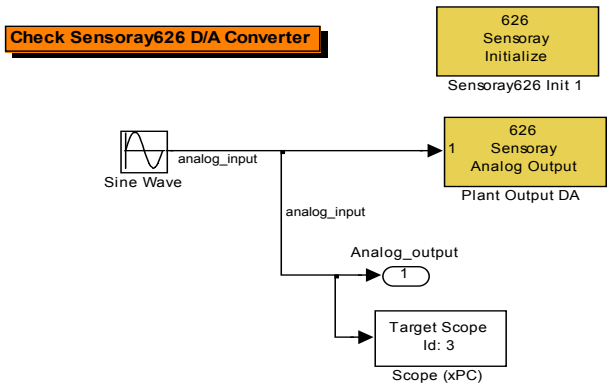

Figure B.12 Simulink block: checking Sensoray 626 D/A converter

The D/A conversion is checked by generating a sine wave in Simulink environment, sending its output via D/A converter, and observing the result on the oscilloscope. After following the procedures in "*How to setup and use of xPC target Tool*" part, the basic Simulink model can be coded by using Real Time Workshop. Under *Configuration Parameters* window in Simulink model, one must open *Real Time Workshop* page and press on the "*Build*" button in the GUI. After *"###Successful completion of xPC target build the model has been built, and downloaded onto the shuttle PC."* message is seen on *Command* window, one may start the application in

the xPC target. If the scope is set up properly and there is no problem with the I/O board, a sine wave should be seen on the scope of oscilloscope. When this exercise was tried at first, the system did not work because of the Sensoray D/A converter driver. An e-mail was sent to MathWorks to inform them about this problem. As a response to this inquiry, they sent the new Sensoray 626 drivers. Up to MATLAB R2006a version, xPC drivers of Sensoray 626 used to have bugs.

This procedure of checking an I/O board is not specific for this thesis. In the Laboratory Report 1 [59] of the Control Systems Course in the Electrical and Computer Engineering Department of University of Calgary, a similar way is followed to check their I/O cards. Before joining DC motor kit to xPC target and power supply, one must follow the start-up procedure that is defined in the next section.

B.2 Close loop application with brushless DC servomotor kit and xPC target

Start-up Procedure of Servo Amplifier BLD 5606 SE4P

The DC power supply voltage within the servo amplifier range ($14\ V \leq V_m \leq 56\ V$) can be used. However, this voltage should be kept as low as possible in order to minimize the electromagnetic interference (EMI) noise. Thus, the optimum power supply voltage V_m [V] is calculated by using the following relation [36],

$$V_m \approx 5 + R \cdot I_{max} + k_E \cdot n_{max} \tag{B.1}$$

where

R [Ohm] is the terminal resistance (phase and phase) constant of motor,

k_E [V/rpm] is the back-emf constant of motor,

I_{max} [A] is the maximum current reached by the motor in specific application,

n_{max} [rpm] is the speed reached by the motor in specific application.

If the values taken from Table 2.7 are used in Equation (B.1), the optimum power supply voltage is found as $V_m = 5 + 1.6\times0.168 + 1.334\times10^{-3}\times28{,}000 = 42.6$ V. The servo amplifier is connected with encoder by following with connection diagram in reference [36] on page 13 and by the help of basic circuit diagram on page 12 in the same reference. Resistor R7 is selected depending on both motor and encoder characteristics. The maximum motor speed must not exceed the encoder speed range. In this study, the maximum motor speed is 28,000 rpm that is smaller than maximum encoder speed, 30,000 rpm. R7 is equal to 16 kΩ depending on R7 table in page 19 of reference [36]. The dimensional drawing in page 12 of the same reference is used to check the place of R7, R1, C1, and C2. The values of the components R1, C1, C2 on SE4P module are checked whether they meet the required values or not. The current limiter is a protection if the motor stalls or if it is overloaded. The system's current limit for the servo amplifier BLD 5606 is calculated by using [36].

$$V_{TP} = \frac{I_{lim} \cdot 1.2}{1.23} \tag{B.2}$$

Control voltage VTP [V] is calculated as 0.6 V by using Equation (B.2). After powering servo amplifier, voltage at point TP is measured by using a voltmeter. TP is shown in Dimension drawing in Servo Amplifier datasheet in reference [36] on page 12. The voltage value is measured 2.64 V and it set to 0.6 V. The start-up procedure of servo amplifier with encoder is completed.

Connection Servo Amplifier with Sansoray626

This merging application started with analyzing BLD 5606-SE4P servo amplifier's connection diagram. It can be observed in Figure B.13 or on page 15 in reference [36]. The connection diagram of the servo amplifier indicates the followings.

- A ground input is needed for power supply at pin number 1.
- A DC power supply voltage (V_m) input is needed at pin number 2.
- Motor speed monitor analog output at pin 15 gives the measurement of motor speed value in terms of Voltage.

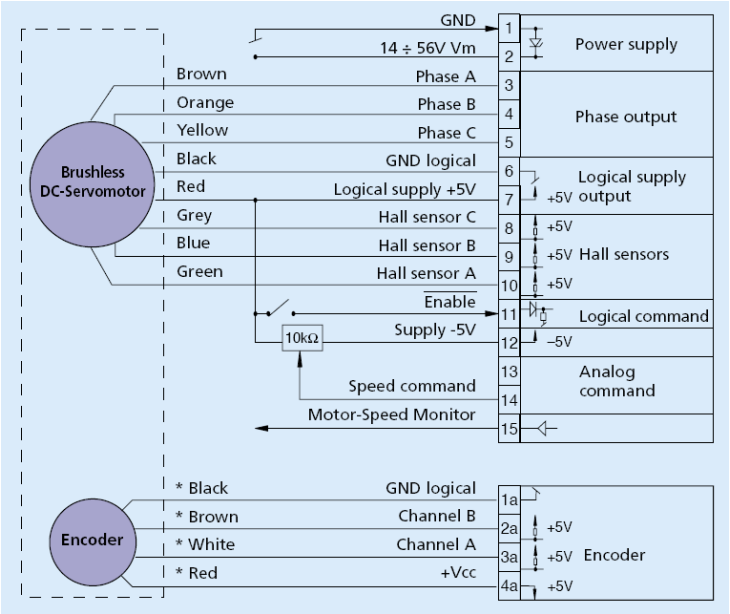

Figure B.13 Connection diagram of BLD 5606-SE4P servo amplifier

The necessary pin number on Sensoray 626 J1 (50 pin IDC ribbon cable connector) should be decided by the help of A/D and D/A connectors table in Sensoray Model 626 Instruction Manual [35]. The numbers of pins, which are used for the connection between servo amplifier and Sensoray 626 are given in Table B.1 and those between servo amplifier and power supply are given in Table B.2.

Table B.1 Connection diagram between servo amplifier and xPC I/O card

BLD 5606-SE4P Servo Amplifier	to Sensoray 626 J1
Pin 1	Pin 39 Ground
Pin 14	Pin 42 (DAC0)
Pin 15	Pin 4 (+AD0)
Pin 15	Pin 3 (-AD0)

Table B.2 Connection diagram between servo amplifier and power supply

BLD 5606-SE4P Servo Amplifier	to Power supply
Pin 1	Ground
Pin 2	+

Calibrating DC motor output

This section includes information about calibrating DC motor output and checking procedure for Sensoray 626 data acquisition card Analog to Digital converter. Using the model given in Figure B.14, one can measure motor speed monitor output of the DC motor. After following the instructions given above about communicating with xPC target and connecting DC motor with xPC target, the calibration gain of the DC motor can be determined.

As seen in block diagram on page 5 in reference [61], the encoder calibration gain is added to simulation after Encoder counter block. For the DC motor, a calibration of its output is needed. Simulink model given in Figure B.15 is downloaded into the xPC target. The purpose of this model is to send constant voltage value, which is 1 V and logged DC motor speed monitor output with time on host PC.

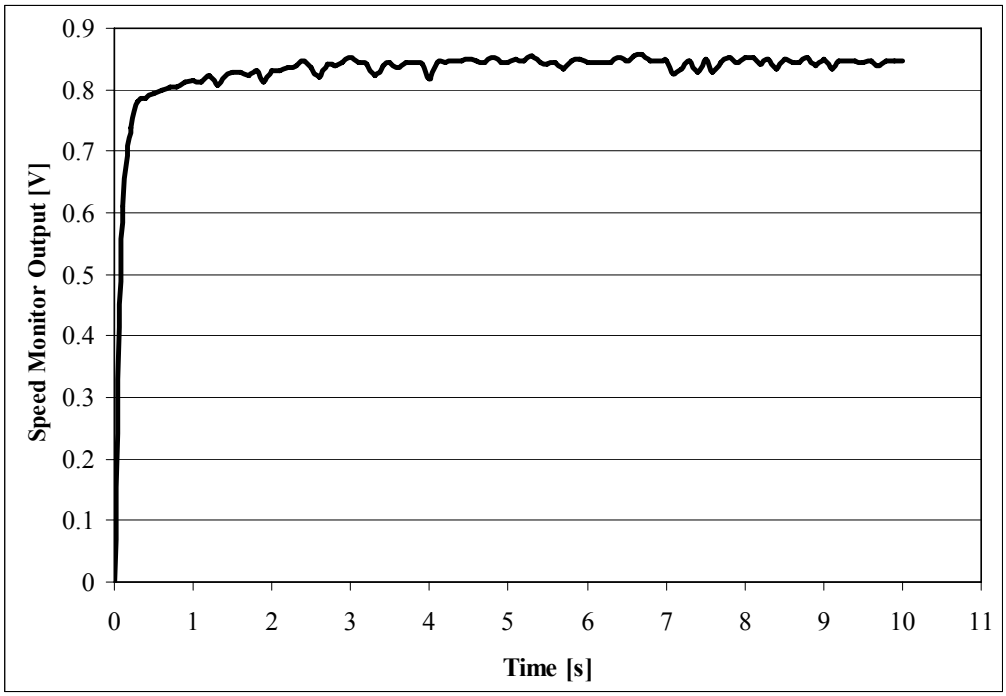

Figure B.14 DC Motor speed monitor test model

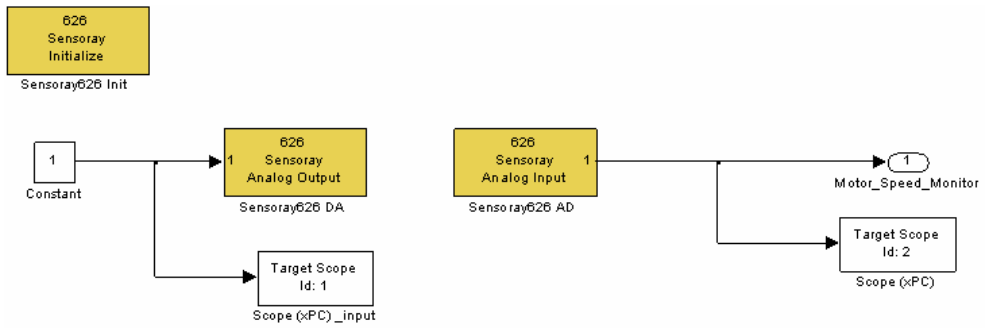

Figure B15 DC motor speed monitor output versus time

To calibrate DC motors, alternatively, one must change R1 on servo amplifier and find a suitable resistance to damp the output at a desired value, or add a motor calibration gain in one's Simulink model.

RS-232 Driver Blocks & RS-232 Message Structures

In this platform, target computers communicate with each other by using an RS-232 serial interface. Therefore, RS-232 drivers should be embedded into both target computers. The necessary C code of the serial communication driver for VxWorks target is given and explained in Chapter 2. For xPC target, this driver is supplied by The MathWorks. Therefore, it is only needed to define RS-232 message structures appropriate to the system and to write these structures in an m-file for using these drivers.

First, RS-232 initialization, send and receive driver blocks should be added to the Simulink model. The procedure of adding these blocks are explained in reference [21] systematically.

One cannot connect to the inputs on the RS-232 send block and the outputs on the RS-232 receive block, because they are not visible until the message structure is loaded into the *MATLAB Workspace*. This should be done before building application. The easiest way to create message structures is to use an *M-file* and load that *M-file* into the *MATLAB Workspace*. For Boeing 747 model, 9 parameters are sent and 4 parameters are received via RS-232 port. Each parameter is one floating-

point number, so “%f” is typed in the M-file. For this setup communication, an extra protocol is added to the *M-file* like a simple error correction. When a data sent or received by xPC target computer, it starts with a number “#” symbol and ends with a dollar “*$*” symbol. By the help of this protocol, the xPC target understands that where the data packages starts and ends when it reads the buffer parameters of RS-232 sent and receive blocks can be seen in below.

```
% RS-232 Send and Receive Block Parameters

% Parameters for the send block.

RS232_Send(1).SendData = '#%f,%f,%f,%f,%f,%f,%f,%f,%f$';

RS232_Send(1).InputPorts = [1 2 3 4 5 6 7 8 9 ];

RS232_Send(1).Timeout = 0.05;

RS232_Send(1).EOM =1;

% Parameters for the receive block.

RS232_Receive(1).RecData = '#%f,%f,%f,%f$';

RS232_Receive(1).OutputPorts = [1 2 3 4 ];

RS232_Receive(1).Timeout = 0.05;

RS232_Receive(1).EOM =1;
```

The xPC target use the serial ports on the target PC for receiving or for sending data. For an extra port, it needs an RS-232 I/O board. There is a trick to send and receive data by using only one serial port on the motherboard. Only one initialize block must be used in the Simulink model, because, the xPC target does not support RS-232 multiplexer.

Generating C codes suitable for VxWorks target

Like xPC target, one can generate C codes from a Simulink model for VxWorks target by using RTWEC. The procedure is explained systematically below. Further information about the options can be found in reference [45]. Fixed-step, discrete (no continuous states) solver should be set with a fixed sample time. For the Boeing 747 model, the sample time is set to 0.05 sec because the model is discretized at this sampling rate. In *the Optimization* page, *inline parameters* should be checked and in configuration, *MATLAB Workspace* should be chosen as a *Source list*.

One can see the parameter loaded in the workspace when the pop up window opens. Desired parameters should be selected to set as *global tunable parameters*, and add to table button should be pressed. *Storage class* of selected parameters is changed by *ImportedExtern*, so one can change these parameters during setup simulation whenever needed. In Real-Time Workshop page, *ert.tlc (Real-Time Workshop Embedded Coder (no auto configuration))* should be chosen as a system target file. C Language should be set and checked HTML report button to generate HTML report. "Generate an example main program" alternative should be chosen for creating VxWorks main code, which is called *ert_main.c*. In addition, VxWorksExample should be set for target operating system.

Finally, *RTW* menu should be opened one more time and the model should be build. Moreover, *"### Successful completion of Real-Time Workshop build procedure for model: B747_controller"* message on Command Window should appear, if RTWEC successfully creates C codes executable in VxWorks target.

Generating Tornado project

After obtaining *modelname_ert_rtw* file (for this case *B747_controller_ert_rtw file*), this model code should be downloaded into the VxWorks target to complete the setup. By the help of Tornado ide, the main function can be generated with adding model codes and extra interface C codes. The name of the Simulink model code is *modelname_step()* according to RTW coding procedure.

<u>To generate Tornado projects</u>;

- Start with opening Tornado ide.
- Choose the target name under Tools => Target Server. In this case, Aitech C772 possessor board is used, so c772_1@T120848650 is taken as a target. T120848650 is the number of this host PC. Please look at Figure B.18
- Press the "*i*" button to connect VxWorks Target. The button is shown in Figure B.16.
- To see the task, which works in the target PC, write "*i*" and press enter.

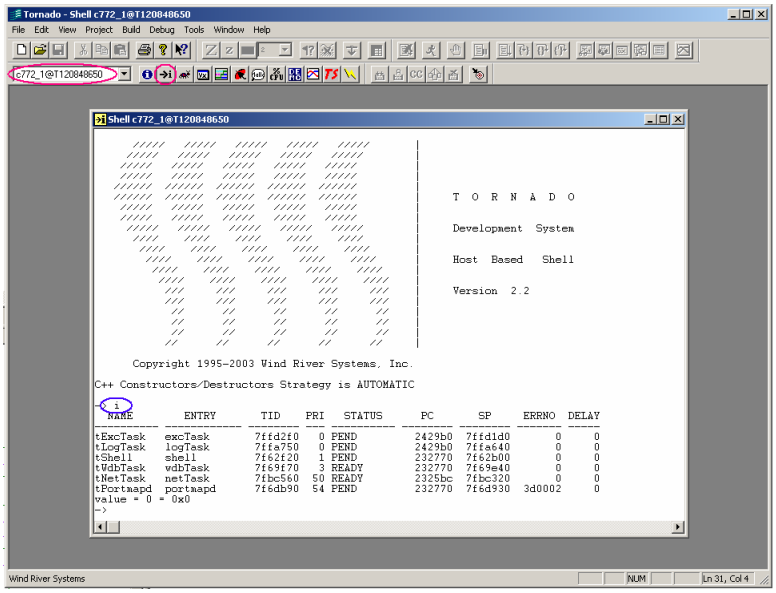

Figure B.16 Tornado Integrated Development Environment screenshot

- Write *moduleShow* to see the application run in the target.
- The model name can be seen in the target console by writing *modelShow* command after downloading the model into the target.
- To download the specific application into target, press button. In this application, a workspace including necessary real times windows library and interfaces codes are created. After compiling, to download all the codes to the target, write the console *modelname.out.*

- To start the task that includes all the defined code, *istart* command should be written on the target console. As shown in Figure B.17, device open successfully is printed on the target console that means it is possible to start the application successfully.

The command, *reboot*, should be written on the console for booting the target, loading, and starting the model code again. Figure B.17 shows the C772 console. For further information about using Tornado IDE, look at Tornado User's Guide [29].

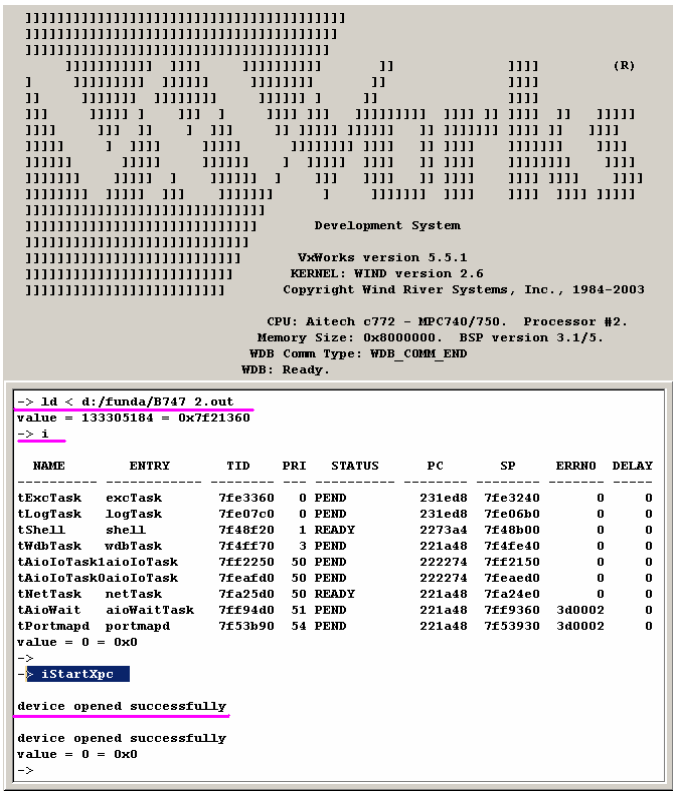

Figure B.17 C772 console on host PC screenshot

APPENDIX C

AV-8B HARRIER II SPECIFICATIONS

Table C.1 AV-8B Harrier II specifications [75] & [76]

History	
First Flight	(YAV-8B) 9 November 1978 (GR.5) 30 April 1985 (GR.7) 29 November 1989
Service Entry	(AV-8B) 12 January 1985 (AV-8BII Plus) June 1993
Crew	**One Pilot**
Estimated Cost	**$23.7 Million [2003$]**
Airfoil Sections	
Wing Root	unknown supercritical (11.5%)
Wing Tip	unknown supercritical (7.5%)
Dimensions	
Length	(AV-8B) 46.33 ft (14.12 m) (GR.7) 47.07 ft (14.36 m)
Wingspan	30.33 ft (9.25 m)
Height	11.65 ft (3.55 m)
Wing Area	230 ft^2 (21.37 m^2)
Weights	
Empty	(AV-8B) 13,970 lb (6,335 kg) (GR.7) 15,705 lb (7,125 kg) (AV-8B Plus) 14,865 lb (6,745 kg)
Normal Takeoff	22,950 lb (10,410 kg)
Max Takeoff	31,000 lb (14,060 kg) [short takeoff] (AV-8B) 20,595 lb (9,340 kg) [vertical takeoff] (GR.7) 19,180 lb (8,700 kg) [vertical takeoff] (AV-8B Plus) 20,755 lb (9,415 kg) [vertical takeoff]
Fuel Capacity	internal: 7,800 lb (3,540 kg) external: 8,070 lb (3,661 kg) in four 300 gal (1,135 L) tanks
Max Payload	(AV-8B) 13,235 lb (6,000 kg) [short takeoff] 4,000 lb (1,815 kg) [vertical takeoff]

Table C.1 AV-8B Harrier II specifications [75] & [76] (Cont'd)

Propulsion	
Powerplant	one Rolls-Royce Pegasus Mk 105 vectored-thrust turbofan
Thrust	21,750 lb (96.75 kN)
Performance	
Max Level Speed	at altitude: Mach 0.98 at sea level: 675 mph (1,085 km/h), Mach 0.89
Initial Climb Rate	Unknown
Service Ceiling	unknown
Range	typical: 1,200 nm (1,110 km) ferry: (AV-8B) 1,800 nm (3,335 km) (GR.7) 1,600 nm (2,965 km)
Endurance	3 hr
g-Limits	+8 / -3
Armament	
Gun	(AV-8B) two 25-mm General Electric GAU-12/U cannons (300 rds total) (GR.7) two 25-mm Aden gun pods (100 rds ea)
Air-to-Air Missile	up to six AIM-9 Sidewinder, AIM-120 AMRAAM, Matra 550 Magic
Air-to-Surface Missile	AGM-12 Bullpup, AGM-62 Walleye, up to four AGM-65 Maverick, AGM-84 Harpoon, Sea Eagle
Bomb	GBU-12/16 Paveway laser-guided, GBU-32/38 JDAM, up to 15 Mk 82 GP, up to five Mk 83 GP, Mk 20 Rockeye, CBU-99 cluster, up to seven BL.755
Other	up to six Matra 155 rocket pods, ECM pods, 30-mm gun pods, recon camera pods
Known Variants	
AV-8A	Original US Harrier based on the British GR.1
YAV-8B	Prototype rebuilt from an AV-8A Harrier
AV-8B	Production attack fighter for the US Marines, later models built to the night attack standard (starting with 167th airframe) including a FLIR and night-vision goggles; about 300 built
TAV-8B	Two-seat trainer; 24 built for the USMC and 2 for the Italian Navy

Table C.1 AV-8B Harrier II specifications [75] & [76] (Cont'd)

EAV-8B	Attack models similar to the AV-8B built for the Spanish Navy; 12 built
T.6	Two-seat trainer for the RAF
GR.7	Improved GR.5 optimized for night attack, equipped with new cockpit displays and a HUD, TV/laser target trackers, and a FLIR; 34 built and all GR.5 models upgraded to the same standard
GR.9	Proposed fighter based on the GR.5 but armed with AIM-120 AMRAAMs, not built
T.10	RAF trainer with night attack capability; 13 built
Known Combat Record	Iraq - Operation Desert Storm (USMC, 1991) Bosnia - Operation Deliberate Force (USMC, 1995) Kosovo - Operation Allied Force (USMC, 1999) Afghanistan - Operation Enduring Freedom (USMC, UK, 2001-present) Iraq - Operation Iraqi Freedom (USMC, 2003-present)
Known Operators	Italy, Marina Militare (Italian Navy Aviation) Spain, Arma Aérea de la Armada Española (Spanish Naval Aviation) United Kingdom (Royal Air Force) United States (US Marine Corps)

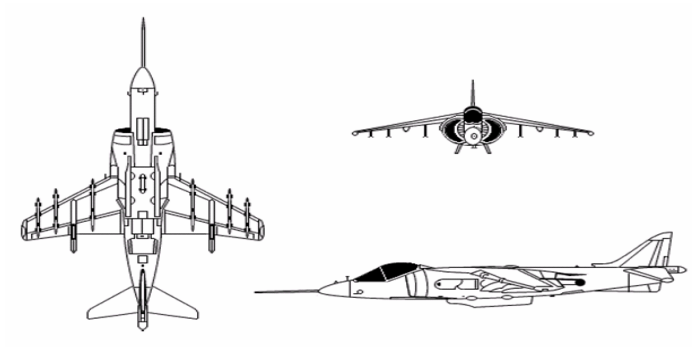

Figure C.1 3-D view of Harrier II at the same time UCAV6 [72]

APPENDIX D

STABILITY DERIVATIVES

Table D.1 Meanings, expressions and units of stability derivatives [64]

Expression	Meaning	unit
$X_u = \frac{-\bar{q}_1 S(C_{D,u} + 2C_{D,1})}{mU_1}$	Forward acceleration per unit change in speed	1/sec
$X_{T,u} = \frac{\bar{q}_1 S(C_{T,x_u} + 2C_{T,x_1})}{mU_1}$	Forward acceleration per unit change in speed (due to thrust)	1/sec
$X_\alpha = \frac{-\bar{q}_1 S(C_{D,\alpha} - C_{L,1})}{m}$	Forward acceleration per unit angle of attack	ft/sec^2/rad.
$X_{\delta_e} = \frac{-\bar{q}_1 S C_{D,\delta_e}}{m}$	Forward acceleration per unit elevator angle	ft/sec^2/rad
$Z_u = \frac{-\bar{q}_1 S(C_{L,u} + 2C_{L,1})}{mU_1}$	Vertical acceleration per unit change in speed	1/sec
$Z_\alpha = \frac{-\bar{q}_1 S(C_{L,\delta} + 2C_{D,1})}{m}$	Vertical acceleration per unit angle of attack	ft/sec^2/rad
$Z_\alpha = \frac{-\bar{q}_1 S\bar{c} C_{L,\dot{\alpha}}}{2mU_1}$	Vertical acceleration per unit rate of change of angle of attack	ft/sec/rad
$Z_q = \frac{-\bar{q}_1 S\bar{c} C_{L,q}}{2mU_1}$	Vertical acceleration per unit pitch rate	ft/sec/rad
$Z_{\delta_e} = \frac{-\bar{q}_1 S C_{L,\delta_e}}{m}$	Vertical acceleration per unit elevator angle	ft/sec^2/rad
$M_u = \frac{\bar{q}_1 S\bar{c}(C_{m,u} + 2C_{m,1})}{I_{yy}U_1}$	Pitch angular acceleration per unit change in speed	rad/sec/ft
$M_{T,u} = \frac{\bar{q}_1 S\bar{c}(C_{m,Tu} + 2C_{m,T1})}{I_{yy}U_1}$	Pitch angular acceleration per unit change in speed (due to thrust)	rad/sec/ft
$M_\alpha = \frac{\bar{q}_1 S\bar{c} C_{m,\alpha}}{I_{yy}}$	Pitch angular acceleration per unit angle of attack	1/sec^2

Table D.1 Meanings, expressions and units of stability derivatives [64] (Cont'd)

$M_{T,u} = \frac{\bar{q}_1 S \bar{c} C_{m,T\alpha}}{I_{yy}}$	Pitch angular acceleration per unit angle of attack (due to thrust)	$1/sec^2$
$M_{\dot{\alpha}} = \frac{\bar{q}_1 S \bar{c}^2 C_{m,\dot{\alpha}}}{2 I_{yy} U_1}$	Pitch angular acceleration per unit rate of change of angle of attack	1/sec
$M_q = \frac{\bar{q}_1 S \bar{c}^2 C_{m,q}}{2 I_{yy} U_1}$	Pitch angular acceleration per unit pitch rate	1/sec
$M_{\delta_e} = \frac{\bar{q}_1 S \bar{c} C_{m,\delta_e}}{I_{yy}}$	Pitch angular acceleration per unit elevator angle	$1/sec^2$

APPENDIX E

UCAV6 PLANT PROPERTIES

Table E.1 UCAV6 plant properties

Operating Point No.	**1**	**2**	**3**
Velocity [ft/s]	10	25	50
Altitude [ft]	100	100	100
Poles	0 -0.88 ±1.45i -0.01 ± 0.10i	0 -1.13 ±1.54i -0.01 ± 0.13i	0 -1.31 ± 1.70i -0.01 ± 0.13i
Zeros	-6.81 5.26 -0.01	-11.23 9.73 -0.01	-13.90 13.09 -0.01
Damping ratios	0.12 0.51	0.12 0.59	0.12 0.61
Undamped natural frequencies [rad/s]	0.1 1.7	0.1 1.9	0.1 2.1
Velocity error coefficient [ft/rad.s]	-3	-5	-28
K, gain [rad/ft]	0.2	0.5	1.8
Operating Point No.	**4**	**5**	**6**
Velocity [ft/s]	73	97	143
Altitude [ft]	100	100	100
Poles	0 -1.61 ± 2.05i -0.01 ± 0.13i	0 -1.80 ± 2.30i -0.01 ± 0.14i	0 -1.89 ±2.39i -0.02 ±0.16i
Zeros	-18.24 17.03 -0.01	-20.98 19.91 -0.01	-21.31 20.75 -0.01
Damping ratios	0.12 0.61	0.13 0.61	0.14 0.62
Undamped natural frequencies [rad/s]	0.1 2.6	0.1 2.9	0.1 3.0
Velocity error coefficient [ft/rad.s]	-79	-126	-183
K, gain [rad/ft]	3	5	10

Table E.1 UCAV6 plant properties (Cont'd)

Operating Point No.	**7**	**8**	**9**
Velocity [ft/s]	185	237	288
Altitude [ft]	100	100	5,000
Poles	0 -1.93 ± 2.41i -0.02 ± 0.18i	0 -2.06 ± 2.36i -0.03 ± 0.22i	0 -2,44 ± 2.21i -0.03 ± 0.22i
Zeros	-24.88 21.03 -0.01	-27.78 26.31 -0.01	-31.06 30.20 -0.01
Damping ratios	0.14 0.62	0.14 0.65	0.01 0.74
Undamped natural frequencies [rad/s]	0.1 3.0	0.2 3.1	0.2 3.2
Velocity error coefficient [ft/rad.s]	-270	-318	-419
K, gain [rad/ft]	14	16	12

Table E.1 UCAV6 plant properties (Cont'd)

Operating Point No.	**10**	**11**	**12**
Velocity [ft/s]	480	668	853
Altitude [ft]	5,000	5,000	5,000
Poles	0 -2.95 ± 2.52i -0.04 ± 0.23i	0 -3.57 ± 2.78i -0.04 ± 0.24i	0 -3.97 ± 3.03i -0.06 ± 0.29i
Zeros	-33.25 29.62 -0.01	-33.60 31.95 -0.02	-34.05 32.93 -0.02
Damping ratios	0.17 0.76	0.18 0.78	0.20 0.79
Undamped natural frequencies [rad/s]	0.2 3.8	0.2 4.5	0.3 5.0
Velocity error coefficient [ft/rad.s]	-852	-2,041	-2,917
K, gain [rad/ft]	50	107	199

APPENDIX F

F-4, F-104, LEARJET24 PLANT PROPERTIES

The reference [64] presents geometric, mass, and flight condition data of several airplanes along with their stability and control derivatives. F-4, F-104, and Learjet24 are chosen because they are jet engine airplanes among them.

The transfer functions of altitude to elevator deflection, the open-loop poles, zeros, and gains and plant properties as well as the damping ratios and undamped natural frequencies of the F-104, F-4, and Learjet are given below.

F-1 F-104 PLANT PROPERTIES

McDonnell F-4 is a twin-engine fighter attack airplane. It is trimmed at U=170 kts

H= 0 ft, M= 0.257.

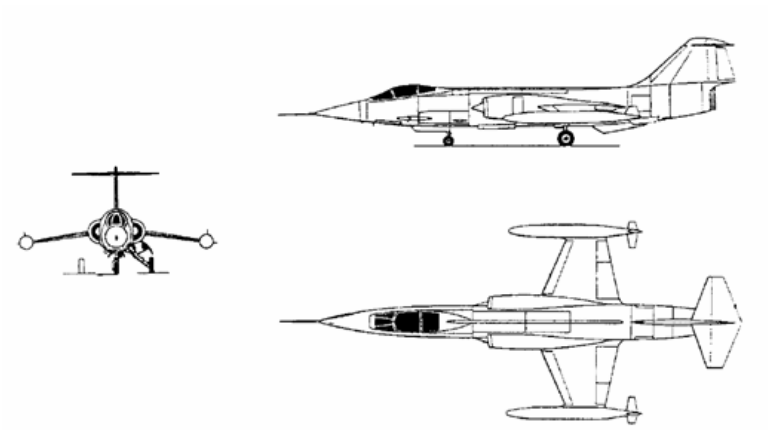

Figure F.1 3-D view of F-104

Table F.1 Plant properties of F-104

Velocity [kts]	170
Altitude [ft]	0
Mach Number.	0.25
Altitude to elevator deflection transfer function	$\frac{h(s)}{\delta_e(s)} = \frac{4250\ s^3 + 510\ s^2 - 114920\ s - 4420}{287\ s^5 + 271\ s^4 + 636\ s^3 + 31\ s^2 + 13\ s}$
Poles	0 -0.45 ± 1.39i -0.02 ± 0.14i
Zeros	-5.24 5.15 -0.04
Damping ratios	0.14 0.31
Undamped natural frequencies [rad/s]	0.14 1.47
K, gain [rad/ft]	14.8
Velocity error coefficient [ft/rad.s]	-340

F-2 F-4 PLANT PROPERTIES

F-104 is a single jet-engine interceptor fighter airplane. It is trimmed at U=518 kts H= 35,000 ft, M=0.9.

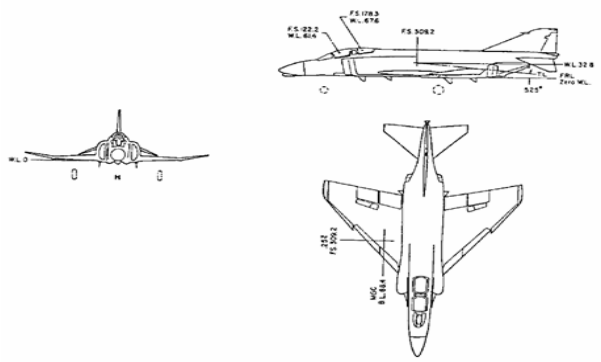

Figure F.2 3-D view of F-4

Table F.2 Plant properties of F-4

Velocity [kts]	518
Altitude [ft]	35,000
Mach Number	0.9
Altitude to elevator deflection transfer function	$\frac{24864\ s^3 + 1554\ s^2 - 2.541 \times 10^6\ s - 12950}{876\ s^5 + 1102\ s^4 + 7106\ s^3 - 4.9\ s^2 - 11\ s}$
Poles	0 -0.63 ± 2.78i 0.04 ± 0.00i
Zeros	-10.13 10.08 0.01
Damping ratios	1.00 0.22
Undamped natural frequencies [rad/s]	0.04 2.84
K, gain [rad/ft]	28.3
Velocity error coefficient [ft/rad.s]	1,177

F-3 Learjet24 PLANT PROPERTIES

Learjet24 is a twin jet-engine airplane, trimmed at U=100 kts, H= 0 ft, M=0.152.

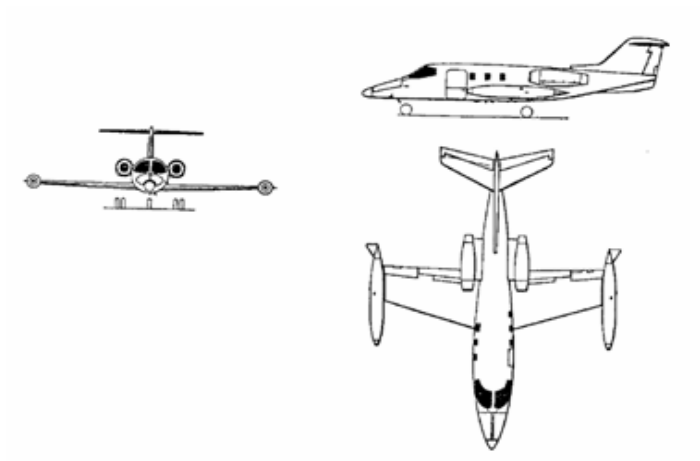

Figure F.3 3-D view of Learjet24

Table F.3 Plant properties of Learjet24

Velocity [kts]	100
Altitude [ft]	0
Mach Number	0.152
Attitude to elevator deflection transfer function	$\frac{780\ s^3 + 300\ s^2 - 29100\ s + 200}{170.6\ s^5 + 305.7\ s^4 + 435\ s^3 + 29\ s^2 + 23\ s}$
Poles	0 -0.88 ± 1.29i -0.01 ± 0.23i
Zeros	-6.30 5.91 0.01
Damping ratios	0.06 0.56
Undamped natural frequencies [rad/s]	0.23 1.56
K, gain [rad/ft]	4.6
Velocity error coefficient [ft/rad.s]	8.7

APPENDIX G

R/C AIRCRAFT PLANT PROPERTIES

The plant properties of RC aircraft at 3 different trim points (20m/s, 23.5m/s and 27m/s) can be monitored in below.

Table G.1 Plant properties of RC aircraft at different velocity_20m/s

	RC-at 20 m/s
Attitude to elevator deflection transfer function	$\frac{4023\ s^3 - 44752\ s^2 - 6085737\ s - 1551691}{68\ s^5 + 1344\ s^4 + 7449\ s^3 + 2635\ s^2 + 3087\ s}$
Poles	0 -9.74 ± 2.92i -0.15 ± 0.65i
Zeros	44.96 33.58 -0.26
Damping ratios	0.95 0.22
Undamped natural frequencies [rad/s]	10.20 0.66
K, gain [rad/m]	59
Velocity error coefficient [m/rad.s]	-503

Table G.2 Plant properties of RC aircraft at different velocity_23.5m/s

	RC-at 23.5 m/s
Attitude to elevator deflection transfer function	$\frac{6438\ s^3 - 81281\ s^2 - 13042971\ s - 4217727}{79\ s^5 + 1829\ s^4 + 11811\ s^3 + 4690\ s^2 + 4906\ s}$
Poles	0 -11.35 ± 3.41i -0.17 ± 0.64i
Zeros	51.90 38.95 -0.32
Damping ratios	0.95 0.26
Undamped natural frequencies [rad/s]	11.90 0.66
K, gain [rad/m]	81
Velocity error coefficient [m/rad.s]	-860

Table G.3 Plant properties of RC aircraft at different velocity_27m/s

	RC-at 27 m/s
Attitude to elevator deflection transfer function	$\frac{9614\ s^3 - 136564\ s^2 - 25230425\ s - 9744643}{90\ s^5 + 2389\ s^4 + 17621\ s^3 + 7749\ s^2 + 7326\ s}$
Poles	0 -12.97 ± 3.89i -0.20 ± 0.63i
Zeros	58.98 -44.39 -0.39
Damping ratios	0.95 0.30
Undamped natural frequencies [rad/s]	13.50 0.66
K, gain [rad/m]	106
Velocity error coefficient [m/rad.s]	-1,329

APPENDIX H

NORMALIZED ALTITUDE TO ELEVATOR DEFLECTION TRANSFER FUNCTIONS FOR UCAV6

The normalized altitude to elevator deflection transfer functions of UCAV6 are given below.

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_1} = \frac{0.20002\,(s+6.813)\,(s-5.261)\,(s+0.01457)}{s\,(s^2+0.02474s+0.01032)\,(s^2+1.769s+2.908)} \tag{H.1}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_2} = \frac{0.49946\,(s+11.23)\,(s-9.74)\,(s+0.006855)}{s\,(s^2+0.03303s+0.01798)\,(s^2+2.267s+3.67)} \tag{H.2}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_3} = \frac{1.8089\,(s+13.9)\,(s-13.09)\,(s+0.007268)}{s\,(s^2+0.03461s+0.01836)\,(s^2+2.635s+4.647)} \tag{H.3}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_4} = \frac{3.0103\,(s+18.24)\,(s-17.04)\,(s+0.01093)}{s\,(s^2+0.03534s+0.01898)\,(s^2+3.228s+6.843)} \tag{H.4}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_5} = \frac{5.4296\,(s+20.98)\,(s-19.92)\,(s+0.009513)}{s\,(s^2+0.03778s+0.01994)\,(s^2+3.614s+8.575)} \tag{H.5}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_6} = \frac{10.0705\,(s+21.31)\,(s-20.76)\,(s+0.01015)}{s\,(s^2+0.04545s+0.0265)\,(s^2+3.791s+9.341)} \tag{H.6}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_7} = \frac{14.3015\,(s+24.89)\,(s-21.03)\,(s+0.01257)}{s\,(s^2+0.054s+0.03649)\,(s^2+3.866s+9.555)} \tag{H.7}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_8} = \frac{15.9312\,(s+27.79)\,(s-26.32)\,(s+0.01336)}{s\,(s^2+0.06674s+0.04953)\,(s^2+4.135s+9.884)} \tag{H.8}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_9} = \frac{17.7753\,(s+31.07)\,(s-30.21)\,(s+0.01421)}{s\,(s^2+0.07313s+0.05197)\,(s^2+4.887s+10.87)} \tag{H.9}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_10} = \frac{49.7469\,(s+33.25)\,(s-29.63)\,(s+0.01477)}{s\,(s^2+0.08394s+0.05617)\,(s^2+5.918s+15.14)} \tag{H.10}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_11} = \frac{107.4537\,(s+33.61)\,(s-31.95)\,(s+0.02241)}{s\,(s^2+0.08968s+0.06182)\,(s^2+7.145s+20.5)} \tag{H.11}$$

$$\left.\frac{h(s)}{\delta e(s)}\right)_{OP\_12} = \frac{199.7888\,(s+34.06)\,(s-32.93)\,(s+0.02982)}{s\,(s^2+0.1229s+0.09153)\,(s^2+7.952s+25.03)} \tag{H.12}$$

APPENDIX I

DIRECT FORM II

The canonical direct form II is presented as the "Control Canonical Form" in Digital Control of Dynamic Systems by Franklin, Powell, and Workman [81]. In general, a direct form realization refers to a structure where the coefficients of the transfer function appear directly as gain blocks. The direct form II realization method is presented as using the minimal number of delay elements, which is equal to n, the order of the transfer function denominator. The canonical direct form II realization can be derived by writing the discrete-time transfer function between the output u (z) and input e (z) as

$$\frac{u(z)}{e(z)} = \frac{u(z)}{h(z)} \cdot \frac{h(z)}{e(z)} \tag{I.1}$$

$$\frac{u(z)}{e(z)} = (b_0 + b_1 z^{-1} + ... + b_m z^{-m}) \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_n z^{-n}} \tag{I.2}$$

$$\frac{u(z)}{h(z)} = (b_0 + b_1 z^{-1} + ... + b_m z^{-m}) \tag{I.3}$$

The block diagram is given in Figure I.1.

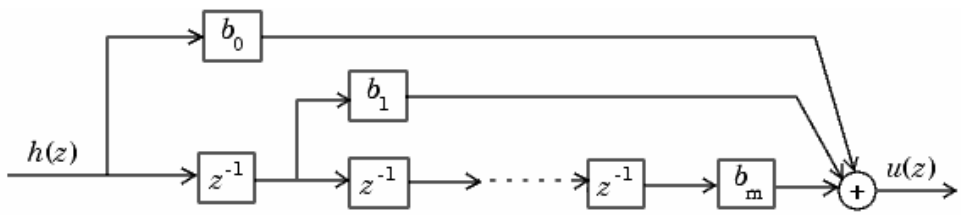

Figure I.1 The block diagram for $\frac{u(z)}{h(z)}$

On the other hand, the block diagram for

$$\frac{h(z)}{e(z)} = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + ... + a_n z^{-n}} \tag{I.4}$$

is given in Figure I.2.

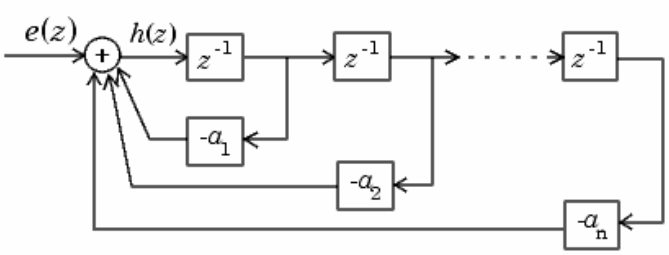

Figure I.2 The block diagram for $\frac{h(z)}{e(z)}$

Combining these two block diagrams yields the direct form II diagram shown in Figure I.3. Notice that the feed forward part (top of block diagram) contains the numerator coefficients and the feedback part (bottom of block diagram) contains the denominator coefficients.

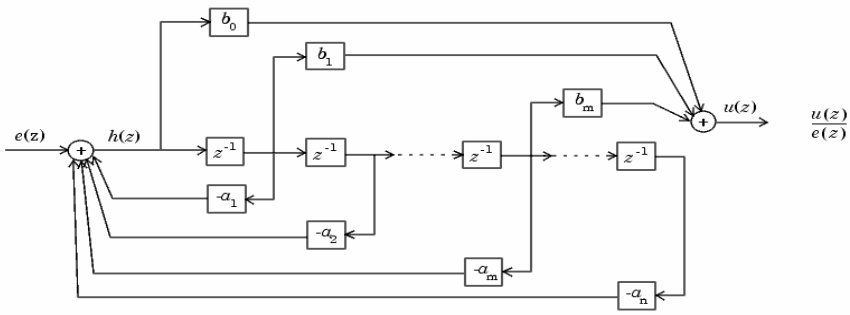

Figure I.3 The block diagram for $\frac{u(z)}{e(z)}$

APPENDIX J

TEST SCENARIOS- DATA STATISTICS

In this Appendix, the methods to obtain data statistics of a signal in Simulink and their logging are described. Table J.1 shows the data statistics the selected flight plans at 100 ft and at 5,000 ft. Moreover, the following tables, which are Table J.2, Table J.3, Table J.4, show the data statistics results of the plant output in both setup and Simulink environments.

Table J.1 Data statistics of flight plans

Properties	**Plant Output at 100 ft**	**Plant Output at 5,000 ft velocity increasing**	**Plant Output at 5,000 ft velocity decreasing**
minimum	-0.6	4,950	-5,229
maximum	111.5	5,017	5,440
mean	99.0	5,000	4,995
median	98.8	5,000	4,998
standard deviation	4.4	4,359	185.6

Table J.2 Data statistics of plant output at 100 ft

	Plant Output at 100 ft		
Properties	**Platform Test 1**	**Platform Test 2**	**Simulink Test**
minimum	-0.6	-0.6	-0.6
maximum	113.6	113.6	111.5
mean	99	99	99
median	98.91	98.91	98.89
standard deviation	4.5	4.5	4.4

Table J.3 Data statistics of plant output at 5,000 ft with increasing airspeed

	Plant Output at 5,000 ft		
Properties	**Platform Test 1**	**Platform Test 2**	**Simulink Test**
minimum	4,950	4,950	4,950
maximum	5,017	5,017	5,017
mean	5,000	5,000	5,000
median:	5,000	5,000	5,000
standard deviation	4,319	4,345	4,359

Table J.4 Data statistics of plant output at 5,000 ft with decreasing airspeed

	Plant Output at 5,000 ft		
Properties	**Platform Test 1**	**Platform Test 2**	**Simulink Test**
minimum	-5.2	-5.2	-5.2
maximum	5,643	5,643	5,440
mean	4,995	4,998	4,995
median	4,998	4,999	4,998
standard deviation	217	193	185

In the *Tools* menu of the Figure window, which includes the plot of the UCAV6 altitude value, *Data Statistics* should be chosen as shown in Figure J.1 to obtain data statistics of the plant output. Data statistics window shows signal statistics both in x and y-axes, as shown in Figure J.2.

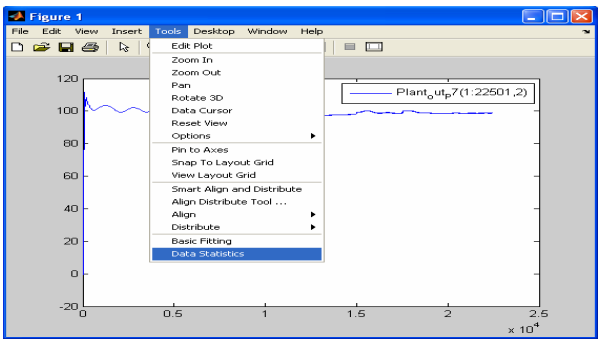

Figure J.1 Obtaining data statistics of the plot

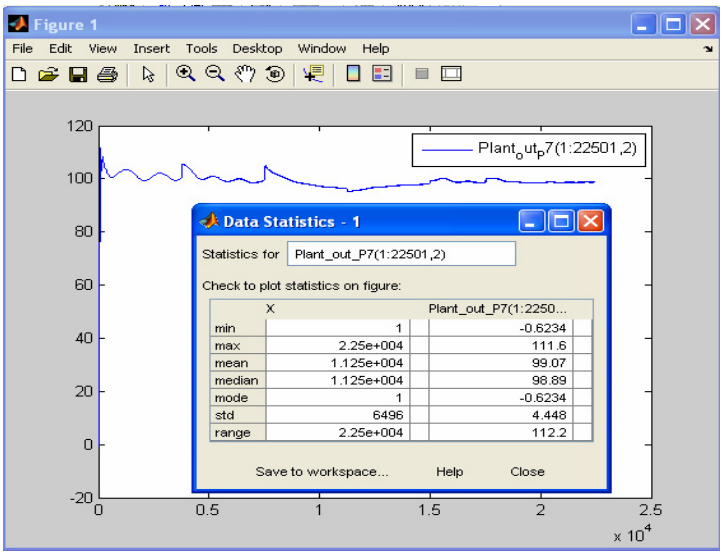

Figure J.2 Data statistics window

The results can be saved in workspace by pressing save to workspace button on the data statistic window. Figure J.3 shows how to save the results on workspace as a mat file.

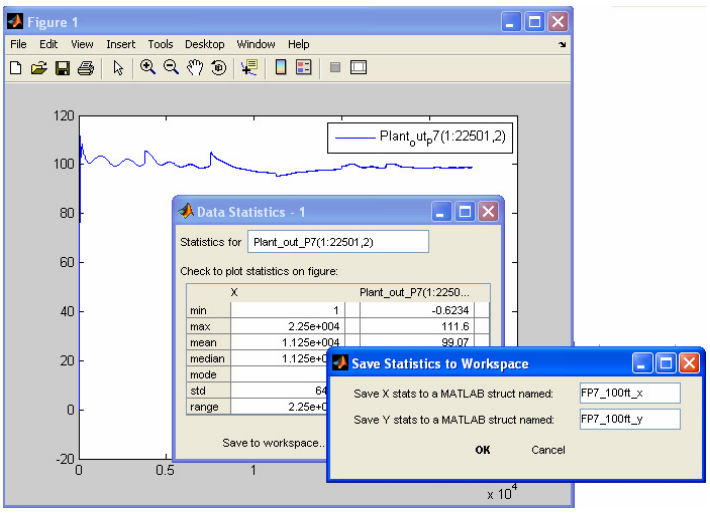

Figure J.3 Saving data statistics results in workspace

APPENDIX K

BOEING 747 PLATFORM TEST RESULTS

The tests results for altitude, h, using the Boeing 747 close loop model in the setup and in Simulink are given in Chapter 6. Here in this Appendix, complementary data these tests are presented in terms of airspeed, angle of attack (α), sideslip angle (β), roll rate about x-axis of body frame (p), pitch rate about y-axis of body frame (q), yaw rate about z-axis of body frame (r), euler heading angle (ψ), euler pitch angle (θ), euler bank angle (φ), distance along x axes of earth fixed coordinate system (X), distance along y axes of earth fixed coordinate system (Y).

As monitored from these figures, the differences of the condition values (such as angle of attack, sideslip angle, etc.) are limited to a small range, hence they can be ignored with respect to the trim values of them. Therefore, it can be said that the behaviors of the plant both in platform and Simulink tests are similar to each other. Table K.1 gives the condition variables and their trim values of the Boeing 747 discrete model.

Table K.1 Condition of Boeing 747 discrete model

Airspeed [m/s]	α [rad]	β [rad]	p [rad/s]	q [rad/s]	r [rad/s]
150	0.046	0	0	0	0
ψ [rad]	θ [rad]	φ [rad]	X [m]	Y [m]	Altitude [m]
0	0.046	0	0	0	61

Airspeed Test Results:

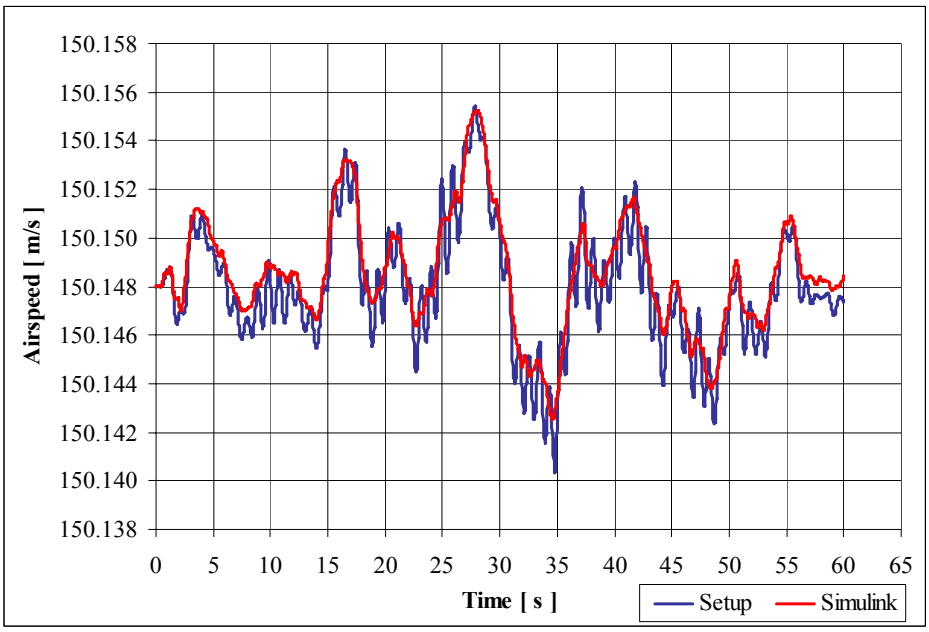

Figure K.1 Airspeed during platform and Simulink tests

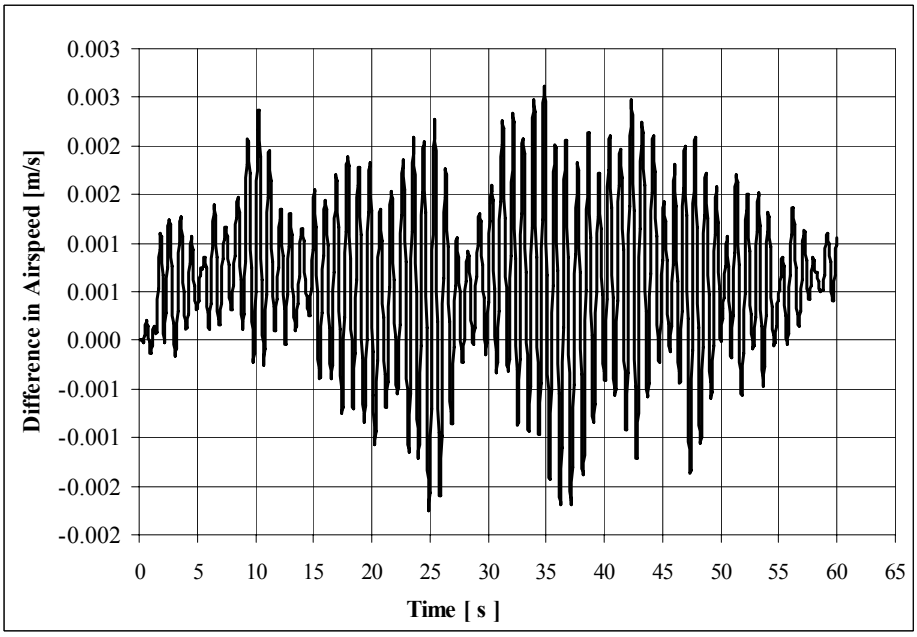

Figure K.2 Difference in airspeed between platform and Simulink tests

Test Results for Angle of Attack, α:

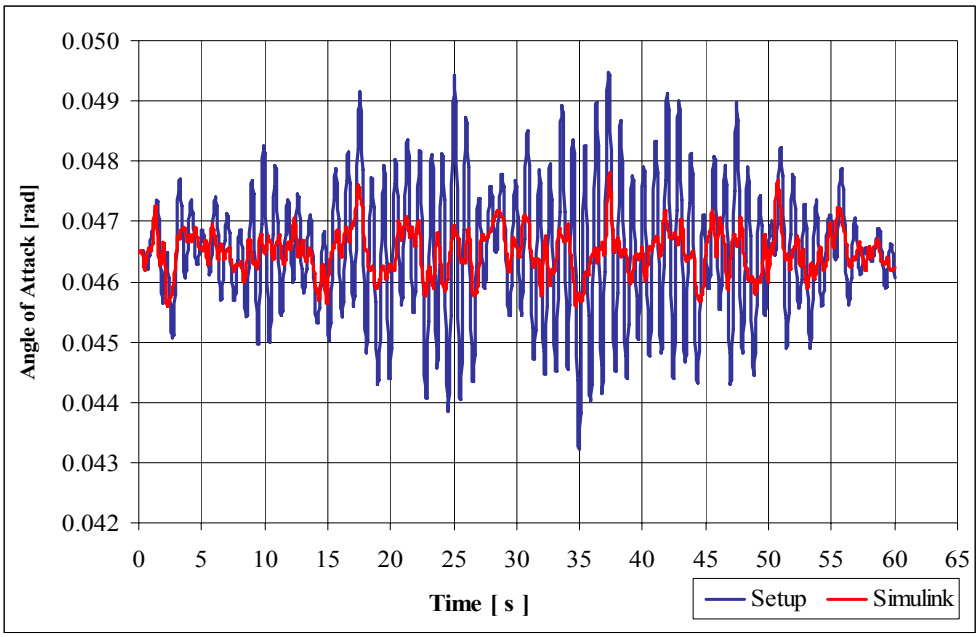

Figure K.3 Angle of attack, α, during platform and Simulink tests

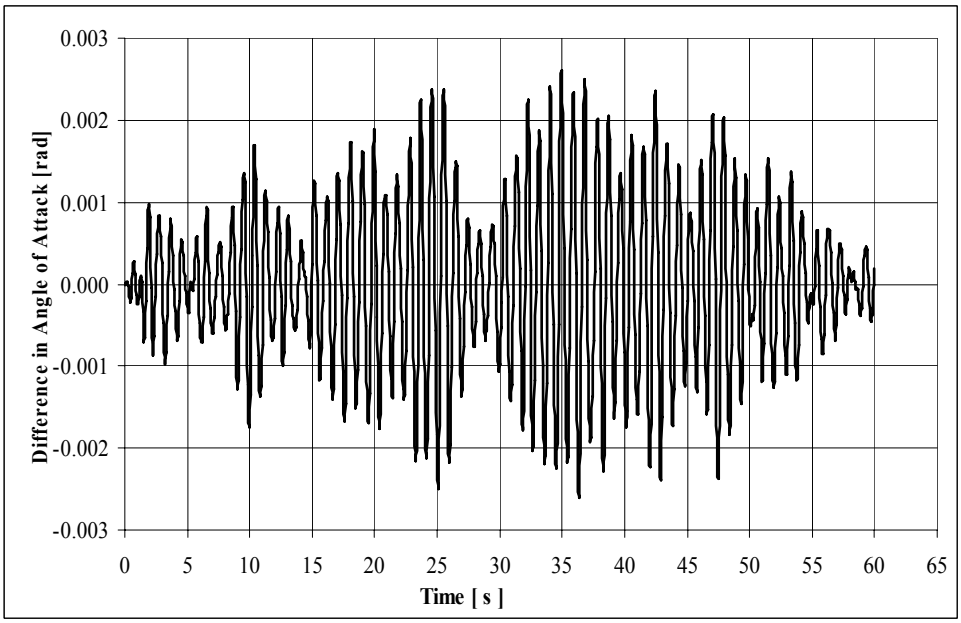

Figure K.4 Difference in angle of attack, α, between platform and Simulink tests

Test Results for Sideslip Angle, β:

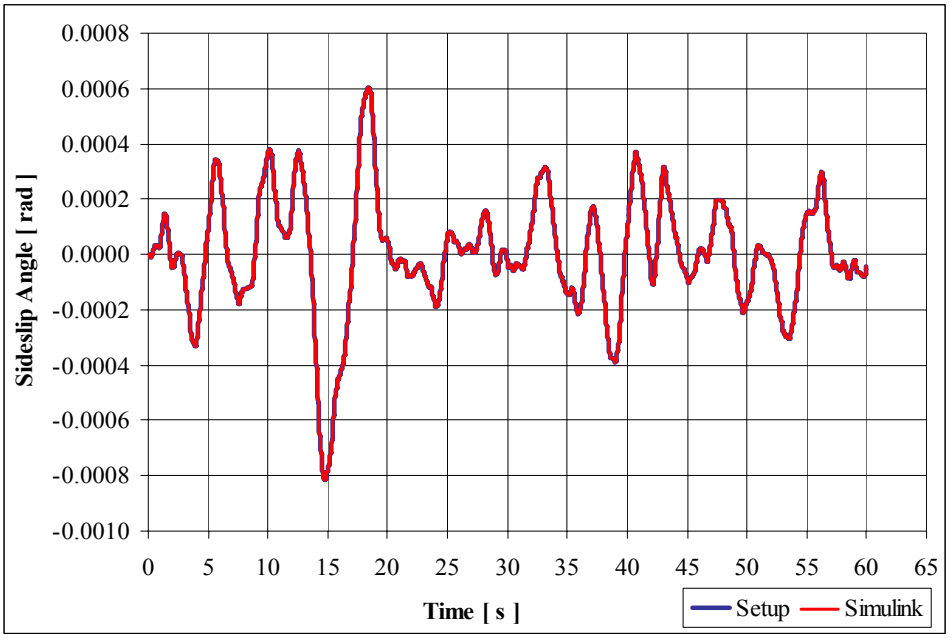

Figure K.5 Sideslip angle, β, during platform and Simulink tests

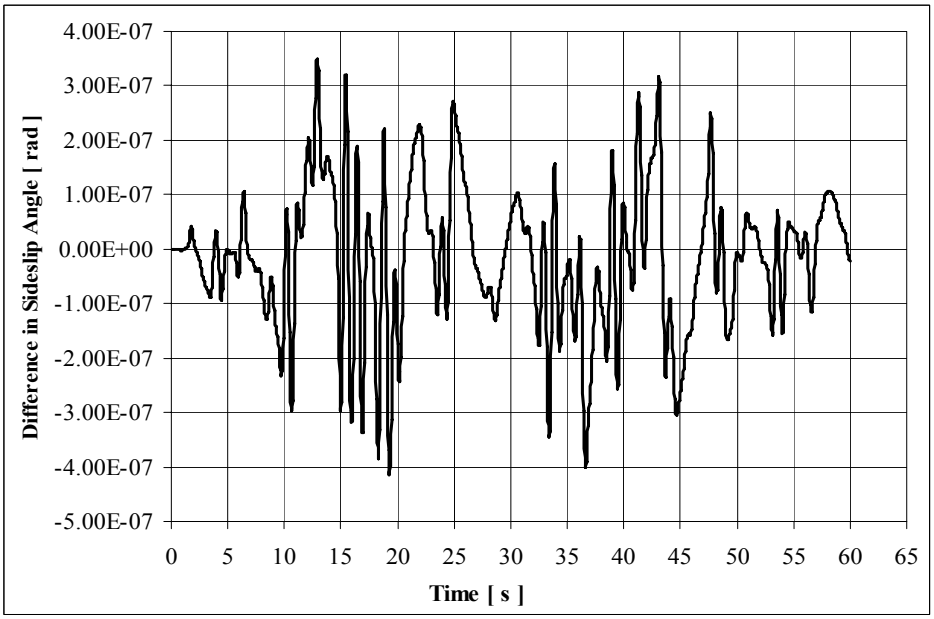

Figure K.6 Difference in sideslip angle, β, between platform and Simulink tests

Test Results for Roll Rate about x-axis of body frame, p:

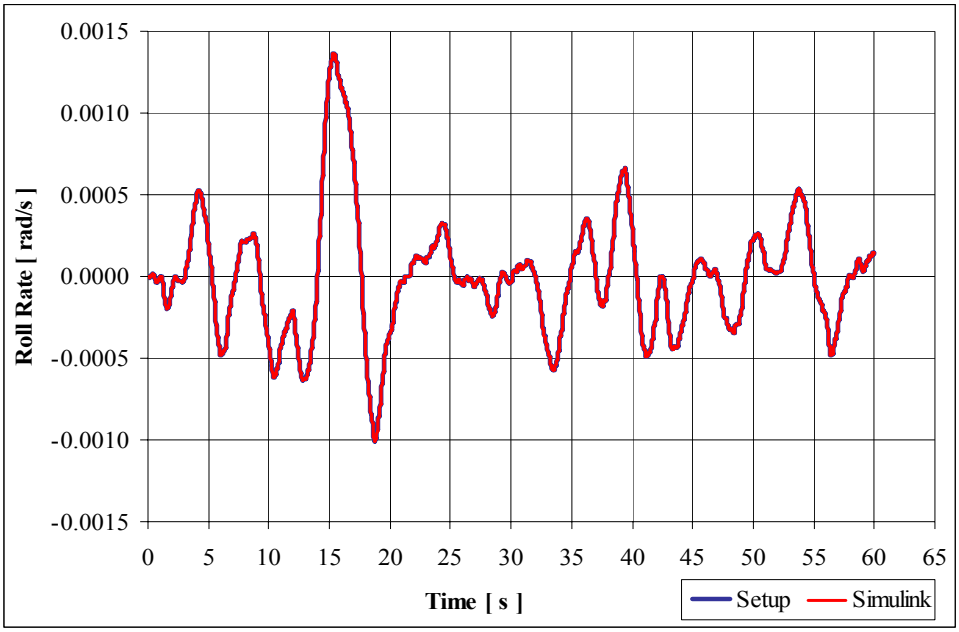

Figure K.7 Roll rate, p, during platform and Simulink tests

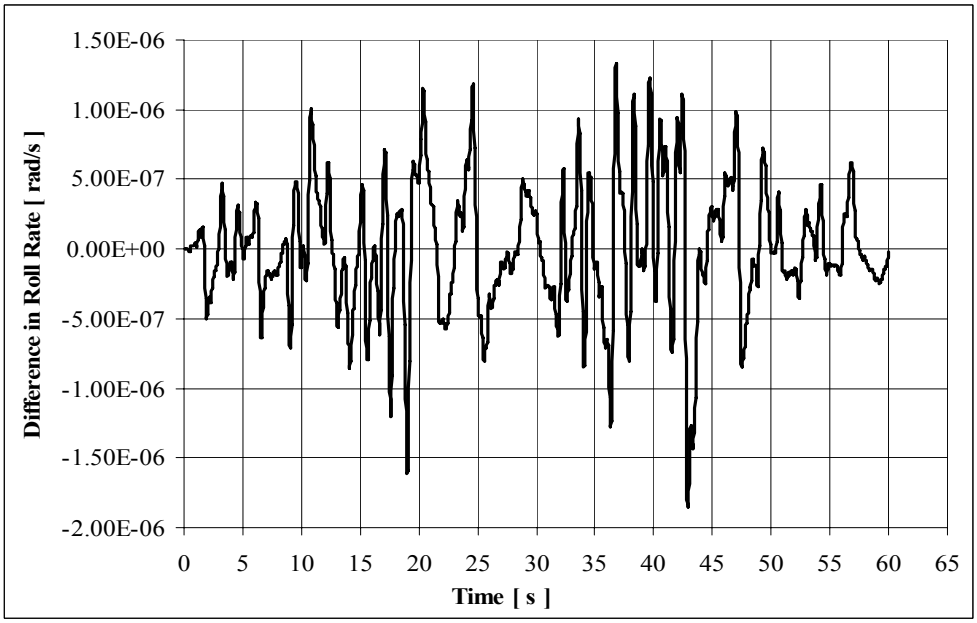

Figure K.8 Difference in roll rate, p, between platform and Simulink tests

Test Results for Pitch Rate about y-axis of body frame, q:

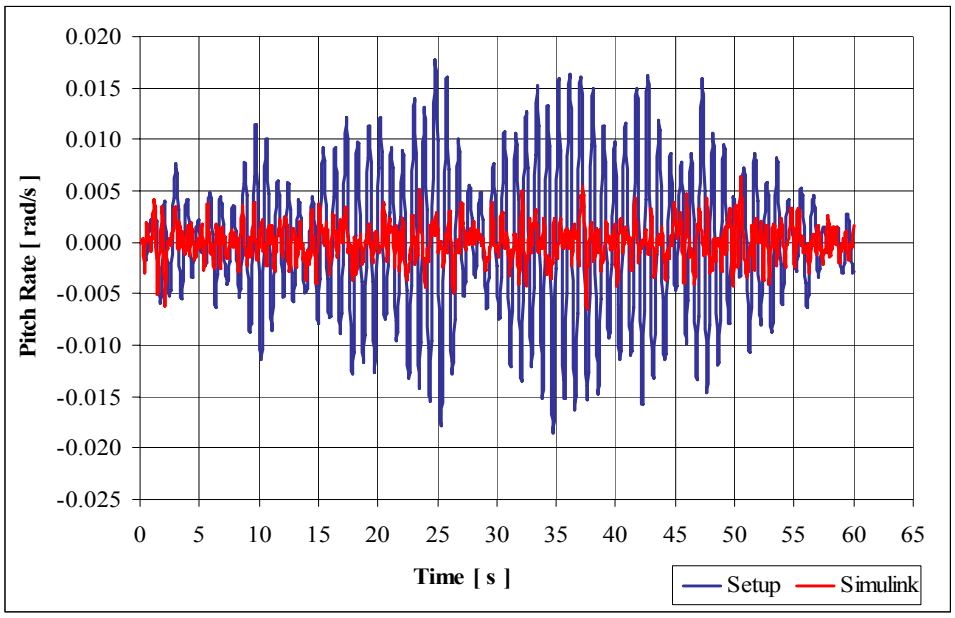

Figure K.9 Pitch rate, q, during platform and Simulink

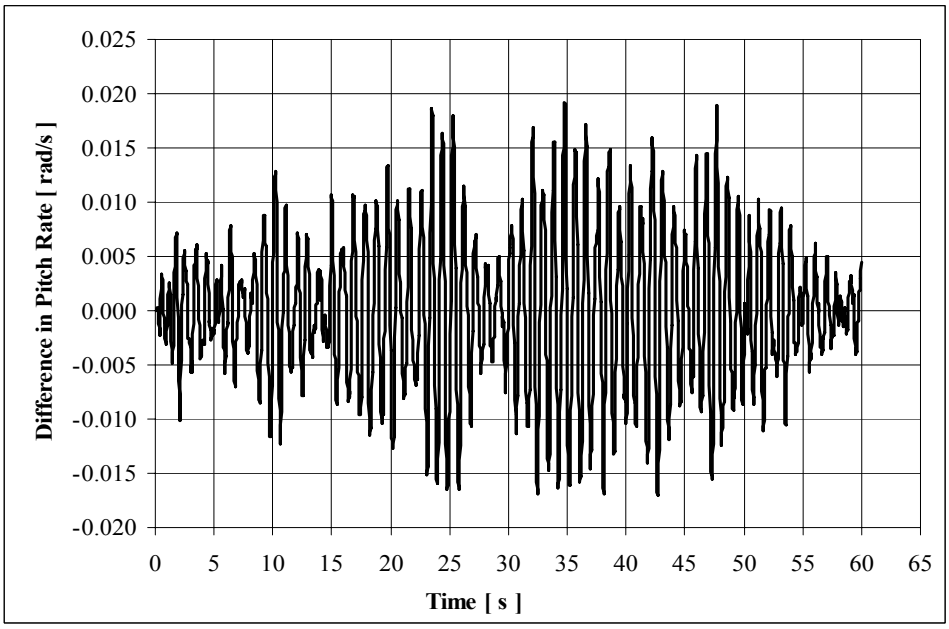

Figure K.10 Difference in pitch rate, p, between platform and Simulink tests

Test Results for Yaw Rate about z-axis of body frame, r:

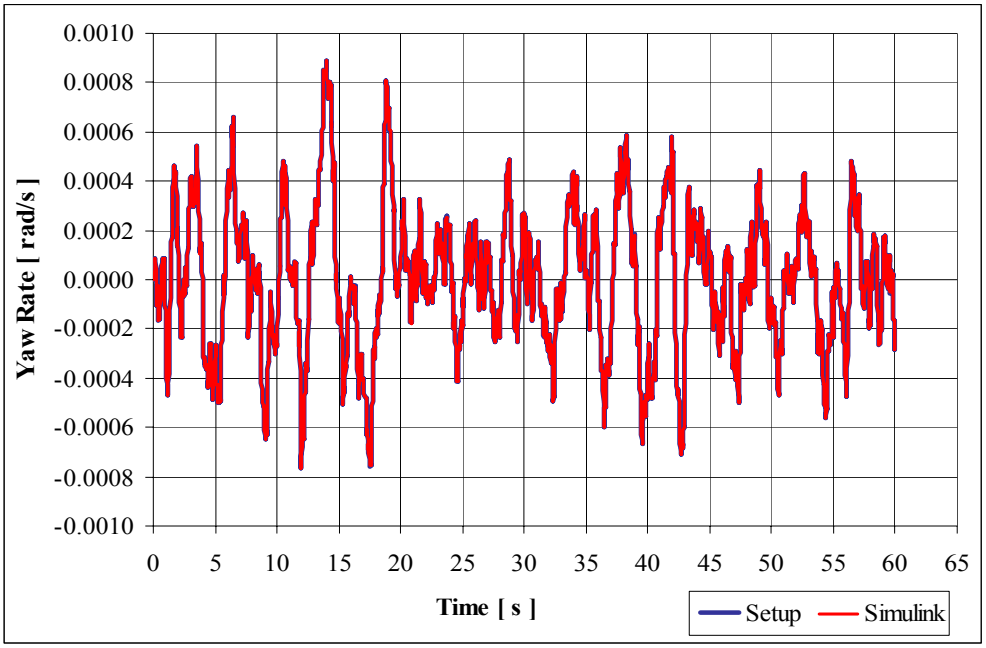

Figure K.11 Yaw rate, r, during platform and Simulink

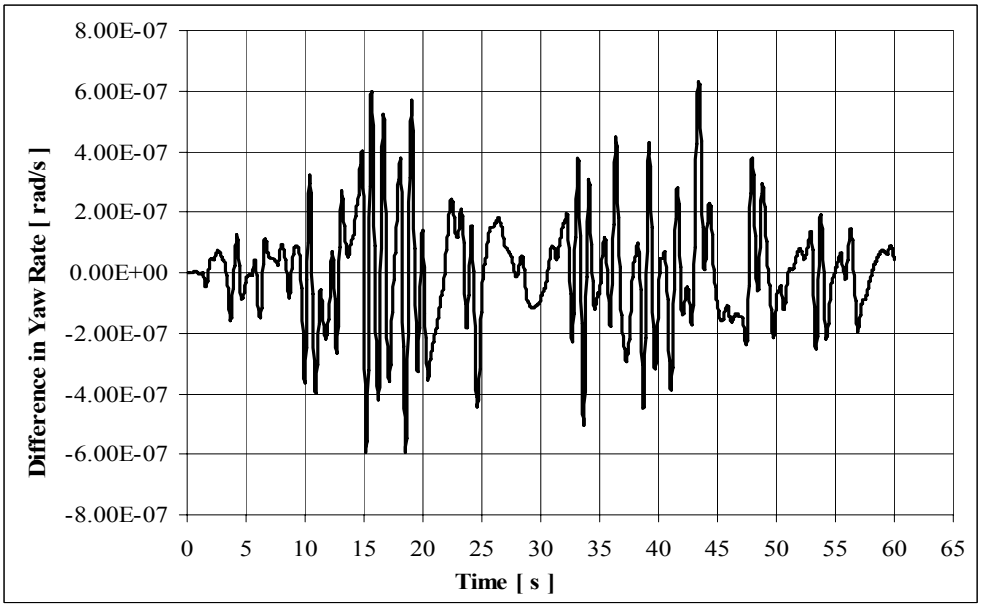

Figure K.12 Difference in yaw rate, r, between platform and Simulink tests

Test Results for Euler Heading Angle, ψ:

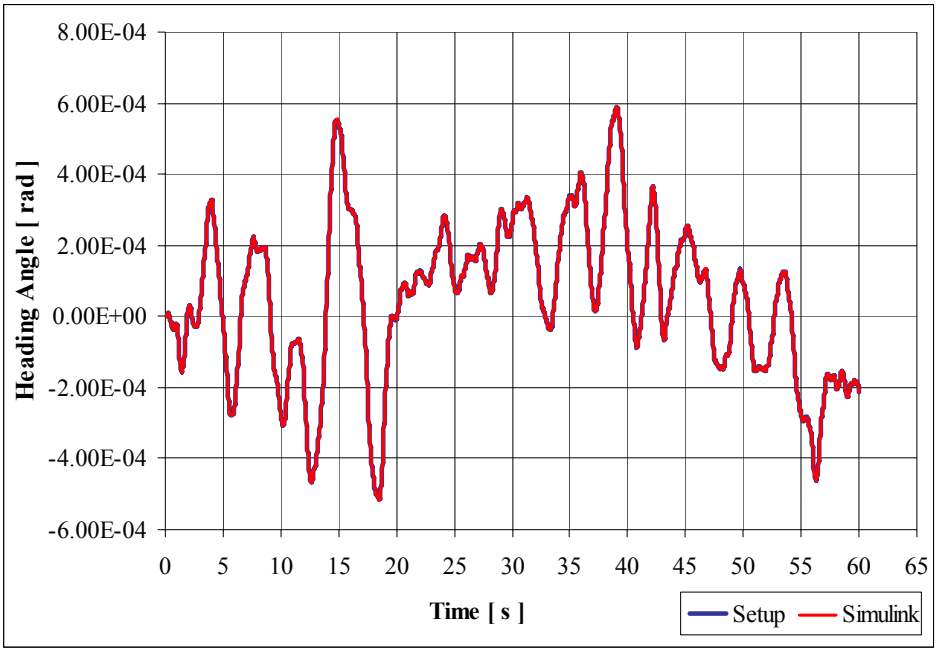

Figure K.13 Heading angle, ψ, during platform and Simulink tests

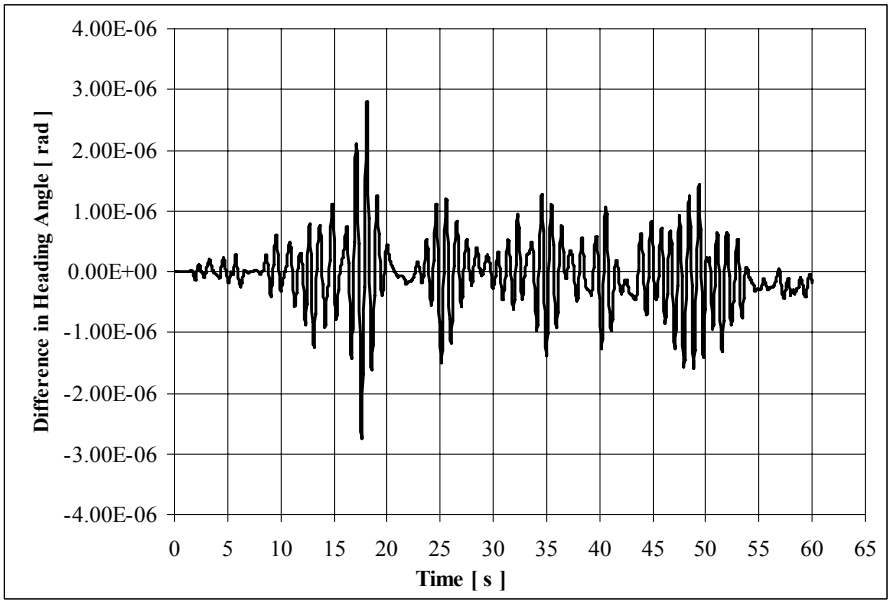

Figure K.14 Difference in heading angle, ψ, between platform and Simulink tests

Test Results for Euler Pitch Angle, θ:

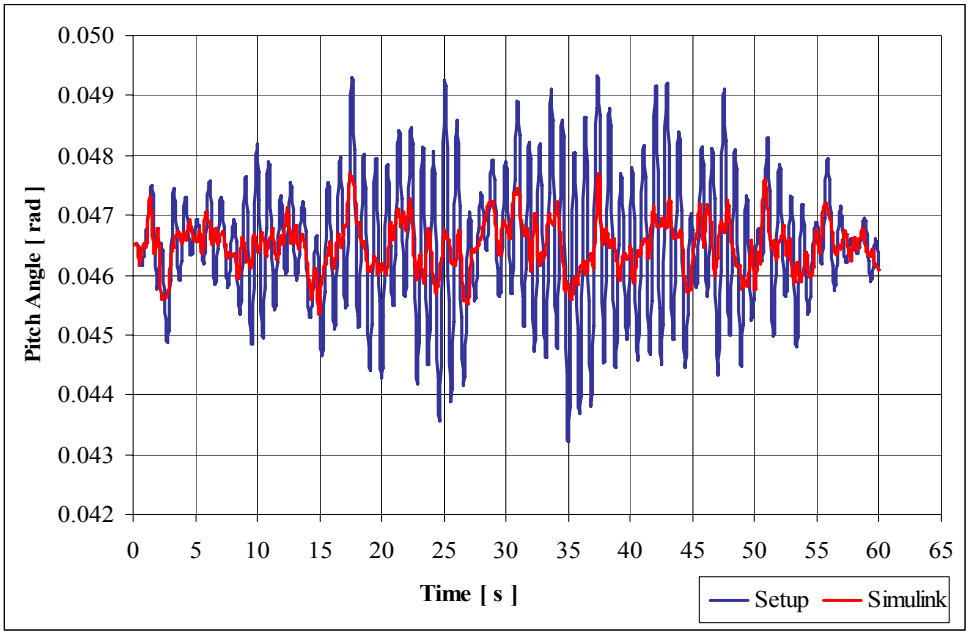

Figure K.15 Pitch angle, θ, during platform and Simulink tests

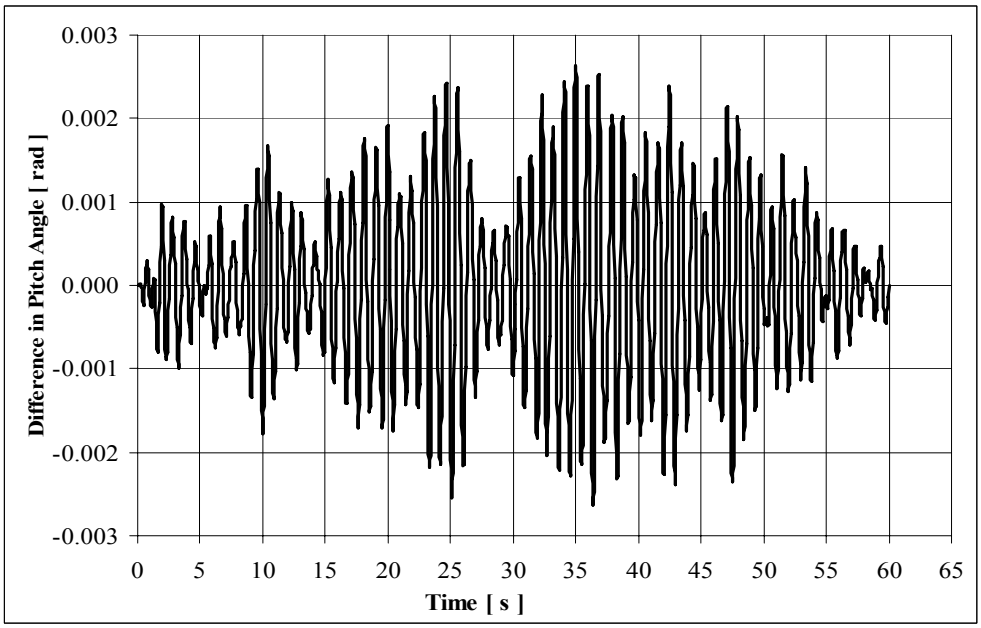

Figure K.16 Difference in pitch angle, θ, between platform and Simulink tests

Test Results for Euler Bank Angle, φ:

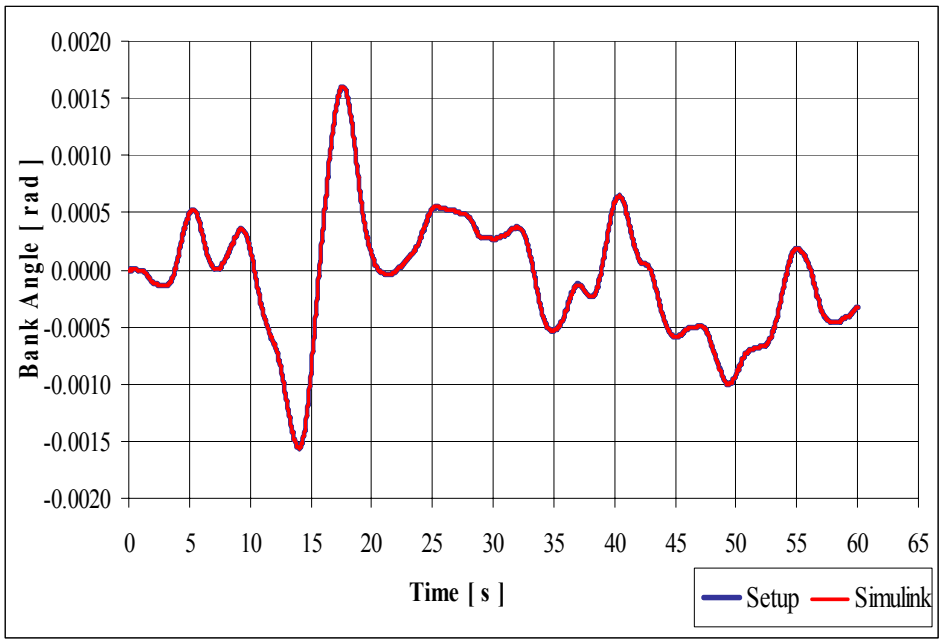

Figure K.17 Bank angle, φ, during platform and Simulink tests

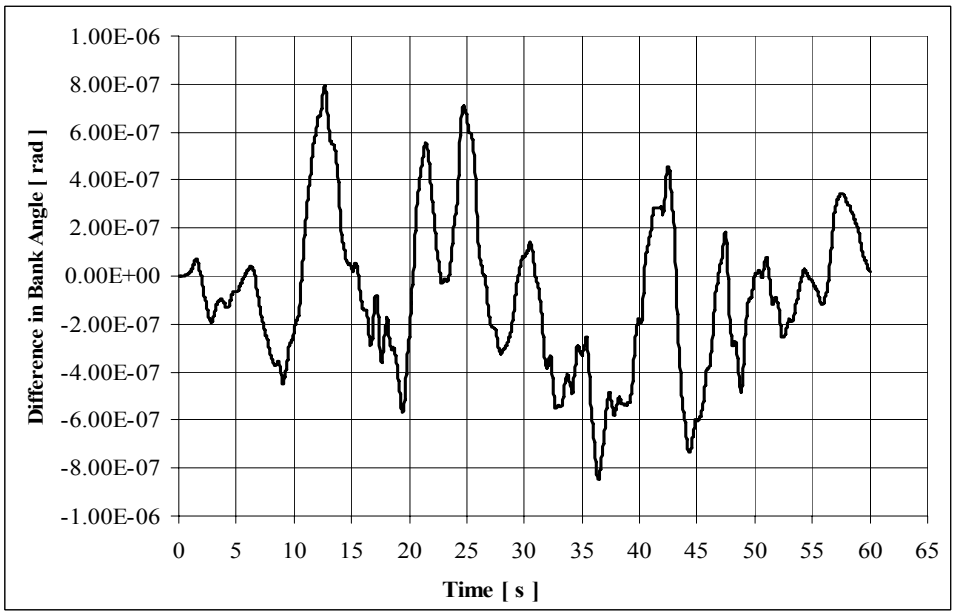

Figure K.18 Difference in bank angle, φ, between platform and Simulink tests

Test Results for distance along x axes of earth fixed coordinate system, X:

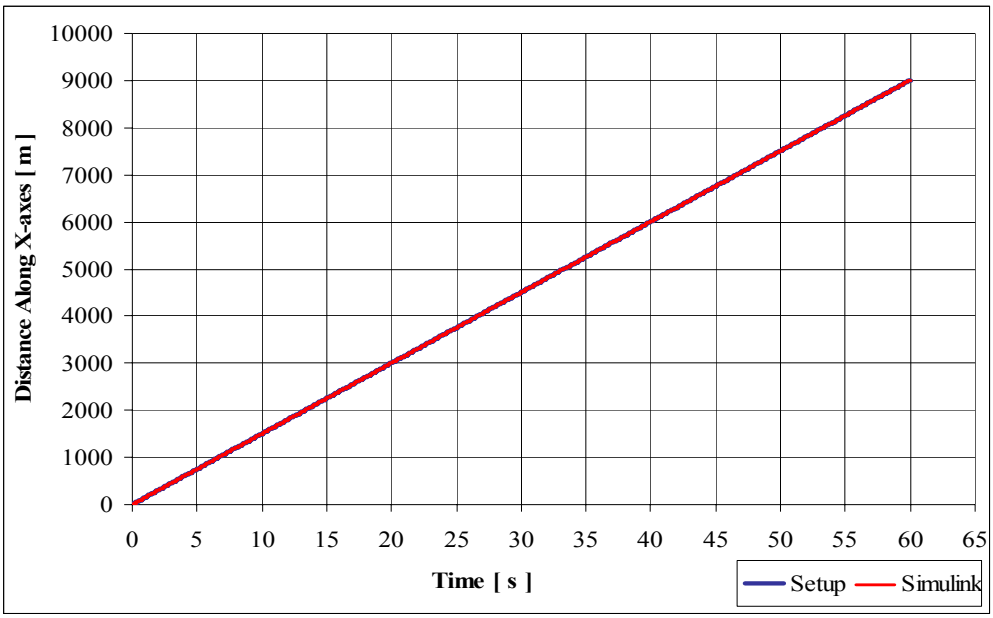

Figure K.19 Distance along x axes, X, during platform and Simulink

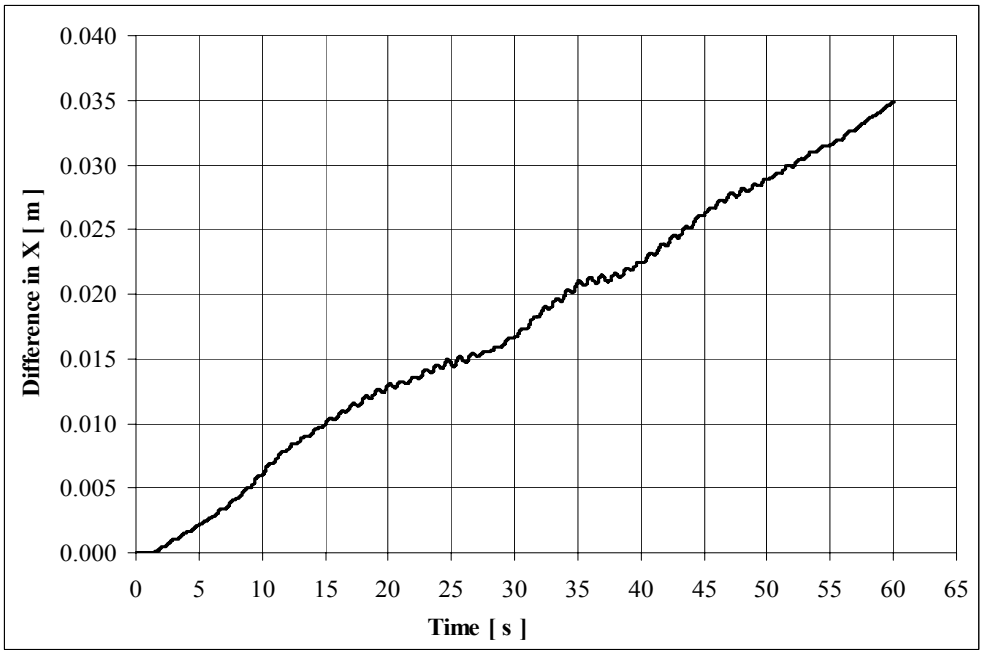

Figure K.20 Difference in distance, X, between platform and Simulink tests

Test Results for distance along y axes of earth fixed coordinate system, Y:

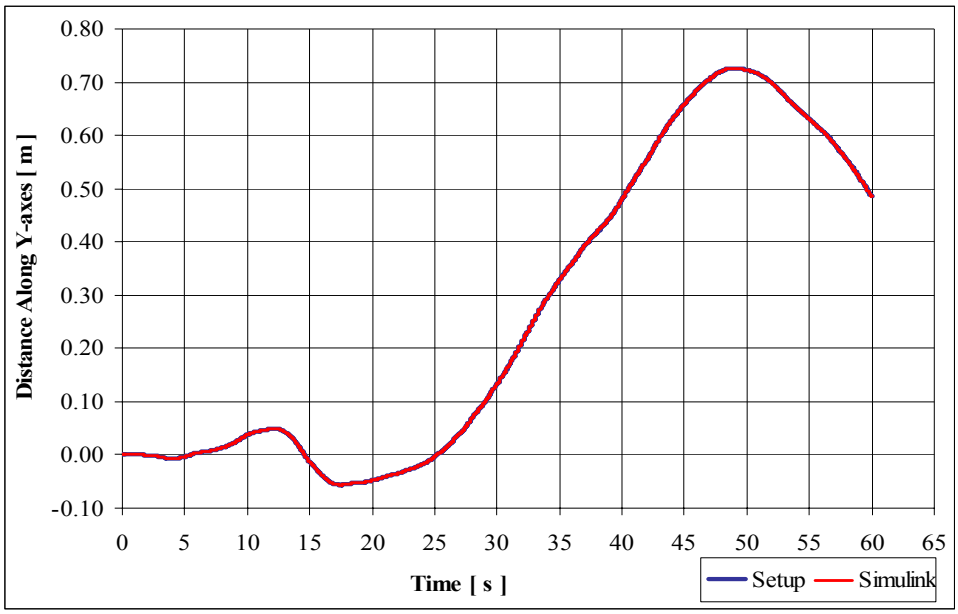

Figure K.21 Distance along y axes, Y, during platform and Simulink

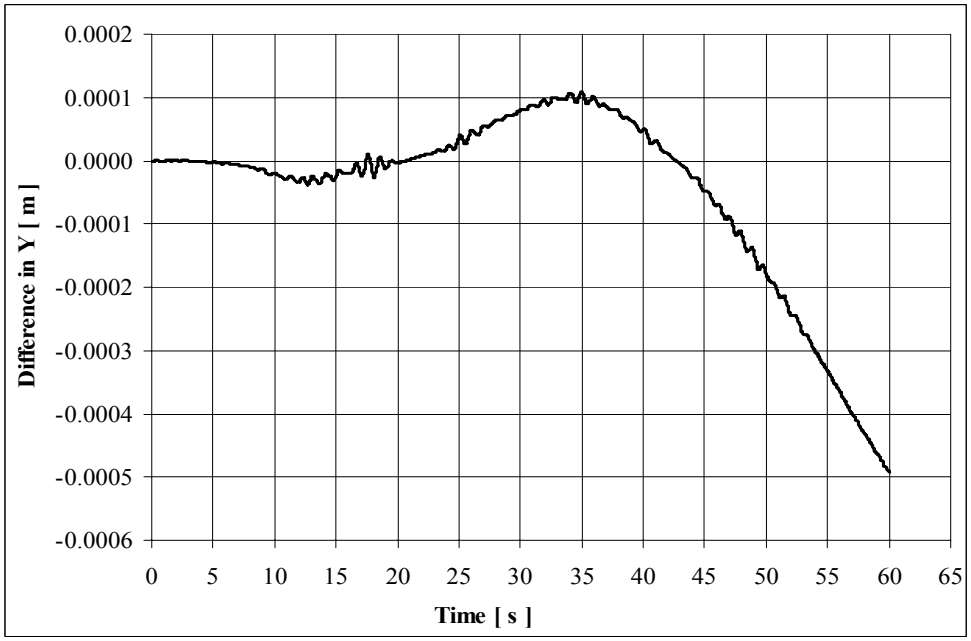

Figure K.22 Difference in distance, Y, between platform and Simulink tests