11-2013

Performance Analysis Of Non-Linear Adaptive Control Laws Using Hardware in the Loop of an Unmanned Aerial System

Brendon Lyons
Embry-Riddle Aeronautical University - Daytona Beach

PERFORMANCE ANALYSIS OF NON-LINEAR ADAPTIVE CONTROL LAWS USING HARDWARE IN THE LOOP OF AN UNMANNED AERIAL SYSTEM

by

Brendon Lyons

A Thesis Submitted to the
Graduate Studies Office
In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Mechanical Engineering

Embry-Riddle Aeronautical University
Daytona Beach, Florida
November 2013

PERFORMANCE ANALYSIS OF NON-LINEAR ADAPTIVE CONTROL LAWS USING HARDWARE IN THE LOOP OF AN UNMANNED AERIAL SYSTEM

by

Brendon Lyons

This thesis was prepared under the direction of the candidate's thesis committee chair, Dr. Hever Y. Moncayo, Department of Aerospace Engineering, and has been approved by the members of his thesis committee. It was submitted to the Department of Mechanical Engineering and was accepted in partial fulfillment of the requirements or the degree of Master of Science in Mechanical Engineering.

Thesis Review Committee:

Hever Y. Moncayo, Ph.D.
Committee Chair

Richard Prazenica, Ph.D.
Committee Member

Yan Tang, Ph.D.
Committee Member

Darris White, Ph.D.
Graduate Program Coordinator,
Mechanical Engineering

Charles Reinholtz, Ph.D.
Department Chair,
Mechanical Engineering

Robert Oxley, Ph.D.
Associate Vice President of Academics

12-4-2013
Date

Acknowlegements

I would like to give thanks to Dr. Hever Y. Moncayo for giving me the opportunity to pursue this exciting line of research. I would also like acknowledge Alfonso Noriega and Israel Moguel for their much appreciated assistance. For without them I would not be able to complete this project. I would also like to thank the thesis review committee Dr. Richard Prazenica and Dr. Yan Tang for their efforts reviewing and assisting me with the construction of my thesis document. Finally, this project would not be possible without the support of an Embry-Riddle Aeronautical University Faculty Internal Grant.

Abstract

Author:	Brendon Lyons
Title	Performance Analysis Of Non-Linear Adaptive Control Laws Using Hardware in the Loop of an Unmanned Aerial System
Institution:	Embry-Riddle Aeronautical University
Degree:	Master of Science in Mechanical Engineering
Year	2011-2013

In practical applications, an Unmanned Aerial System's (UAS) baseline performance is dictated by how well it can follow a given trajectory with limited stress on the actuators. However, these can be insufficient performance metrics when the UAS is allowed to adapt to an unpredicted external influence such as turbulence or actuation failure, while maintaining a satisfactory baseline performance.

In this thesis, different control laws based on the formation flight geometry problem, nonlinear dynamic inversion and an artificial immune system adaptive mechanism , are implemented in hardware-in-the-loop as a precursor for in-flight testing. These controllers are compared based on three performance metrics: trajectory following, control activity and computer task execution time. The controllers chosen for comparison are: Basic Proportional-Integral-Derivative (PID), Outer loop Non-Linear Dynamic Inversion (NLDI), Extended NLDI, and the previous three controllers augmented with an AIS for a total of six controllers. The Extended NLDI augmented with the AIS out-performed all of the other algorithms under falure conditions on a global scale.

Table of Contents

Abstract iii

List of Tables vii

List of Figures viii

1 Introduction 1

1.1 Literature Review 4

1.1.1 Adaptive Control 4

1.1.2 Non-linear Dynamic Inversion 6

1.1.3 Artificial Immune System 6

1.2 Background 7

2 Theory 10

2.1 Equations of Motion 10

2.2 Flight Control Laws 15

2.2.1 FFC Virtual Trajectory Tracking 15

2.2.1.1 Virtual Tracking with PID Control 19

2.2.1.2 Outer Loop - NLDI Controller Design 21

2.2.1.3 Inner Loop - NLDI Controller 24

2.3 Artificial Immune System Adaptive Control Law 32

2.3.1 AIS Augmentation 36

3 UAS Research Platform 37

3.1 Actuators 38

3.2 Motor and Propellor 38

3.3 Electronic Speed Controller 39

3.4 Batteries 40

3.5 Hardware and Sensors 40

3.5.1 Primary Flight Computer 40

3.5.2 Enclosure 41

3.5.3 DC/DC Converter 42

3.5.4 Serial Expansion 42

3.6 Attitude Heading and Reference System 43
3.7 Global Positioning System 44
3.8 Ardupilot Mega Sub System 45
3.8.1 Barometric Pressure Sensor 46
3.8.2 Pitot Tube 46
3.9 Wireless Telemetry 47
3.10 Remote Control Transmitter and Receiver 48
3.11 Servo Controller 48
3.12 Servo Multiplexer 49
3.13 RS-232 to Transister-Transister Logic Converter 49
3.14 Hardware Setup 50
4 Flight Simulation Environment 51
4.1 Aircraft Model 51
4.2 Actuator Model 54
4.3 Sensor Model 56
4.4 Aircraft Failure Model 59
5 Hardware-in-the-Loop Simulation 61
5.1 Real-Time Environment 62
5.1.1 Host Computer 63
5.1.2 Aircraft Target Computer 64
5.1.3 Primary Flight Computer 64
5.2 Real-Time Models 64
5.3 Data Communication 66
5.4 Performance Metrics 68
5.5 Control Laws Performance Analysis 73
5.5.1 Performance Data 75
6 Conclusions and Suggestions 84
6.1 Conclusion 84
6.2 Future Work and Suggestions 85
7 Bibliography 86

Appendix 89
Landing Trajectory Metrics 89

List of Tables

Table 1: Sig Rascal 110 geometric data. 37

Table 2: Digital Datcom input file parameters. 51

Table 3: Preliminary stability derivatives from Digital Datcom. 52

Table 4: Trim conditions and parameters. 54

Table 5: Failure conditions used to test control laws. 60

Table 6: Performance index weights and normalization cut-offs for trajectory tracking. 74

Table 7: Performance index weights and normalization cut-offs for control activity. 74

Table 8: Performance index weights and normalization cut-offs for task execution time. 74

Table 9: Figure 8 trajectory tracking metrics. 75

Table 10: Figure 8 control activity metrics. 76

Table 11: Figure 8 task execution time metrics. 77

Table 12: Figure 8 performance index (PI) data. 78

Table 13: Landing trajectory tracking metrics. 89

Table 14: Landing control activity metrics. 90

Table 15: Figure 8 task execution time metrics. 91

Table 16: Figure 8 performance index (PI) data. 92

List of Figures

Figure 1: NASA X-15 hypersonic research vehicle (NASA, 2000). 5

Figure 2: YF-22 model at West Virginia University. 8

Figure 3: North East Down and Aircraft Body Centered reference frames. 10

Figure 4: Formation Flight Geometry (Moncayo H., et al, 2012). 16

Figure 5: Block diagram of the virtual tracking with PID control system. 19

Figure 6: Two-time-scale dynamic inversion system (Moncayo, Perhinischi, et. al, 2012). 25

Figure 7: Immune feedback controller (Moncayo H., et al, 2012). 35

Figure 8: Basic PID/outer loop NLDI AIS augmentation diagram. 36

Figure 9: Extended NLDI augmented with AIS mechanism. 36

Figure 10: Sig Rascal 110 with the primary flight computer installed. 37

Figure 11: HS- 5625MG digital servos by Hitec. 38

Figure 12: Hacker A-60 16M brushless DC motor. 39

Figure 13: 20x13 propellor by Advanced Precision Composites. 39

Figure 14: Jeti Spin Pro 99 Opto Brushless ESC. 39

Figure 15: Sky Lipo 3300mAh 5 cell Lithium Polymer battery. 40

Figure 16: Athena II SBC from Diamond Systems®. 41

Figure 17: Diamond Systems® Athena II Pandora enclosure. 41

Figure 18: Diamond Systems® Jupiter-MM-SIO DC/DC converter. 42

Figure 19: Diamond Systems® Emerald-MM 4-channel serial port expansion module. 43

Figure 20: 3DM-GX3-25 AHRS by MictroStrain®. 44

Figure 21: LEA-6H GPS by u-blox. 44

Figure 22: APM 2.5 by DIY Drones. 45

Figure 23: BMP085 digital pressure sensor by Bosch Sensortec. 46

Figure 24: Airspeed Kit with the MPXV7002DP differential pressure sensor by Freescale Semiconductor. 47

Figure 25: Xtream-PKG-R RS-232/485 RF Modem by MaxStream. 47

Figure 26: DX8 Transmitter and AR8000 reciever by Spektrum. 48

Figure 27: 18 Channel Servo Controller by Polulu. 49

Figure 28: 4 Channel Servo Multiplexer by Polulu. 49

Figure 29: RS232 Shifter by Sparkfun. 50

Figure 30: Hardware connection diagram. 50

Figure 31: Comparison between the Datcom input and the actual aircraft 51

Figure 32: Linear region of the drag curve. 53

Figure 33: MATLAB's system identification tool. 54

Figure 34: Elevator system response to step input of 25°. 55

Figure 35: Motor system thrust response to 5.75lb input. 56

Figure 36: Example of adding white noise to a signal. 57

Figure 37: Sample of the Z-axis accelerometer at 1g, white noise with a variance of 0.0124g. 57

Figure 38: SIMULINK example of adding brown noise (random walk) to the GPS signal. 58

Figure 39: Sample of the GPS X distance at 16m, brown noise with a variance of 0.094m and a time constant of 150s. 59

Figure 40: Failure selection graphical user interface (GUI). 59

Figure 41: Hardware-in-the-loop simulation setup. 61

Figure 42: The steps taken by xPC Target to compile simulation on to hardware. 62

Figure 43: Smulation environment used for hil simulation with compilation scheme and data flow shown. 65

Figure 44: xPC Target and FlightGear displaying Rascal 110. Testing and Performance Analysis of the Control Laws 66

Figure 45: 3-Dimensional Figure-8 trajectory. 67

Figure 46: Landing trajectory. 67

Figure 47: HIL setup with the host, aircraft computer, and primary flight computer respectively. 68

Figure 48: Summary of Figure 8 trajectory PI_{TT} data. 79

Figure 49: Summary of Figure 8 trajectory PI_{CA} data. 80

Figure 50: Summary of Figure 8 trajectory PI_{TET} data. 81

Figure 51: Summary of Figure 8 trajectory global PI data. 81

Figure 52: Summary of Landing trajectory PITT data. 82

Figure 53: Summary of Landing trajectory PI_{CA} data. 82

Figure 54: Summary of Landing trajectory PI_{TET} data. 83

Figure 55: Summary of Landing trajectory global PI data. 83

LIST OF ACRONYMS

AHRS	Attitude Heading Reference System
AIS	Artificial Immune System
APM	ArduPilot Mega
DCM	Direction Cosine Matrix
ESC	Electronic Speed Controller
FFC	Formation Flying Control
GPS	Global Positioning System
HIL	Hardware In the Loop
LiPo	Lithium Polymer
NLDI	Non-linear Dynamic Inversion
PID	Proportional Integral Derivative
PWM	Pulse Width Modulation
RTOS	Real Time Operating System
SBC	Single Board Computer
TET	Task Execution Time
TTL	Transistor-Transistor Logic
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle

1 Introduction

Aircraft systems are non-linear and complex. Their successful operation depends upon the coordination of many different subsystems. Abnormal conditions of any of these subsystems can easily compromise the safe operation of the aircraft, or even result in a catastrophic failure. Recently, aircraft safety has become a major objective in the aerospace engineering community (White, 2006). In tune with this objective, many Unmanned Aerial Systems (UAS) have been developed to provide an inexpensive, safe and efficient in-flight experimental environment for manned aircraft flight control system development and testing (Moncayo H., et. al , 2012).

Inexpensive UAS platforms coupled with the power of simulation and hardware in the loop (HIL) testing has allowed for the rapid development of advanced algorithms designed to increase aircraft safety. These methods of testing allow for new techniques to be repeatedly tested and fine tuned before they are used in flight. Furthermore, the designer is able to develop these controllers to accommodate various conditions and aircraft states.

Among these advanced algorithms designed to increase aviation safety, research efforts have been focused on the development of adaptive type controllers. Adaptive controllers attempt to adapt a system to either known or unknown disturbances while maintaining global stability. If an adaptive controller is able to adapt to a known disturbance or a model of it, it is known as a feedforward adaptive controller. The other category is feedback adaptive control, where additional information such as a state or a disturbance is fed back into the controller in order to reactively adapt to upset conditions.

This thesis focuses on the latter category of adaptive control using a technique known as feedback linearization or Non-Linear Dynamic Inversion (NLDI) (D. & J., 2001). NLDI allows the calculation of a non-linear control signal from the inversion of the equations governing the system motion in time. If the model is described with a high fidelity, then many non-linearities in the system will be cancelled out (Moncayo H., et. al, 2012). However, no system can be modeled perfectly or parameters obtained precisely, therefore, additional techniques may be used to aid its performance.

There are different alternatives to achieve an adaptation process. Two of the most common configurations are the implementation of a model rederence control, and adapting a baseline controller's gains by using an augmentation system. This thesis focuses on the second method with the design and implementation of an adaptive bio-inspired mechanism based on the artificial immune system (AIS) paradigm. The AIS is a mechanism which emulates the dynamic behavior of the human immune system in response to foreign pathogens (Chen & Wei, 2006).

The basic premise of the AIS mechanism is that as with the human body, it regulates the production of antibodies as a function of pathogen intensity. If the concentration of antibodies are too high, this can also prove toxic to the body and therefore must be regulated accordingly. This dynamic process is repeated until the invading pathogen has been reduced. The immune system metholodogy can be mapped to a mechanical system whereby the error can be seen as the invading pathogen and the system control inputs can be viewed as antibodies (Moncayo H., et. al, 2012).

Research efforts towards the design and development of advanced intelligent fault tolerance flight control systems are being performed by researchers at Embry-Riddle

Aeronautical University (ERAU) and West Virginia University (WVU). Researchers at WVU have developed and tested algorithms which aid aircraft guidance and navigation which are far more robust than the typical waypoint navigation designated formation flying control (FFC) (Campa & Napolitano, 2004). The equations for FFC were developed using a leader aircraft and a follower aircraft. However, if the follower aircraft is replaced with a single point, this method is called virtual trajectory tracking. In this thesis, all of the baseline controllers developed are based on the FFC equations and virtual trajectory tracking.

In Chapter 2 of this thesis, six flight control algorithms are developed,three baseline and the respective controllers augmented with an AIS: Basic PID, Outer loop NLDI, Extended NLDI, PID augmented with an AIS, Outer loop NLDI augmented with an AIS and the Extended NLDI augmented with an AIS. In Chapter 3, the UAV research platform is presented with a description of all subsystems and sensor components. Then, a model the aircraft and all of its systems, including sensors and actuators are developed in Chapter 4. Chapter 5 provides a description of the HIL test setup and an outline of the real time operating system (RTOS). Completion of the HIL test follows collection and analysis of the six control algorithms' performace over the test trajectories in Chapter 6. Finally, the thesis is concluded with an overview of the results and a foreword on future work and suggestions.

The research effort presented in this thesis has resulted in following publications:

- **Lyons, B.,** Moncayo H., Noriega, A., Moguel, I., Perhinschi, M., Hardware-in-the-Loop Simulation of an Extended Non-linear Dynamic Inversion Augmented with an Immune-Based Adaptive Control System, AIAA Guidance, Navigation, and Control Conference, Boston, Massachusetts, August 2013.

- Moncayo, H., Krishnamoorty,K., Wilburn, B., Wilburn, J., Perhinschi , M., **Lyons, B.** Performance Analysis of Fault Tolerant UAV Baseline Control Laws with L1 Adaptive Augmentation, Journal of Modeling, Simulation, Identification, and Control, Columbia International Publishing, Vol.1, No.4, pp. 137-163, 2013.

1.1 Literature Review

1.1.1 Adaptive Control

The seeds for adaptive control were planted in the mid 1950s, a decade where ideas for unconventional flight systems were prevalent. Flight control coupled with a growing interest in process control helped grow the field of adaptive control (Astrom, 1995). Then in the 1960s, theoretical advancements on now better understood theories such as model reference adaptive control, self-tuning control and neural networks were born. However, it would take many years before these theories were established well enough to be used in practical application (Astrom, 1995).

In 1959, one of the first attempts at in flight testing of adaptive control system was the NASA X-15-3 hypersonic experimental vehicle, which featured a Minneapolis Honeywell MH-96 adaptive controller (Zachary T. Dydek, 2010). The X-15-3 hypersonic vehicle featured nearly 200 successful flights from 1959-1968, with the exception of one crash in 1967. This single crash which was caused by a stability issue with the adaptive mechanism, motivated the development of formal techniques and methologies for adaptive control to address this problem (Zachary T. Dydek, 2008).

Since then, adaptive control has been molded into many forms, categories and heirarchies. Adaptive control can be classified under two main categories, feedforward control and feedback control. Feedforward adaptive control is a technique where a system is able to adapt to an external disturbance, before it has time to significantly affect the system. This technique requires state information about the disturbance and the ability to accurately model it. Feedback adaptive control, on the other hand, requires the feedback of a state, external information or a disturbance in the system in order to calculate the required inputs for the system to maintain desired operation.

Figure 1: NASA X-15 hypersonic research vehicle (NASA, 2000).

Feedforward and feedback control can be achieved using direct or indirect methods. Direct methods are ones where parameters required to adapt the system are calculated and used directly. Indirect methods are ones where parameters are calculated and used to aid the controller in adaptation rather than to directly adapt to a disturbance.

1.1.2 Non-linear Dynamic Inversion

Non-linear dynamic inversion (NLDI) is the process in which the equations of a system such as those in equation (1) are inverted and solved for system inputs. Where, $f(x)$ and $g(x)$ are dynamic non-linear state and control functions respectively. The system rates $\dot{x}$ are then replaced with the desired rates $\dot{x}_{des}$ to give the form of equation (2) as a control law (Ito, 2002). This inversion is only possible given that $g(x)^{-1}$ exists.

$$\dot{x} = f(x) + g(x)u \tag{1}$$

$$u = g^{-1}(x)\left[\dot{x}_{des} - f(x)\right] \tag{2}$$

In theory, if the system can be modeled accurately and $\dot{x}_{des} = \dot{x}$, then the nonlinearities of the system are cancelled. However, in practice a system cannot be modeled perfectly and not all disturbances can be accounted for. On the other hand, the nature of NLDI allows it to be used as a baseline test bed for adaptive techniques (Miller, 2011). This is because model information embedded in the control law can be updated online or offline to reflect changes in the system dynamics.

1.1.3 Artificial Immune System

Immunity is a biological reaction based on a feedback mechanism that enables living organisms to resist infections and diseases. The human immune system exhibits a strong resistance to invading pathogens by excreting anitbodies in the same intensity as the foreign material. However, a large concentration of antibodies will become toxic to the organism and as such must be regulated (Chen & Wei, 2006).

The immune system is composed of two types of antibodies and lymphocytes: T-cells and B-cells (E., 1992). B-cells are produced by the bone marrow and can recognize and eliminate the pathogens by generating antibodies. The thymus gland secretes T-cells (lymphocytes) which are composed of T-suppressant cells (Ts-cells) and T-helper cells (Th-cells). These cells stimulate and supress the production of B-cells which effectively adapts the entire immune system in order to create a dynamic balance between pathogens and antibodies.

Applying this model to a control system, relations between the biological immune feedback mechanism and the artificial immune feedback mechanism can be made: the invading pathogens can be seen as the error between the desired state and the actual state, the antibodies generated by the B-cells can be viewed as the control input to the system.

1.2 Background

This thesis is based on previous research efforts at West Virginia University and the Air Force Research Lab (AFRL), to develop a research aircraft as a platform for advanced control techniques with a focus on fault tolerance (Napolitano, 2002). The line of research which this thesis focuses on began with the design of control laws for formation flight or in the case of a single aircraft virtual tracking. These algorithms were successfully tested in flight on the research platform designated as the West Virginia

University model YF-22, shown in Figure 1 (Campa & Napolitano, 2004). The next step in development was to link the navigational states with formation flight and invert them. This step was called the outer loop non-linear dynamic inversion, or simply the "outer loop inversion" (Seanor & Campa, 2005).

Figure 2: YF-22 model at West Virginia University.

Then, (Moncayo, et. al, 2012) extended the non-linear dynamic inversion technique to link the outer loop to the inverted aircraft stability control equations or the "inner loop". This technique was designated as the "extended non-linear dynamic inversion" or extended NLDI. The extended NLDI technique links the "outer loop" navigational states to the states related to aircraft stability and ultimately the control surfaces. These algorithms were sucessfully tested in simulation using the YF-22 aircraft model (Moncayo, Perhinischi, et. al, 2012).

The final development in this line of research is the augmentation of the extended NLDI with an artificial immune system (AIS), which is able to adapt to control surface failures and external disturbances such as turbulence (Moncayo H., et. al, 2012). The

AIS is a self-sufficient feedback control adaptive system and the set of equations are developed independently of the FFC equations.

The aforementioned developments in aircraft control, beginning with formation flying control (FFC) to the adaptive AIS are the basis for the algorithms which are tested and compared in real time using hardware-in-the loop (HIL) on the model Embry-Riddle Aeronautical University (ERAU) Sig Rascal 110 as a precursor to in-flight testing. The control algorithms are tested under actuator failures and their performance evaluated using criterion as described in Chapter 5 of this thesis. The controllers that are tested and compared in this thesis, all based on FFC, are: PID control only, Outer-Loop NLDI control, Extended (both inner and outer-loop) NLDI control and the previous three controllers augmented with an AIS for a total of six controllers. Their performance metrics are based on trajectory tracking, control actuation and task execution time.

2 Theory

2.1 Equations of Motion

In order to simulate the aircraft dynamics of the UAS described in this thesis a non-linear six degree of freedom set of equations of motion were used. The equations are based on an earth fixed inertial reference frame with a mobile reference frame whose motion can be described relative to the inertial reference.

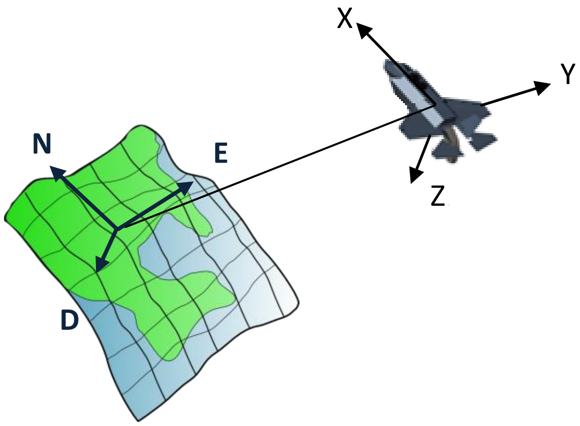


Figure 3: North East Down and Aircraft Body Centered reference frames.

A six degree of freedom model requires twelve independent equations of motion which fall under the following categories:

- Linear Velocities $[u, v, w]$(Velocities along the X, Y and Z axes of ABC respectively).
- Angular Velocities $[p, q, r]$(roll rate, pitch rate and yaw rate)
- Euler Angles $[\phi, \theta, \psi]$(roll, pitch and yaw orientation)
- Inertial Navigation $[N, E, D]$(Inertial X, Y and Z position in NED respectively).

In order to express the ABC reference frame in terms of the inertial reference frame, it is necessary to construct a rotational matrix or a direction cosine matrix (DCM) shown as (Nelson, 1998):

$$R_{NED} = \begin{bmatrix} \cos\theta\cos\phi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\phi & \sin\phi\sin\theta\sin\psi - \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi + \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{3}$$

Where,

R_{NED} is a rotational transformation matrix.

Linear Velocities

Once the ability to transform between the NED and ABC reference frame is achieved, the next step is to obtain the fundamental equations of motion to describe system dynamics utilizing Newton's second law. A full derivation will be skipped and only the most important parts of the derivation will be given. Newton's second law in three space or can be written as

$$\sum \vec{F}_{ext} = m \left. \frac{d\vec{v}}{dt} \right|_I \tag{4}$$

Where,

$\sum \vec{\boldsymbol{F}}_{ext}$ is the sum of all forces acting on the aircraft.

$\left.\frac{d\vec{\mathbf{v}}}{dt}\right|_I$ is the rate of change of the inertial linear velocity in $\mathbb{R}^3$.

m is aircraft mass.

Since the body's accelerations needs to be referenced to the inertial reference frame, solving for external accelerations equation (4) becomes equation (5).

$$\frac{\sum \vec{F}_{ext}}{m} = \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + ru - pw \end{bmatrix} \tag{5}$$

Solving for the aircraft linear body accelerations yields the following set of equations:

$$\begin{aligned} \dot{u} &= rv - qw + \frac{F_{X\,ext}}{m} \\ \dot{v} &= pw - ru + \frac{F_{Y\,ext}}{m} \\ \dot{w} &= pw - ru + \frac{F_{Z\,ext}}{m} \end{aligned} \tag{6}$$

Where,

F_{Next} are the external forces in the N^{th} ABC frame axis.

For an aircraft, the external forces are typically divided into two general categories the aerodynamic forces and the thrust due to the engine. Therefore, F_{Next} can be written as $F_{Next} = F_{Naero} + F_{Nengine}$.

Angular Accelerations

Newton's law as applied to rotational motion (angular momentum) in the body frame must be referenced to the inertial frame as before using the Coriolis identity

$$\sum \vec{\boldsymbol{M}}_{ext} = \frac{d\vec{\boldsymbol{H}}_B}{dt} + \left(\vec{\boldsymbol{\omega}} \times \vec{\boldsymbol{H}}_B\right) \tag{7}$$

Where,

$\vec{\boldsymbol{H}}_B = \overrightarrow{\omega} \times I$

$\sum M_{ext}$ is the sum of the external moments

$\vec{\omega}$ is a vector containing the aircraft angular velocities

I Is the aircraft mass moment of inertia matrix

Assuming symmetry about the X-Z plane, the aircraft inertia matrix can be written as:

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & I_{xz} \\ 0 & I_{yy} & 0 \\ I_{zx} & 0 & I_{zz} \end{bmatrix} \tag{8}$$

Expanding the right hand side of equation (7) yields the following equation:

$$\sum \vec{\boldsymbol{M}}_{ext} = \begin{bmatrix} I_x \dot{p} - I_{xz}\dot{r} + qr\left(I_z - I_y\right) - I_{xz}pq \\ I_y \dot{q} + rp\left(I_x - I_z\right) + I_{xz}\left(p^2 - r^2\right) \\ I_z \dot{r} - I_{xz}\dot{p} + pq\left(I_y - I_x\right) + I_{xz}qr \end{bmatrix} \tag{9}$$

As with the external forces, the external moments are divided into two general categories the aerodynamic forces and the thrust due to the engine. Therefore, M_{Next} can be written as $M_{Next} = M_{Naero} + M_{Nengine}$.

Euler Rates

In order to obtain the Euler rates a simple transformation using the Euler angles and aircraft angular rates is used. These equations are known as the kinematic equations of motion (Nelson, 1998).

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \sin\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{10}$$

Inertial Velocity

In order to complete the 12 equations of motion required by the 6 degree of freedom model of the aircraft, the inertial velocity $\begin{bmatrix} \dot{N} & \dot{E} & \dot{D} \end{bmatrix}$ of the aircraft needs to be obtained. These states are easily obtained by applying the DCM of equation 3 to the aircraft body velocity components $\begin{bmatrix} u & v & w \end{bmatrix}$.

$$\begin{bmatrix} \dot{N} \\ \dot{E} \\ \dot{D} \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\phi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\phi & \sin\phi\sin\theta\sin\psi - \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi + \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{11}$$

Wind Angles

The last set of important variables needed to complete the simulation are the wind angle components and true airspeed or $\begin{bmatrix} V_{TAS} & \alpha & \beta \end{bmatrix}$.

$$
\begin{gathered}
V_{TAS} = \sqrt{{u_r}^2 + {v_r}^2 + {w_r}^2} \\
\alpha = \tan^{-1}\left(\frac{w_r}{u_r}\right) \\
\beta = \sin^{-1}\left(\frac{v_r}{\sqrt{{u_r}^2 + {v_r}^2 + {w_r}^2}}\right)
\end{gathered} \tag{12}
$$

Where,

$$
\begin{aligned}
u_r &= u + u_{wind} \\
v_r &= v + v_{wind} \\
w_r &= w + w_{wind}
\end{aligned}
$$

V_{TAS} is true air speed

α is angle of attack

β is angle of sideslip

2.2 Flight Control Laws

2.2.1 FFC Virtual Trajectory Tracking

Virtual trajectory tracking is derived from FFC with the leader aircraft removed and replaced with a desired trajectory. The formation flight control is based on the geometry in the reference frame of the follower aircraft and its location with respect to the leader aircraft (in inertial space), as shown in Figure 4. This problem can be natually decomposed in the a horizontal tracking problem and a vertical tracking problem (Campa & Napolitano, 2004).

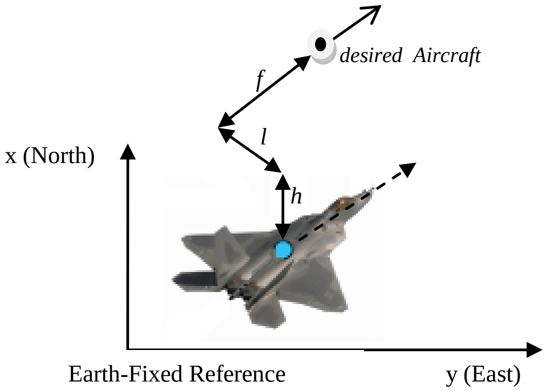


Figure 4: Formation Flight Geometry (Moncayo H., et al, 2012).

$$l = \frac{V_{V_y}\left(x_V - x\right) - V_{V_x}\left(y_V - y\right)}{V_{V_{xy}}} \tag{13}$$

$$f = \frac{V_{V_y}\left(y_V - y\right) - V_{V_x}\left(x_V - x\right)}{V_{V_{xy}}} \tag{14}$$

$$h = z_V - z \tag{15}$$

Where,

l is the lateral error between the virtial point and the follower

f is the forward error between the virtual point and the follower

h is the vertical error between the virtual point and the follower

V_{V_n} is the velocity of the virtual point projected along th n^{th}-axis

n_V is the n^{th}-axis position of the virtual point

n is the n^{th}-axis position of the follower aircraft

Here n defines the x, y and z axes of the Earth-fixed reference frame

In equations (13) and (14), $V_{V_{xy}} = \sqrt{V_{V_x}^2 + V_{V_y}^2}$ is the projection of the desired trajectory velocity onto the x-y plane. A more compact way to write the equations (13) through (15) is:

$$\begin{bmatrix} l \\ f \\ h \end{bmatrix} = \begin{bmatrix} \sin(\chi_V) & -\cos(\chi_V) & 0 \\ \cos(\chi_V) & \sin(\chi_V) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_V - x \\ y_V - y \\ z_V - z \end{bmatrix} \tag{16}$$

Where χ_V is defined by,

$$\cos(\chi_V) = \frac{V_{V_x}}{\sqrt{V_{V_x}^2 + V_{V_y}^2}} \quad , \quad \sin(\chi_V) = \frac{V_{V_y}}{\sqrt{V_{V_x}^2 + V_{V_y}^2}} \tag{17}$$

The relative forward, lateral and vertical speeds of the aircraft are defined as the time derivatives of the forward, lateral, and vertical distance respectively:

$$\dot{l} = \frac{V_{V_x} V_y - V_{V_y} V_x}{V_{V_{xy}}} + \Omega_V f \tag{18}$$

$$\dot{f} = \frac{V_{V_x} V_x - V_{V_y} V_y}{V_{V_{xy}}} + \Omega_V l \tag{19}$$

$$\dot{h} = V_{z_V} - V_z \tag{20}$$

Where,

Ω_V is the induced angular velocity of the virtual leader.

The term Ω_V is considered zero here because it is assumed the virtual point perfectly follows the trajectory. Therefore, equations (18), (19), and (20) can be then written as:

$$\begin{bmatrix} \dot{l} \\ \dot{f} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} V_{xy} \sin\left(\chi - \chi_V\right) \\ V_{V_{xy}} - V_{xy} \cos\left(\chi - \chi_V\right) \\ V_{z_V} - V_z \end{bmatrix} \tag{21}$$

Equations (16) and (21) form the basis of the Virtual Trajectory Tracking control laws and are developed in the following sections to show how they are implemented with the PID, Outer Loop NLDI, Inner Loop NLDI cotrol systems.

2.2.1.1 Virtual Tracking with PID Control

The Virtual Tracking with PID control system is divided into three sections (Campa & Napolitano, 2004): calculation of the trajectory tracking variables, the outer loop controller which relates the trajectory tracking variables to aircraft attitude and throttle commands, and the inner loop controller which relates the aircraft state commands to the actuator position required to achieve the new state.

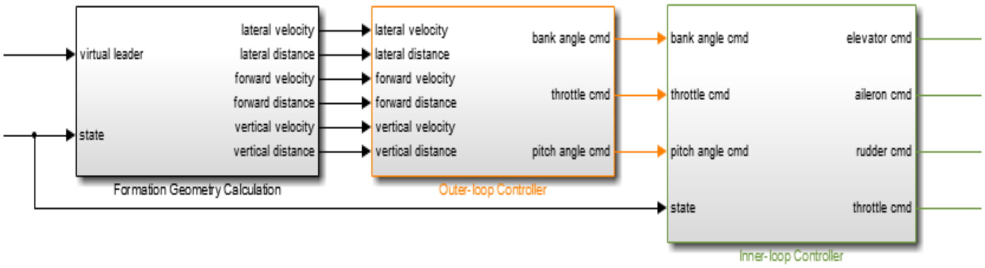


Figure 5: Block diagram of the virtual tracking with PID control system.

As described by (Campa & Napolitano, 2004), the FFC problem can be decoupled into a horizontal plane and a vertical plane tracking problem. The simplest controller in the PID control scheme is the vertical tracking scheme. Using the vertical distance and its rate of change as inputs, this controller provides the desired pitch angle θ_d, which is then taken as a desired attitude by the longitudinal inner controller:

$$\theta_d = K_{\dot{Z}}\dot{\delta}_Z + K_Z\delta_Z \qquad (22)$$

Where,

δ_Z is the vertical distance error between the virtual point and the follower

$\dot{\delta}_Z$ is the vertical velocity error between the virtual point and the follower

K_Z is the proportional gain for the vertical distance error

$K_{\dot{Z}}$ is the proportional gain for the vertical velocity error

The lateral and forward tracking control schemes are fomulated in the same way with bank angle and throttle command as their outputs respectively (Campa & Napolitano, 2004).

$$\phi_d = K_{\dot{l}}\dot{l} + K_l l \tag{23}$$

$$\delta_T = K_{\dot{f}}\dot{f} + K_f f \tag{24}$$

Actuator command is provided by utilizing the classical PID control algorithms by taking the linear combination of the proportional, integral and derivative of the error of the desired aircraft state. The PID control logic used in this thesis is shown in the equations (25)-(27) in discrete form.

$$\delta_e = \left(k_{P\theta} + \frac{k_{I\theta}}{z-1} + k_{D\theta}\frac{z-1}{z} \right)(\theta_d - \theta) - k_{Pq} q \tag{25}$$

$$\delta_a = \left(k_{Pa\phi} + \frac{k_{Ia\phi}}{z-1} + k_{Da\phi}\frac{z-1}{z} \right)(\phi_d - \phi) - k_{Pap} p - k_{Par} r \tag{26}$$

$$\delta_r = \left(k_{Pr\phi} + \frac{k_{Ir\phi}}{z-1} + k_{Dr\phi}\frac{z-1}{z} \right)(\phi_d - \phi) - k_{Prp} p - k_{Prr} r \tag{27}$$

Where,

δ_n is the nth control surface: elevator, aileron and rudder respectively.

k_{Pm} is the mth proportional gain.

k_{Im} is the mth integral gain.

k_{Dm} is the mth derivative gain.

The PID controller is designed to counteract the effects of high rates and simultaneously achieve the desired reference angle.

2.2.1.2 Outer Loop - NLDI Controller Design

The next step in the controller design is to invert the “outer loop” of the FFC equations. Once again, the FFC control problem is divided into two problems, the vertical and the horizontal tracking problems. The vertical tracking problem is the basic linear equation defined in equation (22) which controls the aircraft altitude.

The outer loop inversion is performed on the lateral and forward controllers, where the bank angle and throttle commands be some function of the azimuth difference between the aircraft and the virtual leader, the lateral and forward errors and their rates or written precisely as (Seanor & Campa, 2005):

$$\begin{bmatrix} \delta_T \\ \phi_d \end{bmatrix} = f\left(\chi - \chi_L, \begin{bmatrix} f \\ l \end{bmatrix}, \begin{bmatrix} \dot{f} \\ \dot{l} \end{bmatrix} \right) \tag{28}$$

The lateral and forward equations in (16) and (21) do not contain δ_T and ϕ_d, therefore their derivative must be taken until they appear in the equation, otherwise these parameters have no affect on l, or f (Hedrick & Girard, 2005). Here, only the second derivative of l, and f along with several substitutions to achieve this were required.

$$\begin{bmatrix} \ddot{l} \\ \ddot{f} \end{bmatrix} = \begin{bmatrix} \sin(\chi - \chi_V) \\ -\cos(\chi - \chi_V) \end{bmatrix} \dot{V}_{xy} + \begin{bmatrix} \cos(\chi - \chi_V) \\ \sin(\chi - \chi_V) \end{bmatrix} V_{xy}(\Omega - \Omega_V) + \begin{bmatrix} f \\ -l \end{bmatrix} \dot{\Omega}_V + \begin{bmatrix} \dot{f} \\ -\dot{l} \end{bmatrix} \Omega_V \tag{29}$$

$$\dot{V} = \left(\frac{cos\alpha \cos\beta}{m} \right) T - \left(\frac{\bar{q}S\left(C_D \cos\beta - C_Y \sin\beta\right)}{m} + g\sin\gamma \right) \equiv \omega_1 T - \omega_2 \tag{30}$$

Assuming no lateral acceleration or a coordinated turn condition for the follower aircraft the approximation $\Omega = \dot{\chi} \equiv \dot{\psi} \equiv \dfrac{g}{V} tan\phi$ can be made.

$$\begin{bmatrix} \ddot{l} \\ \ddot{f} \end{bmatrix} = \begin{bmatrix} V_{xy}\cos\left(\chi-\chi_V\right) & \frac{V_{xy}}{V}\omega_1 \sin\left(\chi-\chi_V\right) \\ V_{xy}\sin\left(\chi-\chi_V\right) & -\frac{V_{xy}}{V}\omega_1 \cos\left(\chi-\chi_V\right) \end{bmatrix} \begin{bmatrix} \frac{g}{V}\tan\phi \\ T_0 + K_T\delta_T \end{bmatrix} + \frac{V_{xy}}{V}\omega_2 \begin{bmatrix} -\sin\left(\chi-\chi_V\right) \\ \cos\left(\chi-\chi_V\right) \end{bmatrix} - \Omega_V V_{xy} \begin{bmatrix} \cos\left(\chi-\chi_V\right) \\ \sin\left(\chi-\chi_V\right) \end{bmatrix} + \Omega_V \begin{bmatrix} \dot{f} \\ -\dot{l} \end{bmatrix} \tag{31}$$

Where,

$$\omega_1 = \frac{1}{m} cos\alpha \cos\beta$$

$$\omega_2 = \frac{\bar{q}S}{m}\left(C_D\cos\beta - C_Y\sin\beta\right) + \sin\gamma$$

Now the equations can be solved for δ_T and ϕ_d

$$\begin{bmatrix} \frac{g}{V}\tan\phi \\ T_0 + K_T\delta_T \end{bmatrix} = \frac{1}{V_{xy}} \begin{bmatrix} \cos\left(\chi-\chi_V\right) & \sin\left(\chi-\chi_V\right) \\ \frac{V}{\omega_1}\sin\left(\chi-\chi_V\right) & -\frac{V}{\omega_1}\cos\left(\chi-\chi_V\right) \end{bmatrix} \begin{bmatrix} \ddot{l}_d \\ \ddot{f}_d \end{bmatrix} + \begin{bmatrix} \Omega_V \\ \frac{\omega_2}{\omega_1} \end{bmatrix} + \frac{\Omega_V}{V_{xy}} \begin{bmatrix} \dot{l}\sin\left(\chi-\chi_V\right) - \dot{f}\cos\left(\chi-\chi_V\right) \\ -\frac{V}{\omega_1}\dot{l}\cos\left(\chi-\chi_V\right) - \frac{V}{\omega_1}\dot{f}\sin\left(\chi-\chi_V\right) \end{bmatrix} \tag{32}$$

By imposing $\alpha = \alpha_0$, $\beta = 0$ and $\Omega = constant$, the lateral NLDI control law is:

$$\phi_d = \arctan\left\{ \frac{1}{g\cos\gamma}\left[\ddot{l}_d\cos\left(\chi-\chi_V\right) + \ddot{f}_d\sin\left(\chi-\chi_V\right)\right] + \frac{V}{g}\Omega_V + \left[\dot{l}\sin\left(\chi-\chi_V\right) - \dot{f}cos\left(\chi-\chi_V\right)\frac{\Omega_V}{g\cos\gamma}\right]\right\} \tag{33}$$

and the forward control law is:

$$\begin{aligned}\delta_T \quad = \quad & \frac{m}{K_T \cos\gamma}\left[\ddot{l}_d \sin\left(\chi-\chi_V\right)-\ddot{f}\cos\left(\chi-\chi_V\right)\right] \\ & +\frac{1}{K_T}\left(\frac{1}{2}\rho V^2 S\left(C_{D0}+C_{D\alpha}\alpha_0\right)+mg\sin\gamma-T_0\right) \\ & -\frac{m}{K_T \cos\gamma}\Omega_V\left[\dot{l}\cos\left(\chi-\chi_V\right)+\dot{f}\sin\left(\chi-\chi_V\right)\right]\end{aligned} \tag{34}$$

It can be seen that if equation (35) holds true, then the nonlinearities of the system are canceled. The lateral and forward accelerations can be controlled via compensator type equations (36).

$$\begin{bmatrix}\ddot{l} \\ \ddot{f}\end{bmatrix}=\begin{bmatrix}\ddot{l}_d \\ \ddot{f}_d\end{bmatrix} \tag{35}$$

$$\begin{aligned}\ddot{l}_d &= -K_{ls}\dot{l}-K_l l \\ \ddot{f}_d &= -K_{fs}\dot{f}-K_f f\end{aligned} \tag{36}$$

The bank and pitch commands are then sent to the inner loop which uses PID control identical to equations (25) through (27).

2.2.1.3 Inner Loop - NLDI Controller

The final step in the baseline controller design is to develop the equations for the inner loop inversion of the FFC or the “NLDI inner loop”. Once again, given a nonlinear system described by equation (1), an inversion is possible only if $g(u)^{-1}$ exists. To ensure invertibility, a two-time-scale inversion is implemented as shown in Figure 6 (Moncayo, Perhinischi, et. al., 2012). The two-time-scale inversion consists of "slow mode" and the "fast mode" states of the aircraft. The slow mode utilizes Euler angles $[\phi_d \quad \theta_d \quad \psi_d]^T$ and outputs desired aircraft angular rates $[p_d \quad q_d \quad r_d]^T$. The fast mode takes the desired angular rates and calculates the deflections $[\delta a \quad \delta e \quad \delta r]^T$ required to achieve these states.

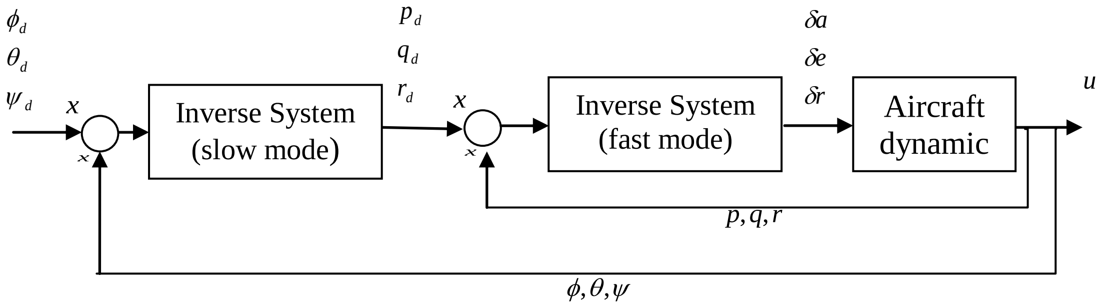


Figure 6: Two-time-scale dynamic inversion system (Moncayo, Perhinischi, et. al, 2012).

2.2.1.3.1 Slow Mode System

Inversion of the slow mode system begins with the kinematic equations which relates the Euler rates to the aircraft angular rates, reproduced in equation (37).

$$\begin{aligned}\dot{\phi} &= p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \\ \dot{\theta} &= q\cos\phi - r\sin\phi \\ \dot{\psi} &= q\sin\phi\sec\theta + r\cos\phi\sec\theta\end{aligned} \tag{37}$$

Next, equation (37) is solved for $\begin{bmatrix} p & q & r \end{bmatrix}^T$, then the angular rates $\begin{bmatrix} p & q & r \end{bmatrix}^T$ are replaced with desired angular rates $\begin{bmatrix} p_d & q_d & r_d \end{bmatrix}^T$, and finally the Euler Rates replaced with pseudo Euler rates $\begin{bmatrix} U_\phi & U_\theta & U_\psi \end{bmatrix}^T$. Pseudo Euler rates are required since it is not possible to measure the states $\begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$ using any sensor.

$$\begin{bmatrix} p_d \\ q_d \\ r_d \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix}^{-1} \begin{bmatrix} U_\phi \\ U_\theta \\ U_\psi \end{bmatrix} \tag{38}$$

Where the pseudo Euler rates are defined as:

$$\begin{bmatrix} U_\phi \\ U_\theta \\ U_\psi \end{bmatrix} = \begin{bmatrix} K_\phi\left(\phi_d - \phi\right) \\ K_\theta\left(\theta_d - \theta\right) \\ K_\psi\left(\psi_d - \psi\right) \end{bmatrix} \tag{39}$$

As with the outer loop equation, the pseudo rates are regulated using proportional control. However, outer-NLDI controller does not provide the yaw angle ψ , so the third element in equation (39) could not be calculated. This is solved assuming a coordinated turn condition such as:

$$\dot{\psi} = \frac{g}{V}\tan\phi \tag{40}$$

Then, the derivative of the yaw angle is be replaced with the pseudo control U_{ψ}, and the roll angle with the desired angle ϕ_d :

$$U_{\psi} = \dot{\psi} = \frac{g}{V}\tan\phi_d \tag{41}$$

Finally, with equation (39) the desired angular rates are calculated and sent to the fast mode to compute the required control surface deflections (Moncayo, Perhinischi, et. al, 2012).

2.2.1.3.2 Fast Mode System

The fast mode system of the NLDI inner loop equations are developed by relating the aircraft angular motion to the control surface deflections using the aerodynamic moments (Moncayo H., et. al , 2012). Here, it is useful to express the aerodynamic moments in terms of the aerodynamic coefficients C_l, C_m, C_n:

$$\bar{M}_A = \begin{bmatrix} L_A \\ M_A \\ N_A \end{bmatrix} = \bar{q}S \begin{bmatrix} bC_l\left(x,\delta\right) \\ \bar{c}C_m\left(x,\delta\right) \\ bC_n\left(x,\delta\right) \end{bmatrix} \tag{42}$$

Where,

S is the wing area.

$\bar{q}$ is the dynamic pressure.

b is the wing span.

$\bar{c}$ is the mean aerodynamic chord.

δ is a placeholder for relvant actuators.

L_A is the aerodynamic moment along the aircraft X-axis or the rolling moment.

M_A is the aerodynamic moment along the aircraft Y-axis or the pitching moment.

N_A is the aerodynamic moment along the aircraft Z-axis or the yawing moment.

The left hand side of equation (42) is typically written in terms of angular accelerations $\left[\dot{p} \quad \dot{q} \quad \dot{r}\right]^T$. Furthermore, assuming aircraft symmetry about the X-Z plane, the moments of inertia I_{xy} and I_{xz} can be considered zero. Therefore, the moment equation can br written as:

$$\begin{aligned}
\dot{p} &= \frac{I_{yy}-I_{zz}}{I_{xx}}qr+\frac{I_{xz}}{I_{xx}}\left(\dot{r}+pq\right)+L_A \\
\dot{q} &= \frac{I_{zz}-I_{xx}}{I_{yy}}pr+\frac{I_{xz}}{I_{yy}}\left(r^2+p^2\right)+M_A \\
\dot{r} &= \frac{I_{xx}-I_{yy}}{I_{zz}}pq+\frac{I_{xz}}{I_{zz}}\left(\dot{p}+qr\right)+N_A
\end{aligned} \tag{43}$$

Solving the equation (43) for the aerodynamic moments $\begin{bmatrix}L_A & M_A & N_A\end{bmatrix}^T$, and replacing them with desired moments $\begin{bmatrix}L_{Ad} & M_{Ad} & N_{Ad}\end{bmatrix}^T$, and the angular accelerations $\begin{bmatrix}\dot{p} & \dot{q} & \dot{r}\end{bmatrix}^T$ with the pseudo angular accelerations $\begin{bmatrix}U_p & U_q & U_r\end{bmatrix}^T$, the moment equation can be rewritten as:

$$\begin{bmatrix}L_{Ad}\\ M_{Ad}\\ N_{Ad}\end{bmatrix}=\begin{bmatrix}-I_{xz}pq+\left(I_{zz}-I_{yy}\right)qr\\ -I_{xz}\left(p^2-r^2\right)+\left(I_{xx}-I_{zz}\right)pr\\ -I_{xz}qr+\left(I_{yy}-I_{xx}\right)pq\end{bmatrix}+\begin{bmatrix}I_{xx} & 0 & -I_{xz}\\ 0 & I_{yy} & 0\\ -I_{xz} & 0 & I_{zz}\end{bmatrix}\begin{bmatrix}U_p\\ U_q\\ U_r\end{bmatrix} \tag{44}$$

Where the pseudo angular accelerations are defined as

$$\begin{bmatrix}U_p\\ U_q\\ U_r\end{bmatrix}=\begin{bmatrix}K_p\left(p_d-p\right)\\ K_q\left(q_d-q\right)\\ K_r\left(r_d-r\right)\end{bmatrix} \tag{45}$$

Once again, proportional control is used to control the pseudo rates. The next step in the fast mode design process is to develop the right side of equations (42). Using the standard perturbative techniques to expand the aerodynamic moment coefficient functions $\left[C_l\left(x,\delta\right) \quad C_m\left(x,\delta\right) \quad C_n\left(x,\delta\right)\right]^T$, they can be written as (Moncayo, Perhinischi, et al, 2012):

$$\begin{aligned} C_l\left(x,\delta\right) &= C_{l0} + C_{l\beta}\beta + \frac{b}{2V}\left(C_{lp}p + C_{lr}r\right) + C_{l\delta a}\delta_a + C_{l\delta r}r \\ C_m\left(x,\delta\right) &= C_{m0} + C_{m\alpha}\alpha + \frac{c}{2V}C_{mq}q + C_{m\delta e}\delta_e \\ C_n\left(x,\delta\right) &= C_{n0} + C_{n\beta}\beta + \frac{b}{2V}\left(C_{np}p + C_{nr}r\right) + C_{n\delta a}\delta_a + C_{n\delta r}r \end{aligned} \tag{46}$$

It is apparent that the aileron, elevator and rudder control commands $\left[\delta_a \quad \delta_e \quad \delta_r\right]^T$ now appear in equation set (46). In order to complete the fast mode inversion, the equation must be solved for these variables.

The longitudinal inversion is completed by solving for δ_e in the equation for $C_m\left(x,\delta\right)$ in equation set (46) yields the following expression.

$$\delta_e = \frac{C_m\left(x,\delta\right)C_{m0} - C_{m\alpha}\alpha - \frac{\bar{c}}{2V}C_{mq}q}{C_{m\delta e}} = \frac{\frac{M_{ad}}{\bar{q}S\bar{c}} - C_{m0} - C_{m\alpha}\alpha - \frac{\bar{c}}{2V}C_{mq}q}{C_{m\delta e}} \tag{47}$$

Where,

V is the aircraft's true airspeed.

The lateral-directional control equations require some manipulation in order to solve for their control commands. By inspection of the equations for $C_l\left(x,\delta\right)$ and

$C_n(x,\delta)$, one can see that the modes are coupled since cross terms for δ_a and δ_r appear in both equations. However, it is possible to obtain explicit linear expressions for δ_a and δ_r following these manipulations (Moncayo H., et. al, 2012):

$$b_1 = C_{l\delta a}\delta a + C_{l\delta r}\delta r \quad and \quad b_1 = C_{n\delta a}\delta a + C_{n\delta r}\delta r \tag{48}$$

Placing equation (48) into equation (46) for the rolling and yawing moment expressions gives:

$$\begin{aligned} b_1 &= C_l(x,\delta) - C_{l0} - C_{l\beta} - \frac{b}{2V}\left(C_{lp}p + C_{lr}r\right) = \frac{L_{Ad}}{\bar{q}Sb} - C_{l0} - C_{l\beta} - \frac{b}{2V}\left(C_{lp}p + C_{lr}r\right) \\ b_2 &= C_n(x,\delta) - C_{n0} - C_{n\beta} - \frac{b}{2V}\left(C_{np}p + C_{nr}r\right) = \frac{N_{Ad}}{\bar{q}Sb} - C_{n0} - C_{n\beta} - \frac{b}{2V}\left(C_{np}p + C_{nr}r\right) \end{aligned} \tag{49}$$

All the terms in the equation (49) are known, and hence the terms b_1 and b_2 are known. Finally, solving for δ_a and δ_r in the equation (46), results in:

$$\begin{aligned} \delta_a &= \frac{C_{l\delta r}b_2 - C_{n\delta r}b_1}{C_{l\delta r}C_{n\delta a} - C_{n\delta r}C_{l\delta a}} \\ \delta_r &= \frac{C_{n\delta r}b_2 - C_{l\delta r}b_1}{C_{l\delta r}C_{n\delta a} - C_{n\delta r}C_{l\delta a}} \end{aligned} \tag{50}$$

2.3 Artificial Immune System Adaptive Control Law

The artificial immune system (AIS) adaptive control law is based on a mathematical model of the biological immune system's ability to adaptively respond to an invading antigen. At time k, amount of antigens generated can be defined as $\varepsilon(k)$, the stimulated output from the helper cells as $T_h(k)$, and the effect of the suppressor sells on the B-cells as $T_s(k)$. Thus, the total stimulation of the B-cells $B(k)$ by the suppressor cells and the helper cells can be modeled as (Chen & Wei, 2006) :

$$
\begin{aligned}
B(k) &= T_h(k) - T_s(k) \\
T_h(k) &= k_1 \varepsilon(k) \\
T_s(k) &= k_2 f\left[\Delta B(k)\right]\varepsilon(k)
\end{aligned}
\tag{51}
$$

Where,

k_1 is the helper cell T_h reaction rate.

k_2 is the suppressor cell T_s reaction rate.

$f\left[\Delta B(k)\right]$ is a nonlinear function corresponding to the concentration change of the B-cells.

Finally substituing equations for $T_h(k)$ and $T_s(k)$ for the equation for $B(k)$,it is apparent that a feedback mechanism is formed:

$$B(k) = K\left(1 - \eta f\left[\Delta B(k)\right]\right)\varepsilon(k) \tag{52}$$

Where,

$K = k_1$

$\eta = \frac{k_2}{k_1}$ or the level of interaction between the T_h and T_s cells.

The relation between the immune feedback mechanism of equation (43) and a conventional feedback controller is as follows (Moncayo H., et. al , 2012): Let $y_d(k)$ be the desired system output and $y(k)$ be the current output of the system at time step k. Then, the error $e(k) = y_d(k) - y(k)$ can be seen as the antigen invading the system $\varepsilon(k)$. Furthermore, the total antibodies produced by the B-cell stimulation corresponds to the control input $u(k)$. Making these substitutions, equation (52) can be written as a discrete form feedback equation:

$$u(k) = K\left(1 - \eta f\left(\Delta u(k)\right)\right)e(k) \tag{53}$$

According to (Moncayo H., et al, 2012), the immune non-linear function $f\left(\Delta u(k)\right)$ can be defined as

$$f\left(\Delta u(k)\right)=1-\frac{2}{exp\left(\left(\Delta u(k)\right)^2/a\right)+exp\left(-\left(\Delta u(k)\right)^2/a\right)} \tag{54}$$

Where,

a is an operational factor of antibody concentration.

$\Delta u(k)$ is the change of the controller output.

Placing equation (54) in equation (53) creates a non-linear proportional controller which gain changes based on the output of the feedback controller. However, since proportional control is not sufficient to fully control a non-linear system with noise and non-linear disturbances, an additional control technique is required to minimize these disturbances. Therefore, the immune controller of equation (53) will be fused with a conventional PID controller which are expected to synergistically improve performance (Moncayo H., et al, 2012). Using the discrete representation of the linear PID controller, the immune based PID controller and be written as:

$$u(k)=k_p\left(1-\eta f\left(\Delta u(k)\right)\right)\left(1+\frac{k_i}{z-1}+k_d\frac{z-1}{z}\right)e(k) \tag{55}$$

By the definition of a PID controller and the definition of η the PID gains and η need to be positive for $\left(k_p,k_i,k_d\right)>0 \quad and \quad \eta>0$ for negative feedback control.

Additionally, it can be seen that must hold true for a stable system. Therefore, an upper limit on η must be placed to ensure this is always the case:

$$0 \le \eta \le \frac{1}{\sup f\left(\Delta u(k)\right)}, \quad \text{for all } k \tag{56}$$

Where,

$\sup f\left(\Delta u(k)\right)$ is the supermum of the set of values within $f\left(\Delta u(k)\right)$.

The final scheme for the immune based PID controller is shown in Figure 7 below.

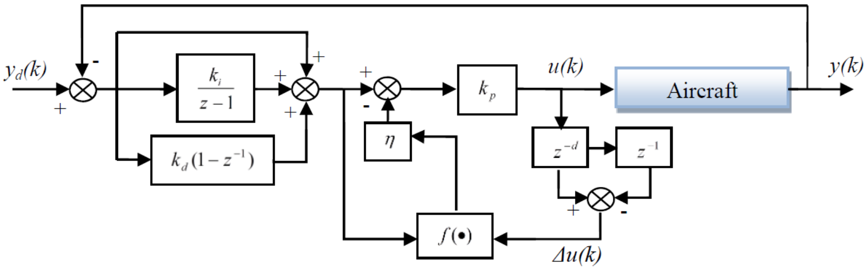


Figure 7: Immune feedback controller (Moncayo H., et al, 2012).

The AIS adaptive control system is a modular replacement for any PID control system. As such, the three control algorithms described in this paper (virtual tracking with PID, outer loop NLDI, extended NLDI) utilizes the AIS feedback controller in their inner loops. The inner loops were chosen because the failures tested in this thesis were actuator failures which are only compensated directly by the robustness of the inner loop controller of the aircraft.

2.3.1 AIS Augmentation

The three baseline controllers are augmented with the AIS feedback mechanism to create a non-linear adaptive controller. The focus of this thesis is adaptation to actuator failures, and as such, the adaptive mechanism must have access to the fastest information available in order to compensate quickly to upset conditions. Therefore, the AIS feedback mechanism is placed in the inner loop of the the three baseline controllers. Figure 8 shows the schematic of the inner loop of the basic PID and outer loop NLDI augmented with the AIS. For the extended NLDI controller the fast mode system is the section of the inner loop augmented as shown in Figure 9.

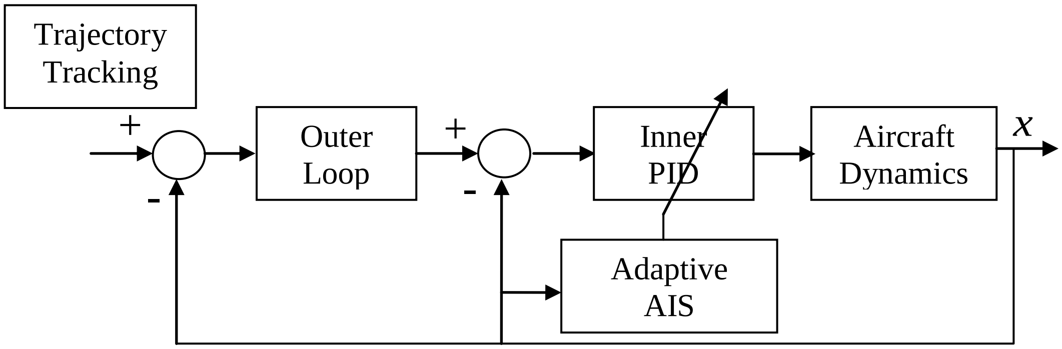


Figure 8: Basic PID/outer loop NLDI AIS augmentation diagram.

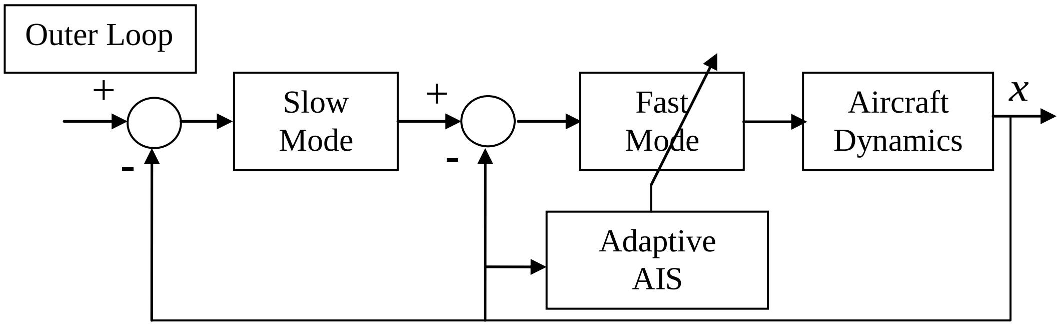


Figure 9: Extended NLDI augmented with AIS mechanism.

3 UAS Research Platform

The UAS airframe used to test the control laws in this thesis is the Sig Rascal 110 (Figure 10). The aircraft is an all electric tail dragger with flaps and a fixed landing gear. This aircraft was chosen because of its stable flying characteristics and large space to house the electronics which will eventually be used in flight. Furthermore, a host of institutions use this aircraft for basline and advanced UAS research (Ma & Stephanyan, 2006) (Choon Seong, 2008) (Xiao, 2009). A list of specifications is presented in Table **Error! Reference source not found.**.

Table 1: Sig Rascal 110 geometric data.

Parameter	**Value**
Mass	$7.00kg$
Wingspan	$2.80m$
Wing Area	$0.981m^2$
MAC	$0.351m$
I_{xx}	$2.64kg\cdot m^2$
I_{yy}	$2.10kg\cdot m^2$
I_{zz}	$2.59kg\cdot m^2$

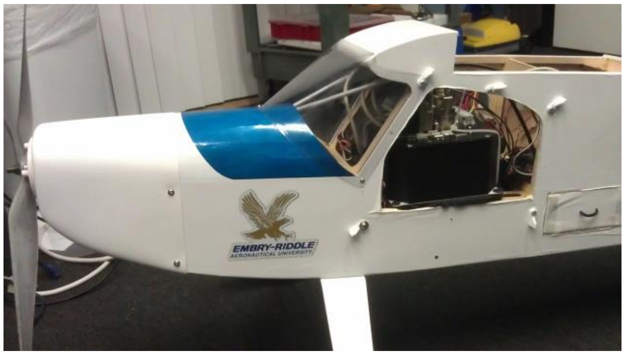


Figure 10: Sig Rascal 110 with the primary flight computer installed.

3.1 Actuators

The recommended actuator for use with the Rascal 110 was the HS-5625MG Digital Servo by Hitec. The HS- 5625MG (shown in Figure 11) is digital metal gear servo designed for high speed and high torque applications. It has an operating voltage of 4.8-6V and a rotation of 180. The maximum torque output by the HS-5625MG is 131 oz-in at 6V.

Figure 11: HS- 5625MG digital servos by Hitec.

3.2 Motor and Propellor

The motor propeller combination recommended for the Rascal 100 was the Hacker A-60 16M (Figure 12) motor and the 20x13 propeller (Figure 13) by Advanced Precision Composites (APC) respectively. The motor is a brushless DC or BLDC motor which draws up to 60A of current and operates at 215Kv (215 rpm/V). The 20x13 name of the APC propeller implies that the propeller's diameter is 20 inches with a pitch of 13 degrees at 25% of the radius length.

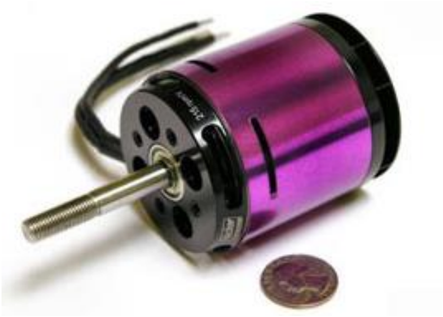


Figure 12: Hacker A-60 16M brushless DC motor.

Figure 13: 20x13 propellor by Advanced Precision Composites.

3.3 Electronic Speed Controller

The electronic speed controller (ESC) used to turn the Hacker A-60 motor was the Jeti Spin Pro 99 Opto Brushless ESC shown in Figure 14. The ESC supports a continuous current draw of 99A and a max current draw of up to 109A.

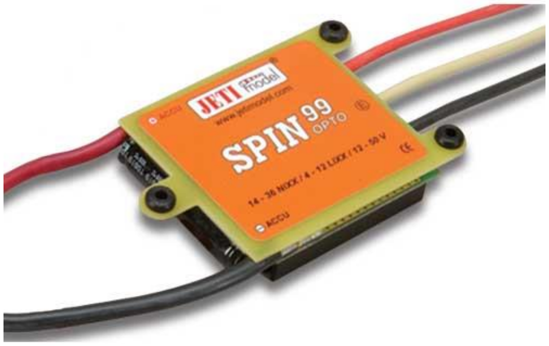


Figure 14: Jeti Spin Pro 99 Opto Brushless ESC.

3.4 Batteries

The batteries used to power the Hacker A-60 and 20x13 APC propellor combination were two Sky Lipo 3300mAh 5 cell Lithium Polymer (LiPo). The batteries were used in series for a total of 10 cells. Fully charged the batteries operated at a total of 42V at a current draw of 60 amps for a maximum power consumption of 2520W.

Figure 15: Sky Lipo 3300mAh 5 cell Lithium Polymer battery.

3.5 Hardware and Sensors

3.5.1 Primary Flight Computer

The primary flight computer is a high performance single board computer, the Athena II SBC from Diamond Systems® shown in Figure 16. The system features a 256MB soldered DRAM for a higher resistance to shocks and vibration and an auto calibrating analog and digital I/O and an 800MHz Pentium® III processor. the system I/O includes an Ethernet port, four RS-232 ports (expanded with the Pandora enclosure), four USB 1.1 ports, Ultra DMA IDE controller with support for two hard disk drives and keyboard and mouse support.

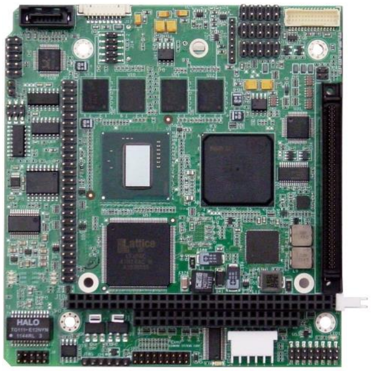


Figure 16: Athena II SBC from Diamond Systems®.

3.5.2 Enclosure

The Pandora enclosure was needed to protect the Athena II SBC and any additional PC/104 boards required to expand Athena's capabilities. Furthermore, the enclosure provided a convenient input/output panel which allowed external devices to be connected to the Athena II with safety and ease, see Figure 17.

Figure 17: Diamond Systems® Athena II Pandora enclosure.

3.5.3 DC/DC Converter

A DC/DC converter to allow the Athena II SBC and sensors to be powered on board the Rascal 110. For this role, the Jupiter-MM-SIO DC/DC converter was chosen (Figure 18). The Jupiter allowed for the Athena's input voltage to be expanded from only 5V DC to a range of 7-30V DC. The expansion module allowed for a 5V 10A output source to be used to power on board sensors.

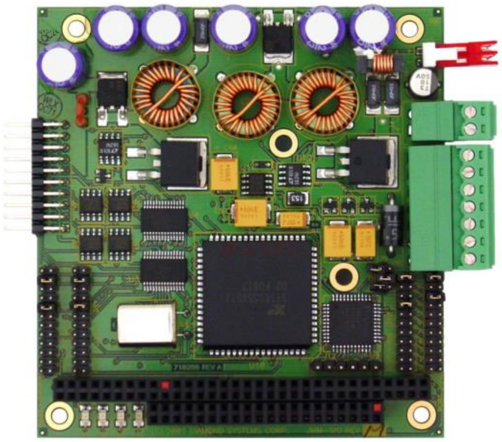


Figure 18: Diamond Systems® Jupiter-MM-SIO DC/DC converter.

3.5.4 Serial Expansion

A serial expansion board was required to accommodate the number of serial based sensors on board the Rascal 110. For this, the Diamond Systems® Emerald-MM 4-channel serial port PC/104 module was chosen (Figure 19). As the title describes, this expansion module provided an additional 4 RS-232 serial ports to be used with the Athena II SBC.

Figure 19: Diamond Systems® Emerald-MM 4-channel serial port expansion module.

3.6 Attitude Heading and Reference System

The attitude heading reference system (AHRS) used in this application was the 3DM-GX3-25 (Figure 20) by MictroStrain®. The 3DM-GX3-25 is a micro-electronic mechanical system or MEMS device. It contains a 3-axis accelerometer, 3-axis, 3-axis magnetometer, temperature sensor and an on-board processor used for sensor fusion. The accelerometers feature a default +/- 5g measurement range of acceleration with options up to 50g. The gyroscope has a default angular rate measurement range of +/- 300^{o}/s and an optional maximum of 1200 ^{o}/s. Finally the magnetometer is able to measure a magnetic field strength of up to +/-2.5Gauss. All of the sensors are sampled with a 16bit Analog to Digital resolution.

Figure 20: 3DM-GX3-25 AHRS by MictroStrain®.

3.7 Global Positioning System

The global positioning system (GPS) used for this application was the u-blox 6 LEA-6H by u-blox (Figure 19).The GPS features a horizontal accuracy of up to 2.5m, a velocity accuracy of up to 0.1m/s and a heading accuracy of 0.5 ^{o}. The GPS has a default sample rate of 5Hz.

Figure 21: LEA-6H GPS by u-blox.

3.8 Ardupilot Mega Sub System

The ArduPilot Mega (APM) 2.5, shown in Figure 22, is a micro controller autopilot system produced by DIY Drones. It was designed to be used as a cheap alternative for flying micro aerial vehicles (MAVs) autonomously. The APM contains both an ATMEGA 2560 16MHz processor and an ATMEGA 32U-2 8MHz processor for normal processing and usb functions respectively. Furthermore, the APM has a host of sensors and auxiliary functions, however, most of these functions are made redundant by better hardware described in this thesis. The APM’s role in this UAV is to read pulse-width (PWM) modulation signals from the remote control receiver described in section I. and to relay static pressure and temperature information from its embedded barometric pressure sensor.

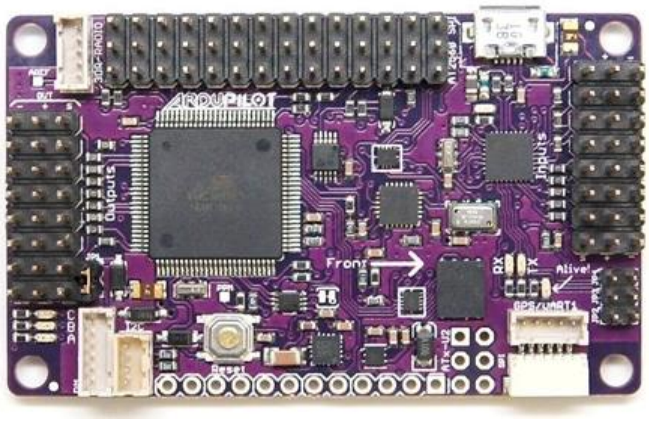


Figure 22: APM 2.5 by DIY Drones.

3.8.1 Barometric Pressure Sensor

The barometric pressure sensor used is the BMP085 digital pressure sensor from Bosch Sensortec, shown in Figure 23. As stated in the previous section this sensor was embedded in the APM 2.5 package. The sensor’s pressure range is from 300 to 11 hPa which equates to 9000m above and 500m below sea level. The temperature range is from 0^{o}C to 65^{o}C.

Figure 23: BMP085 digital pressure sensor by Bosch Sensortec.

3.8.2 Pitot Tube

The differential pressure sensor is the MPXV7002 series piezoelectric transducer by freescale^{TM} semiconductor shown in Figure 24. It has a pressure range of -2.0kPa to 2.0kPa and a sensitivity of 1V/kPa. The differential pressure sensor is used to calculate an estimate of the aircraft’s true airspeed from dynamic pressure.

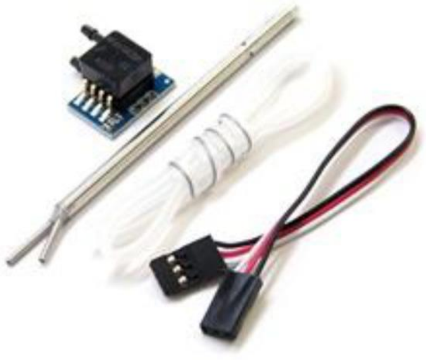


Figure 24: Airspeed Kit with the MPXV7002DP differential pressure sensor by Freescale Semiconductor.

3.9 Wireless Telemetry

The wireless telemetry used to communicate flight data to the ground station was the Xtream-PKG-R RS-232/485 RF Modem by MaxStream, shown in Figure 25. The Xtream has a throughput data rate of up to 9600bps and an interface data rate selectable between 9600bps and 57600bps with a range of up to 5km.

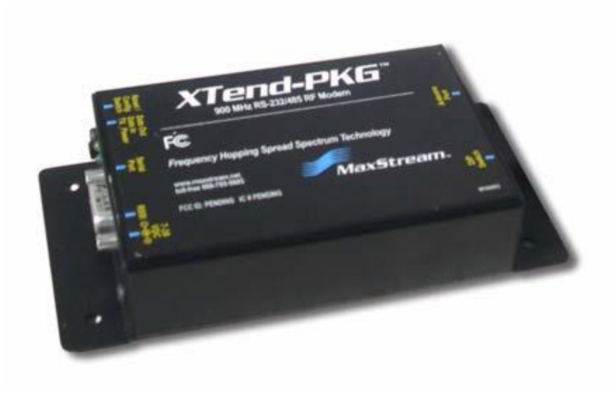

XTend-PKG™
900 MHz RS-232/485 RF Modem
Frequency Hopping Spread Spectrum Technology
MaxStream.

Figure 25: Xtream-PKG-R RS-232/485 RF Modem by MaxStream.

3.10 Remote Control Transmitter and Receiver

The transmitter used to control the Rascal 110 was the DX8 8 channel transmitter by Spektrum, shown in the left hand side of Figure 26. The transmitter transmits commands at a frequency of 2.4Ghz. The receiver used in pair with the DX8 8 channel transmitter was the AR8000 8channel receiver by Spektrum, shown in the right hand side of Figure 26. Once the receiver receives the commands via wireless 2.4GHz radio, it emits a pulse width modulation (PWM) signal to a servomechanism or other PWM reading device.

Figure 26: DX8 Transmitter and AR8000 reciever by Spektrum.

3.11 Servo Controller

A servo controller is necessary to control UAV actuators autonomously. A simple solution for this application is the 24channel Maestro Servo Controller by Polulu shown in Figure 27. The servo controller receives commands via RS-232 serial. It is capable of outputting PWM signals with a resolution of up to 0.25μs to drive the servo actuators on the Rascal 110.

Figure 27: 18 Channel Servo Controller by Polulu.

3.12 Servo Multiplexer

A servo multiplexer was required as a fail-safe hard switch between the pilot's commands and those of the primary flight computer. For this, the 4 Channel Servo Multiplexer by Polulu was chosen, shown in Figure 28. The aircraft required more than 4 channels of PWM for control and switching, therefore two of these devices are used in unison for a total of 8 switchable channels.

Figure 28: 4 Channel Servo Multiplexer by Polulu.

3.13 RS-232 to Transister-Transister Logic Converter

The RS232 to transister-transister logic (TTL) (Figure 29) converter is required to convert the TTL logic of the APM 2.5 and the Polulu servo controller so that they may communicate with the Athena II SBC in the RS-232 serial protocol. TTL is a binary digital logic which converts high and low voltages to data bits read by the device.

Figure 29: RS232 Shifter by Sparkfun.

3.14 Hardware Setup

Figure 30 shows how the hardware is connected to the Athena II SBC.

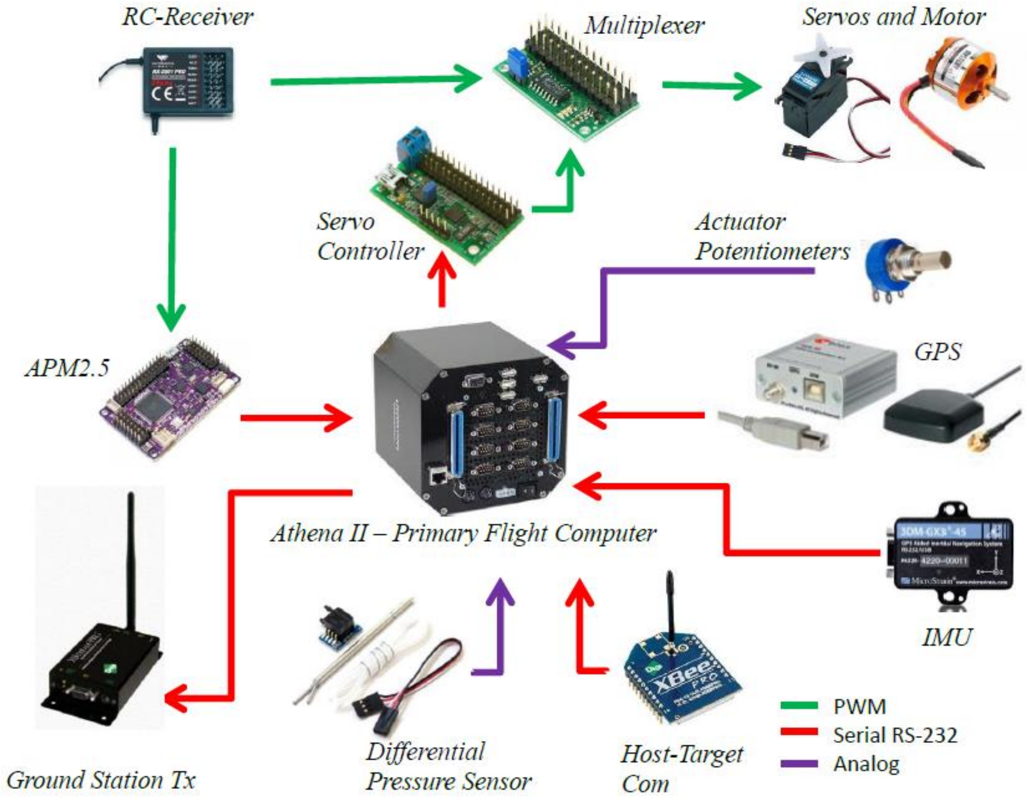


Figure 30: Hardware connection diagram.

4 Flight Simulation Environment

4.1 Aircraft Model

The first step in constructing a preliminary aerodynamic model of the Rascal was to generate its stability derivatives using Digital Datcom. To do this, a geometry based on (Choon Seong, 2008) was used. The MATLAB routine PlotDatcom3d was then used to verify that the geometry matched the actual aircraft. Figure 31 shows a comparison between the Datcom input geometry and the actual aircraft, while Table 2 lists the values of the main parameters used.

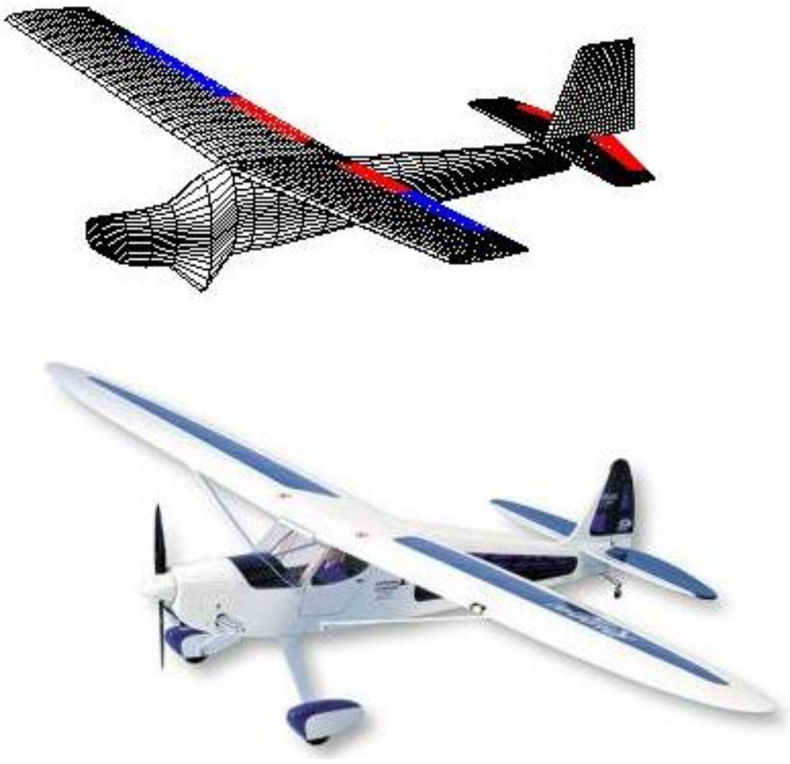


Figure 31: Comparison between the Datcom input and the actual aircraft.

Table 2: Digital Datcom input file parameters.

Parameter	**Value**	**Units**
Speed	68.10	ft/s
Altitude	0	ft
Chord	1.25	ft
Wing Area	10.57	ft^2
Span	9.17	ft
Weight	15.74	lb

It can be seen in Figure 31 that the input geometry approximates the actual aircraft, making it suitable for a preliminary stability analysis. Once this was determined, Digital Datcom was run and the preliminary set of stability derivatives was extracted from the output file. For simplicity, the stability derivatives used were the ones obtained at an angle of attack of 0 ^{o}. Table 3 shows these stability derivatives.

Table 3: Preliminary stability derivatives from Digital Datcom.

Longitudinal Stability Derivatives (per rad)		**Lateral – Directional Stability Derivatives (per rad)**	
C_{L0}	0.4940 rad^{-1}	$C_{y\beta}$	-0.3198 rad^{-1}
$C_{L\alpha}$	5.9730 rad^{-1}	C_{yp}	-0.1138 rad^{-1}
C_{Lq}	4.8850 rad^{-1}	$C_{l\beta}$	-0.1002 rad^{-1}
C_{D0}	0.0310	C_{lp}	-0.5087 rad^{-1}
C_{m0}	0.0323	$C_{n\beta}$	0.0127 rad^{-1}
$C_{m\alpha}$	-0.3217 rad^{-1}	C_{np}	-0.0380 rad^{-1}
C_{mq}	-11.000 rad^{-1}	C_{nr}	-0.0378 rad^{-1}

Another important stability derivative that was not obtained directly from the Digital Datcom output file was the drag curve slope, $C_{D\alpha}$. However, Digital Datcom does provide approximate values of the coefficient of drag at different angles of attack. These values were used to construct the linear region of the drag curve. A trend line was added and the slope was found. Figure 32shows the drag curve constructed. From this plot it was found that the drag curve slope was 0.0092 per ^{o} or 0.527 per rad.

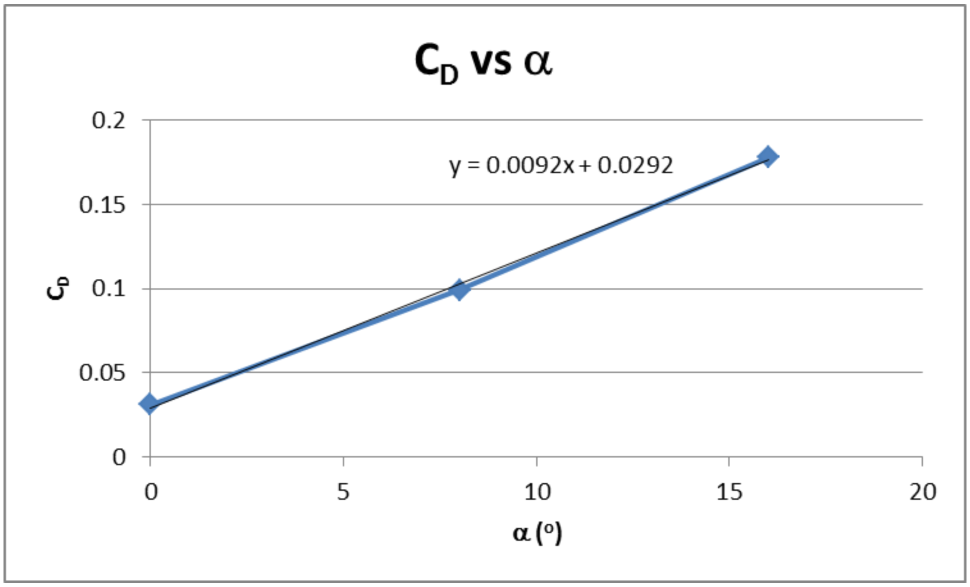


Figure 32: Linear region of the drag curve.

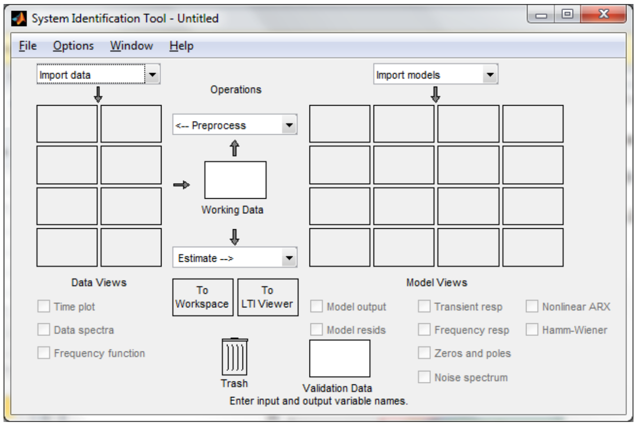
These stability derivatives found were then used in the Simulink model. This model was then trimmed and the appropriate parameters and flight condition listed on Table 4.

Table 4: Trim conditions and parameters.

Parameter	**Value**	**Units**
Altitude	100.00	m
Speed	29.20	m/s
Angle of Attack	-3.80	^{o}
Elevator Deflection	3.49	^{o}
Thrust	35.00	N

4.2 Actuator Model

An essential part of the aircraft model is the actuator model. The actuators were modeled based on a first order time domain transfer function. The actuators (the control surfaces and the electric motor system) received a step input. Figure 34 shows the elevator system response to a step input of 25^{o}. The measured data was logged and placed in MATLAB’s system identification tool shown in Figure 32. The model of the system was estimated in the time domain as a first order linear model. The Laplace transform shown in equation 57 has a time constant of 0.033s (1/30) and a delay of 0.1s. All other control surfaces were assumed to follow the same model.



Enter input and output variable names.

Figure 33: MATLAB’s system identification tool.

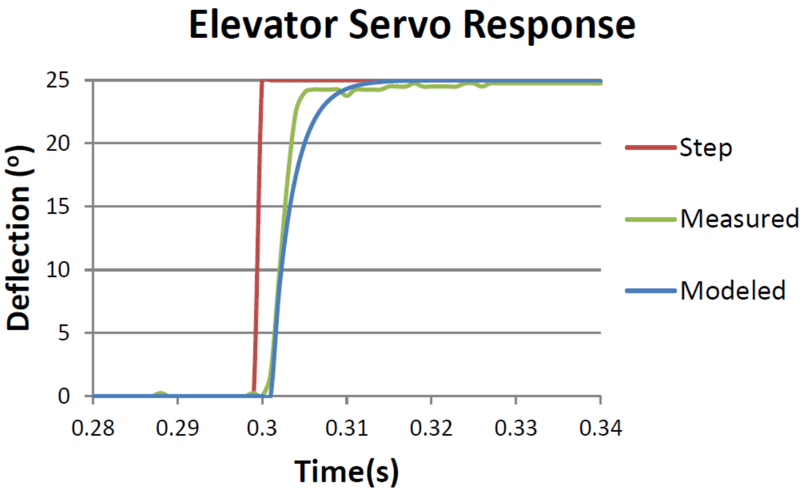


Figure 34: Elevator system response to step input of 25^{o}.

$$\frac{30}{s+30}e^{-0.1s} \tag{57}$$

The aircraft's engine was also modeled as a first order system using MATLAB's system identification tool. The resulting model shown in equation 58 as a Laplace transform has a time constant of 0.0201s (1/49.75) and a delay of 0.1s. A visualization of the experimental data and its model is shown in Figure 35.

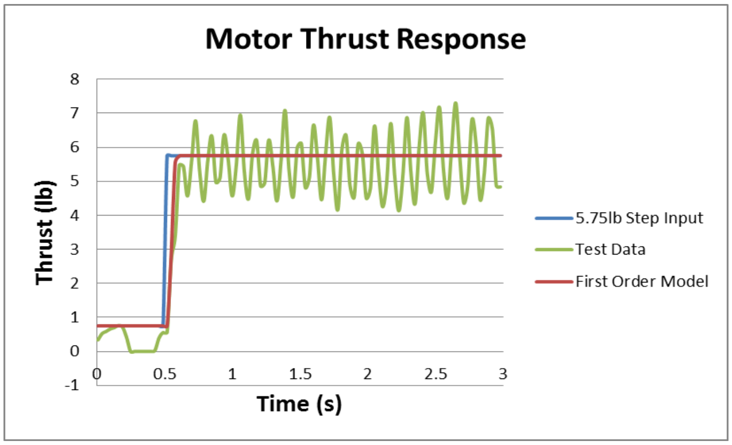


Figure 35: Motor system thrust response to 5.75lb input.

$$\frac{49.75}{s+49.75}e^{-0.1s} \tag{58}$$

4.3 Sensor Model

The sensors on the Sig Rascal aircraft (with exception to the GPS) were modeled as a white noise signal or a signal that follows a zero mean, normal Gaussian distribution. White noise is defined as a random signal whose samples are independent of time. The signal's probability distribution function is shown in equation 59. Using this model only requires the signal variance under operating conditions. Figure 37 shows an example of the white noise present in accelerometer data under operating conditions.

$$w(\mu,\sigma^2)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{59}$$

Where,

x is the input signal

μ is the signal's mean

σ is the signal's standard deviation

In SIMULINK, the implementation of this is simply the linear addition of a noise block shown in Figure 36.

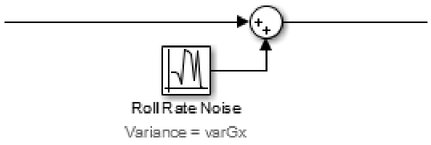


Figure 36: Example of adding white noise to a signal.

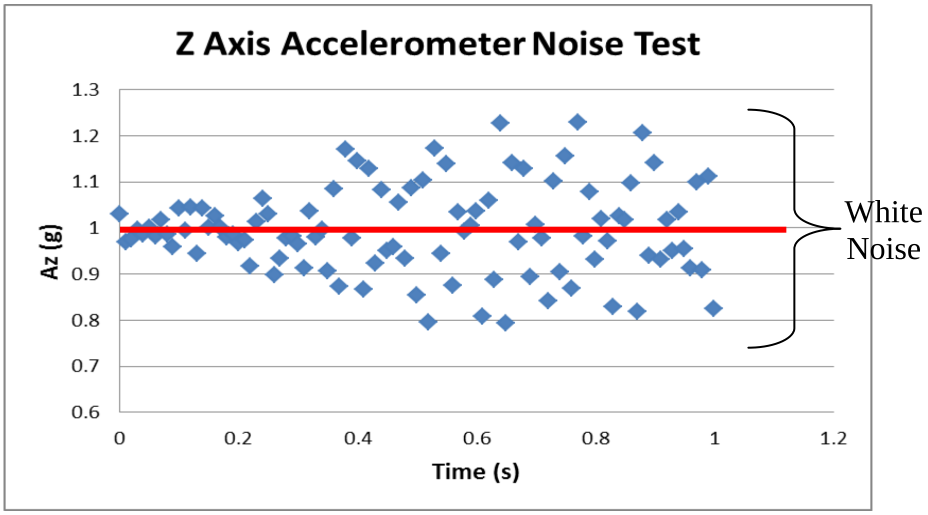


Figure 37: Sample of the Z-axis accelerometer at 1g, white noise with a variance of 0.0124g.

The GPS noise is modeled as colored noise. Colored noise is a random signal whose samples are dependent on time. A GPS's colored noise is brown noise, or random walk. Brown noise is obtained from the integration of white noise over time, expressed in equation 60. The SIMULINK implementation of this is to estimate the brown noise as a Guass-Markov process modeled in equation 60, by performing a discrete integration of the white noise plus an exponential decay according to the time constant of the GPS (Beard & McLain, 2012). Figure 39 shows an example of the brown noise present in GPS measurements.

$$B(t) = e^{K_{GPS}T_s} + w_{GPS} \tag{60}$$

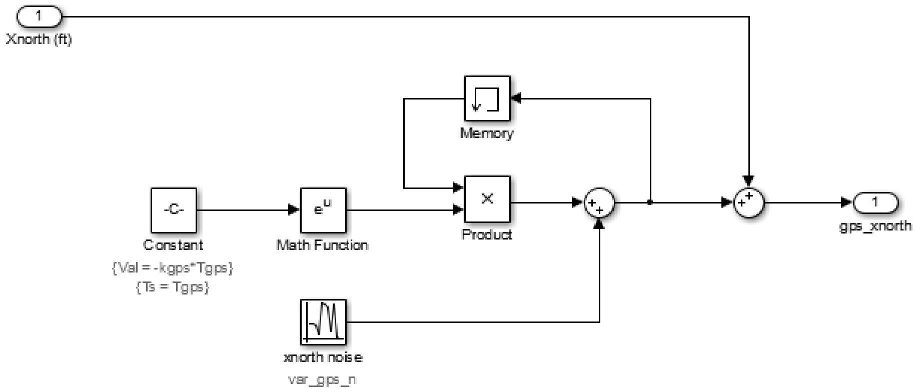


Figure 38: SIMULINK example of adding brown noise (random walk) to the GPS signal.

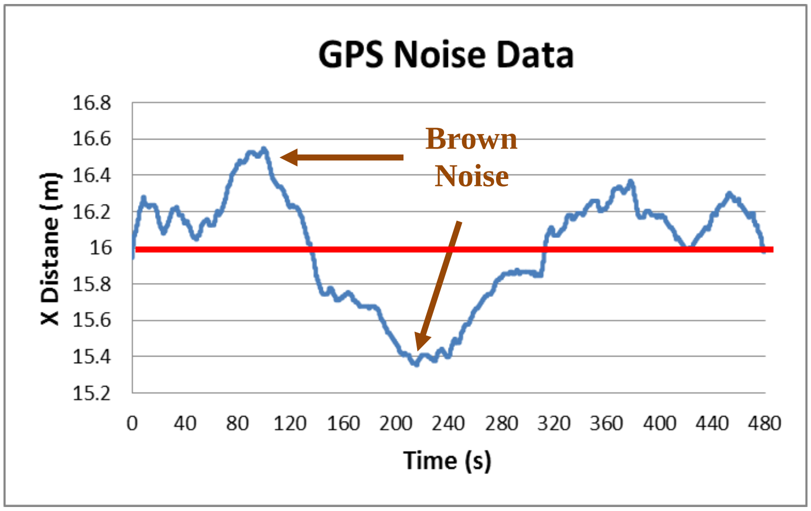
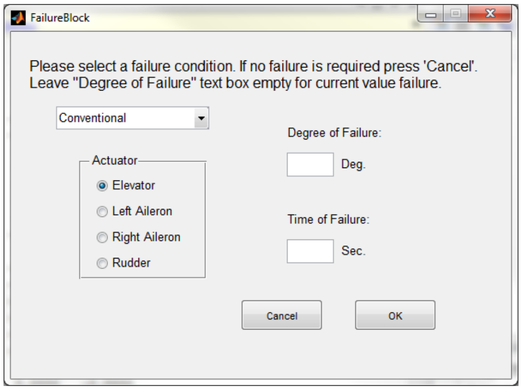


Figure 39: Sample of the GPS X distance at 16m, brown noise with a variance of 0.094m and a time constant of 150s.

4.4 Aircraft Failure Model

Several abnormal conditions (aircraft sub-system failures) have been modeled and implemented within the Rascal 110 simulation environment. Upon loading the simulation model, a GUI allows selection of a control surface failure scenario. The failure setup GUI for the Rascal 110 aircraft is shown in the Figure 40.



Please select a failure condition. If no failure is required press 'Cancel'.
Leave "Degree of Failure" text box empty for current value failure.

Conventional

Actuator
● Elevator
○ Left Aileron
○ Right Aileron
○ Rudder

Degree of Failure: Deg.

Time of Failure: Sec.

Figure 40: Failure selection graphical user interface (GUI).

The GUI provides the ability to impart a failure of any magnitude on any surface of the aircraft at any time the user selects. However, in order to minimize the massive amount of data collected from each flight test, only nominal and two failure conditions are investigated in this thesis as shown in Table 5. The failure conditions occur after 10 seconds from the beginning of the trajectory, when either the left or right aileron is stuck at an angle of 2 ^{o}, 5 ^{o} or 8^{o}.

Table 5: Failure conditions used to test control laws.

Severity	**Left Aileron**	**Right Aileron**
	[degrees]	**[degrees]**
Low	2	2
Medium	5	5
High	8	8

5 Hardware-in-the-Loop Simulation

The HIL simulation is the most important stage of testing control laws before implementation in flight. As such, it is important to have a system that can operate in real-time and a flight computer that can perform the computations required to maintain flight and achieve performance objectives with the designed control laws. Three independent computer systems were used in the HIL simulation as illustrated in Figure 41: the host, the aircraft simulator, and the primary flight controller.

The role of the host computer is to host the control algorithms, the aircraft dynamics model, the aircraft visualization window and to connect to and download code to both targets. The creation of the control algorithms and aircraft dynamics model is done in SIMULINK by MathWorks®. The real time operating system (RTOS) used to execute the code is xPC Target by MathWorks®. Finally, the flight is visualized using the open source tool FlightGear.

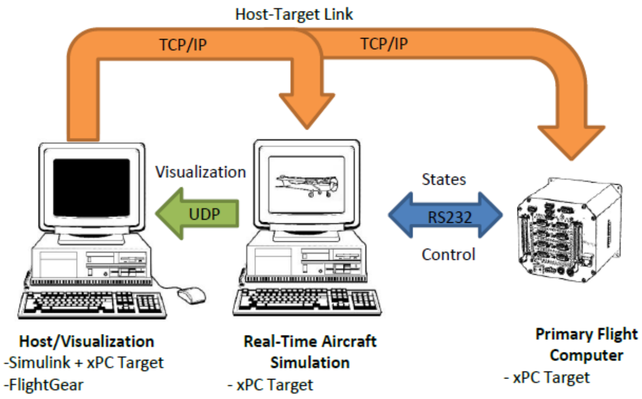


Figure 41: Hardware-in-the-loop simulation setup.

5.1 Real-Time Environment

The real time environment used is the MathWorks® xPC Target system. xPC Target is a real-time operating system that enables the user to load SIMULINK models on to physical systems and execute them in real-time. In the setup shown in Figure 40, the system environment consists of two modules. The first one is a host computer with Matlab/SIMULINK used to create models and the second one is a separated target computer that runs real time applications.

The host computer runs a user generated SIMULINK model, with the xPC Target toolbox active. A C compiler is then called to generate real-time code. Then, the host computer sends this information to both of the computers to execute the code with two separate instances of the xPC Target RTOS pre-loaded. The steps used to accomplish this are shown in the Figure 42 below.

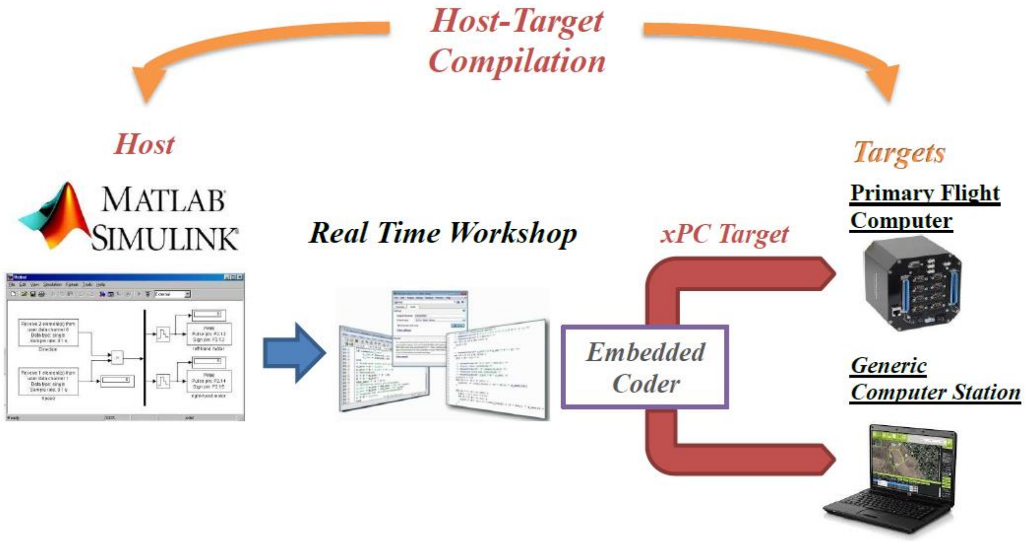


Figure 42: The steps taken by xPC Target to compile simulation on to hardware.

The real-time environment xPC Target provides the resources track and log a parameter called task execution time. According to the xPC Target documentation by MathWorks®, the task execution time is defined as "...an average of the measured CPU times, in seconds, to run the model equations and post outputs during each sample interval. Task execution time is nearly constant, with minor deviations due to cache, memory access, interrupt latency, and multirate model execution". However, during the time in which the processor appears idle, background processes with lower priorities are being completed such as serial communication, TC/IP communication and graphics processing (MathWorks, 2013).

This thesis uses task execution time statistics to be discussed later, along with other metrics, as a benchmark to analyse the performance of the control laws developed.

5.1.1 Host Computer

The host computer is a generic desktop computer which serves as the core functionality in the hardware-in-the-loop simulation setup. It is responsible for establishing contact with both the aircraft simulation and primary flight computers as well as generating and deploying their respective Simulink® models. The final role of the host computer is to visualize the aircraft using FlightGear.

5.1.2 Aircraft Target Computer

The aircraft target computer is a generic desktop computer which its main purpose is to continuously run the aircraft dynamics in real time. Additionally, the aircraft target computer must accept commands from and provide information to the primary flight computer in order to perform close loop operation. This information flow is achieved via serial RS232. Finally, data is sent to the host computer via UDP for FlightGear visualization.

5.1.3 Primary Flight Computer

The primary flight computer's role in the HIL setup is to execute the control law algorithms in real time. The primary flight computer creates a closed loop system with the aircraft target computer via RS-232.

5.2 Real-Time Models

A two-part simulation environment was developed specifically for HIL purposes. The right hand side of Figure 43 is an independent simulation model which holds the aircraft aero and sensors models and is compiled on the target generic computer station. The left hand side of Figure 43 is the other independent model which holds the pre-recorded reference trajectory path and the adaptive control laws. This model is compiled and loaded on to the Athena II SBC primary flight computer.

The aircraft model and the control laws are both run in real-time asynchronously at a clock speed of 0.002*s* or 500Hz. The models were effectively synchronized using a flag sent from the aircraft model to the primary flight computer. This means that the primary flight computer did not begin processing any data until the aircraft model was started and the flag was received. Subsequently, this introduces some lag into the system, however, the test run times were very short (the longest run lasted 265*s*) and did not pose any discernible effects in the system performance.

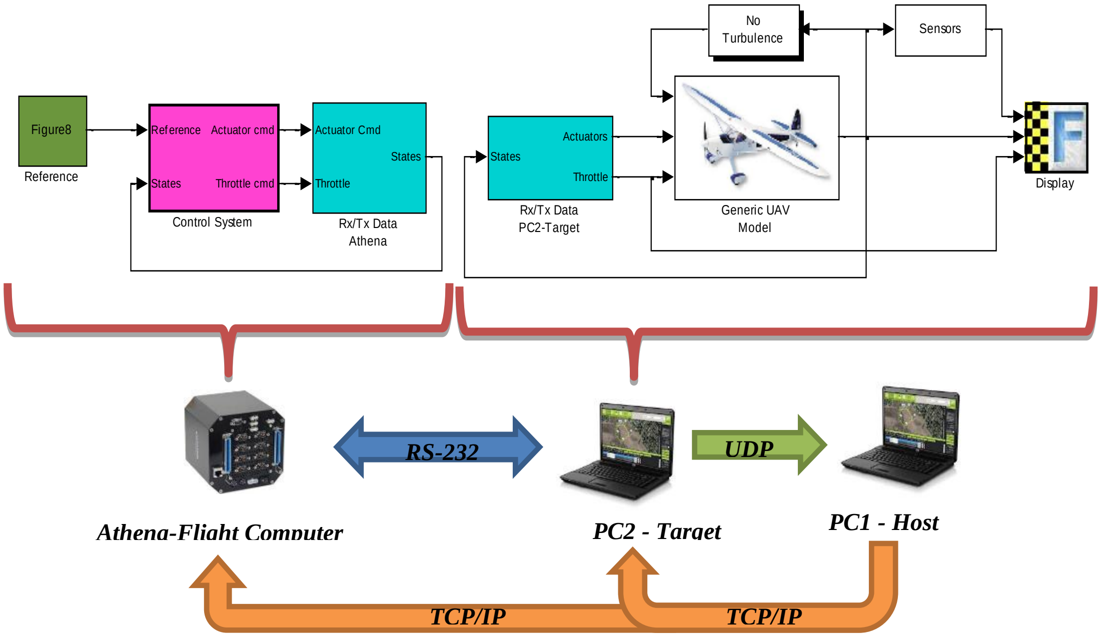
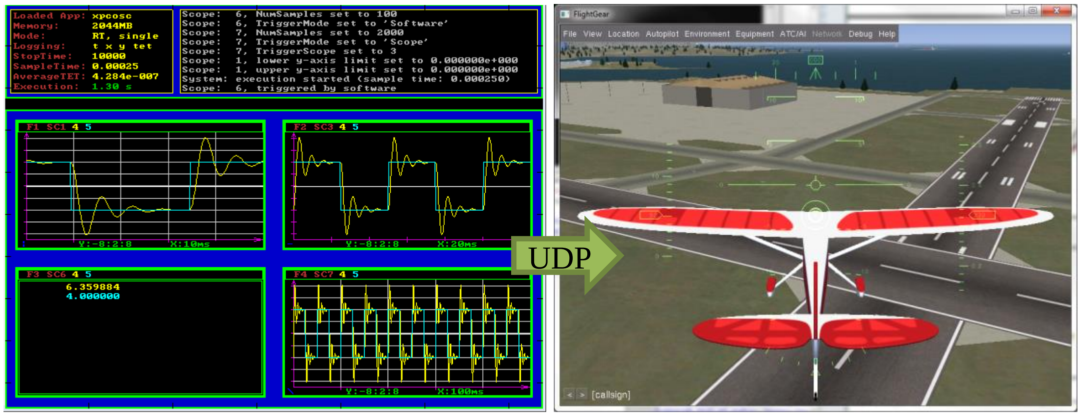


Figure 43: Smulation environment used for hil simulation with compilation scheme and data flow shown.

5.3 Data Communication

The data communication in Figure 43 is established in a simple manner. First, the host loads the Simulink® block diagrams to both of the target computers via ethernet cables using the TCP/IP protocol. Once started from the host, the aircraft simulation computer exchanges data with the primary flight computer via a serial link established between the two. The signals sent from aircraft computer to the primary flight computer are only the states required to allow the control laws to perform their task. The data sent from the primary flight computer to the aircraft computer are the control actuation commands required for the aircraft simulation to operate. Simultaneously, the aircraft computer sends selected state data via UDP to the host computer to be visualized in FlightGear. Figure 44 shows this scheme.



```
Loaded App: xpcosc
Memory:     2044MB
Mode:       RT, single
Logging:    t x y tet
StopTime:   10000
SampleTime: 0.00025
AverageTET: 4.284e-007
Execution:  1.30 s
```

```
Scope:  6, NumSamples set to 100
Scope:  6, TriggerMode set to 'Software'
Scope:  7, NumSamples set to 2000
Scope:  7, TriggerMode set to 'Scope'
Scope:  7, TriggerScope set to 3
Scope:  1, lower y-axis limit set to 0.000000e+000
Scope:  1, upper y-axis limit set to 0.000000e+000
System: execution started (sample time: 0.000250)
Scope:  6, triggered by software
```

xPC Target with Rascal 110 Model FlightGear with Rascal 110 Model

Figure 44: xPC Target and FlightGear displaying Rascal 110.

Testing and Performance Analysis of the Control Laws

To demonstrate the functionality of the control algorithms discussed in this thesis, a series of simulation tests were performed over a variety of flight scenarios at normal and abnormal conditions. These flight scenarios were then compared over two pre-recorded trajectories, a three dimensional figure-8 (Figure 44) and a landing (Figure 45).

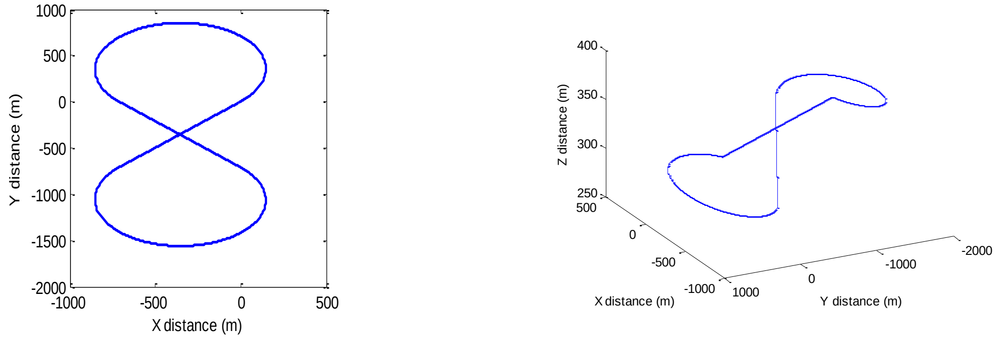


Figure 45: 3-Dimensional Figure-8 trajectory.

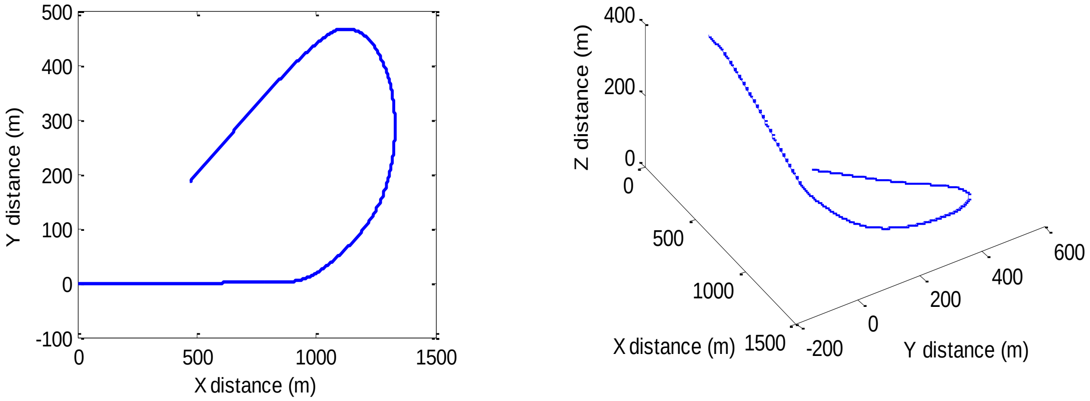


Figure 46: Landing trajectory.

The HIL simulation was setup as shown in Figure 46. The fundamental frequency was set to be 500*Hz* on both the aircraft and primary flight computers. Both models were then loaded on to their respective computers, and the process as outlined in Chapter 5.

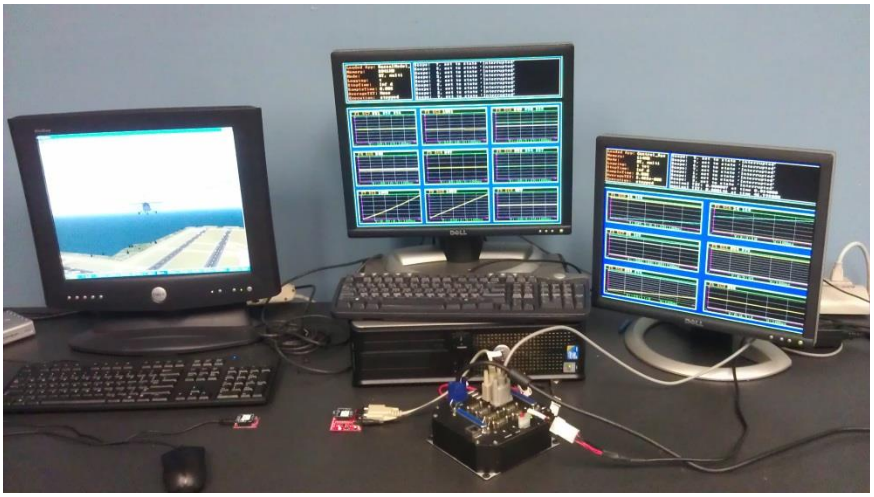


Figure 47: HIL setup with the host, aircraft computer, and primary flight computer respectively.

5.4 Performance Metrics

The performance evaluation criteria are based upon trajectory tracking error, control activity and task execution time. For the trajectory tracking performance, the root of the squared error on the horizontal, vertical and combined trajectory are calculated. Then, a trajectory tracking specific performance vector (Moncayo H., et. al, 2012) can be defined as:

$$\begin{aligned} PV_{TT} &= \left[tt_i \quad | \quad i = 1, 2, \ldots, 9 \right] \\ &= \left[\overline{e}_{XY} \quad \overline{e}_{Z} \quad \overline{e}_{XYZ} \quad e_{\max XY} \quad e_{\max Z} \quad e_{\max XYZ} \quad \hat{e}_{XY} \quad \hat{e}_{Z} \quad \hat{e}_{XYZ} \right]^T \end{aligned} \tag{61}$$

Where,

$\overline{e}_k(t)$ is the absolute mean along the kth trajectory.

$e_{\max k}(t)$ is the absolute maximum along the kth trajectory.

$\hat{e}_k(t)$ is the standard deviation along the kth trajectory.

It is desirable for the trajectory tracking algorithm to supply commands which are gradual and do not cause the control surfaces to saturate. If the total duration of the flight simulation is *T*, the following two control activity-related evaluation parameters can be defined (Moncayo, Perhinischi, et. al, 2012):

The integral of control command/deflection rate of change

$$I\dot{\delta}_c = \frac{1}{T}\int_0^T \left|\dot{\delta}_c\right| dt \tag{62}$$

and the control command/deflection saturation index

$$S_{\delta_c} = \frac{100}{T}\int_0^T \tilde{\delta}_c(t)\, dt \tag{63}$$

and

$$\tilde{\delta}_c(t) = \begin{cases} 0 & \delta_c < \delta_{c\max} \\ 1 & \delta_c > \delta_{c\max} \end{cases} \tag{64}$$

Where,

δ_c the control command/deflection of the relevant control surface.

In this case the relevant deflections are from the elevator δ_e, aileron δ_a and the rudder δ_r. The relevant control command is the throttle command δ_t.

Let a control activity specific performance vector be defined as (Moncayo H., et. al, 2012):

$$\begin{aligned} PV_{CA} &= \left[ca_i \quad | \quad i = 1,2,...,8 \right] \\ &= \left[I\dot{\delta}_e \quad I\dot{\delta}_a \quad I\dot{\delta}_r \quad I\dot{\delta}_t \quad S_{\delta_e} \quad S_{\delta_a} \quad S_{\delta_r} \quad S_{\delta_t} \right]^T \end{aligned} \tag{65}$$

Finally, it is important that the flight computer does not use an excessive amount of task execution time for a given control law algorithm.

Let the task execution time be denoted as $\Upsilon(t)$. The following task execution time evaluation parameters are defined:

- Average task execution time:

$$\bar{\Upsilon}(t) = mean\left(\Upsilon(t)\right) \tag{66}$$

- Maximum task execution time:

$$\Upsilon_{\max}(t) = \max\left(\Upsilon(t)\right) \tag{67}$$

- Standard deviation of the task execution time:

$$\hat{\Upsilon}(t) = STD\left(\Upsilon(t)\right) \tag{68}$$

Let a task execution time specific performance vector be defined as:

$$PV_{TET} = \left[tet_i \quad | \quad i = 1, 2, 3\right] = \left[\bar{\Upsilon} \quad \Upsilon_{max} \quad \hat{\Upsilon}\right]^T \tag{69}$$

In this paper, the six trajectory tracking algorithms were compared over two trajectories at many operating conditions. Consequently, an evaluation scheme is needed to reduce the vast performance data into a simplified and meaningful metric. For this purpose, a performance index is formulated for each tracking algorithm, based upon a weighted sum of the normalized components of each of the various performance parameters.

A tracking performance index is computed as a weighted average of the individual tracking error components, with weights assigned based upon subjective relative importance. Likewise, a surface activation performance index is computed as a weighted average of the individual surface parameters, also with weights subjectively assigned. A total performance index for the tracking algorithm test combines the tracking error and surface activation performance indices using another weighted average . This is summarized in the following equations (Moncayo, Perhinischi, et. al, 2012):

A trajectory tracking specific performance index can be defined as:

$$PI_{TT} = w_{TT} \cdot PV_{TT} \tag{70}$$

A control activity specific performance index can be defined as:

$$PI_{CA} = w_{CA} \cdot PV_{CA} \tag{71}$$

A task execution time performance index can be defined as:

$$PI_{TET} = w_{TET} \cdot PV_{TET} \tag{72}$$

Finally, a trajectory tracking global performance index can be defined as:

$$PI_{UAV} = \overline{w}_{TT} \cdot PV_{TT} + \overline{w}_{CA} \cdot PV_{CA} + \overline{w}_{TET} \cdot PV_{TET} \tag{73}$$

Note that w_{TT} , w_{CA} , w_{TET} , $\overline{w}_{TT}$, $\overline{w}_{CA}$ and $\overline{w}_{TET}$ are normalization and desirability weights.

5.5 Control Laws Performance Analysis

The nominal and upset condition performance metrics of the six controllers were tested in HIL simulation using the two different pre-recorded trajectories described in the previous section. For these trajectories, a nominal test was conducted, as well as stuck surface failures on both ailerons. For the stuck surface failures, a given surface was locked at the specified deflection 10 seconds after the beginning of the test.

From these tests, the analysis resulted in a large amount of data which were condensed using the performance indices described in the section on Performance Metrics. Once all of the data were collected, they were normalized, placed in performance vectors and performance indices were computed using the normalization cut-off values and desirability weights provided in the tables below.

The global weights were chosen to penalize the trajectory tracking the most and the task execution time (TET) performance the least. This configuration was chosen because after many tests, it was concluded that the control algorithms' TET did not saturate the fundamental sample time of the simulation of 0.002s. The slowest algorithm had a TET of approximately 0.00025s or 12.5% or the fundamental sample time. Therefore, an algorithm performing poorly with respect to all the others (in terms of TET) would not have an effect on its overall performance.

The normalization cutoffs were chosen based on a worse case scenario of the worst control algorithm's performance. However, the values had to be cut off at a reasonable point. For example, the maximum XY cutoff was chosen to be 100 meters because five of the six controllers were within this range. The sixth controller's performance index would be set to 0 since its performance lied outside of this range.

Table 6: Performance index weights and normalization cut-offs for trajectory tracking.

	Trajectory Tracking Performance									Global PI Weight
	Max			Mean			Standard Deviation			
	XY	Z	XYZ	XY	Z	XYZ	XY	Z	XYZ	
Norm Cut-Off	100	40	100	80	20	80	20	10	20	
w_{TT}	0.10	0.10	0.10	0.20	0.20	0.20	0.03	0.03	0.03	
$\overline{w}_{TT}$										0.75

Table 7: Performance index weights and normalization cut-offs for control activity.

	Control Activity Performance								Global PI Weight
	Surface Activation Index				Saturation Index				
	Elevator	Aileron	Rudder	Throttle	Elevator	Aileron	Rudder	Throttle	
Norm Cut-Off	2.00	2.00	2.00	75	50	50	50	50	
w_{CA}	0.15	0.30	0.20	0.15	0.05	0.05	0.05	0.05	
$\overline{w}_{CA}$									0.20

Table 8: Performance index weights and normalization cut-offs for task execution time.

	Task Execution Time Performance			Global PI Weight
	Max	Mean	Standard Deviation	
Norm Cut-Off	2.00E-03	1.00E-03	5.00E-03	
w_{TET}	0.35	0.50	0.15	
$\overline{w}_{TET}$				0.05

The following set of tables are a condensed form of the metrics chosen to represent the total performance of the control algorithms under nominal and failure conditions. The data is divided into the two trajectories described in the previous section, Figure 8 and Landing respectively.

5.5.1 Performance Data

Figure 8 Performance Summary

Table 9: Figure 8 trajectory tracking metrics.

Failure	Control Algorithm	Max			Mean			Standard Deviation		
		XY	**Z**	**XYZ**	**XY**	**Z**	**XYZ**	**XY**	**Z**	**XYZ**
		[m]	**[m]**	**[m]**	**[m]**	**[m]**	**[m]**	**[m]**	**[m]**	**[m]**
Left Aileron Stuck at 2°	PID	283.97	10.95	284.01	115.90	5.30	116.11	58.08	2.32	57.95
	NLDI	85.59	23.92	86.15	77.65	8.02	78.15	11.20	3.30	11.08
	Extended NLDI	78.68	2.29	78.69	67.50	0.56	67.51	11.12	0.43	11.12
	AIS PID	46.55	1.28	46.57	19.13	0.46	19.14	8.17	0.33	8.16
	AIS NLDI	47.53	3.06	47.61	25.82	0.62	25.83	3.55	0.45	3.55
	AIS Extended NLDI	26.96	2.29	26.96	24.57	0.56	24.58	1.75	0.43	1.74
Left Aileron Stuck at 5°	PID	263.01	11.26	263.03	110.77	5.70	111.03	53.34	4.42	53.29
	NLDI	88.00	23.94	88.92	78.62	7.87	79.22	12.14	5.78	12.15
	Extended NLDI	78.68	2.29	78.69	67.48	0.56	67.49	11.11	0.43	11.11
	AIS PID	46.55	1.28	46.57	20.09	0.49	20.10	7.98	0.36	7.97
	AIS NLDI	47.94	3.06	48.02	25.79	0.73	25.81	3.62	0.53	3.63
	AIS Extended NLDI	26.96	2.29	26.96	24.56	0.56	24.57	1.73	0.43	1.73
Left Aileron Stuck at 8°	PID	231.38	16.28	231.38	105.81	7.07	106.24	48.03	6.26	47.99
	NLDI	90.72	24.25	91.99	78.38	9.24	79.22	13.16	7.51	13.46
	Extended NLDI	78.63	2.29	78.63	67.47	0.56	67.47	11.11	0.43	11.11
	AIS PID	46.48	1.29	46.50	21.48	0.56	21.49	8.14	0.38	8.14
	AIS NLDI	47.40	3.06	47.48	25.79	0.87	25.81	3.57	0.67	3.58
	AIS Extended NLDI	26.94	2.29	26.94	24.53	0.56	24.55	1.71	0.43	1.71
Right Aileron Stuck at 2°	PID	297.50	11.93	297.53	111.26	5.20	111.59	61.81	1.79	61.47
	NLDI	84.18	23.92	85.18	77.71	7.98	78.19	10.45	2.72	10.27
	Extended NLDI	78.63	2.29	78.63	67.47	0.56	67.47	11.10	0.43	11.10
	AIS PID	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	AIS NLDI	47.40	3.06	47.48	25.71	0.67	25.72	3.52	0.33	3.53
	AIS Extended NLDI	26.94	2.29	26.94	24.54	0.56	24.55	1.71	0.43	1.71
Right Aileron Stuck at 5°	PID	298.28	12.89	298.29	102.40	5.64	103.01	63.62	3.51	62.99
	NLDI	86.52	23.95	88.13	77.84	8.36	78.40	10.20	3.90	10.09
	Extended NLDI	78.60	2.29	78.60	67.47	0.56	67.47	11.08	0.43	11.08
	AIS PID	46.54	1.30	46.55	19.09	0.48	19.10	8.94	0.29	8.94
	AIS NLDI	47.99	3.06	48.07	25.71	0.73	25.73	3.61	0.39	3.61
	AIS Extended NLDI	26.90	2.29	26.90	24.55	0.56	24.56	1.71	0.43	1.70
Right Aileron Stuck at 8°	PID	287.81	15.00	287.81	85.14	6.55	86.40	64.77	5.71	63.67
	NLDI	88.87	24.25	91.35	77.98	9.13	78.72	10.31	5.83	10.39
	Extended NLDI	78.59	2.29	78.59	67.46	0.56	67.47	11.07	0.43	11.07
	AIS PID	46.55	1.28	46.57	20.13	0.52	20.14	9.45	0.28	9.45
	AIS NLDI	47.99	3.06	48.07	25.74	0.83	25.76	3.60	0.53	3.61
	AIS Extended NLDI	26.94	2.29	26.94	24.53	0.56	24.54	1.71	0.43	1.71
Nominal	PID	279.61	12.13	279.65	106.18	6.39	106.57	56.55	1.76	56.22
	NLDI	85.78	23.94	86.37	78.58	9.20	79.19	10.80	3.04	10.68
	Extended NLDI	78.74	2.30	78.74	67.49	0.56	67.49	11.11	0.43	11.11
	AIS PID	46.39	1.28	46.41	19.08	0.47	19.10	8.35	0.31	8.33
	AIS NLDI	47.92	3.06	48.00	25.86	0.60	25.87	3.62	0.45	3.63
	AIS Extended NLDI	26.94	2.29	26.94	24.54	0.56	24.55	1.71	0.43	1.71

Table 10: Figure 8 control activity metrics.

Failure	Control Algorithm	Integral of Control Surface Rate of Change				Saturation Index			
		Elevator	Aileron	Rudder	Throttle	Elevator	Aileron	Rudder	Throttle
		[rad]	[rad]	[rad]	[%]	[%]	[%]	[%]	[%]
Left Aileron Stuck at 2°	PID	0.0665	0.8509	0.2878	0.17	0.00	0.00	0.00	0.00
	NLDI	0.0976	0.7890	0.2461	32.79	0.00	0.00	0.00	0.55
	Extended NLDI	0.6623	0.1228	0.0190	28.49	0.00	0.00	0.00	0.00
	AIS PID	0.2382	0.2106	0.0065	17.83	0.00	0.00	0.00	0.00
	AIS NLDI	0.2556	0.3955	0.0393	45.01	0.00	0.00	0.00	0.00
	AIS Extended NLDI	0.6684	0.6210	0.0826	45.77	0.00	0.00	0.00	0.00
Left Aileron Stuck at 5°	PID	0.0676	0.8443	0.2782	0.17	0.00	0.00	0.00	0.00
	NLDI	0.0990	0.7482	0.2441	32.48	0.00	0.07	0.00	0.55
	Extended NLDI	0.6939	0.1259	0.0194	28.40	0.00	0.00	0.00	0.00
	AIS PID	0.2353	0.2019	0.0062	17.21	0.00	0.00	0.00	0.00
	AIS NLDI	0.2807	0.3909	0.0373	48.09	0.00	0.00	0.00	0.00
	AIS Extended NLDI	0.7134	0.6231	0.0855	47.06	0.00	0.00	0.00	0.00
Left Aileron Stuck at 8°	PID	0.0600	0.6067	0.1859	0.17	0.00	0.10	0.00	0.00
	NLDI	0.1043	0.8449	0.2525	35.29	0.00	0.31	0.00	0.58
	Extended NLDI	0.6894	0.1312	0.0199	28.15	0.00	0.00	0.00	0.00
	AIS PID	0.2382	0.2119	0.0064	17.87	0.00	0.00	0.00	0.00
	AIS NLDI	0.2582	0.3555	0.0233	43.65	0.00	0.04	0.00	0.00
	AIS Extended NLDI	0.6945	0.6258	0.0875	46.58	0.00	0.00	0.00	0.00
Right Aileron Stuck at 2°	PID	0.0659	0.8039	0.2675	0.17	0.00	0.00	0.00	0.00
	NLDI	0.1081	0.8720	0.2854	36.74	0.00	0.00	0.00	0.55
	Extended NLDI	0.6879	0.1250	0.0211	28.42	0.00	0.00	0.00	0.00
	AIS PID	0.0000	0.2192	0.0065	18.71	0.00	0.00	0.00	0.00
	AIS NLDI	0.2693	0.3986	0.0277	46.58	0.00	0.00	0.00	0.00
	AIS Extended NLDI	0.6560	0.6154	0.0812	43.04	0.00	0.00	0.00	0.00
Right Aileron Stuck at 5°	PID	0.0588	0.6896	0.2222	0.17	0.00	0.00	0.00	0.00
	NLDI	0.1127	0.9391	0.2989	38.85	0.00	0.00	0.00	0.54
	Extended NLDI	0.6354	0.1149	0.0172	28.14	0.00	0.00	0.00	0.00
	AIS PID	0.2359	0.2270	0.0069	18.50	0.00	0.00	0.00	0.00
	AIS NLDI	0.2468	0.3460	0.0234	42.31	0.00	0.00	0.00	0.00
	AIS Extended NLDI	0.6544	0.6183	0.0806	43.95	0.00	0.00	0.00	0.00
Right Aileron Stuck at 8°	PID	0.0655	0.7076	0.2023	0.17	0.00	0.00	0.00	0.00
	NLDI	0.0970	0.8222	0.2495	34.18	0.00	0.00	0.00	0.55
	Extended NLDI	0.6448	0.1169	0.0172	28.11	0.00	0.00	0.00	0.00
	AIS PID	0.2336	0.2242	0.0067	18.30	0.00	0.00	0.00	0.00
	AIS NLDI	0.2397	0.3521	0.0229	41.98	0.00	0.00	0.00	0.00
	AIS Extended NLDI	0.7063	0.6220	0.0846	45.71	0.00	0.00	0.00	0.00
Nominal	PID	0.0667	0.8268	0.2665	0.17	0.00	0.00	0.00	0.00
	NLDI	0.1108	0.9419	0.2903	39.45	0.00	0.00	0.00	0.54
	Extended NLDI	0.7214	0.1297	0.0199	28.22	0.00	0.00	0.00	0.00
	AIS PID	0.2272	0.1917	0.0065	17.19	0.00	0.00	0.00	0.00
	AIS NLDI	0.2466	0.3614	0.0303	41.76	0.00	0.00	0.00	0.00
	AIS Extended NLDI	0.7392	0.6218	0.0840	47.22	0.00	0.00	0.00	0.00

Table 11: Figure 8 task execution time metrics.

Failure	Control Algorithm	Max	Mean	Standard Deviation
		$\times 10^{-5}$[s]	$\times 10^{-5}$[s]	$\times 10^{-5}$[s]
Left Aileron Stuck at 2°	PID	13.22	2.36	0.92
	NLDI	15.25	2.93	0.99
	Extended NLDI	21.10	3.60	1.00
	AIS PID	16.75	3.49	1.02
	AIS NLDI	17.06	4.11	1.05
	AIS Extended NLDI	25.32	6.52	3.49
Left Aileron Stuck at 5°	PID	13.16	2.39	0.88
	NLDI	14.22	2.94	0.93
	Extended NLDI	16.40	3.62	1.09
	AIS PID	16.60	3.48	1.10
	AIS NLDI	19.21	4.09	1.07
	AIS Extended NLDI	25.62	6.50	3.45
Left Aileron Stuck at 8°	PID	13.54	2.35	0.82
	NLDI	13.33	2.96	0.98
	Extended NLDI	15.51	3.61	1.04
	AIS PID	16.86	3.53	1.05
	AIS NLDI	16.83	4.04	1.15
	AIS Extended NLDI	24.99	6.48	3.39
Right Aileron Stuck at 2°	PID	13.65	2.39	0.89
	NLDI	14.44	2.84	0.94
	Extended NLDI	14.84	3.62	1.00
	AIS PID	18.75	3.60	1.05
	AIS NLDI	20.19	4.15	1.15
	AIS Extended NLDI	23.23	6.36	3.09
Right Aileron Stuck at 5°	PID	17.40	2.36	0.86
	NLDI	13.63	2.86	0.89
	Extended NLDI	17.74	3.62	0.99
	AIS PID	16.18	3.54	1.06
	AIS NLDI	16.22	4.11	1.10
	AIS Extended NLDI	24.74	6.38	3.25
Right Aileron Stuck at 8°	PID	13.52	2.37	0.88
	NLDI	13.13	2.82	0.90
	Extended NLDI	15.32	3.63	1.04
	AIS PID	15.81	3.54	1.02
	AIS NLDI	17.43	4.11	1.14
	AIS Extended NLDI	27.43	6.38	3.08
Nominal	PID	13.96	2.40	0.88
	NLDI	15.07	2.87	0.93
	Extended NLDI	14.45	3.59	0.97
	AIS PID	16.74	3.49	0.99
	AIS NLDI	15.42	4.08	1.19
	AIS Extended NLDI	24.79	6.37	3.17

Table 12: Figure 8 performance index (PI) data.

Failure	Control Algorithm	PI_{TT}	PI_{CA}	PI_{TET}	Global PI
Left Aileron Stuck at 2°	PID	0	0.84	0.96	0
	NLDI	0.25	0.78	0.96	0.39
	Extended NLDI	0.46	0.87	0.94	0.56
	AIS PID	0.77	0.91	0.95	0.81
	AIS NLDI	0.75	0.83	0.95	0.77
	AIS Extended NLDI	0.80	0.76	0.92	0.80
Left Aileron Stuck at 5°	PID	0	0.84	0.96	0
	NLDI	0.23	0.79	0.96	0.38
	Extended NLDI	0.46	0.87	0.95	0.56
	AIS PID	0.77	0.92	0.95	0.81
	AIS NLDI	0.75	0.82	0.95	0.77
	AIS Extended NLDI	0.80	0.75	0.92	0.80
Left Aileron Stuck at 8°	PID	0	0.89	0.96	0.01
	NLDI	0.20	0.77	0.96	0.35
	Extended NLDI	0.46	0.87	0.95	0.56
	AIS PID	0.76	0.91	0.95	0.80
	AIS NLDI	0.75	0.84	0.95	0.77
	AIS Extended NLDI	0.80	0.75	0.92	0.80
Right Aileron Stuck at 2°	PID	0	0.85	0.96	0
	NLDI	0.26	0.76	0.95	0.39
	Extended NLDI	0.46	0.87	0.96	0.56
	AIS PID	0.00	0.80	0.95	0.00
	AIS NLDI	0.75	0.82	0.94	0.77
	AIS Extended NLDI	0.80	0.76	0.93	0.80
Right Aileron Stuck at 5°	PID	0	0.87	0.96	0
	NLDI	0.24	0.74	0.96	0.38
	Extended NLDI	0.46	0.88	0.95	0.56
	AIS PID	0.77	0.91	0.95	0.81
	AIS NLDI	0.75	0.84	0.95	0.78
	AIS Extended NLDI	0.80	0.76	0.92	0.80
Right Aileron Stuck at 8°	PID	0	0.87	0.96	0
	NLDI	0.22	0.78	0.96	0.37
	Extended NLDI	0.46	0.88	0.95	0.56
	AIS PID	0.77	0.91	0.95	0.80
	AIS NLDI	0.75	0.84	0.95	0.78
	AIS Extended NLDI	0.80	0.75	0.92	0.80
Nominal	PID	0	0.84	0.96	0
	NLDI	0.24	0.74	0.96	0.37
	Extended NLDI	0.46	0.87	0.96	0.56
	AIS PID	0.77	0.92	0.95	0.81
	AIS NLDI	0.75	0.84	0.95	0.78
	AIS Extended NLDI	0.80	0.75	0.92	0.80

The following is a summary of the performance data for the Figure 8 trajectory for control laws ordered in increasing complexity. The summary is a direct average of all of the performance indexes over all of the failure conditions. Firstly, it should be noted that the baseline PID controller had a PI_{TT} of 0. This means that the PID control algorithm was unable to complete the trajectory within the parameters set by Table 6. However, the artificial immune system did ensure the PID was able to compensate for failires much better than it's baseline counterpart, as with all the other control algorithms. The trajectory tracking performance index has a very clear trend, the performance index gets better with each successive level of non-linear dynamic inversion.

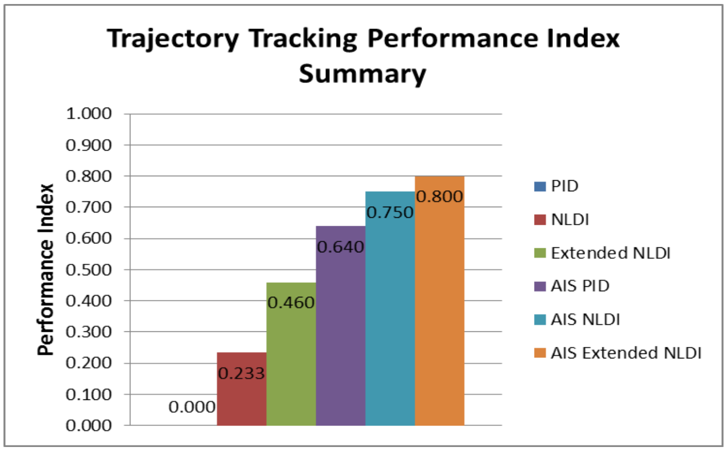


Figure 48: Summary of Figure 8 trajectory PI_{TT} data.

The PI_{CA} varies significantly more than the PI_{TT} . However, a trend is clear, on average the PI_{CA} decreases with algorithm complexity. This is because, on average, each successive dynamic inversion uses more control actuation in order to follow the trajectory.

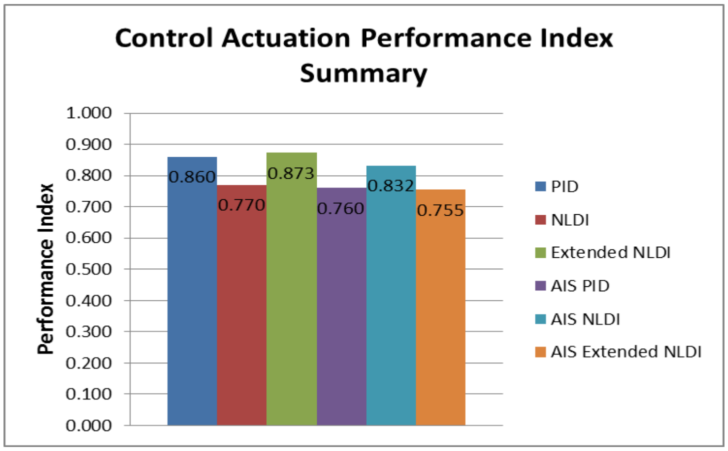


Figure 49: Summary of Figure 8 trajectory PI$_{CA}$ data.

The PI_{TET} has an inverse relationship with the PI_{TT} , it steadily decreases with increasing control algorithm complexity. This trend is due to the fact each successive step of dynamic inversion adds additional equations which requires more processing power and hence a longer task execution time. However, one should note that in each case, the AIS performed as well or worse than the unaugmented systems. This suggests the AIS feedback mechanism requires a significant amount of processing when compensating for failures.

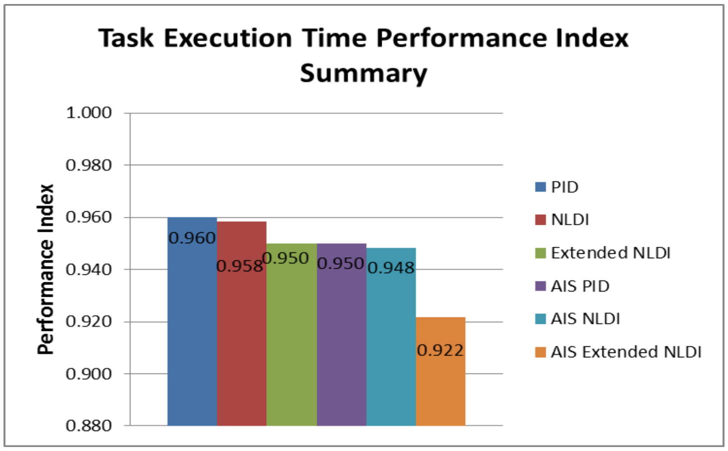


Figure 50: Summary of Figure 8 trajectory $\mathrm{PI}_{\mathrm{TET}}$ data.

The global performance of the control algorithms for the Figure 8 trajectory were as expected, the baseline PID being the worst with a *PI* of 0.002 and the extended NLDI augmented with the AIS being the best with a *PI* of 0.800.

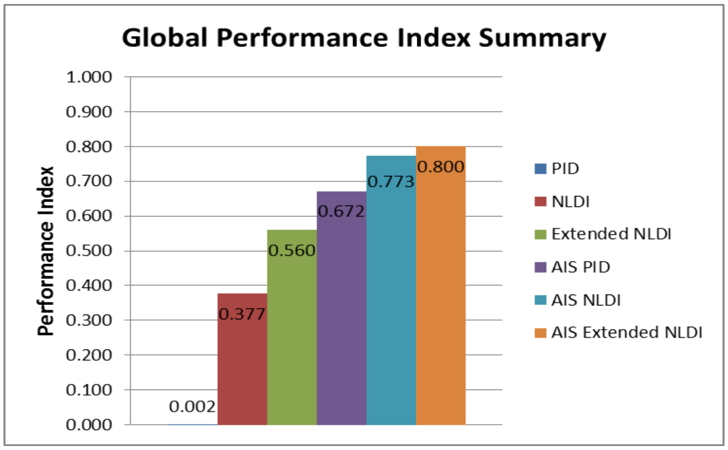


Figure 51: Summary of Figure 8 trajectory global PI data.

Landing Performance Summary

The following is a summary of the performance data for the Landing trajectory. The summary is a direct average of all of the performance indexes over all of the failure conditions.

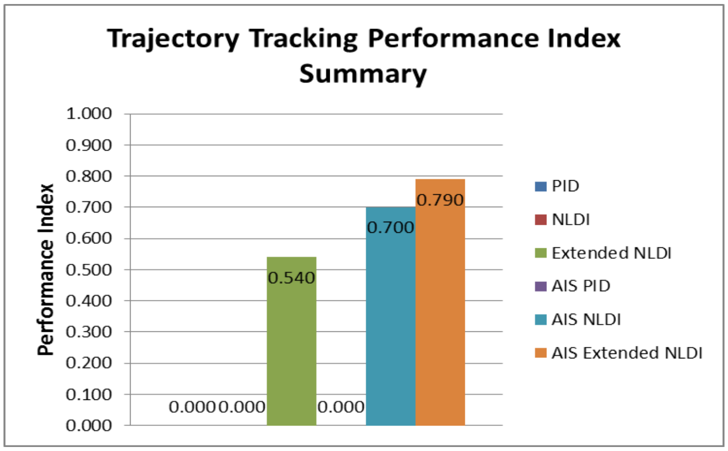


Figure 52: Summary of Landing trajectory PITT data.

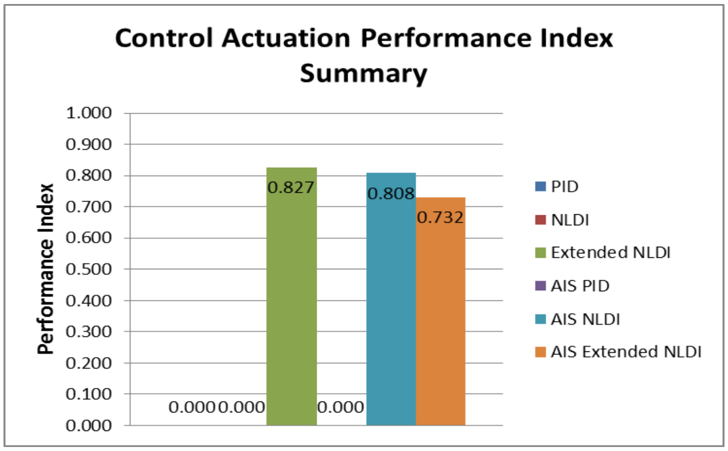


Figure 53: Summary of Landing trajectory $\mathrm{PI}_{\mathrm{CA}}$ data.

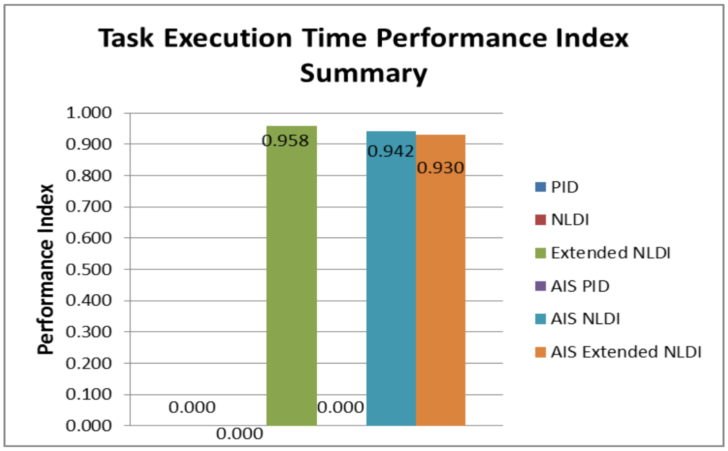


Figure 54: Summary of Landing trajectory PI$_{TET}$ data.

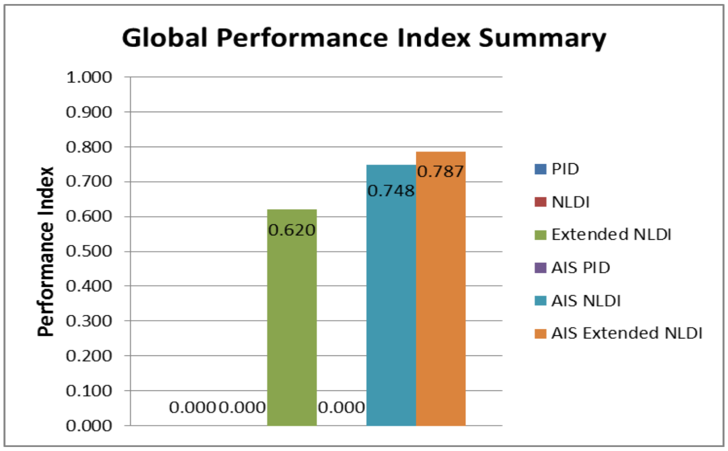


Figure 55: Summary of Landing trajectory global PI data.

6 Conclusions and Suggestions

6.1 Conclusion

In this paper, HIL testing of several baseline controllers as well and their AIS counterparts were presented. The results show that in all cases, with the exception of landing, the AIS augmented systems are able to outperform the baseline performance on a global scale. Under nominal conditions and failures, all of the controllers were able to successfully complete the Figure 8 pattern within the cutoffs provided with the exception of the baseline PID controller. The Landing pattern was a more difficult trajectory to follow which was made manifest due to the fact that only 3 of the six controllers were able to complete it in nominal and failure conditions, the Extended NLDI baseline controller, the Outer NLDI augmented with AIS controller, and the Extended NLDI augmented with the AIS controller.

Every controller tested was able to complete each trajectory with failures in real time. In other words, the task execution time used was always less than the fundamental time of 0.002s imposed by the system, less than or equal 12.5% of that time to be precise. This provides a strong basis to pursue a comparison of these controllers for in flight testing.

6.2 Future Work and Suggestions

The next step in the UAV design process is to accurately model the aerodynamic parameters of the Sig Rascal 110 airframe via wind tunnel testing or parameter identification. Once this model is complete, preleminary testing in simulation and in HIL should be performed on a simple controller such as the PID controller under nominal and failure conditions.

After the controller is proven reliable in flight, the controller should be augmented with the AIS feedback controller and the UAV’s performance re-evaluated. If the immune based controller’s performance trends with those seen in HIL and simulation then the other 4 controllers (outer NLDI, extended NLDI, both outer and extended augmented with the AIS) can be developed and tested in flight as well, with the expectation of improved performance with increasing algorithm complexity.

Finally, one should consider the additional task execution time that reading sensors and sending data via telemetry on board the aircraft requires. If it is deemed that the performance reduction is negligible then the controllers may be ported directly to the aircraft flight computer for in flight testing. Otherwise, the reduction of the fundamental sample rate from 500Hz may be required.

7 Bibliography

Astrom, K. J. (1995). Adaptive Control Around 1960. *3*.

Beard, R. W., & McLain, T. W. (2012). *Small Unmanned Aircraft Theory and Practice.* Princeton: Princeton University Press.

Campa, G., & Napolitano, M. (2004). Design of Control Laws for Maneuvered Formation Flight. *AIAA Control Conference* (pp. 2344-2349). Boston: AIAA.

Chen, W., & Wei, H. (2006). Compensatory Controller Based on Artificial Immune System. *International Conference on Mechatronics and Automation* (pp. 1608-1613). Luoyang: IEEE.

Choon Seong, C. (2008). *Generic UAV Modeling to Obtain its Aerodynamic Control and Derivatives.* Monterey: Naval Post Graduate School.

D., I., & J., V. (2001). *Re-Entry Vehicle Flight Controls Design Guidelines: Dynamic Inversion, Final Technical Report.* Texas: Texas A&M University.

Dydek, Z. T. (2010, June). Adaptive Control and the NASA X-15-3 Flight Revisited. *IEEE Control Systems Magazine,* pp. 32-48.

E., B. (1992). *Immunology, A Short Course.* New York: Wiley-List Publications.

Hedrick, J., & Girard, A. (2005). *Control of Nonlinear Dynamic Systems: Theory and Applications.* Retrieved October 2, 2013, from Berkley.edu: www.me.berkeley.edu/ME237/

Ito, D. (2002). *Reentry Vehicle Flight Controls Design Guidlines: Dynamic Inversion.* Houston: NASA.

Ma, L., & Stephanyan, V. (2006). Flight Test Bed for Visual Tracking of Small UAVs. *AIAA Guidance, Navigation, and Control Conference and Exhibit* (pp. 1-10). Keystone: AIAA.

MathWorks. (2013). *xPC Target User's Guide.* Retrieved from Mathworks Support: http://www.mathworks.com/support/product/XP/productnews/xpc_ target_ ug_ Nov_ 07_ 2003.pdf

Miller, C. (2011). Nonlinear Dynamic Inversion Baseline Control law: Flight-Test Results for the Full Scale Advanced Systems Testbed F/A-18 Airplane. *AIAA Guidance, Navigation and Control Conference.* 2011.

Moncayo, H. Y., & Perhinischi, M. G. (2012). Extended Nonlinear Dynamic Inversion Control laws for Unmanned Air Vehicles. *AIAA Guidance, Navigation and Control Conference* (pp. 1-17). Minneapolis: AIAA.

Moncayo, H., & et, a. (2012). UAV Adaptive Control Laws Using Non-Linear Dynamic Inversion Augmented with an Immunity-based Mechanism. *AIAA Guidanance, Navigation and Control Conference* (pp. 1-18). Minneapolis: AIAA.

Napolitano, M. R. (2002). *YF22 Model With On-board On-line Learning Microprocessors-Based Algorithms for Autopilot and Fault-Tolerant Flight Control Systems.* Arlington: Air Force Research Lab.

NASA. (2000, February 24). *X - 15 Hypersonic Research at the Edge of Space*. Retrieved September 27, 2013, from NASA History: http://history.nasa.gov/x15/cover.html

Nelson, R. C. (1998). Flight Stability and Automatic Control. McGraw-Hill.

Seanor, B., & Campa, G. (2005). NLDI Guidance Control Laws for Close Formation Flight. *American Control Conference* (pp. 2972-2977). Portland: AIAA.

Sherman, R. (2013, September 24). *F-16 Fighting Falcon*. Retrieved from Federation of American Scientists: http://www.fas.org/man/dod-101/sys/ac/f-16.htm

White, J. (2006). NASA's Aviation Safety Program. *44th Annual AIA Aerospace Sciences Meeting*. Reno: AIAA.

Xiao, X. (2009). *The Control of Mobile Sensing Platforms to Perform Estimation of Mobile Targets.* Ann Arbor: ProQuest, LLC.

Zachary T. Dydek, e. (2008). Adaptive Control and the NASA X–15 Program: A Concise History, Lessons Learned, and a Provably Correct Design. *American Control Conference* (pp. 2957-2962). Seattle: IEEE.

Appendix

Landing Trajectory Metrics

Table 13: Landing trajectory tracking metrics.

Failure	Control Algorithm	Max			Mean			Standard Deviation		
		XY	Z	XYZ	XY	Z	XYZ	XY	Z	XYZ
		[m]	[m]	[m]	[m]	[m]	[m]	[m]	[m]	[m]
Left Aileron Stuck at 2°	PID	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Extended NLDI	68.23	3.21	68.24	46.40	1.70	46.45	19.78	0.89	19.76
	AIS PID	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	AIS NLDI	47.76	3.18	47.77	32.48	1.69	32.51	13.84	0.88	13.83
	AIS Extended NLDI	28.87	3.14	28.92	19.76	1.68	19.86	7.44	0.86	7.42
Left Aileron Stuck at 5°	PID	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Extended NLDI	68.10	3.21	68.11	46.33	1.70	46.38	19.73	0.89	19.71
	AIS PID	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	AIS NLDI	47.67	3.19	47.68	32.43	1.69	32.46	13.81	0.88	13.80
	AIS Extended NLDI	28.87	3.14	28.92	19.76	1.68	19.86	7.44	0.86	7.43
Left Aileron Stuck at 8°	PID	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Extended NLDI	68.11	3.21	68.12	46.32	1.70	46.37	19.75	0.89	19.73
	AIS PID	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	AIS NLDI	47.68	3.19	47.68	32.43	1.69	32.46	13.83	0.88	13.81
	AIS Extended NLDI	28.87	3.14	28.92	19.78	1.68	19.88	7.44	0.86	7.43
Right Aileron Stuck at 2°	PID	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Extended NLDI	68.20	3.21	68.18	46.33	1.70	46.35	19.74	0.87	19.72
	AIS PID	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	AIS NLDI	47.71	3.18	47.54	32.42	1.69	32.49	13.84	0.88	13.81
	AIS Extended NLDI	28.93	3.14	28.98	19.82	1.68	19.91	7.45	0.86	7.44
Right Aileron Stuck at 5°	PID	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Extended NLDI	68.19	3.21	68.20	46.36	1.70	46.40	19.76	0.89	19.74
	AIS PID	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	AIS NLDI	47.73	2.25	47.74	32.45	1.69	32.48	13.83	0.87	13.82
	AIS Extended NLDI	28.87	3.14	28.92	19.78	1.68	19.87	7.44	0.86	7.43
Right Aileron Stuck at 8°	PID	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Extended NLDI	68.07	3.21	68.08	46.27	1.70	46.31	19.72	0.89	19.70
	AIS PID	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	AIS NLDI	47.65	3.18	47.66	32.39	1.69	32.42	13.80	0.87	13.79
	AIS Extended NLDI	28.80	3.14	28.85	19.74	1.68	19.83	7.42	0.86	7.41
Nominal	PID	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Extended NLDI	68.06	3.21	68.07	46.28	1.70	46.33	19.73	0.89	19.71
	AIS PID	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	AIS NLDI	47.64	3.19	47.65	32.40	1.69	32.43	13.81	0.88	13.80
	AIS Extended NLDI	28.87	3.14	28.92	19.78	1.68	19.87	7.44	0.86	7.43

Table 14: Landing control activity metrics.

Failure	**Control Algorithm**	**Integral of Control Surface Rate of Change**				**Saturation Index**			
		Elevator	**Aileron**	**Rudder**	**Throttle**	**Elevator**	**Aileron**	**Rudder**	**Throttle**
		[rad]	**[rad]**	**[rad]**	**[%]**	**[%]**	**[%]**	**[%]**	**[%]**
Left Aileron Stuck at 2°	PID	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	NLDI	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	Extended NLDI	1.2247	0.1081	0.0890	29.65	0.00	0.00	0.00	0.00
	AIS PID	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	AIS NLDI	1.2373	0.0757	0.0623	20.76	0.00	0.00	0.00	0.00
	AIS Extended NLDI	1.2401	0.4966	0.2275	45.64	0.00	0.00	0.00	0.00
Left Aileron Stuck at 5°	PID	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	NLDI	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	Extended NLDI	1.1912	0.0978	0.0727	28.47	0.00	0.00	0.00	0.00
	AIS PID	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	AIS NLDI	0.8338	0.0684	0.1009	19.93	0.00	0.00	0.00	0.00
	AIS Extended NLDI	1.1759	0.4270	0.1965	42.65	0.00	0.00	0.00	0.00
Left Aileron Stuck at 8°	PID	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	NLDI	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	Extended NLDI	1.1802	0.0996	0.0737	27.94	0.00	0.00	0.00	0.00
	AIS PID	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	AIS NLDI	0.8261	0.0697	0.1516	19.56	0.00	0.00	0.00	0.00
	AIS Extended NLDI	1.1712	0.4406	0.2011	42.72	0.00	0.00	0.00	0.00
Right Aileron Stuck at 2°	PID	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	NLDI	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	Extended NLDI	1.2989	0.0820	0.0801	30.89	0.00	0.00	0.00	0.00
	AIS PID	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	AIS NLDI	1.2393	0.0719	0.1211	21.62	0.00	0.00	0.00	0.00
	AIS Extended NLDI	1.1853	0.4539	0.2098	43.09	0.00	0.00	0.00	0.00
Right Aileron Stuck at 5°	PID	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	NLDI	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	Extended NLDI	1.1867	0.1105	0.8798	30.10	0.00	0.00	0.00	0.00
	AIS PID	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	AIS NLDI	0.8307	0.0773	0.1615	21.07	0.00	0.00	0.00	0.00
	AIS Extended NLDI	1.2290	0.5023	0.2287	45.01	0.00	0.00	0.00	0.00
Right Aileron Stuck at 8°	PID	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	NLDI	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	Extended NLDI	1.1777	0.1099	0.0814	27.41	0.00	0.00	0.00	0.00
	AIS PID	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	AIS NLDI	0.8244	0.7694	0.0570	19.19	0.00	0.00	0.00	0.00
	AIS Extended NLDI	1.1693	0.4870	0.2238	41.46	0.00	0.00	0.00	0.00
Nominal	PID	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	NLDI	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	Extended NLDI	1.2478	0.1066	0.0878	30.32	0.00	0.00	0.00	0.00
	AIS PID	0.0000	0.0000	0.0000	0.00	0.00	0.00	0.00	0.00
	AIS NLDI	1.1935	0.0746	0.0615	21.22	0.00	0.00	0.00	0.00
	AIS Extended NLDI	1.1337	0.4028	0.1809	39.13	0.00	0.00	0.00	0.00

Table 15: Figure 8 task execution time metrics.

Failure	**Control Algorithm**	**Max**	**Mean**	**Standard Deviation**
		$\times 10^{-5}$[s]	**$\times 10^{-5}$[s]**	**$\times 10^{-5}$[s]**
Left Aileron Stuck at 2°	PID	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00
	Extended NLDI	14.23	3.60	0.96
	AIS PID	0.00	0.00	0.00
	AIS NLDI	18.50	4.67	1.24
	AIS Extended NLDI	23.53	6.32	2.88
Left Aileron Stuck at 5°	PID	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00
	Extended NLDI	10.50	3.64	0.95
	AIS PID	0.00	0.00	0.00
	AIS NLDI	13.65	4.74	1.23
	AIS Extended NLDI	24.91	6.42	3.32
Left Aileron Stuck at 8°	PID	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00
	Extended NLDI	14.76	3.64	1.23
	AIS PID	0.00	0.00	0.00
	AIS NLDI	19.19	4.73	1.60
	AIS Extended NLDI	24.10	6.44	3.30
Right Aileron Stuck at 2°	PID	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00
	Extended NLDI	14.66	3.55	1.07
	AIS PID	0.00	0.00	0.00
	AIS NLDI	19.06	4.61	1.40
	AIS Extended NLDI	17.21	6.36	2.92
Right Aileron Stuck at 5°	PID	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00
	Extended NLDI	10.76	3.64	0.96
	AIS PID	0.00	0.00	0.00
	AIS NLDI	13.98	4.74	1.24
	AIS Extended NLDI	17.31	6.38	2.92
Right Aileron Stuck at 8°	PID	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00
	Extended NLDI	17.41	3.66	1.07
	AIS PID	0.00	0.00	0.00
	AIS NLDI	22.63	4.75	1.39
	AIS Extended NLDI	23.21	6.52	3.30
Nominal	PID	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00
	Extended NLDI	10.29	3.53	0.94
	AIS PID	0.00	0.00	0.00
	AIS NLDI	13.38	4.59	1.22
	AIS Extended NLDI	16.38	6.13	2.84

Table 16: Figure 8 performance index (PI) data.

Failure	Control Algorithm	PI_{TT}	PI_{CA}	PI_{TET}	Global PI
Left Aileron Stuck at 2°	PID	0.00	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00	0.00
	Extended NLDI	0.54	0.82	0.96	0.62
	AIS PID	0.00	0.00	0.00	0.00
	AIS NLDI	0.70	0.81	0.94	0.76
	AIS Extended NLDI	0.79	0.72	0.93	0.78
Left Aileron Stuck at 5°	PID	0.00	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00	0.00
	Extended NLDI	0.54	0.83	0.96	0.62
	AIS PID	0.00	0.00	0.00	0.00
	AIS NLDI	0.70	0.81	0.94	0.76
	AIS Extended NLDI	0.79	0.74	0.92	0.79
Left Aileron Stuck at 8°	PID	0.00	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00	0.00
	Extended NLDI	0.54	0.83	0.96	0.62
	AIS PID	0.00	0.00	0.00	0.00
	AIS NLDI	0.70	0.81	0.94	0.76
	AIS Extended NLDI	0.79	0.74	0.92	0.79
Right Aileron Stuck at 2°	PID	0.00	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00	0.00
	Extended NLDI	0.54	0.82	0.96	0.62
	AIS PID	0.00	0.00	0.00	0.00
	AIS NLDI	0.70	0.81	0.95	0.70
	AIS Extended NLDI	0.79	0.74	0.94	0.79
Right Aileron Stuck at 5°	PID	0.00	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00	0.00
	Extended NLDI	0.54	0.83	0.96	0.62
	AIS PID	0.00	0.00	0.00	0.00
	AIS NLDI	0.70	0.81	0.94	0.76
	AIS Extended NLDI	0.79	0.72	0.94	0.78
Right Aileron Stuck at 8°	PID	0.00	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00	0.00
	Extended NLDI	0.54	0.83	0.95	0.62
	AIS PID	0.00	0.00	0.00	0.00
	AIS NLDI	0.70	0.80	0.94	0.75
	AIS Extended NLDI	0.79	0.73	0.93	0.79
Nominal	PID	0.00	0.00	0.00	0.00
	NLDI	0.00	0.00	0.00	0.00
	Extended NLDI	0.54	0.82	0.96	0.62
	AIS PID	0.00	0.00	0.00	0.00
	AIS NLDI	0.70	0.84	0.95	0.77
	AIS Extended NLDI	0.79	0.76	0.94	0.79