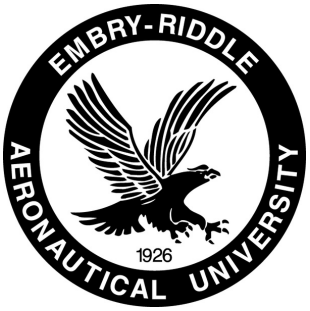
8-2020

UAS Model Identification and Simulation to Support In-Flight Testing of Discrete Adaptive Fault-Tolerant Control Laws

Mansi Subhash Bakori

UAS Model Identification and Simulation To Support In-Flight Testing of Discrete Adaptive Fault-Tolerant Control Laws



Mansi Subhash Bakori

Department of Electrical, Computer, Software and Systems Engineering (ECSSE)

Embry-Riddle Aeronautical University

Supervisor

Dr. Hever Moncayo

In partial fulfillment of the requirements for the degree of

Master of Science in Unmanned and Autonomous Systems Engineering

August 2020

UAS MODEL IDENTIFICATION AND SIMULATION TO SUPPORT IN-FLIGHT TESTING OF DISCRETE ADAPTIVE FAULT-TOLERANT CONTROLS

By

MANSI SUBHASH BAKORI

This Thesis was prepared under the direction of the candidate's Thesis Committee Chair, Dr. Hever Moncayo, Department of Aerospace Engineering, and has been approved by the members of the Thesis Committee. It was submitted to the Office of the Senior Vice President for Academic Affairs and Provost, and was accepted in the partial fulfillment of the requirements for the Degree of Master of Science in Aerospace Engineering.

THESIS COMMITTEE

Chairman, Dr. Hever Moncayo

Richard Stansbury
Digitally signed by Richard Stansbury
Date: 2020.08.21 06:52:31 -04'00'

Member, Dr. Richard Stansbury

Member, Dr. Yan Tang

Digitally signed by Timothy A. Wilson
Date: 2020.08.17 16:11:55 -04'00'

Chair,
Dr. Timothy A. Wilson

Date

Maj Mirmirani
Digitally signed by Maj Mirmirani
Date: 2020.08.18 09:17:54 -04'00'

Dean of the College of Engineering,
Dr. Maj Mirmirani

Date

Christopher Grant
Digitally signed by Christopher Grant
Date: 2020.08.20 16:00:23 -04'00'

Associate Provost of Academic Support,
Dr. Christopher Grant

Date

DEDICATION

I dedicate this thesis to my parents and grandparents for unconditional love, guidance, support and for believing in me…– Mr. Subhash Bakori, Mrs. Kamal Bakori, Mr.Ravjibhai Bakori, Mr. Jamnadas Javia and Mrs. Sushilaben Javia

UAS MODEL IDENTIFICATION AND SIMULATION TO SUPPORT IN-FLIGHT TESTING OF DISCRETE ADAPTIVE FAULT-TOLERANT CONTROL LAWS

Abstract

by Mansi Subash Bakori
Master of Science in Unmanned and Autonomous Systems Engineering
Embry-Riddle Aeronautical University
August 2020

In mission-critical applications of unmanned and autonomous aerial systems(UAS), it is of significant importance to develop robust strategies for fault-tolerant systems that can countermeasure system degradation and consequently support the integration into the National Airspace (NAS). This thesis research illustrates the results of systems identification that is performed using DATCOM followed by the flight test data. This data is acquired from conducting an intensive flight testings program of a fixed-wing UAS to determine the state-space model of the aircraft. A discrete state-space system is reconstructed from these models to derive Auto-Regressive Moving-Average (ARMA) models used to design a Discrete Direct and Indirect Model Reference Adaptive Control. Description of the UAS, sub-systems, and integration is presented in this thesis along with analysis of results from numerical simulation to support the design, development, and validation of adaptive control laws for fault tolerance.A set of performance metrics are defined to perform the analysis in terms of control effort, tracking performance, and reconfiguration of control laws under commonly occurring failures such as partial control surface damage, pilot-induced oscillations, and uncertain ice accretion.

ACKNOWLEDGMENT

Though the following dissertation is an individual effort, I believe that it epitomizes my years of hard work, propensity to move forward and most importantly, people who have assisted me in this journey. Recapitulating my journey, I have nothing but utmost appreciation and gratitude for the valuable support, guidance and efforts of people. My passion for contributing to aerospace unmanned industry has been reinforced by a strong organizational resilience and academic guidance. Therefore, I would like to extend my gratitude to my Research advisor Dr. Hever Moncayo for directing me with his experience and aptitude and for being an exceptional coach in my scholastic life.

My sincere appreciation towards my Advisory Committee , Dr.Richard Stansbury and Dr. Yan Tang for giving your valuable time and insights. An exceptional thanks to Michael Potash and Bill Russo for your enormous assistance with hardware and calmly noting every one of my inquiries. I would also like to offer gratitude to Dr. Gordan Leishman and Dr. Zheng Zang for their distinctive contribution in helping me calibrate air-data probe and allowing me to use the Subsonic Wind-Tunnel Facility.

I acknowledge assistance from Rob, Austin, Ethan and Jeremy for being ever ready to be the test-pilot and not complaining about early morning flights. I likewise recognise contributions from Jorge Begue for providing helping hand in assembling the UAV.

On board of this journey, there was unprecedented team of talented individuals that provided valuable expertise and their benevolent support in the advancement of the unmanned

aerial vehicle thus I will be everlastingly thankful to my lab colleagues turned friends: Angelica, Sharath, Andrei, Diana, Christoph, Sherwin, Juan and Nolan.

Last however not the least, ardent gratitude to my companions turned family here; Omkar, Suyash, Ravi, Karan, Harsh, Farhan, Ashwini, Hardik, Varad, Akshay and Pratik They gave unending enthusiasm, motivation and consolation to achieve what I call it as one of my best scholarly accomplishments so far..

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Nomenclature

Symbols

k	denotes the current sample
$k+1$	denotes one sample in future
s_d	Damage parameter
Ts	sampling time
x	states of the aircraft

Abbreviations

ADP	Air Data Probe
AMA	Academic of Model Aeronautics
AOA	angle of attack
APC	Aircraft Pilot Coupling
AR	AutoRegressive
$ARMA$	Auto Regressive Moving Average
BEC	battery eliminator circuit

CFD	Computational Fluid Dynamics
CTS	Continuous Time System
$DATCOM$	DATa COMpendium
DCS	Discrete Control System
EKF	Extended Kalman Filter
$ERAU$	Embry-Riddle Aeronautical University
FBW	fly by wire
FMU	Flight Management Unit
GBI	Generalized Bezout Identity
GNC	Guidance, Navigation and Control
IAE	Integral Absolute Error
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
ISE	Integral Square Error
ISR	Intelligence, Surveillance and Reconnaissance
MA	Moving Average
$MAVlink$	Micro Air Vehicle LINK
$MIMO$	Multi-input Multi-output
$MRAC$	Model Reference Adaptive Control

OBC	On-Board Computer
ODE	Ordinary Differential Equation
OSD	Office of Secretary of Defense
PEM	Prediction Error Minimization
PID	Parameter IDentification
PIO	Pilot Induced Oscillations/Pilot-in-Loop Oscillations
PWM	Pulse Width Modulation
$RFCS$	Reconfigurable Flight Control System
$RMSE$	Root Mean Square Error
RPV	Remotely Piloted Vehicle
$RTOS$	Real Time Operating System
SBC	Single Board Computer
$SEAD$	Suppression of Enemy Air Defense
$SIDPAC$	Systems IDentification Program for AirCraft
$SysID$	Systems Identification Toolbox
UA	Unmanned Aerial
UAS	Unamnned Aerial System
UAV	Unmanned Aerial Vehicle
ZOH	Zero Order Hold

Subscripts

α	Angle of Attack
β	Angle of Sideslip
ρ	Density
P_1	Pressure at port number 1
P_{static}	Static Pressure
P_{total}	Total Pressure
y_m	m stands for reference model

Chapter One

Introduction

1.1 Research Background

The importance of Unmanned Aerial Vehicle (UAV) or more general Unmanned Aerial System (UAS) is revealed by many different statistics published in official and unofficial reports like the manufacturing of more than 600 unmanned aerial systems of various sizes by 250 manufacturers in 42 nations, and this includes solely commercial and government organizations(K. K. Bhamidipati, Daniel Uhlig and Natasha Neog, 2008).Geological surveying, fire monitoring, rescue missions, wildfire mapping, law enforcement, aerial imaging are some parts of civilian applications of UAS, and Intelligence, Surveillance and Reconnaissance (ISR), Suppression of Enemy Air Defense (SEAD), and high-value asset recovery scenarios are some examples of military applications. Simultaneously the importance of UAV safety and reliability is attracting more attention in this field. An acknowledgement of this importance related to UAV roadmap for 2005-2030 years is reported by The Office of the Secretary of Defense (OSD) by stating that "Improving UA (Unmanned Aircraft) reliability is the single most immediate and long-reaching need to ensure their success."

In early years, UAVs were completely controlled by human operator from the ground known as Remotely Piloted Vehicles (RPVs) and the last decade has witnessed unprecedented interactions between technological developments in computing, control, and communications.

These developments led to the design and implementation of interacting dynamical systems such as cooperative as well as networked unmanned multi-vehicle systems. Advances in sensor systems, on-board computational platforms, energy storage, and other enabling technologies have made it possible to build a huge variety of UAVs for a range of different mission scenarios. Many of the mission scenarios of interest, such as persistent surveillance, are inherently long duration and require coordination of multiple cooperating UAVs in order to achieve the mission objectives. In these types of missions, a high level of autonomy is desired due to the logistical complexity and expense of direct human control of each individual vehicle. On the other hand, although military/civilian researches and implementation results around the world have underscored the potential utility of unmanned aerial vehicles but still, most of their successes have occurred in a setting that allows a relatively large margin for errors, thereby, such sophisticated control systems should meet increased performance and safety requirements. In other words, although the knowledge of control is trying to save the pilot's life as the most valuable part of manned aerial vehicle, by replacing it with different controller architectures in the control loop, but in parallel, it is an important task for control engineers also to save the UAV system itself, in the case of fault/failure occurrence in either hardware or software components.Furthermore, the use of unmanned aircraft for verification and validation of flight control laws has become an appealing option due to the high cost and risks associated with flight testing programs of manned aircraft for research scientists. Failures and faults in manned or unmanned aircraft can lead to catastrophic consequences resulting loss of life and the aircraft itself. As the operation of UAV becomes more common, the implementation of technologies with fault-tolerant control capabilities have become a required part of the integrated Guidance, Navigation and Control (GNC) architecture.

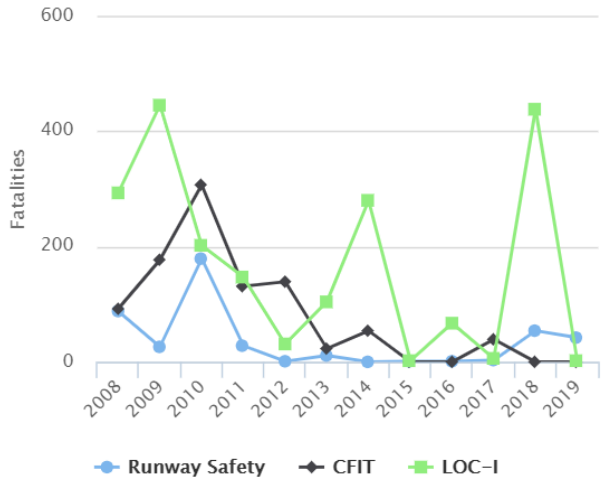


Figure 1.1 Share of Fatal Accidents By risk Category Source: ICAO Accident-Statistics

1.2 Motivation

Nearly half of the current-generation of unmanned surveillance aircraft has been lost. According to aircraft accident statistics for world wide commercial jet fleet, 1926 out of 2200 accidents analysed, were caused due to loss of control in flight resulting large number of fatalities. This loss rate is about 10 times worse than manned combat aircraft. This fact reveals the importance of fault-tolerant control of UAVs. The consequences of a minor fault in a system component can be catastrophic particularly for many safety-critical systems, including aircraft. Therefore, the demand on reliability, safety and fault tolerance is generally high.

The chief goal of this research thesis is to present analysis and results obtained for generating a non-linear mathematical model of a UAS used as research platform to support the design, development and in-flight testing of a set of fault-tolerant flight control laws previously implemented in simulation. The model identification is performed within an intensive flight research program initiated at Embry-Riddle Aeronautical University (ERAU) to demonstrate navigation and control of aerial systems under nominal conditions while assessing the overall system health with detection and evaluation of abnormal conditions, and

Figure 1.2 Asiana Airline Flight 214 , Soucre: Wings Herald

accommodation of the upset conditions. As complex systems are controlled by computers, which are discrete in nature, of particular interest is the design of discrete-time adaptive control laws. In this study, a discrete model reference adaptive control is designed and its performance is validated using numerical simulations of the UAS model. Then, the compensation capabilities are verified considering three types of faults: partial loss of actuator elevator, pilot-induced oscillations(PIO) and uncertain icing condition.

1.3 Research Objective

The objective of this research is to identify a model of an Unmanned Aerial Vehicle, the Rascal 110 from flight data that accurately represent its dynamic behavior. This mathematical model finds its application within a fault-tolerant flight control system for commonly occurring failures.

1.4 Literature Review

1.4.1 Reconfigurable Flight Control

Reliability and survivability play a vital role in modern high performance aircraft that leads to designing of reconfigurable flight control system (RFCS) that can automatically reconfigure during the occurrence of the faults. In certain, unexpected scenarios, aircraft has a control surface damage or locked/jammed control surfaces during which its dynamics alters with non-linearities being produced. These non-linearities can deteriorate aircraft's performance tremendously. In this situation, RFCS can redistribute and coordinate the control effort among remaining control surfaces, is desired to retain stability or satisfactory flight performance of the aircraft when it is physically possible. In recent years, for this purpose, adaptive controls are being studied extensively.

One such efforts was taken D.G. Ward, J.F. Monaco and M. Bodson (1998)[4]. Real-time parameter identification and control reconfiguration algorithm were implemented and evaluated on series of flight tests. The author discusses about the challenges associated with poor information content of the signals used for identification and its need for autonomy, reliability and fast adaption. Modified Sequential Least Square algorithm was developed to overcome the problem of identifying time-varying parameters in the system that is often insufficiently excited during on-line estimation. The identification technique was evaluated using Lockheed Martin's Non Linear 6-DOF simulation of the F-16 and flight data provided by Calpan. A series of flight tests resulted in a landing under a simulated failure condition and on-line reconfigurable control and demonstrated the success of the identification algorithm in determining the parameters of the aircraft in real-time.

In 2009, a team of researchers [5] performed parameter identification that was particularly used for fault-tolerance purpose. Their research efforts described the extraction of the mathematical model of WVU(West Virginia University) YF-22 unmanned research aircraft under nominal and failure conditions for both linear and non-linear models. These

models were developed using flight test data collected during the performance of doublets that were injected by on-board computer. For identification of nominal longitudinal model, 'n4sid' function of MATLAB System Identification Toolbox®was used which estimates a state-space model using a subspace based identification method. For nominal-lateral identification, prediction-error minimization 'pem' was used. The methods and equations for identification of decoupled non-linear longitudinal and lateral model was implemented. Simulation results were presented to compare the results from measured flight test data.

Continuing this work, Kerri Philips in her dissertation[7] derived a non-linear model for YF-22. From this analysis, the stability and control derivatives were extracted to determine the aerodynamic forces and moments on each aircraft. These aerodynamics were next introduced into a simulation environment to validate the accuracy of the identified mathematical models. The author performed several simulation studies to validate the accuracy of the models for each research platform, focusing on both nominal and primary control surface failure conditions where applicable. Later her results, the model outputs were compared to the measured flight data from the two respective research platforms to validate the accuracy of the estimated parameters.

In 2012, J. Lee, H. S. Choi, S. Lee, E. T. Kim and D. Shin[8], worked on building a fault-tolerant controller that can compensate for actautor failures during aircraft autolanding. Acoridng to the authors, the last component in control -action implementation on an airplane are the actuators and they play vital role in delivering necessary power to change the controlled variable. Also, it is not ideal to have multiple redundant actuators on airplanes due to cost-effectiveness. Actuators are heavy and bulky for operating control surfaces under large aerodynamic forces hence fault-tolerant control design considering faults in actuators is an active area of research. In this research, in order to achieve robust performance of small jet aircraft, authors adopted time delay control(TDC) scheme to design fault-tolerant flight control system. They apply this to the automatic landing problem under actuator failures. The performance was validated via a 6-DOF non-linear simulation. The results proved that

aircraft can perform auto-landing without degradation of performance in case of single and multiple faults.

1.4.2 Model Reference Adaptive Control

Model reference adaptive control(MRAC) is usually used in situations where conventional feedback controllers may not perform well due to changes in process dynamics, non-linear actuators, alterations in environmental conditions and other types of disturbances.
Adaptive controls in discrete system is very rarely found in the controls literature. S.Hyung and Y.Kim [2005] used discrete MRAC to create a reconfigurable flight control. The systems identification was performed using Auto regressive(AR) model since it can handle variable structure systems. Discrete MRAC was utilized for fault-tolerant control systems that can adapt to the reference system since reference system is viewed as ideal nominal system for the aircraft.

The faulty case considered here is control surface damage of F-16. The 60 percent damage of elevator was induced at 4 sec in the numerical simulation. The derived results stand as a testament for Linear Quadratic Tracking controller that follows the reference command even after the instance of the fault occurrence. The appendix of the paper also proves the stability of the discrete adaptive control.

Experimental results on actuator fault-tolerant control for a quadcopter Unmanned Aerial Vehicle system was presented by A. Chamseddine , Y. Zhang, C. Rabbath, C, Fulford and J.Apkarian[2011] . The control strategy to achieve this goal is based on Model Reference Adaptive Control(MRAC). Three different schemes of MRAC were discussed and implemented with their pros and cons namely the MIT rule MRAC, the Conventional MRAC(C-MRAC) and the Modified MRAC (M-MRAC). These MRAC schemes were compared to Linear Quadratic Regulator baseline controller. The primary advantage of MRAC is that it

does not require explicit information about fault location and or amplitude and thus, fault detection and identification is not needed to detect, isolate and identify the occurred faults. The fault scenarios that were experimented on Qball-X4 included partial effectiveness loss in the total thrust, partial effectiveness loss in the 4th rotor and partial damage of the 4th propeller. Conventional MRAC out of all the three schemes proved to be capable of rapidly reacting to faults without any priori knowledge about the system or fault-location. Also, the results clearly showed that depending on actuator constraints, only a certain amplitude of actuator fault can be reconfigured. Beyond this limit, system cannot be stable anymore.

1.4.3 Pilot-in-Loop Oscillation

Pilot Induced Oscillations (PIO), sometimes referred to as Pilot Involved Oscillations and, more recently, as unfavourable Aircraft-Pilot Couplings (APC), are rare, unexpected, and unintended excursions in aircraft attitude and flight path caused by anomalous interactions between the pilot and the aircraft. Many researches have been conducted to implement the advanced control systems that could provide great potential for improvement in airplane's performance. Different techniques involve using L1 Adaptive Control(M. Santone, C. Cao, 2012), control allocation techniques(Y. Yildz, I. V. Kolmanovsky, 2010), etc. There are several causes of PIO occurrence some of them are, due to actuator rate saturation, actuator delay in response to pilot input, dynamic coupling.

I. Alcala, F. Gordillo and J. Aracil presented phase compensation design for prevention of PIO due to actuator rate saturation in 2004. They developed a simple and effective solution for type 2 pilot-induced oscillations due to rate limit in the control surface. All aircraft control surfaces have restrictions when the actuators are operating at their maximum capacity. One of these limitations is known as rate limit and it relates to the maximum speed at which an actuator can follow changes in the input signal. Furthermore, in fly-by-wire (FBW) the

control signals are rate limited by software before feeding the control surface in order to avoid stress in the actuator. The proposed method uses a non-linear filter that compensates the phase of the control signal before feeding the actuator. It is developed with a rate-limited feedback and a phase-lead network for compensating the phase lag .

The structure of this filter has advantages over previous realizations that allow tuning simplicity considering limit cycle prevention as control specification. Simulation results demonstrate the good performance of the proposed compensation.

Y.Yildiz, I. V. Kolmanovsky's research efforts illustrates a control allocation technique that can help pilots recover from PIO. PIO are described as unwanted aircraft oscillation that occur due to joint enterprise between the aircraft and pilot. There are several other ways of causing PIO. They included rate saturated actuators, high gain pilot or controller, system delays and phase lags. This paper particularly concentrates on PIO caused by rate saturated actuators. It proposes control allocator that reduces the effective time delay by minimizing the phase shift between the commanded and the actual attitude accelerations. CAPIO functions by minimizing the error between the derivatives of desired total control effort and the achieved noise, digital realization of the derivative. The integration of CAPIO with a PIO detector are important points to be addressed for a successful technology transfer. Simulation results are reported, which demonstrate phase shift minimization and recovery from PIOs.

L1 Adaptive control, as mentioned earlier, has been used to suppress PIO. Combined efforts of C. Wang, M. Santone and C. Cao presents the L1 adaptive controller that has been introduced to suppress the PIO, which is caused by rate limiting and pure time delay in 2012. According to the authors, due to its architecture, the L1 adaptive controller will achieve a desired response with fast adaptation. The analysis of PIO with its categories and its suppression by L1 adaptive controller are given in detail in the journal paper. The authors show the system modeling and PIO modeling in simulation. The numerical simulations

results indicate that the L1 adaptive control is efficient in solving this kind of problem.

1.4.4 Icing Condition

Many aircraft accidents have occurred due to uncertain ice accretion on the aircraft's body. Pilots generally have a very less time to escape the harmful icing condition. Sometimes experienced pilots do not even realise that they have flown into such harmful condition. Also sometimes ice accretion goes unnoticed during the cruise condition. In this thesis research[K. M. Rankin, 2014], author mentioned few examples of air-crash owing to effects of icing condition. Effects of ice on the equations of motion of airplane are studied and modelled here which author describes it as a challenging task since there are many shapes and areas on the aircraft where ice can form where it can have some or no effect on control surface. It can even lead to total loss of control of aircraft.

The iced aircraft models are based on data taken from the effects of ice accretion on a DeHavilland Twin Otter aircraft. Statespace models for, with and without icing are derived for Cessna 208 Super Cargomaster where change in stability and control parameters due to icing are mentioned with explicit percentage change.

Author designed the classical fixed gain controller to show its inadequate performance during the ice accretion.This is followed by designing and implementation of adaptive output tracking control algorithm in simulation for normal and ice conditions. The adaptive control algorithm illustrates the effectiveness of adaptive control to handle the alterations in aircraft's dynamics.

1.4.5 Calibration of Air Data Probe

Air data probe(ADP) is very important to successfully complete an aircraft's mission and is derived from the air surrounding the aircraft.These air data encompass, indicated and true airspeed, pressure altitude, ambient air temperature, angle of attack(AOA) and sideslip, Mach number, and rate of climb (E.A Haering, 1995).There are several ways to calibrate the ADP, some include using inertial measurements, inertial navigation system(INS), wind-tunnel method, etc.

In 1982, A.A. gerner and C.L. Maurer at Air Force Academy submitted the research paper at AIAA 20th Aerospace Sciences Meeting that describes the method of calibrating seven-hole ADP upto 80 degrees for compressible and incompressible airflow [9]. The probe measures total and static pressures that are used to calculate the angle of attack(α) and angle of sideslip(β) of the aircraft. The authors explain the experimental procedure and apparatus needed to carry out in 1 by 1 foot blowdown wind tunnel

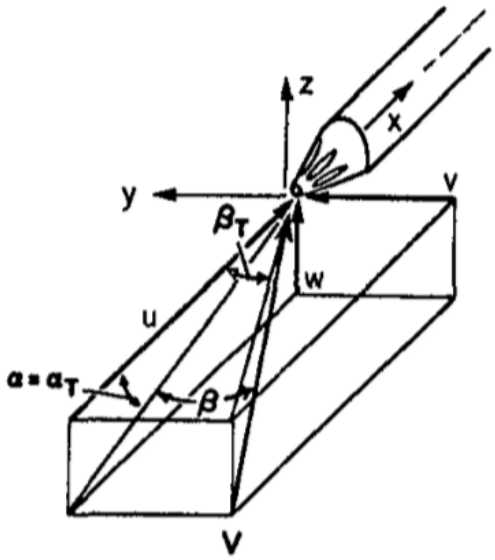


Figure 1.3 Low Angle Reference System - Source: A.A Gerner, C.L. Maurer U.S. Air Force Academy

For this research [9], only low flow angles and incompressible theory was point of interest where mainly the probe's axis are less than or equal to 30 degrees. The author considers tangential plane as a reference system which is slightly different from wind reference frame where α - β system exists. To conserve the air in the blow down wind tunnel, the authors ended up using least squares to sample large amount of data therefore reducing it to minimum number of data sets required for a polynomial fit. The probe calibration is represented by three-variable third order polynomials that yields all the desirable output quantities that are computed from the pressures measured by probe.

1.5 Thesis Outline

This thesis is organized as follows. The first chapter provides brief introduction and motivation behind choosing this topic research with some literature review. Chapter two presents an overview of discrete control theory and how it is different from continuous time systems. Chapter three describes the process of developing and instrumenting the UAV test bed. Chapter four presents a detailed procedure of calibrating the seven hole air data probe. This is followed by detailed description of how identification of mathematical model was carried out in Chapter five. Chapter six illustrates the numerical simulations and results performed from implementing direct and indirect MRAC.

Chapter Two

Discrete Control Theory

2.1 Introduction

Recently, there has been an increase in usage of digital controllers in control systems. Digital control or discrete control system (DCS) are used for achieving optimal performance in form of maximum productivity, maximum profit, minimum cost or minimum energy. Also, they behave more robust to environment disturbances when compared to continuous time systems (CTS). Discrete-time controls are the ones in which one or more variables can change only at discrete instant of time. These are called samples and the rate at which it gets updated is called sample rate. Samples are denoted by letter k in this thesis.

DCS operate in Z-domain while CTS in S-domain for computing response of the dynamic system. Z-domain transfer functions are computed by difference equations whereas S-domain transfer functions use ordinary differential equation(ODE). Difference equations are suitable for computer systems as they operate upon discrete samples of system input (past and present) and previous system output. Differential equation requires continuous knowledge of input and output and solving them on computer requires numerical methods which is essentially a difference equation. Since continuous time system uses ODE, it requires higher computing processors than discrete controls.

There are three general approaches for implementing DCS on a continuous plant system.

1.Design CTS controller for a continuous plant and convert to an equivalent DCS considering the effects of sample time.

2.Convert a continuous plant system and continuous sensors to discrete plant and discrete sensors followed by designing a discrete controller for them.

3.Tune a discrete controller with a continuous plant model

This thesis research follows the 2nd approach.

There are five methods to convert continuous time system to discrete. They are

1. Zero-order hold (ZOH): exact discretization in the time domain for staircase inputs.It holds the previous sample value with zero slope or zeroth order polynomial and step up the value with next sample time. ZOH is used in this research for converting continuous plant system to discrete plant.

2. First order hold (FOH): provides the exact match for discretization in the time domain for piecewise linear inputs. This method is also referred to as ramp invariant method because it produces the same ramp response between CTS and DCS.

3. Impulse : The impulse-invariant mapping produces a discrete-time model with the same impulse response as the continuous time system

4. Tustin Approximation or Bilinear approximation: yields the best frequency-domain match between the continuous-time and discretized systems. If the system has important dynamics at a particular frequency that the transformation needs to preserve, use the Tustin method with frequency prewarping. This method ensures a match between the continuous- and discrete-time responses at the prewarp frequency. By default, the Tustin method rounds any time delay to the nearest multiple of the sample time. Therefore, for any time delay tau, the integer portion of the delay, $k*Ts$, maps to a delay of k sampling periods in the discretized model. This approach ignores the residual fractional delay, tau $k*Ts$.

5. Zero-Pole Matching equivalent: The stability of continuous time closed loop controller that is linearly time-invariant is determined by placement of closed-loop poles in s-plane.

The poles that are on left half of the s-plane i.e. negative real part are stable which mean they will showcase the exponential decay. The poles that are on right half of s-plane are unstable. Complex variable z and s are related .

$$z = e^{Ts} \tag{2.1}$$

Hence the locations of poles of closed-loop pulse transfer function can help determine the stability of linear-time invariant discrete time closed loop system. Sampling period T affects the dynamic behavior of the discrete-time control system. In terms of poles and zeros in z plane their locations depend on the sampling period. It means that alteration in sampling period modifies poles and zero locations in z plane causes the response behavior to change. [K.O.,Discrete-Time Control Systems,2013] This method involves taking poles of CTS and mapping those poles in discrete time. If the time constant of the system is slow compared to sample time then each of the method stated above is similar.

2.1.1 Stability Analysis of Discrete Controller

For discrete control, sampling period holds an important place. If sampling period is too long where Nyquist sampling theorem is not satisfied then it leads to frequency folding and aliasing. This alters the location of poles and zeros.

The stability of linear-time invariant single input-single output discrete-time control systems is determined by the position of closed-loop poles in the z plane or the roots of characteristic equation. Lets consider a system that has the following transfer function.

$$\frac{C(z)}{R(z)} = \frac{G(z)}{1 + GH(z)} \tag{2.2}$$

The characteristic equation is

$$P(z) = 1 + GH(z) = 0 \tag{2.3}$$

The roots of characteristic equation will determine the system's stability. For the system to be stable, the roots must lie within the unit circle in the z-plane. Any closed-loop pole outside the unit circle makes the system unstable. If a pole lies on the unit circle that is $z = 1$ then the system is critically stable. System also becomes critically stable if the single pair of the conjugate complex poles lies on the unit circle in the z- plane. Any multiple close-loop poles on the unit circle makes the system unstable. Zeros of closed-loop system do not affect the system stability and therefore may be located anywhere in the z-plane.

Hence to summarize for a discrete closed-loop system is unstable, if the poles lies outside the unit circle and/or any multiple-closed loop pole lies on the unit circle in the z plane.

2.2 Introduction to Model Reference Adaptive Control

When something or someone 'adapts to' a situation it means to adjust to a new situation so when it comes to controllers, standard ones are unalterable and cannot be used in systems that have varying parameters over time. Thus the requirement for adaptive controllers.An adaptive control system resembles any other control system which has the capability to mutually adjust itself based on the inputs from the system in consideration of the system uncertainty. The parameters that are altered are called adaptive parameters and the mechanism of adjustment, described by mathematical equations is the adaptive law. Figure 2.1 describes the general architecture of MRAC. In this research thesis, MRAC approach is used to solve mid-air occurrence of faults/failures in the fixed wing UAV. MRAC is used for making a closed loop controller which adjusts the variables of the system dynamically by comparing the output of the plant with a standard reference response. Further, MRAC is classified into two types:

1) Direct Control: In this controller, the system adjusts itself to the error signal which is described as the difference between the plant and the reference response. The controller parameter T of the controller C(T) is updated in real time by adaptive law.

2) Indirect Control: The basic philosophy of this approach is to estimate the parameters of the unknown plant from input output data and, in turn, use these estimates to adjust the parameters for a controller so that the transfer function of the controlled plant evolves to that of the model (Kumpati S. N., L S. Valavani,1976).

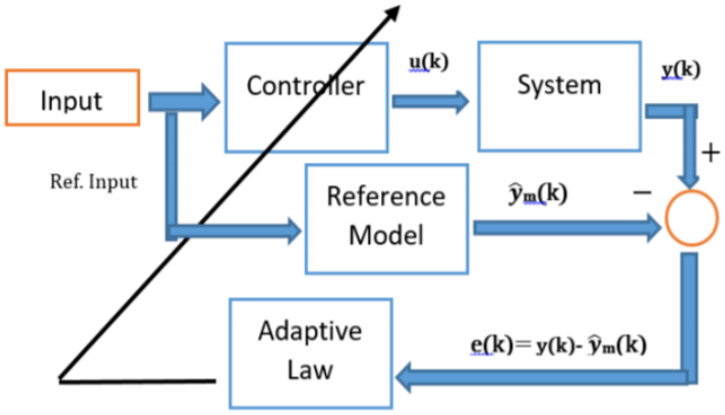


Figure 2.1 General Structure of MRAC

2.3 Design of a Discrete Direct Model Reference Adaptive Control

Discrete MRAC using system identification is one approach that can be used as a fault-tolerant control system. In this work, the identified aircraft model was converted to a discrete system. Later a Discrete Direct MRAC was designed and implemented and its capabilities of compensating actuator failures, PIO and uncertain ice accretion are evaluated. Consider the following discrete linear system plant in state-space in equation 2.4. The ouput model for that equation can be defined as:

$$x(k+1) = Ax(k) + Bu(k) \tag{2.4}$$

$$y(k) = Cx(k) \tag{2.5}$$

where x(k) $\in R^n, u(k) \in R^m, y(k) \in R^p$ and A, B, C are system, input and output matrices, m is number of inputs, n is number of states and p is $m \times n$ respectively. Consider a reference model for the system to follow and represented as:

$$x_m(k+1) = A_m x_m(k) + B_m r_m u(k) \tag{2.6}$$

$$y_m(k) = C_m x_m(k) \tag{2.7}$$

The objective of the control law is to determine adequate control input to make the output $y(k)$ follow the reference model output $y_m(k)$. The output error can be then defined as:

$$e(k) = y(k) - y_m(k) \tag{2.8}$$

For a Multi-Input-Multi-Output (MIMO) system, the linear discrete-time model is considered as follows

$$A(z^{-1}) = I + a_1 z^{-1} + ... + a_n z^{-n} \tag{2.9}$$

$$B(z^{-1}) = b_0 + b_1 z^{-1} + ... + b_m z^{-m} \tag{2.10}$$

where z^{-1} is a shift operator, $u(k)$ is the input vector, $a's$ and $b's$ are control and stability parameters and $y(k)$ is an output vector. A variety of parametric model structures are available to modeling an unknown system. Parametric models describe systems in terms of differential equations and transfer functions. The system used here was defined using an ARMA model. In the statistical analysis of time series, ARMA models provide a parsimonious description of a (weakly) stationary stochastic process in terms of two polynomials, one for the Auto Regression (AR) and the second for the Moving Average (MA). Both solve

linear regression in analytic form. The ARMA model is expressed by the past input-output data and the dimension of the system model is determined by the data size of the gathering input-output information. It is natural that the data size must be larger than the system dimension, and the accurate size of dimension needs not to be known[4] . The discrete system is stable if every pole of every transfer function has a magnitude less than 1. All poles of all transfer functions must exist inside the unit circle on the Z plane. The roots for $A(z^{-1})$, $B(Z^{-1})$, $A_m(Z^{-1})$ and $B_m(Z^{-1})$ are within the unit circle and therefore they are stable. Combining equation (6), (7) and (8), an ARMA model can be written as

$$y(k+1) = b_0u(k)+b_1u(k-1)+b_2u(k-2)...+b_m(k-m)-a_1y(k)-a_2y(k-1)..-a_ny(k-n+1) \tag{2.11}$$

With the known number of inputs and outputs, m and n, respectively. The coefficient values of $A(Z^{-1})$ and $B(Z^{-1})$ polynomial equations are to be exactly determined since they are affected by changes in the system. The objective of Direct MRAC is to generate the appropriate input $u(k)$ at the k^{th} step to make system output $y(k)$ follow the reference $y_m(k)$.

A system tracking problem can be formulated by using a desired tracking dynamics with a delay d greater than 1:

$$D(Z^{-1})[y_m(k+d) - y(k+1)] = 0 \tag{2.12}$$

$$A(Z^{-1})y(k+1) = B(Z^{-1})u(k) \tag{2.13}$$

and using generalized Bezout Identity defined as $D(z^{-1})$:

$$D(Z^{-1}) = z^{-d}R(Z^{-1}) + S(Z^{-1})A(Z^{-1}) \tag{2.14}$$

Within this approach the system tracking is achieved while estimating on-line the unknown parameters within the control laws $u(k)$ for the Discrete DMRA:

$$u(k) = \frac{1}{b_0}[D(Z^{-1})y_m(k+d) - R(Z^{-1})y(k) - B_s^*(Z^{-1})u(k-1)] \tag{2.15}$$

where,

$$B_s^*(Z^{-1}) = b_1 + b_2 Z^{-1} + b_3 Z^{-1} + ... + b_m Z^{-1} \tag{2.16}$$

From which we have:

$$D(Z^{-1}) y_m(k+d) = \hat{b_0} u(k) + B_s^*(Z^{-1}) u(k-1) + \hat{R}(Z^{-1}) y(k) \tag{2.17}$$

where in a matrix form,

$$\begin{aligned} D(Z^{-1}) y_m(k+d) = [\hat{b_0}\ \hat{B^*_{s1}} .... \hat{B^*}_{s(d+m-1)} \hat{R}_0\ \hat{R}_1 ... \hat{R}_{n-1}] ... \\ [u(k)\ u(k-1) .. u(k-d-m+1)\ y(k)\ y(k-1) ... y(k-n+1)]' \end{aligned} \tag{2.18}$$

and

$$P^T(k) = [\hat{b_0}\ \hat{B^*_{s1}} .... \hat{B^*}_{s(d+m-1)}\ \ \hat{R}_0\ \hat{R}_1 ... \hat{R}_{n-1}] \tag{2.19}$$

$$M(k) = [u(k)\ u(k-1) .. u(k-d-m+1)\ y(k)\ y(k-1) ... y(k-n+1)]' \tag{2.20}$$

Thus

$$D(Z^{-1})\ y_m(k+d) = P^T(k) M(k) \tag{2.21}$$

A priori tracking error in Discrete DMRAC can be defined as:

$$e(k+d) = D(z^{-1})\ [y_m(k+d) - y(k+d)] \tag{2.22}$$

where

$$D(z^{-1})\ [y_m(k+d)] = \hat{P^T}(k) M(k) \tag{2.23}$$

$$D(z^{-1})\ [y(k+d)] = P^T(k) M(k) \tag{2.24}$$

Consider residual error $e(k)$ defined as follows.

$$e(k) = P^T M(k-d) - \hat{P^T}(k-d) M(k-d) \tag{2.25}$$

Unfortunately this error has built-in inherent delay, therefore adaptive signal error is introduced that is defined as:

$$e^*(k) = e(k) - [(P^T - \hat{P^T}(k-d)) M(k-d)] = (P^T - \hat{P^T}(k)) M(k-d) \tag{2.26}$$

Therefore,

$$\hat{P}(k) = \hat{P}(k-1) + F(k-1)M(k-d)e^*(k) \tag{2.27}$$

$$F(k) = F(k-1) - \frac{F(k-1)M(k-d)M^T(k-d)F(k-1)}{1 + M^T(k-d)F(k-1)M(k-d)} \tag{2.28}$$

Solving for the adaptive signal error:,

$$e^*(k) = [P^T - \hat{P}^T(k-1)]M(k-d) - e^{*T}(k)[F(k-1)M(k-d)]^T M(k-d) \tag{2.29}$$

Rearranging the terms,

$$e^*(k) = \frac{P^T M(k-d) - \hat{P}^T(k-1)]M(k-d)}{1 + M^T(k-d)F(k-1)M(k-d)} \tag{2.30}$$

The general architecture for the designed Direct MRAC is shown in Fig. 2.2.

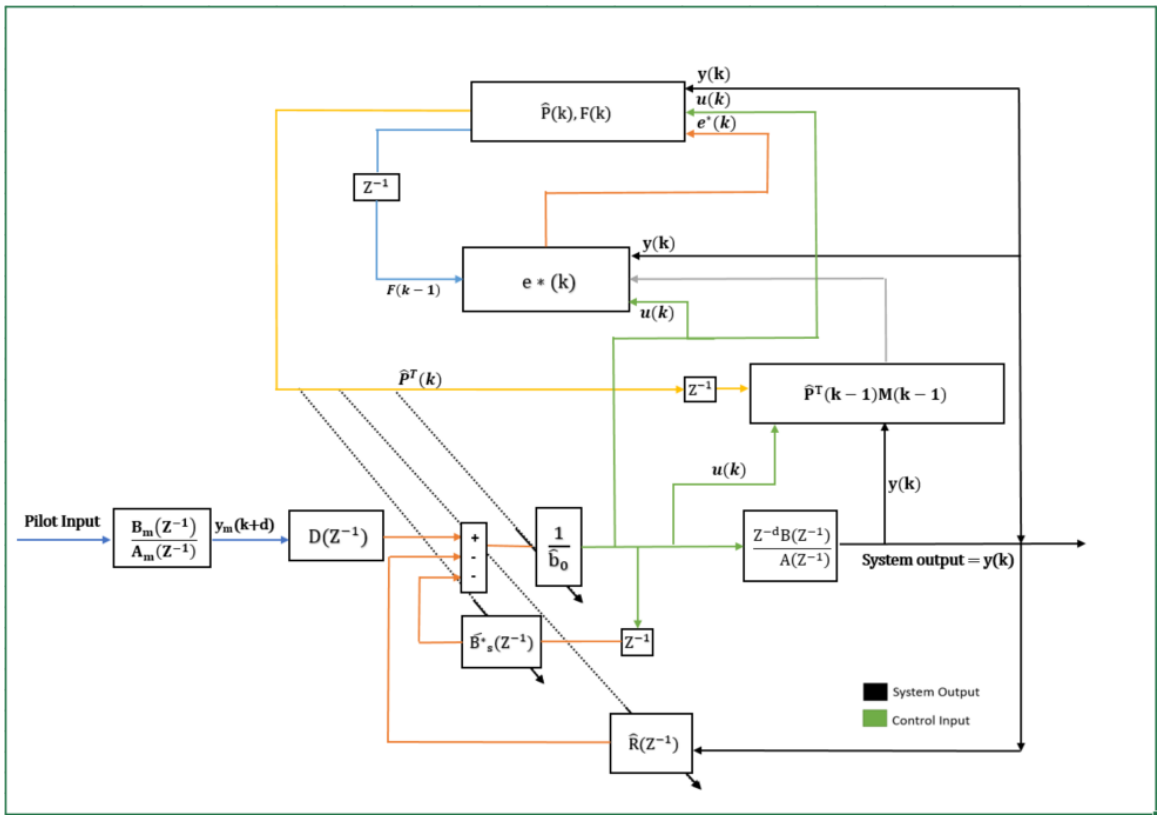


Figure 2.2 Discrete Direct Model Reference Adaptive Control Architecture

2.4 Design of a Discrete Indirect Model Reference Adaptive Control

The approach for Indirect MRAC is quite different from the Direct MRAC. Infact within the indirect MRAC, the coefficients of polynomials $A(Z^{-1})$ and $B(Z^{-1})$ of the ARMA (m,n) model describing the actual system $y(k)$ are estimated. This is followed by application of Bezout identity

$$D(z^{-1}) = z^{-d}R(z^{-1}) + S(z^{-1})A(z^{-1}) \tag{2.31}$$

where $\hat{\hat{R}}(Z^{-1})$ and $\hat{\hat{S}}(Z^{-1})$ which are both function of $A(Z^{-1})$. $\hat{\hat{B}}_s(Z^{-1})$ is found using $\hat{B}(Z^{-1})$ and $\hat{\hat{S}}(Z^{-1})$. Finally the expression $u(k)$ is derived similarly as the direct MRAC when the delay is greater than one.

$$u(k) = \frac{1}{\hat{b_0}}[D(z^{-1})y_m(k+d) - \hat{\hat{R}}(Z^{-1})y(k)-$$

$\hat{b_0}$ has only one sign of estimation because it is direct estimate while $\hat{\hat{R}}(Z^{-1})$ and $\hat{\hat{S}}(Z^{-1})$ has double sign because they are function of estimates. Although the overall indirect MRAC algorithm is more complicated than the direct MRAC algorithm(since it involves all the steps outlined above), the parameter estimation part is considerably simpler and it does not involve the problems associated with the delay. For parameter estimation, most recent available output is used that is'$y(k)$' Hence the equation 2.11 becomes

$$y(k) = Z^{-d}y(k+d) \tag{2.33}$$

$$Z^{-d}y(k+d) = b_0u(k-d)+b_1u(k-d-1)+b_2u(k-d-2)...+b_m(k-d-m)-a_1y(k-1)-a_2y(k-2)..-a_ny(k-n) \tag{2.34}$$

The above equation can be written in matrix form as follows.

$$y(k) = \begin{bmatrix} b_0 b_1 ... b_m a_1 a_2 ... a_n \end{bmatrix} \begin{bmatrix} u(k-d) \\ u(k-d-1) \\ \\ ... \\ u(k-d-m) \\ -y(k-1) \\ -y(k-2) \\ ... \\ -y(k-m) \end{bmatrix} \tag{2.35}$$

where first part of matrix product is $P^T(k)$ and second is $M(k-1)$ Similarly the estimate $\hat{y}(k)$ will be given by:

$$\hat{y(k)} = \hat{P}^T(k)M(k) \tag{2.36}$$

where *a's* and *b's* are estimated that form the part of $\hat{P}^T(k)$

$$\hat{P}(k) = \hat{P}(k-1) + F(k-1)M(k-d)[y(k) - \hat{P}^T(k-1)M(k-1)] \tag{2.37}$$

with

$$F(k) = F(k-1) - \frac{F(k-1)M(k-d)M^T(k-d)F(k-1)}{1 + M^T(k-d)F(k-1)M(k-d)} \tag{2.38}$$

where $F(k)$ is symmetric matrix of $(m+n+1)$ by $(m+n+1)$. Once the $\hat{P}(k)$ is computed, the algorithm proceeds as explained earlier. Therefore $\hat{P}(k)$ yields estimated control and stability matrix coefficients while Genearilized Bezout's Identity(GBI) produces input sequence(equation **??**).

Figure 3.8 shows the architecture of discrete Indirect MRAC.

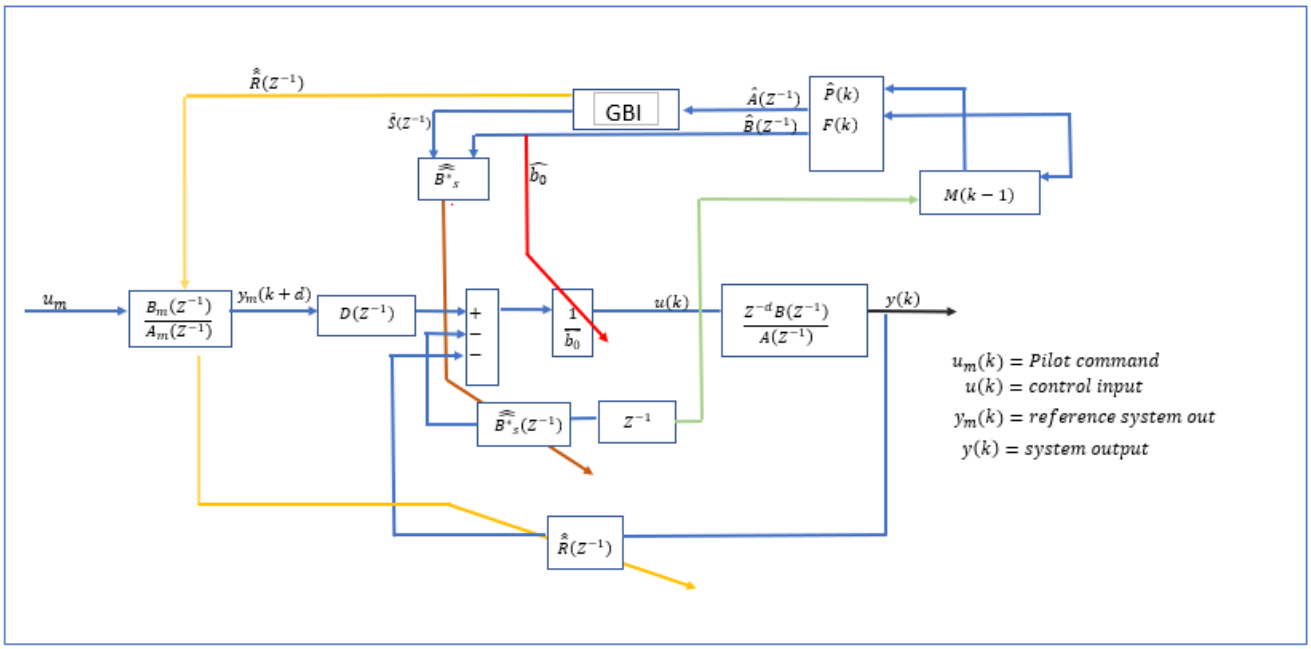


Figure 2.3 Discrete Direct Model Reference Adaptive Control Architecture

Chapter Three

Development of Research Testbed

Combining interest and theory in to a practical problem ensured a steep learning curve, demanding using earlier experience and expanding ones horizon. This chapter highlights the electronic components that goes on the aircraft to make it autonomous and capable for collecting the flight data required for systems identification.

3.1 Airframe

Rascal 110 is a single engine, high wing, 9 ft wingspan, balsa wood airplane. Originally designed as a radio-controlled airplane, the Rascal 110 was chosen for conversion to UAS mainly due to its high gross weight and ample cabin space. "Take-offs and landings are spectacular in their smoothness, and when it comes to what goes on in the air, the Rascal 110 is positively elegant"(Tower hobbies, Rascal 110 manual) .

Figure 3.1 Rascal 110 Airframe

3.2 Propulsion

The power plant was chosen as a AXI 5345/18 HD GOLD Line electric motor with a 20"x13" propeller for longer flight time. The motor is a brushless DC motor that can draw up to 75 amps and operates at 171 Kv (171 RPM/V). It can handle up to a 12-cell lithium polymer (Li-Po) battery. The 20x13 APC propeller means it has a diameter of 20 inches and a pitch of 13 degrees at 25 percent of the length of the radius.The electronic speed controller(ESC) selected with this motor is Jeti Spin 99 Pro Opto Brushless.This ESC can support a continuous draw of 99 amps and a max current draw of 109 amps. It is important that the proper ESC, motor, propeller, and battery combination is selected to meet the needs of the desired performance. There are 4 batteries used on board the Rascal. Two 6-cell Li-Po batteries connected in series generate approximately 50V to power the AXi 5345/18HD motor. The servos and on-board computer are powered by two 3-cell (11.1V) LiPo batteries that uses battery eliminator circuit(BEC) for converting it to 5V. BEC is used for delivering full electrical power that is needed for electronics without any voltage drops.

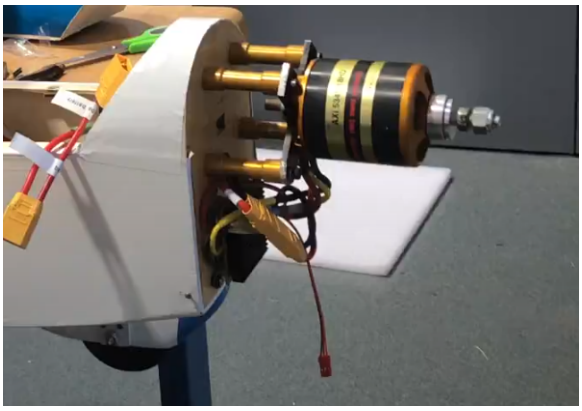


Figure 3.2 Motor mounting on the aircraft

3.3 On-board Electronics

3.3.1 Transmitter and Receiver set-up

The Transmitter used to by the RC pilot is a FrSky 7-channel receiver. The transmitter is integrated with RF Long range module This transmitter is used due to its many channels and ability to integrate with Long range RF module. The receiver obtains the signal from the transmitter and then feeds the signal to both the Pixhawk and an 8-channel RC/RX multiplexer by Cytron Technologies. The multiplexer board can receive signals in PWMs. Its function is to switch signals from RC mode to On-board Computer(OBC)/Autonomous mode.

3.3.2 On-Board Flight Computer (OBC)

The primary flight computer is a high performance, single board computer(SBC): PCM-3356 by Advantech. This is a PC-104 type computer that has a capability to join other boards to perform necessary processes. The PCM-3356 is the primary(and a companion) computer of the this UAV system. It gives the vehicle the ability to process and save large amounts of data and can be used to run the real-time algorithms. The system features an AMD LX800/500 MHz and LX600/366 MHz processor by Geode™. Also included are three RS-232/422 ports and four USB 2.0 ports. In addition to the use of the PCM-3356, an analog board MM 16R-AT by Diamond Systems® was assembled to provide inputs for analog sensors and convert it to digital output. The computer with analog board were assembled and then placed inside a 3D printed enclosure with input/output ports to allow for external devices to be easily connected to the PCM-3356.PCM 3356 is loaded with external bootable flash drive with a real-time simulink code. The PCM 3356 is connected to Pixhawk Cube 2 that is the flight computer.

Pixhawk cube 2 is a micro controller that has built-in sensors.These built-in sensors

Figure 3.3 PCM 3356 and Analog board enclosed in 3D Printed Box

include accelorometer, gyroscope, and GPS. In this research experiment, primary function of Pixhawk is to receive servo commands from transmitter through receiver send them to OBC. Second purpose of Pixhawk is using as data acquisition system. It records and sends all the values from sensors to OBC for redundancy purpose, more like a back-up system in case the other external sensors fail to record the data. Pixhawk comes with built-in Extended Kalman Filter (EKF) that computes Euler angles.

In order for OBC to read messages from Pixhawk, it has to first go to TTL(Transistor to Transistor Logic) to RS232 converter. This is because Pixhawk communicates in TTL which is a binary logic that uses voltages between 0V and +5V while RS-232 port on most PCs typically read voltages from -13V to +13V. The RS-232 to TTL converter changes voltages so that the two systems can communicate.

3.3.3 Servo Set-up

Pololu mini maestro 18 is a servo controller that delivers the control commands from on-board computer to servos in pulse-width modulation (PWMs) through RS232 to TTL converter. It actuates the servos on the aircraft with resolution of $0.25\mu s$

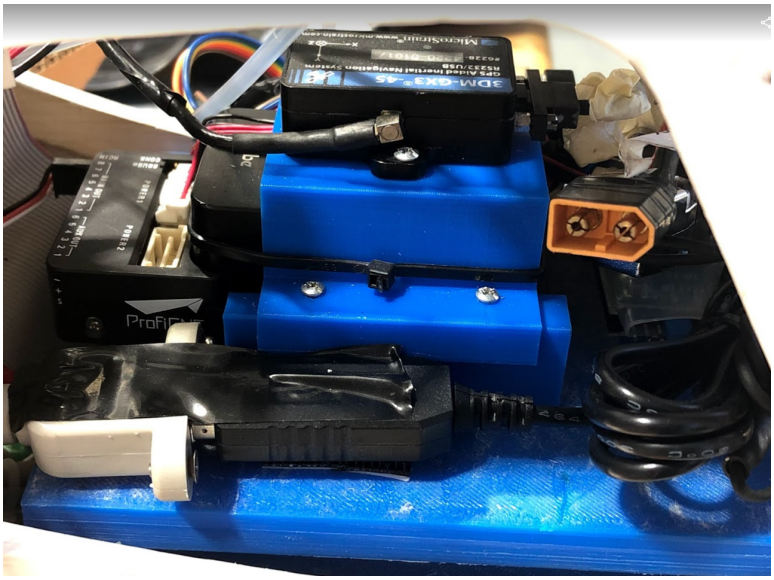


Figure 3.4 IMU, Pixhawk and OBC Set-up

3.3.4 Sensors Package

MicroStrain® 3DM-GX4-45™ INS is one of the sensors that is read by the PCM-3355 on-board the aircraft. This sensor provides highly accurate measurements of the aircraft attitude ($\pm 0.8^{o}$), angular rates, and accelerations. It uses an EKF to provide more accurate results and, to compute GPS location ($\pm$5m), velocities ($\pm$0.1m/s) as well as pressure altitude under high g maneuvers. This sensor is the selected because it is the smallest and for its ease of use, light weight, high accuracy, and has the lowest power of GPS/INS available. The MicroStrain® automatically compensates for vehicle noise and vibration, and does not need field calibration due to automatic magnetometer calibration and anomaly rejection that is estimated and compensated by the EKF. The 3DM-GX4-45™ architecture has been carefully designed to substantially eliminate common sources of error such as hysteresis induced due to temperature changes and sensitivity to supply voltage variations. For redundancy , sensors from Pixhawk are used It has 3 sets of Inertial Measurement Unit (IMU) sensors for extra redundancy and two sets of IMU are vibration-isolated mechanically, reducing the effect of frame vibration to state estimation IMUs are temperature-controlled by on-board heating resistors, allowing optimum working temperature of IMUs. The entire flight management

unit(FMU) and IMU are housed in a relatively small form factor (a cube). It comes with RS-232 and USB 2.0 communication interfaces and both of these are connected to PC/104 computer. A detachable antenna is plugged to the sensor and position on top of the aircraft to obtain GPS satellite links.

Figure 3.5 Attitude, Gyroscope, Accelerometer, GPS sensor

3.4 Real-Time Operating System

xPC Target is a host-target solution for prototyping, testing, and deploying real-time systems using standard PC hardware. "It is an environment that uses a target PC, separate from the host PC, for running real-time applications" It enables the user to load MATLAB Simulink® models on to physical systems and execute them in real-time.

There are different ways to configure this on target computer i.e. through Ethernet cable and stand alone mode. In former, a host computer is needed to build and start/stop you model. SDK 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 Real-time Operating System (RTOS) pre-loaded while in the later the Target PC runs completely autonomously. The model is no longer downloaded from the host, but is stored on the CF boot disk. When the computer starts, the model begins

running automatically, with an indefinite stop time.

Figure 3.6 Real-Time Operating Software

3.4.1 MAVLink Communication

The Micro Air Vehicle Link (MAVLink) is a communication protocol for unmanned systems(eg.drones).It is developed in MATLAB Simulink to establish a communication between Pixhawk Cube and PCM-3356. Pixhawk sends RC messages from transmitter to PCM-3356. It is a light weight, well-established message binary serialization protocol. Binary serializa-

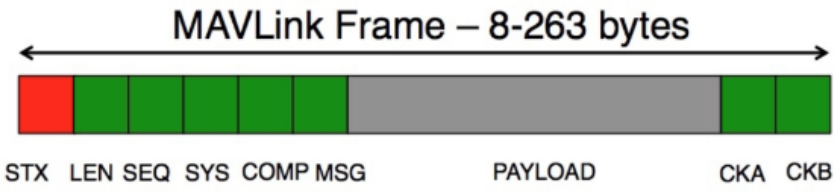


Figure 3.7 MAVlink Message frame

tion means that the content of the message is transformed into a sequence of bytes to be transmitted through the network. The receiver(in this case is OBC - PCM 3356) of the serialized message performs its de-serialization(i.e. decryption) in the opposite direction to reconstitute the original message sent. Each MAVlink message has a header appended to

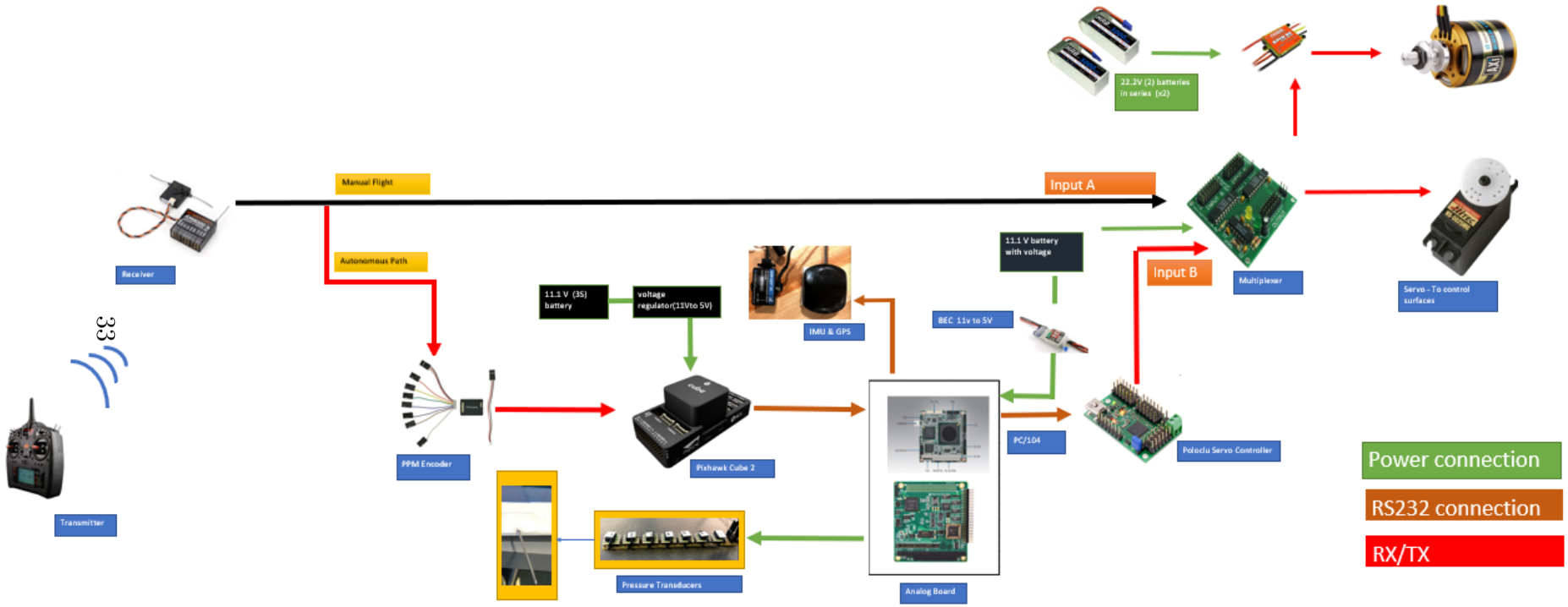
the payload message. The header contains the information related to the message whereas payload contains the data itself that has to be transmitted.

3.5 Fail-safe

Most vital part of any aircraft operation is designing a fail-safe for its flight. This helps the RC pilot recover the unmanned aircraft if there is any kind of glitch in the autonomous system. For this reason, Rascal 110 was instrumented to operate in two ways, first is manual flight where a pilot has direct control over the aircraft(fail-safe mode) and second is autonomous mode. Cytron Multiplexer 8 channel is used for this purpose. It switches the signal from one input to another for servo output. Autonomous mode passes through flight computer to servos while manual flight from transmitter-receiver to servos which allows RC pilot to abort the test anytime during a critical situation and land safely.

Figure 3.8 Multiplexor

Figure 3.9 summarizes the signal flow and the assembly of all the electronics that goes on Rascal 110.


22.2V (2) batteries in series (x2)
Manual Flight
Input A
Receiver
Autonomous Path
11.1 V (3S) battery
voltage regulator(11Vto 5V)
11.1 V battery with voltage
Input B
Multiplexer
Servo - To control surfaces
IMU & GPS
BEC 11v to 5V
PPM Encoder
Pixhawk Cube 2
PC/104
Pololu Servo Controller
Transmitter
Pressure Transducers
Analog Board
Power connection
RS232 connection
RX/TX

Chapter Four

Development and Calibration of Multihole Air Data Probe

An air data probe (ADP) is an effective light weight solution that has an ability to compute angle of attack, angle of sideslip and true airspeed for aircraft based on the static and total pressure measured. The basic principle of operation, which most multihole probes have in common, is the ability to determine velocity magnitude and direction from a measured pressure differential (Zillac,1989). These probes are an extension on the pitot-tube notion – i.e. the knowing the relative position of each pressure port allows calculation of both a flow magnitude and direction (Crawford, 2011). Three-hole probes are measure a 2-dimensional flow i.e. a single flow angle. Five and seven-hole probes are capable of fully measuring a 3-dimensional velocity field i.e. two flow angles. The two extra holes on the 7-hole probe allows to measure higher angles of attack almost to 80 degrees relative to their axis. When combined with computerized data acquisition system, they are capable of taking data rate of nearly two data points per second. Additionally, these probes are too small to disturb any flow stream that they are measuring but due to probe's small size, it suffers from from inherent manufacturing defects. As a result, the probe must be calibrated before it can be used for measuring any flight data. (A.A. Gerner and C.L. Maurer, 1982)

4.1 Method of Calibration

A seven-hole probe was built and calibrated in house. The probe is 12 inch in length with 0.25 in total radius. Figure 4.1 and 4.2 shows the catia sketch of the air-data probe assembly and assembly of sting, rod and mounting block with pneumatic tubes respectively.Microscopic view of the ADP is seen in Figure 4.4. In-flight, the flow stream pressure at each hole of ADP are recorded by pressure transducer.These pressure transducers are shown in Figure 4.3. LP-series-Analog pressure sensors were used for this purpose. They are surface mountable pressure sensor package with compensated analog output suitable for ultra-low pressure sensing application. They measure pressure ranging from -0.15 to 0.15 psi. They are available in two different types, Gage and Differential. Later type was used for this research. The differential pressure is measured in an analog electrical signal. These signals are sent to analog-digital board where using the transfer function formula from Merit Data sheet, voltage is converted to meaningful pressure reading. Analog-Digital board supplies 5V of power to these sensors.

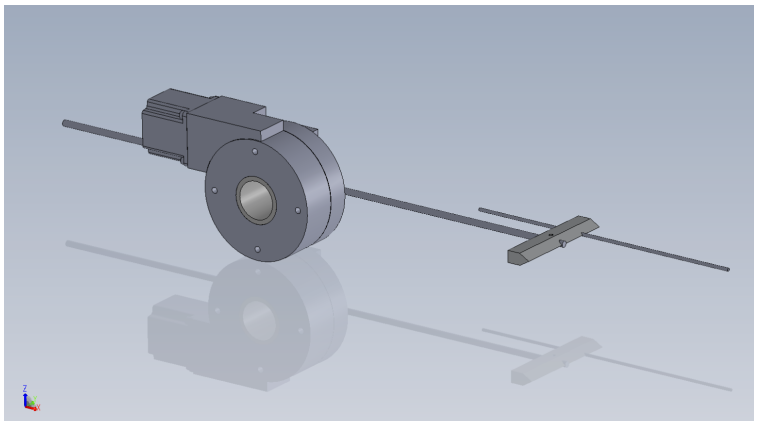


Figure 4.1 Catia Sketch of Air-Data Probe Assembly inside the Wind-Tunnel

For instance, voltage measured by pressure transducer is 3.70V

Pmin = -0.15psi , Pmax = 0.15psi, Vout = 3.70V

Vmin = 0.5V , Vmax = 4.5V

Figure 4.2 ADP is assembled with sting, rod and mounting block with pneumatic tubes

$$P_{psi} = (P_{max} - P_{min}).\frac{V_{out} - V_{min}}{V_{max} - V_{min}} + P_{min} \tag{4.1}$$

where P_{psi} is pressure recorded in pound per square inch (PSI).

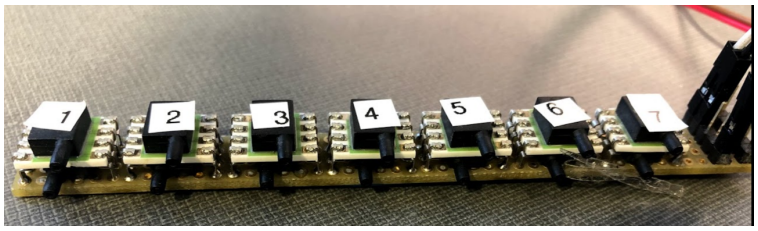


Figure 4.3 Differential Pressure Sensors

$$P_{psi} = (0.15 - (-0.15)).(\frac{3.25 - 0.5}{4.5 - 0.5}) + (-0.15) \tag{4.2}$$

$$P_{psi} = 0.5625 \tag{4.3}$$

4.2 The Wind Tunnel Testing

The ADP calibration was performed using new sub-sonic FluiDyne Wind Tunnel in Micaplex Research Park at ERAU. The 7-hole ADP was calibrated from the range -30 to 30 degrees in both angle of attack α and angle of sideslip β. The calibration matrix was nominally in

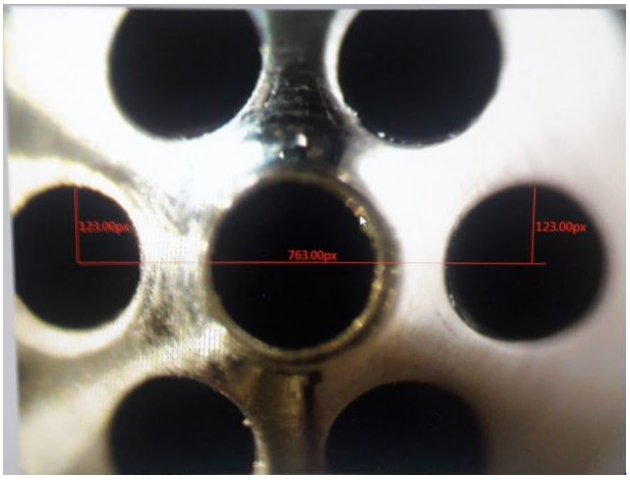
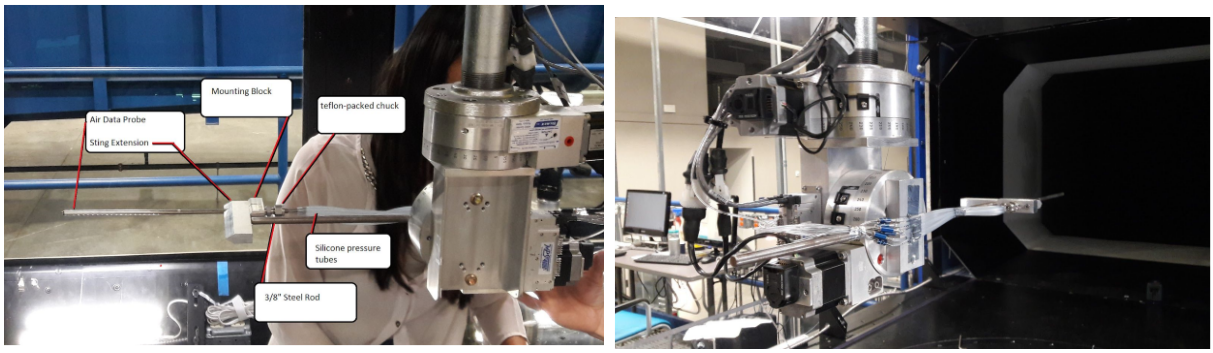


Figure 4.4 Microscopic view of ADP for ensuring a 180 degree Orientation in the Mounting block

steps of 2-degree increments (961 measurement points), with possible refinement in the step size based on the initial outcomes from the tests. For the wind tunnel experiment, different set of multi-channel pressure sensor bank and pressure tube adapters were used to calibrate the ADP. This was done by mounting a special angular calibration fixture from the ceiling of the wind tunnel, this fixture being used to precisely position the angle of attack and angle of sideslip in the flow of the wind tunnel. The test set-up is shown in Figure 4.5.The sting rod was the main device that held the ADP out into the slipstream of the wind tunnel. The steel rod (Figure 4.5) was originally was 1/4" thick, but was increased to 3/8". The thicker rod would further ensure that there would be no unnecessary oscillation inside the wind tunnel. The wind tunnel can deliver a maximum flow speed of up to 420 ft/s. The test was done at a flow speed of 150 ft/s. The fixture is controlled through LabView. It is very important to align the port number 1 and 4 with X-Z plane of Wind Tunnel test section. The data sample rate was 50Hz and each angle increment was sampled for 15 seconds with a 5 second rest time. From the measurements of the corresponding 7 pressures at each angle of attack and angle of sideslip, then the full calibration matrix for the ADP can be calculated. This calibration matrix is then used in flight to measure the angle of attack and angle of sideslip of the aircraft. Two tests were conducted for this process.



(a) **(b)**

Figure 4.5 ADP setup in test section of Wind Tunnel

Test 1: This is the Partial sweep. This was performed by increasing one angle (angle of attack or angle of sideslip) over the full range to obtain a general idea of the calibration curve and the quality of the curve fit. This test required less than 30 minutes of wind-on time, and verified that the calibration fixture was operated properly that allowed an examination of the general nature of the pressure responses and the expected calibration curves. Test 2: This is a full sweep test. This test examined the full range of +/- 30 degrees for angle of attack and sideslip at 2-degree increments (961 measurement points). Each calibration point lasted approximately 20 seconds to provide accurate calibration data; 5 seconds of probe move and wait time and 15 seconds for the pressure data to be recorded. The pressure data are then ensemble averaged over the 15 seconds. The results were normalized with dynamic pressure. This test lasted for 4hours wind-on time. Figure 4.6 shows the air-data probe assembled on the left wing of the airplane. The probe was placed in a position that assured that there was no prop wash affecting the measurements.

4.3 Post-Processing data

The pressure data for each each hole on ADP is collected and post-processed using MATLAB® and MATLAB® SIMULINK. For low flow angles, it is advisable to define dimensionless pressure coefficients which utilizes all seven measured probe pressures and are sensitive

Figure 4.6 ADP setup in the wing

to change in flow angularity with respect to probe's x-axis. The pressure coefficients that are sensitive to change in angle of attack in x-z plane is defined as

$$C_{\alpha 1} = \frac{P_4 - P_1}{P_7 - Avg(P_{1-6})} \tag{4.4}$$

In equation 4.4, numerator is angular flow difference in opposite port pressures while denominator pseudo dynamic pressure since this is obtained from the difference of 7th i.e. central port and average six surrounding pressure from P1 to P6. Central port gives the total pressure while P1 to P6 yields approximated static pressure. Other coefficient of pressure are

$$C_{\alpha 2} = \frac{P_3 - P_6}{P_7 - Avg(P_{1-6})} \tag{4.5}$$

$$C_{\alpha 3} = \frac{P_2 - P_5}{P_7 - Avg(P_{1-6})} \tag{4.6}$$

These coefficients are then resolved into the α_T - β_T reference system where α_T is defined as projection on vertical plane of the angle between the velocity vector and the probe's axis. β_T is defined as projection on the horizontal plane of the angle between probe's axis and the relative wind. For resolving the above coefficients, contribution of each coefficient is weighed in determining alpha and beta.

$$C_\alpha = \frac{1}{3}(2C_{\alpha 1} + C_{\alpha 2} - C_{\alpha 3}) \tag{4.7}$$

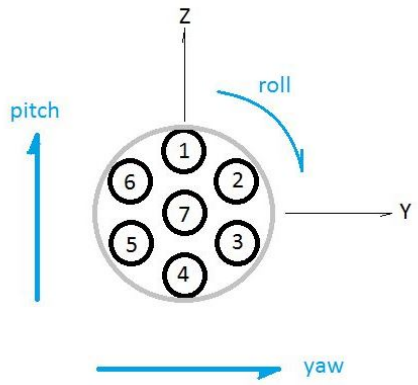


Figure 4.7 Numbering of ADP holes to compute Angle of Attack(Pitch Axis) , Sidelsipe (Yaw Axis)

$$C_\beta = \frac{1}{\sqrt{3}}(C_{\alpha 2} + C_{\alpha 3}) \tag{4.8}$$

In equation 4.7, it contains all the three coefficients to determine C_α where $C_{\alpha 1}$ has the maximum significance because it directly falls on axis of interest. The equation4.8 has equal weights from $C_{\alpha 2}$ and $C_{\alpha 3}$ while it does not take into account $C_{\alpha 1}$ since it perpendicular to axis of β_T.

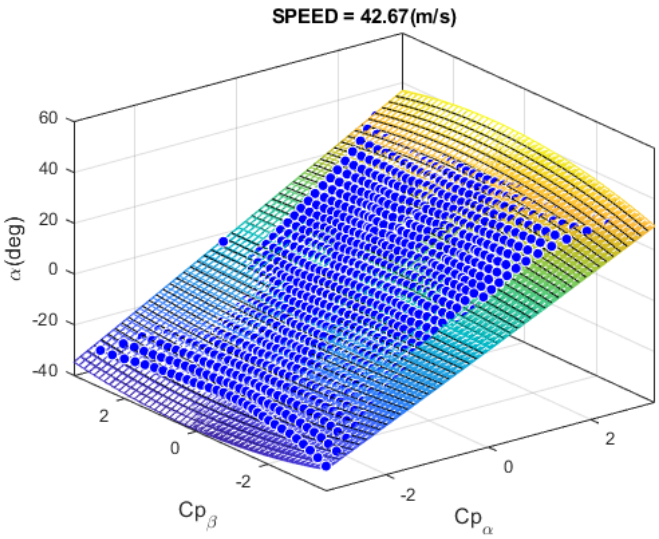


Figure 4.8 Angle of Attack versus Coefficient of Pressure (alpha and beta)

The MATLAB was used to post-process the data from Wind-tunnel. The data was processed using the curve fitting tools.Mainly four plots were generated from this process.

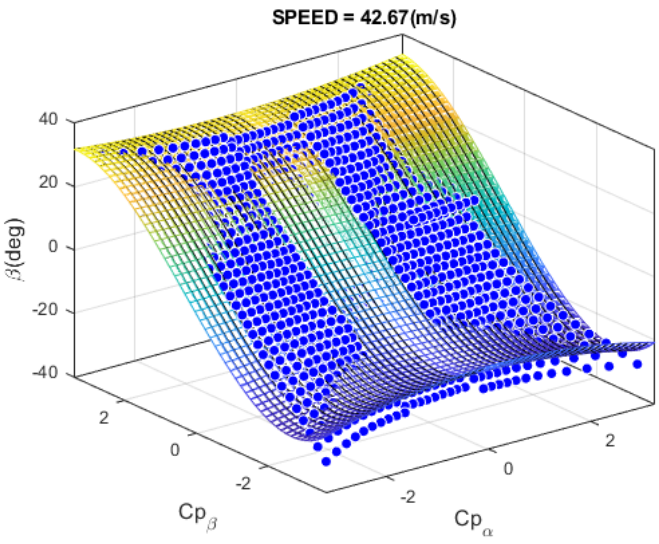


Figure 4.9 Angle of Sideslip versus Coefficient of Pressure (alpha and beta)

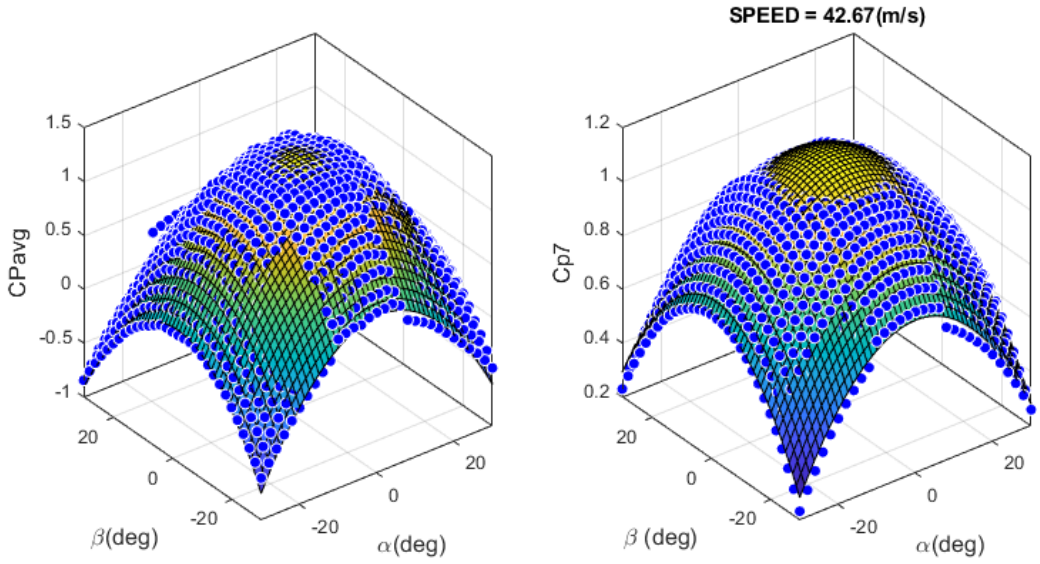


Figure 4.10 Average Pressure Coefficient(Left) and Total Pressure Coefficient(Right) versus Alpha and Beta

First important plot is angle of attack versus coefficient of pressures for alpha and beta (Figure 4.8) followed by angle of sideslip versus coefficient of pressures for alpha and beta (Figure 4.9). The other two graphs that were generated are total pressure versus alpha, beta and Average coefficient of pressure versus alpha,beta (Figure 4.10).

The blue points on graphs show the data set obtained from the wind-tunnel. From the

measurements of the corresponding seven pressures at each angle of attack and angle of sideslip, then the full calibration matrix for the ADP can be calculated. This calibration matrix is then used with a separate set of pressure transducers to measure the angle of attack and angle of sideslip of the aircraft during actual flight.

The aircraft is also mounted with pitot-static tube from 3DR. It measures the static pressure of the incoming air stream. The tube from pitot-static is connected to the static ports of the pressure transducer. This helps in determining the true airspeed of the aircraft. The true airspeed is computed from the dynamic pressure measured which is the difference between the total and static pressure(according to Bernoulli's principle).

$$P_{total} - P_{static} = \frac{1}{2}\rho V^2 \tag{4.9}$$

where ρ is the density at which the aircraft is flying, V is the airpeed or relative velocity of the aircraft Hence,

$$V = \sqrt{\frac{2(P_{total} - P_{static})}{\rho}} \tag{4.10}$$

Chapter Five

Systems Identification and Flight Testing

The determination of the parameters describing the aerodynamic behavior of the aircraft from flight data is one of the approaches(Marcello Napolitano ,2012) that are available to extract a system's mathematical model. The drawback of this method is that it requires prior estimation of aircraft's information to conduct the test flights. Nevertheless, identifying system through test flights has its own advantages like it validates the methods that were previously used for the estimation of aerodynamic coefficients, evaluating performance of flight control laws to comprehend closed-loop system and also testing and assessing the new modified aerodynamic coefficients. The overall idea of this process is to know the input and record the output from flight test and using these two set of information, identify the mathematical model. It is known as Parameter IDentification(PID). A conceptual block diagram of the PID process is shown in Figure 5.1. Once initial estimates from flight data is computed, a technique is used to minimize the error between actual flight data and the output of the state-space/mathematical model.This is known as validation technique.

5.1 Rascal 110 DATCOM Analysis

DATCOM stands for DATa COMpendium is a digital tool that is used for preliminary design operations for an aircraft. This is the computational fluid dynamics (CFD) based analysis

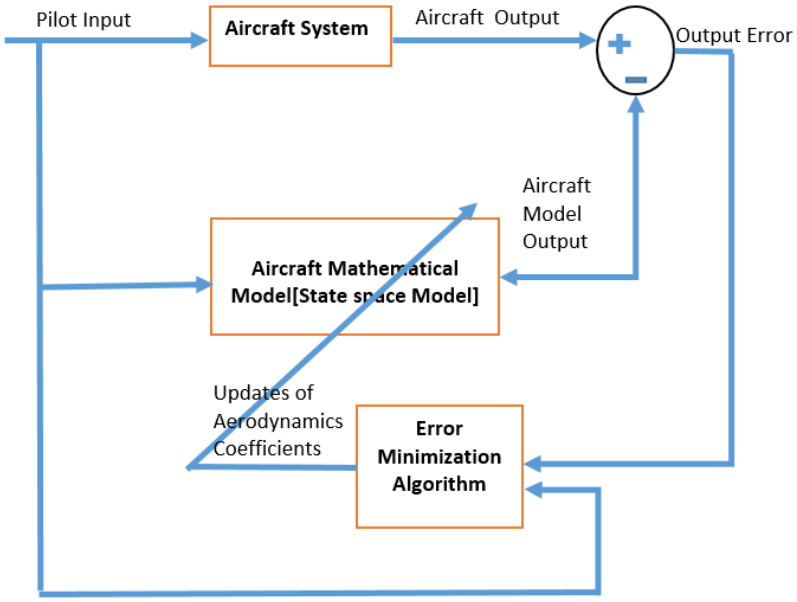


Figure 5.1 Parameter Identification Process

of conventional airplane subsonic configurations featuring wing,fuselage, horizontal tail and vertical tail. It is a low budget approach for initial estimation. Digital DATCOM computes static stability, high lift and control devices using the massive wind tunnel results that are mainly obtained by US Air force and NASA. These wind tunnel results analyze virtually all possible aircraft configuration at subsonic, transonic and supersonic speeds. The DATCOM requires geometric properties, general flight conditions, mass properties and inertia properties of the aircraft model that are shown in Table 5.1. In previous research conducted at Advanced Dynamics and Control Lab, ERAU, Digital DATCOM analysis on Rascal 110 was performed (Lyons, Bredon 2013). The stability and dynamics derivatives obtained are shown in Table 5.2. Using the derivatives obtained in Table 5.2, trim conditions of the aircraft were obtained in MATLAB Simulink as shown in the table 5.3

The aircraft system is linearized at trim conditions resulting in a longitudinal and lateral state space matrix model. State-space is set of first-order differential equations represented mathematically as Equation 5.1. In this case coefficients in differential equations are made

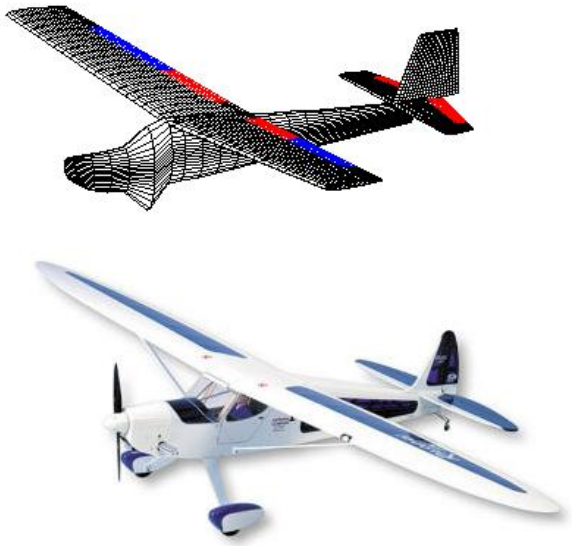


Figure 5.2 3D DATCOM Model

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

Table 5.1 Digital Datcom Input File Parameters

of aerodynamic stability derivatives, mass, and inertia characteristics of the airplane. These state-space models are computed in continuous time system which are then converted to discrete system with sampling time = 0.01 as shown in Equations 5.3 and 5.2 using Zero-hold method. To validate this step, state-space model was also computed analytically using lateral and longitudinal linearized set of equations (Nelson,2018).

Comparing Table 5.4 and 5.5, the Eigen values, natural frequency and damping ratio of all the dynamic modes are very similar from simulation and analytical. These models were further used design the flight test inputs.

Longitudinal Stability Derivatives	per rad	Lateral/Directional Stability Derivatives	per rad
C_{L_o}	0.4940	C_{γ_β}	-0.3198
C_{L_α}	5.9730	C_{γ_p}	-0.1138
C_{L_q}	4.8850	C_{L_β}	-0.1138
C_{D_O}	0.0310	C_{L_P}	-0.5087
C_{M_O}	0.0323	C_{N_β}	0.0127
C_{M_α}	- 0.3217	C_{N_P}	-0.0380
C_{M_q}	-11.000	C_{N_r}	-0.0378

Table 5.2 Stability Derivatives

Parameter	Value	Units
Altitude	275	m
Airspeed	31	m/s
Angle of Attack	-6.7	degrees
Elevator Deflection	-6.3	degrees
Thrust	19.27	N

Table 5.3 Trim Conditions for Rascal in Simulation

$$x(k+1) = Ax(k) + Bu(k) \tag{5.1}$$

$$\begin{bmatrix} \alpha(k+1) \\ v(k+1) \\ q(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} 0.7286 & -0.0003616 & 0.015 & 3.734e-05 \\ -0.5503 & 0.9966 & -0.007061 & -0.1958 \\ -0.4864 & 8.953e-05 & 0.8854 & -5.397e-06 \\ -0.00523 & 5.504e-07 & 0.01885 & 1 \end{bmatrix} \begin{bmatrix} \alpha(k) \\ v(k) \\ q(k) \\ \theta(k) \end{bmatrix} + \begin{bmatrix} 0.07551 \\ -0.02303 \\ 7.785 \\ 0.07944 \end{bmatrix} \delta_e(k) \tag{5.2}$$

Dynamic Mode	Eigenvalue	Natural Frequency (rad/s)	Damping ratio
Short Period	-9.02±2.25i	9.3	0.970
Phugoid	-0.0419±0.445i	0.447	0.11
Dutch Roll	-0.527±3.16i	3.2	1.90
Roll	-11.9	-	-

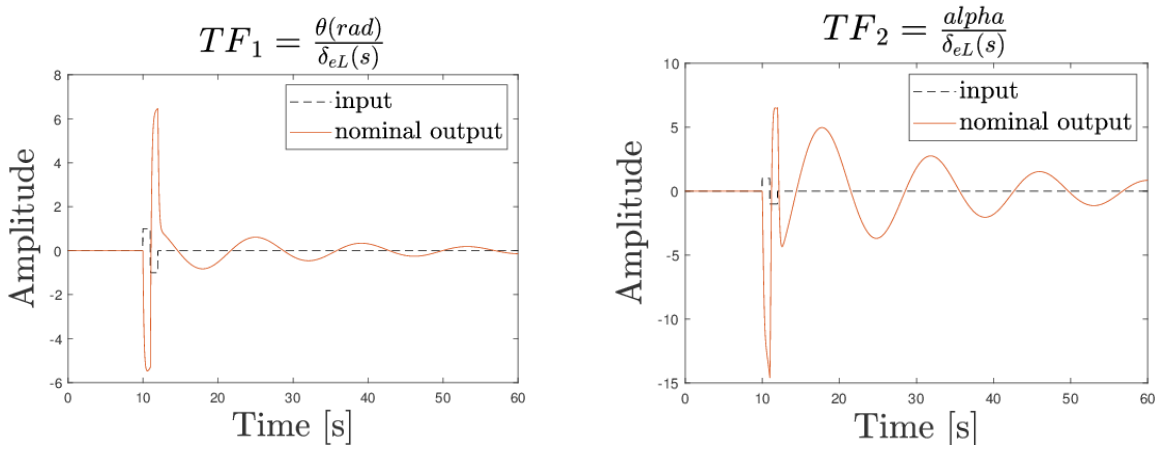
Table 5.4 ANALYTICAL: Eigenvalues, Natural Frequency, and Damping of Rascal's dynamic models

Dynamic Mode	Eigenvalue	Natural Frequency (rad/s)	Damping ratio
Short Period	-10.7±2.02i	10.9	0.983
Phugoid	-0.066±0.216i	0.226	0.295
Dutch Roll	-0.428±3.56i	3.59	1
Roll	-14.6	-	-

Table 5.5 SIMULATION: Eigenvalues, Natural Frequency, and Damping ratio of Rascal's dynamic models

$$\begin{bmatrix} \beta(k+1) \\ p(k+1) \\ r(k+1) \\ \phi(k+1) \end{bmatrix} = \begin{bmatrix} 0.985 & 0.0002299 & -0.01968 & 0.00735 \\ -0.7618 & 0.7937 & 0.007885 & -0.002937 \\ 0.1168 & -0.01558 & 0.9815 & 0.0004262 \\ -0.007932 & 0.01786 & 5.371e-05 & 1 \end{bmatrix} \begin{bmatrix} \beta(k) \\ p(k) \\ r(k) \\ \phi(k) \end{bmatrix} + \begin{bmatrix} 0.0004802 & 0.01058 \\ 3.599 & 0.07604 \\ -0.04921 & -1.06 \\ 0.03737 & 0.0008049 \end{bmatrix} \begin{bmatrix} \delta_a(k) \\ \delta_r(k) \end{bmatrix} \tag{5.3}$$

Figure 5.3 shows the responses of two calculated transfer functions from simulation derived state-space longitudinal model. The two transfer functions plots the responses of pitching angle and and angle of attack with respect to change in elevator deflection.



(a) Change in Pitching angle due to an elevator input

(b) Change in angle of attack due to an elevator input

Figure 5.3 Output Responses from Transfer Functions of Linear Model

5.2 Flight Testing

Various set of flight tests were performed with the UAS research platform. The flight tests are part of an extensive flight test program initiated at ERAU to support research on the design, development, and testing of intelligent adaptive flight control systems and health management algorithms. The 'Academy of Model Aeronautics' (AMA) Daytona Beach field was chosen for the flight test program. Approximately 1400 ft long and 1300 ft wide, the field has enough space to perform the necessary maneuvers. It has a single and hard-surface runway located on the east side. Figure 5.6 shows a satellite image of the field. The flight trajectory selected for the flight tests was a rectangle, as shown in Figure 5.6, with maneuvers taking place during the long straight portions of the track. These longer portions allow the aircraft to respond to the stimuli for enough time so that the aircraft's dynamic response can be observed.

The maneuvers were coded and automatically injected from the OBC. It injects the two types of elevator maneuvers that can possibly excite the short period and phugoid mode of the airplane. The frequency of the maneuvers was based on a preliminary model and was refined as new data became available. The maneuvers consisted of a combination of

Figure 5.4 AMA'S Daytona Beach RC Flying Field

Figure 5.5 Test Flight in progress

doublets, 3-2-1-1 ans step. During each flight, after the pilot enters the maneuver section of the track, the maneuver switch is activated. The OBC then performs a state check that determines if a maneuver can be safely injected at that specific moment. If so, the OBC injects the maneuver's signal to the actuators. For safety reasons, the pilot has, at all times, the ability to override the flight computer and control the aircraft manually. Five seconds prior to the injection of doublet, aircraft is trimmed to ensure that there will not be any additional excitation. The flight tests were conducted early in the morning to avoid any kind of disturbances from high winds. This provided an additional means that during the flight,

aircraft's control surfaces were very close to the trim position. Once the OBC injected the doublet, six to ten seconds were provided to record the aircraft's responses. Sample flight segment of aircraft's responses during the performance of doublet are shown below. Figure 4.5a shows a typical doublet maneuver generated by the OBC in the elevator. The plots generated in Figure 5.7 and 5.8 are the unfiltered data collected during test-flights. Notice how the OBC maintains an autonomously steady state level flight before the maneuver is injected.

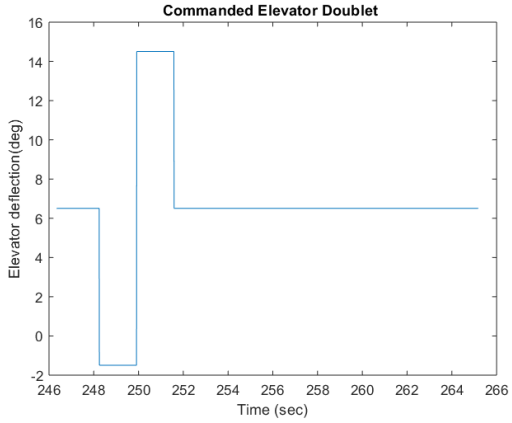
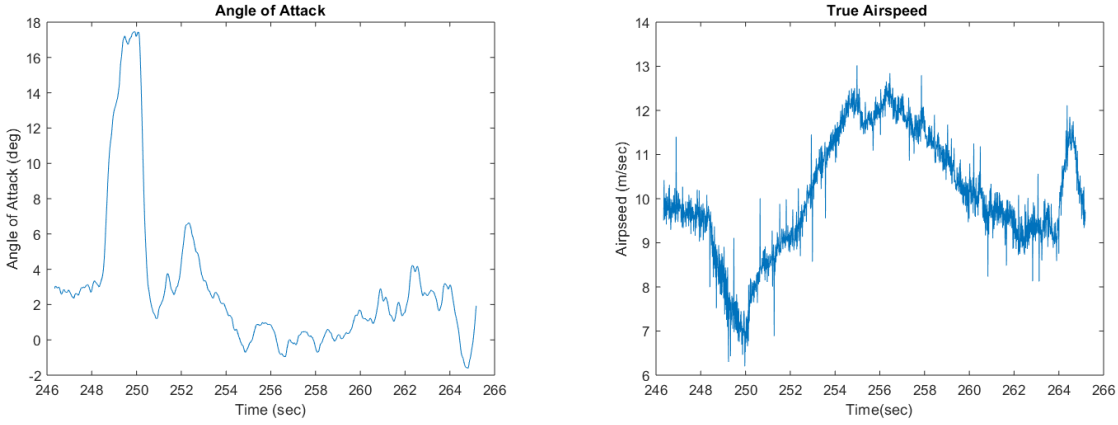
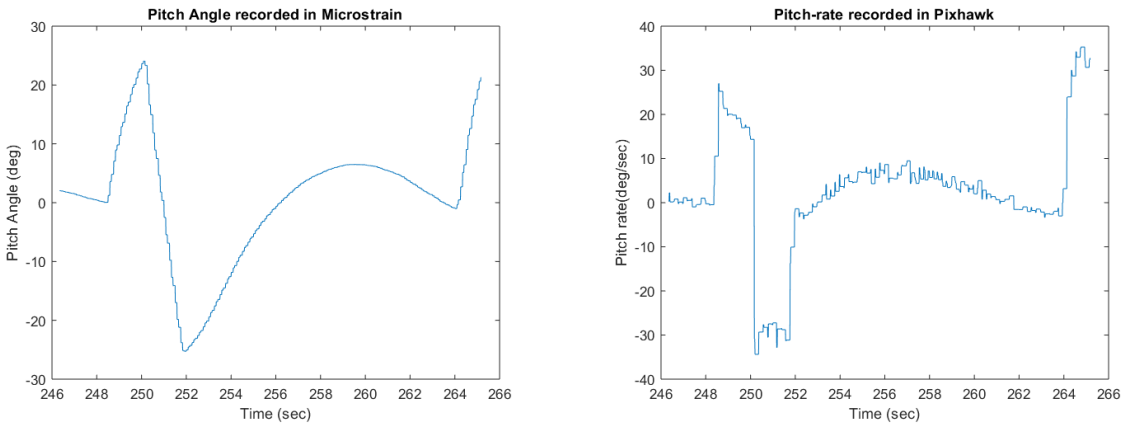


Figure 5.6 Doublet injected by OBC



(a) Angle of Attack **(b)** True Airspeed

Figure 5.7 Responses recorded during and post doublet



(a) Pitch Angle **(b)** Pitch-rate

Figure 5.8 Responses recorded during and post doublet

5.3 Parameter Identification of Rascal 110

Identifying the mathematical model or transfer functions of the system based on its input and output data is known as parameter/systems identification. In this research, input were provided by elevator commands to excite the natural frequencies of the dynamic modes of the aircraft while output were responses recorded from the various sensors during this excitation. The identification process can be conducted off-line or online. Online or real-time estimation are conducted during the flight, mostly using frequency-domain methods. For off-line parameter identification, either time-domain or frequency domain methods can be implemented. PID can be performed using several ways such as SIDPAC(Systems IDentification Program for AirCraft), Systems Identification Toolbox from MATLAB, etc. The key to a successful parameter identification is the apt designing of the experiment that includes proper selection of sensors for data acquisition and proper selection of PID maneuvers The following data channels were used for longitudinal model identification: angular rates from gyroscope, linear accelerations, attitude angles, air-data probe and engine parameters. The calibration of all the sensors were performed in laboratory at ground level and alignment of IMU to aircraft's X axis to ensure the provision of correct data. PID maneuver selection should be done in a manner that can excite the appropriate states to be measured. For example the pilot inputs

determine the operating point of the system and which modes will be excited. The elevator steps excite the short period and phugiod mode while rudder-aileron combination excite the dutch roll. Sometimes there are restrictions on the maneuvers performed for PID such as the frequency and amplitude of the doublet. The time is also significant limitation factor that in specific circumstances during higher angle of attack approach , the airplane cannot sustain there for longer time hence shorter time will be available to collect the data.

Doublet inputs are two sided pulses that are mostly used in this research for systems identification. These doublets were designed at six different frequencies that fell in range of natural frequencies of short period and phugoid obtained rom initial estimation of the state-space from DATCOM. This thesis focuses mainly on computing longitudinal state-space.

5.3.1 Systems Identification Toolbox

The linear parameter identification is performed using Matlab® System Identification (SysID) toolbox. The toolbox estimates the the linear and non-linear models of dynamic systems from input and output data. The SysID toolbox performs four major steps to estimate the model. They are:

1. Collect and pre-process the input data
2. Select a Model Structure
3. Find the best model in a structure.
4. Evaluate the resulting model.

With this tool, segmented flight data was imported into the toolbox. Many flight segments were available for this process but the best were the once that had captured maximum information about airplane dynamics that is short period and phugoid. The very first step is importing the input and output data to SysID toolbox and specifying whether it is time domain or frequency domain. In this case, time domain with 0.01 sec sampling time was

selected. This is sampling time at which data is collected in OBC. Also the user can select the input properties like how the behaviour of input signal was between the samples during data acquisition. It is used when transforming models from discrete-time to continuous-time and when re-sampling the data. Zero-order hold was selected, which indicates that the input was piecewise-constant during data acquisition. Once this is established, input-ouput data can be imported from Matlab workspace. For input, elevator deflection and for output-angle of attack, pitch rate, pitch angle and true airspeed were imported. These data can be pre-processed by removing the means or normalizing the data from measured and input signal. User can also perform the filtering to remove outliers or noise from sensor data.This step is mainly performed for preparing the data for identification as shown in Figures 5.9 and 5.10.

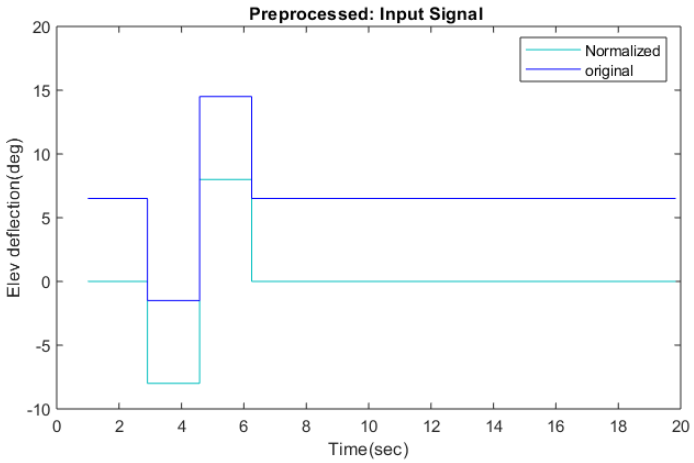
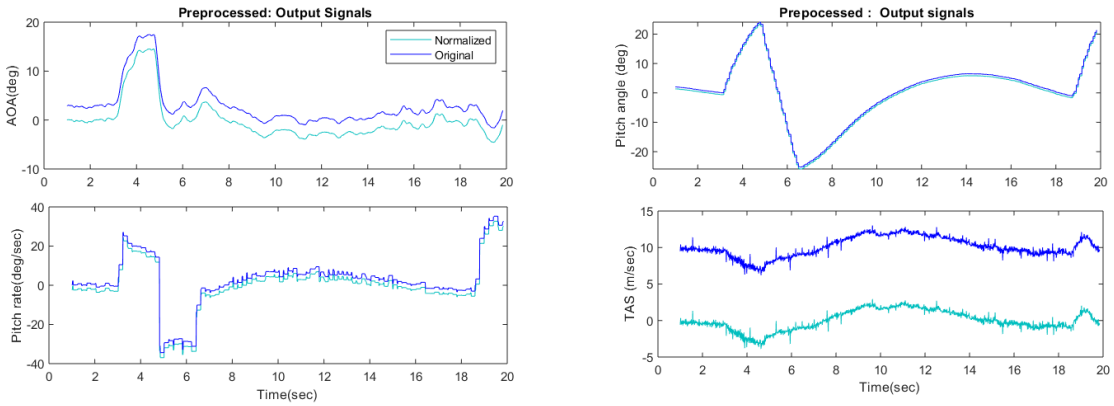


Figure 5.9 Input Signal pre-processing

There were to two sets of data selected, one for estimation while other for validation. Estimation data was dropped into working data and the other set was dropped into validation data box as shown in Figure 5.11

There are different models available for end-user which can be quickly ran through to check best fit model. Of all the models that were tested in this research, state-space and



(a) Angle of Attack(top) and Pitch rate **(b)** Pitch Angle(top) and True Airspeed

Figure 5.10 Output Data Preprocessing

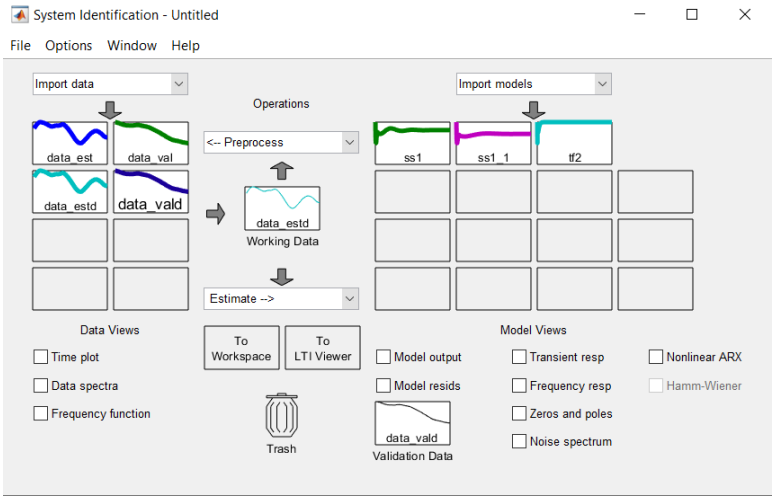
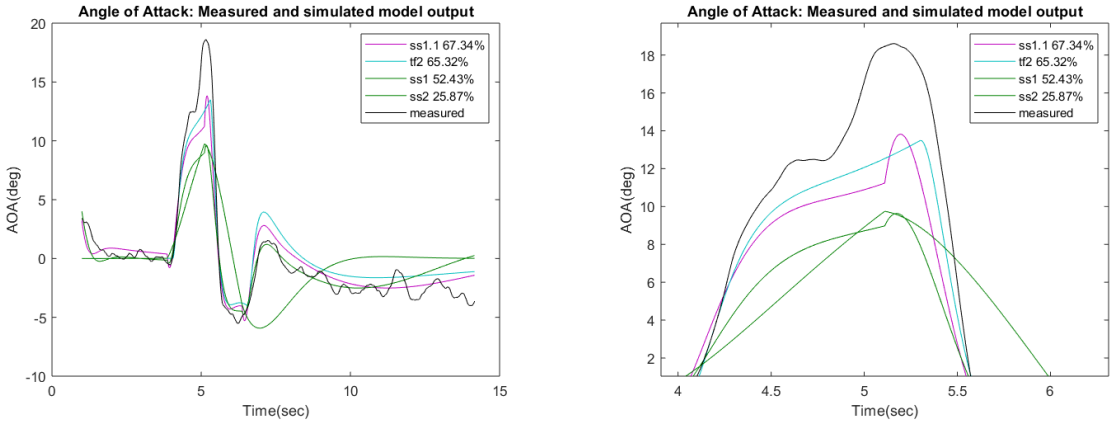


Figure 5.11 System Identification Toolbox

transfer function did fairly well. The difference between these two types is transfer function representation, the model order is related to the number of poles and zeros but for state-space representation, the model order corresponds to the number of states. Once the model was selected, the data was run through the model and posted in the Model Views section. Model Output allows the user to graphical comparison of the simulated output of the channels to the measured data. These models can be evaluated on basis of the best fit percentage that is the accuracy measurement between 0 to 100 percent to give the user a quantitative measure of how the model is performing. The model information output is a state space

representation for which the eigenvalues and eigenvectors are used to assess the stability of the modeled system.

The different processed models are shown below where state-space and transfer function models are compared with best fit percentages. *'ss1'* represents state-space model estimated using 'N4SID' function with prediction focus. 'N4SID' is non-iterative subspace method for estimating linear state space models. This also includes disturbance component K. This model is further refined using 'Refine Existing Model' with simulation focus named *'ss1.1'* in the legend. Comparing overall estimation from all the models, refined state-space model (ss1.1) performs better looking at the best fit values across all the states.



(a) Angle of Attack **(b)** Zoomed-in

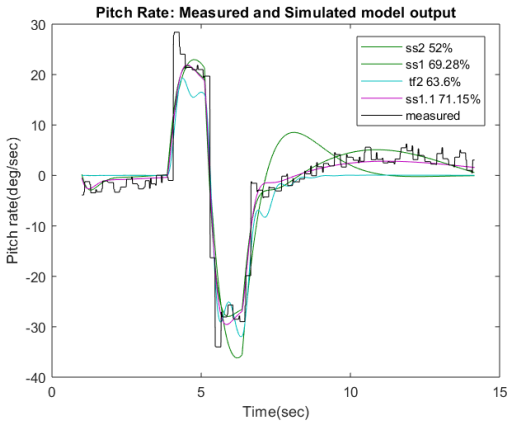
Figure 5.12 Measured and Simulated Output response for Angle of Attack

Longitudinal State-space Model

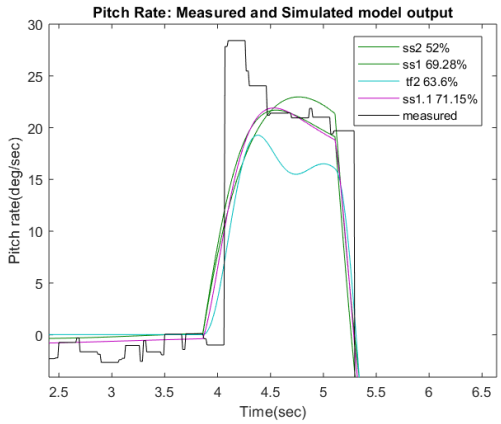
The discrete state-space(ss1.1) that was estimated from the toolbox was imported to Matlab to compute the open-loop poles. The state-space representation is in form of Equations 5.4 and 5.5 where 'K' is the disturbance component

$$x(k+Ts) = Ax(k) + Bu(k) + Ke(k) \tag{5.4}$$

$$y(k) = Cx(k) + Du(k) + e(k) \tag{5.5}$$

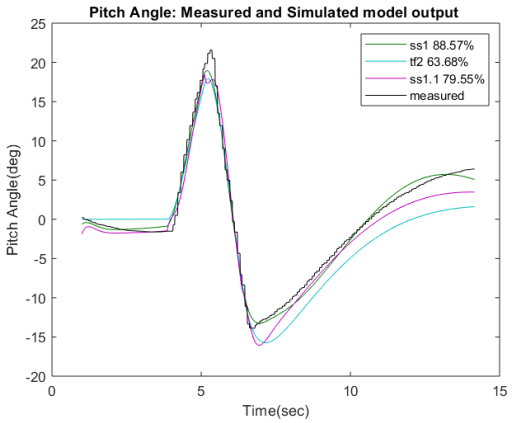


(a) Pitch Rate

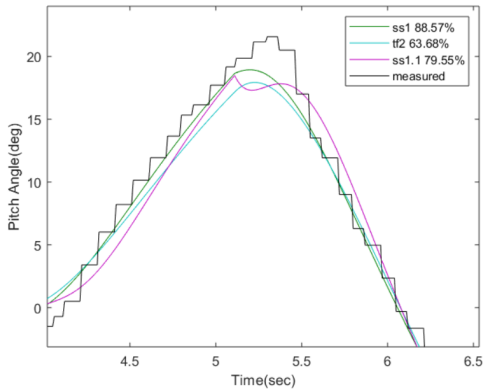


(b) Zoomed-in

Figure 5.13 Measured and Simulated Output response for Pitch Rate



(a) Pitch Angle



(b) Zoomed-in

Figure 5.14 Measured and Simulated Output response for Pitch Angle

$$\begin{bmatrix} \alpha(k+1) \\ q(k+1) \\ \theta(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} 0.9763 & -0.0211 & 0.0300 & 0.0159 \\ -0.0083 & 0.9909 & 0.0120 & -0.0003 \\ 0.0181 & 0.0522 & 0.9435 & -0.0897 \\ -0.0299 & -0.0124 & 0.0307 & 1.0063 \end{bmatrix} \begin{bmatrix} \alpha(k) \\ q(k) \\ \theta(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} -0.0002 \\ -0.0001 \\ 0.0011 \\ -0.0002 \end{bmatrix} \delta_e(k) + Ke(k) \tag{5.6}$$

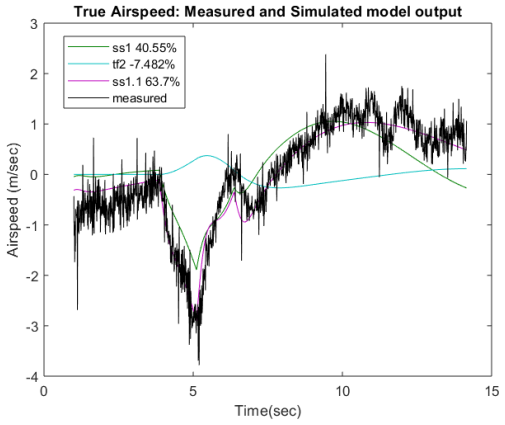
where,

$$K = \begin{bmatrix} -0.0106 & -0.0000 & -0.0000 & -0.0000 \\ -0.0025 & 0.0000 & 0.0019 & -0.0000 \\ -0.1701 & 0.0001 & 0.0015 & 0.0083 \\ -0.0071 & -0.0022 & 0.0007 & -0.0038 \end{bmatrix} e(k) \tag{5.7}$$

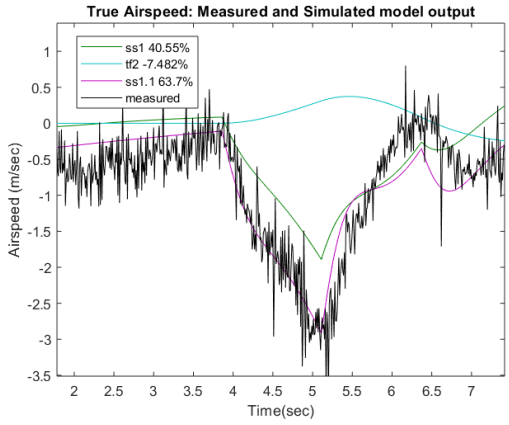
The eigenvalues are shown in Table 5.6 with damping ratio and natural frequencies of short and long period dynamics.

Dynamic Modes	Eigenvalue	Natural Frequency (rad/s)	Damping ratio
Phugoid	-0.197 ±0.346i	0.398	0.495
Short period	-3.96 ± 4.03i	5.65	0.253

Table 5.6 Eigenvalues, Natural Frequency, and Damping obtained from stat-space



(a) Pitch Angle



(b) Zoomed-in

Figure 5.15 Measured and Simulated Output response for Pitch Angle

Chapter Six

Numerical Simulation and Performance Analysis

The Rascal 110 modeling procedures were designed with the end goal of application within a fault-tolerant flight control system handling actuator failures on the primary control surfaces. After the nominal longitudinal model was derived through DATCOM and flight testing, extensive simulation studies were conducted to validate the discrete Direct and Indirect MRAC for commonly occurring upset conditions in the aircraft. All the simulations had 60 seconds duration with the inputs consisting of the two sides pulses or doublets. Only one failure occurred at a time. Pilot commands has also been characterised in a mathematical model to give ideal handling of an actual aircraft. For this purpose, a reference model has been used. A pilot reference model architecture like that presented by Perez et. al. (2015) was used to take the stick commands from the pilot $[\delta_{lat}, \delta_{long}, \delta_{dir}]$.

6.1 Failure Modeling

There are several types of possible upset conditions that may disturb the dynamics of the UAS outside bounds of nominal design. Some of them are faults in actuator. Actuator are the last component in the control-action hence they play a very important role in the

airplane, eg. delay in actuators that can cause PIO. Rate saturated actuators, system delays, high gain pilot/controller and phase lags are another known reasons that leads to PIO.

Rate Saturation: when actuators are rate-saturated due to aggressive pilot commands, high gain flight control systems or some anomaly in the system, the effective delay in the control loop may increase depending on the nature of the cause. This effective delay manifests itself as a phase shift between the commanded and actual system signals. This tends to make pilot compensate with faster responses and can induce PIO. This often worsens the situation.

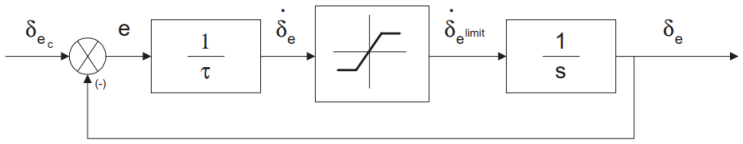


Figure 6.1 Actuator model with rate saturation

Partial Loss of Control Surface: Partial loss of elevator has been modeled in this research. This type of failure includes (partial) destruction(parts missing,holes) and/or deformation. The aerodynamic efficiency(aerodynamic forces and moments) of a control surface is altered at the beginning of the disturbance when compared to the output of the nominal aircraft response with the same pilot input. A surface damage parameter s_d models the magnitude of the failure through the ratio between the efficiency parameter after and before failure occurring moment which is modeled in equation 6.4

$$s_d = \frac{E_{u_k AfterFailure}}{E_{u_k BeforeFailure}} \tag{6.1}$$

where $s_d \in [0,1]$ with $s_d = 1$ for 'no failure' case and $s_d = 0$ for a failure involving a completely missing surface. The control matrix A remains unchanged during such failures however the the control surface inputs that is B matrix changes as shown in equation 6.2

$$x(k+1) = Ax(k+1) + Bu(k) * [1 - sddeL] \tag{6.2}$$

where $sddeL$ is change in control surface effectiveness. For this case $sddeL = 0.7$ i.e. 30 percent loss of effectiveness in elevator.

Icing Condition: Icing condition on the airplane control surfaces changes the shape of the airfoil that changes the aerodynamics of the system. The accounts of ice effects was developed by Bragg et al 2001. He presented in his work for DeHavilland Twin Otter, the effects of icing on individual performance, stability or control parameter. The aerodynamic coefficient under icing effect is computed in the following equation

$$C_{(A)iced} = (1 + \eta_{ice} k'_{c(A)}) C_{(A)} \tag{6.3}$$

$$C_{(A)iced} = (1 + f_{ice}) C_{(A)} \;\; where, f_{ice} = \eta_{ice} k'_{c(A)} \tag{6.4}$$

where $C_{(A)iced}$ is the control and stability parameter after ice accretion has taken into account while $C_{(A)}$ is the nominal parameter before ice accretion. η_{ice} and $k'_{c(A)}$ is the icing severity and coefficient icing factor that depends on the aircraft information respectively. The model described by Bragg is not very accurate since there are still more factors that can be taken into consideration. This icing factor is based on calculations obtained from DeHavilland Twin Otter that are applied to control and stability parameters of Rascal 110. The change of stability and control parameters due to icing are listed in Table 6.1.

For instance , the nominal M_α is given in the Equation 6.5.

$$M_\alpha = \frac{\bar{q} S \bar{c} C_{m\alpha}}{I_{yy}} \tag{6.5}$$

where $\bar{q}$ is the dynamic pressure, S is the wing area, and $\bar{c}$ is the mean geometric chord of the aircraft. After taking ice accretion into account, the M_α can be modeled in following way:

$$M_\alpha = \frac{\bar{q} S \bar{c} (1 + f_{ice}) C_{m\alpha}}{I_{yy}} \tag{6.6}$$

In the similar way, rest of the stability and control parameters of A and B matrices of state-space can be modelled.

Longitudinal Parameters	f_{ice}
ΔZ_0	0
ΔZ_α	-0.10
ΔZ_q	-0.012
$\Delta Z_{\delta e}$	-0.095
ΔZ_{M_0}	0
ΔZ_{M_α}	-0.099
ΔZ_{M_q}	-0.035
$\Delta Z_{M_{\delta e}}$	-0.10

Table 6.1 Change in Control and Stability Parameters

6.2 Constant gain and Open-loop Response

Using rate-saturated gain in one of the actuators, pilot-in-loop oscillation were simulated for constant gain controller and open-loop system. Figure 6.3 and 6.2 show the response of the open-loop plant and the effect of a linear controller during the PIO scenario. It is clear that even a linear control feedback will lead to an undesired oscillations and consequently to a failure in the aircraft hence they are incapable during such events of disturbances.

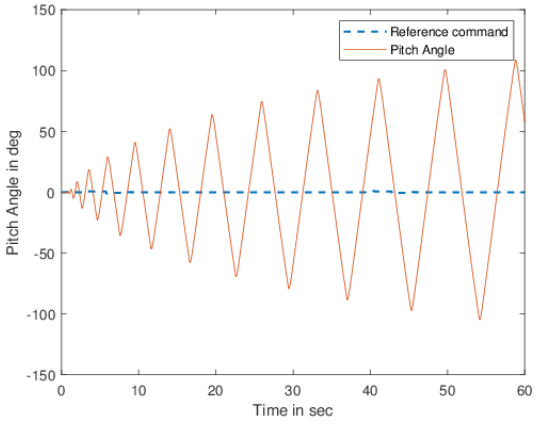


Figure 6.2 Constant Gain Controller Response under PIO Condition

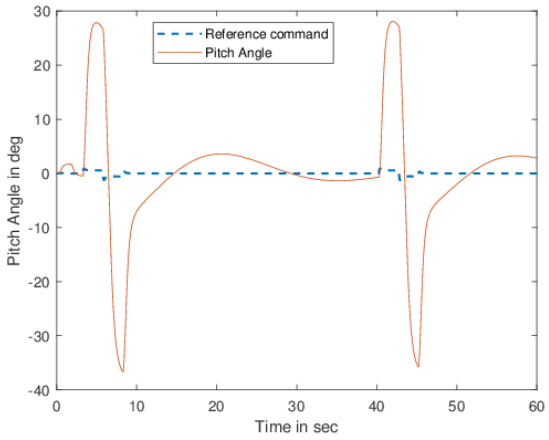


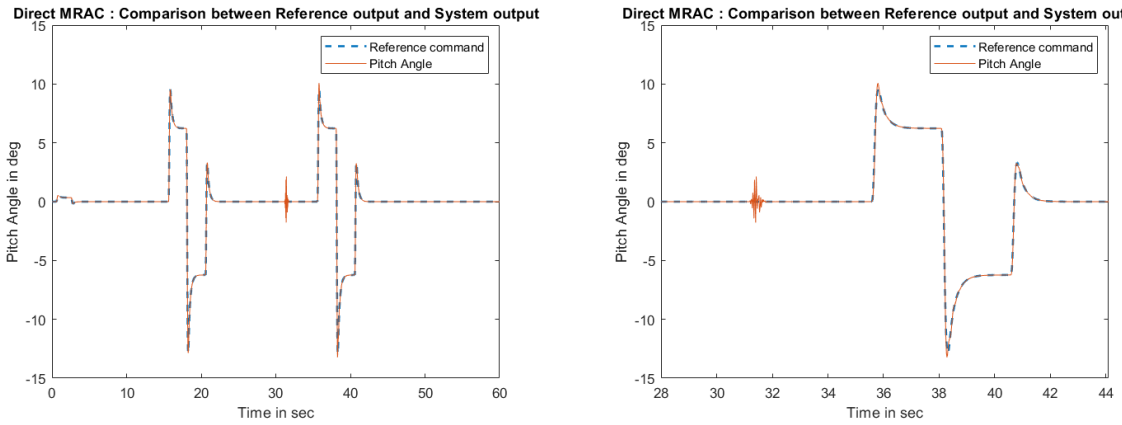
Figure 6.3 Open-Loop Response under PIO Condition

6.3 Discrete Direct MRAC Results

Discrete Direct MRAC was implemented using the equations from Section 2.3. Performance of Direct MRAC was evaluated on the basis of the failures mentioned in Section 6.1.

Control Surface Damage Analysis

The partial control surface failure was injected at 30th second before the second doublet was commanded. The output aircraft response is pitch rate.

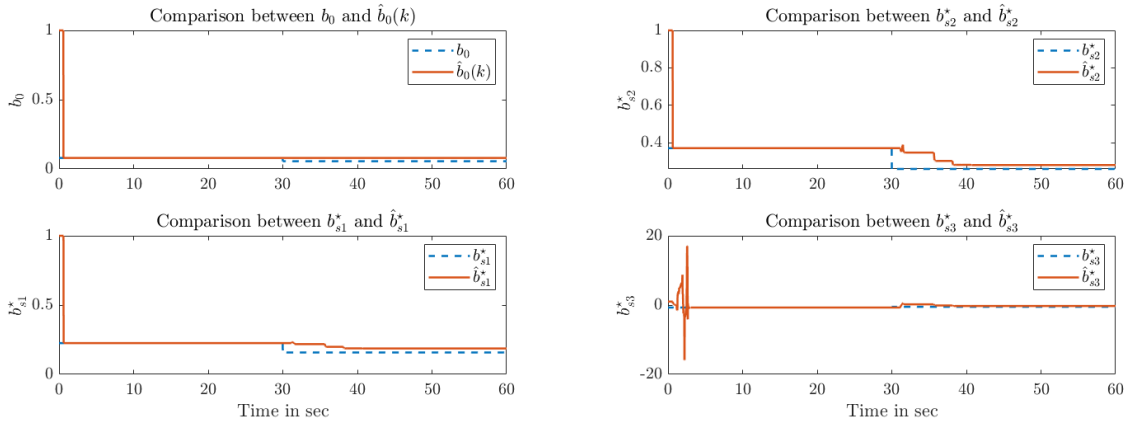


(a) Pitch Rate response **(b)** Zoomed-in

Figure 6.4 Output Response for partial Elevator Damage

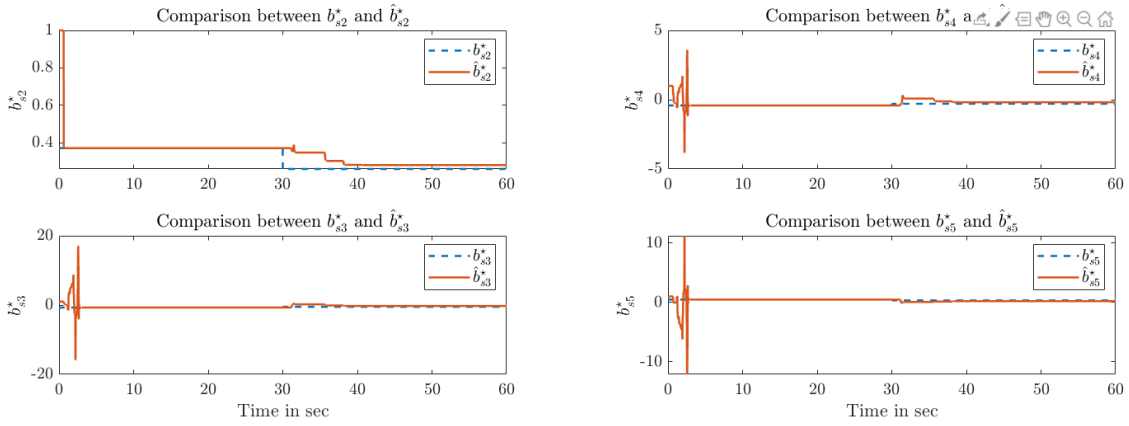
Figure 6.4a shows the full 60 seconds of simulation and Figure 6.4 shows from 30th to

44 seconds. Irregular oscillations are observed at 30th second indicating the partial damage to control surface. Up until 30 seconds, aircraft's response follows the reference command indicating the Direct MRAC controller is working. Figure 6.4a and 6.4b show that at the instant of fault, the aircraft takes about 1 second to reconfigure and make the system follow the reference command once again.



(a) **(b)**

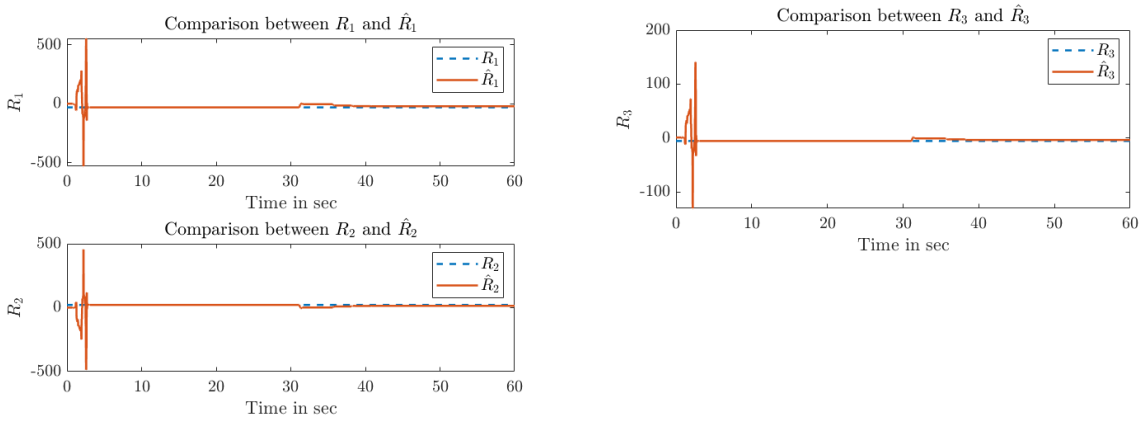
Figure 6.5 Control Surface Damage:Bezout's Coefficients



(a) **(b)**

Figure 6.6 Control Surface Damage:Bezout's Coefficients

Figure 6.5b, 6.6 and 6.7 shows the estimated Generalized Bezout Identity (GBI) Coefficients using equation 2.15 These are unknown control parameters that are estimated. Notice that the parameters change at the moment where the fault occurs, as a compensatory action.

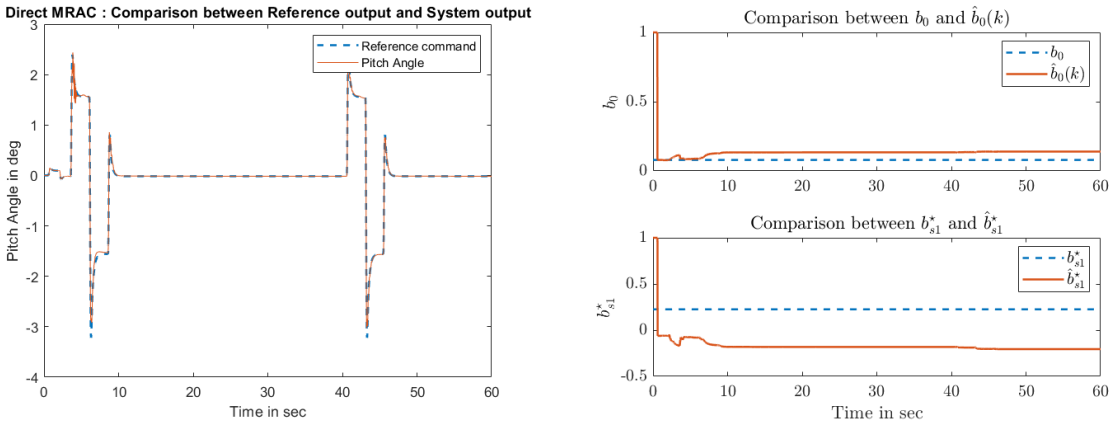


(a) **(b)**

Figure 6.7 Control Surface Damage: Bezout's Coefficients

Pilot-Induced Oscillation Analysis

This sub section shows the results produced when Direct MRAC's performance was evaluated under PIO. Rate saturation in actuator is causing unwanted oscillation in the beginning of the first commanded doublet as seen Figure 6.8. These oscillations are due to compensatory action generated from the controller.

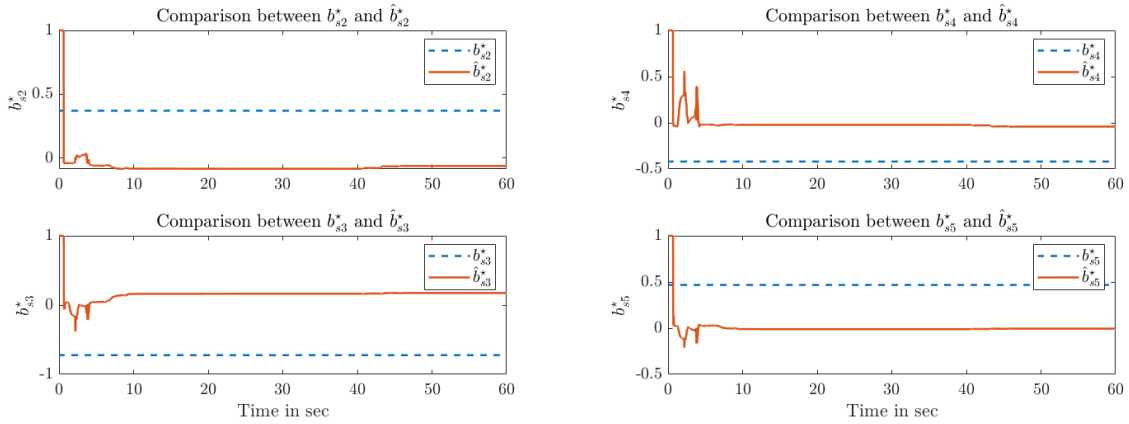


(a) Output Response **(b)** Bezout's Coefficient

Figure 6.8 Pilot-Induced Oscillation

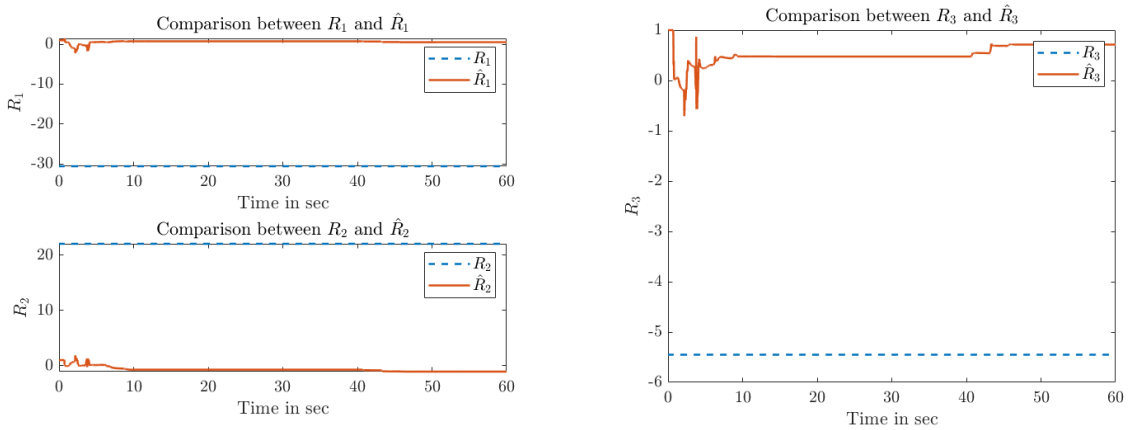
Figure 6.9 and 6.10 represents the estimated bezout coefficients for PIO condition.At the beginning, the parameters change due to the occurrence of PIO and resume to the nominal

response after reconfiguration.



(a) **(b)**

Figure 6.9 PIO: Bezout's Coefficients

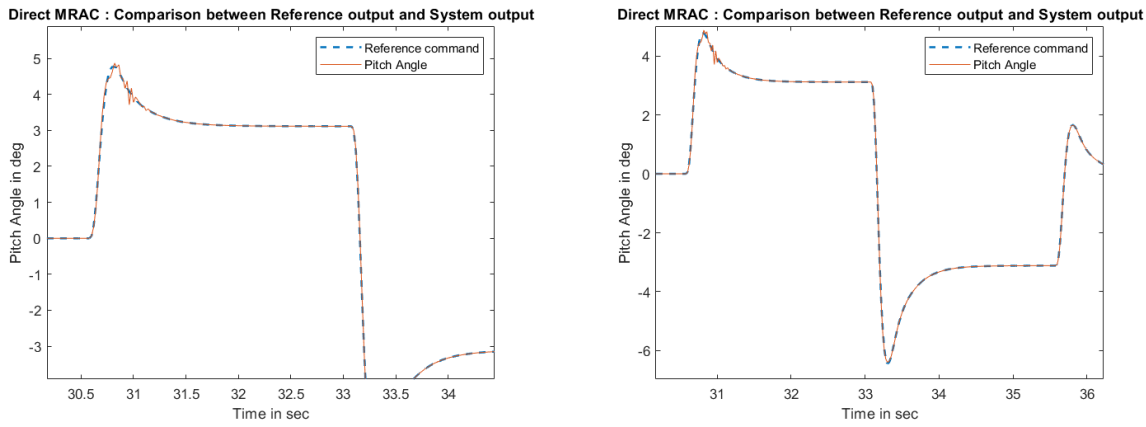


(a) **(b)**

Figure 6.10 PIO: Bezout's Coefficients

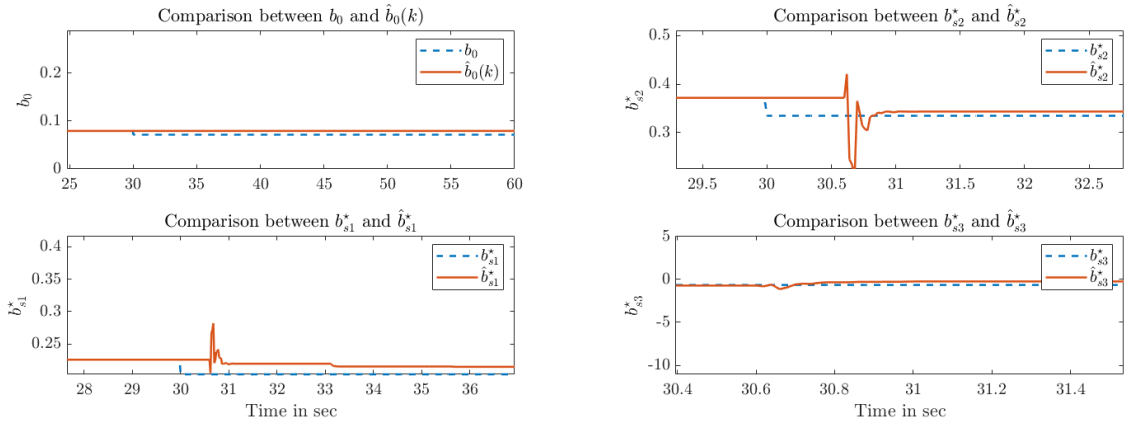
Icing Condition Analysis

Icing conditions are modeled in failure modeling section and results are produced on this section. Figure 6.11 shows the output response of the aircraft during ice accretion.The uncertainties(oscillations) produced by this particular disturbance is very less. This is because the icing condition modelled is not very severe. Figures 6.12, 6.13 and 6.14 shows the change in parameters due to ice accretion.



(a) **(b)**

Figure 6.11 Output Response during Icing condition



(a) **(b)**

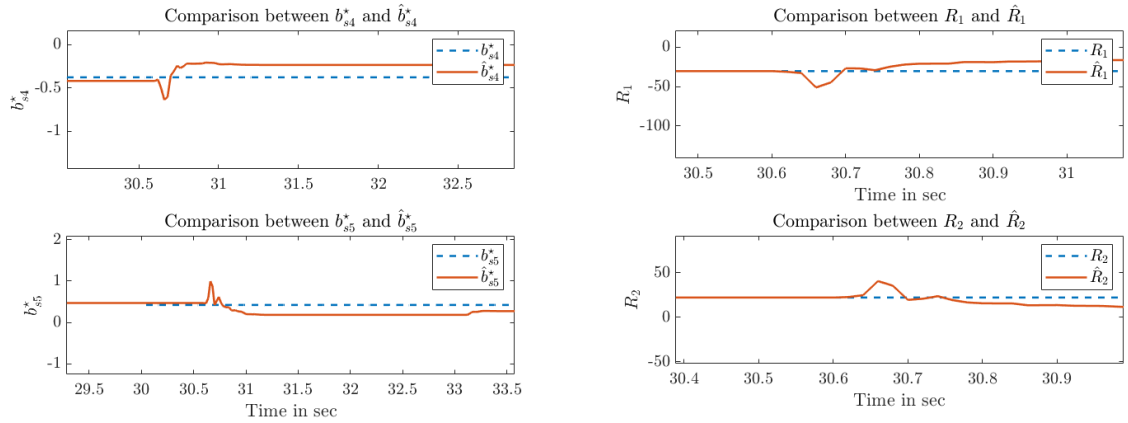
Figure 6.12 Icing condition: Bezout Coefficient

6.4 Discrete Indirect MRAC Results

Control Surface Damage Analysis

Discrete Indirect MRAC Controller is evaluated for partial elevator damage in Figure 6.15.

The Figure 6.9, 6.17, 6.18 and 6.19 are estimated polynomial and bezout coefficients. Polynomial coefficients are direct representation of the aircraft's system that are derived in ARMA model. These coefficients shows the change of values at the instant of control surface damage.



(a) **(b)**

Figure 6.13 Icing condition: Bezout Coefficient

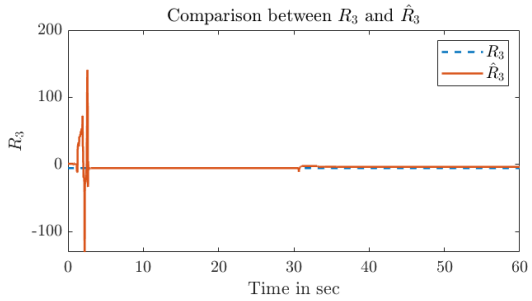
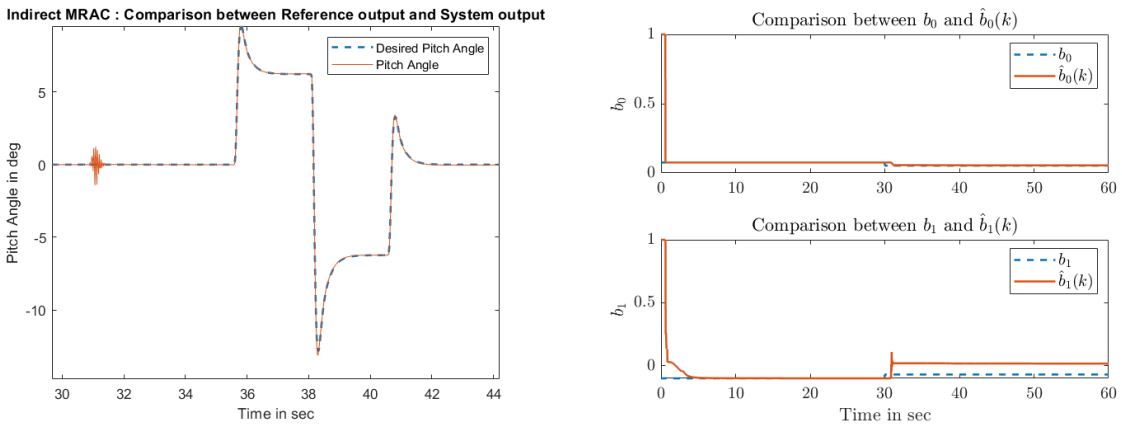


Figure 6.14 Icing condition: Bezout Coefficient

Pilot-Induced Oscillation Analysis

PIO is modeled into Indirect MRAC controller by saturating the actuator. Figure 6.20a shows the oscillations at the beginning of doublet which lasts few seconds followed aircraft returning to the reference trajectory.

This is also evident in the polynomial and bezout coefficients estimation in Figure 6.21,6.22,6.23 and 6.24 . During the PIO occurrence, these coefficients adapt to new values for to make aircraft follow the reference command.



(a) Output Response **(b)** Polynomial coefficient

Figure 6.15 Aircraft response in Partial Elevator damage and Polynomial Coefficient(right)

Icing Condition Analysis

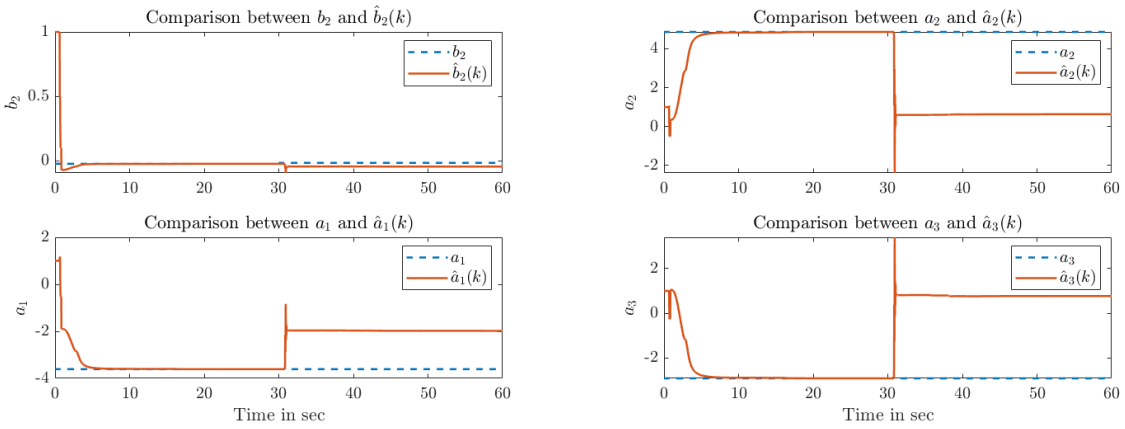
Icing Conditions produces disturbances in the beginning of second doublet as seen the output response produced in Figure 6.25. The estimated coefficients alters at the 30th second where the fault due to Icing condition occurs as observed in figure 6.26, 6.27 and 6.28.

6.5 Performance Analysis of Direct and Indirect MRAC

All the measures require a fixed experiment to be performed on the system (i.e. a fixed set-point or disturbance change). The following metrics were implemented to further analyze the performance of both controllers: integral square error (ISE), integral absolute error (IAE) and Root Mean Square(RMSE).

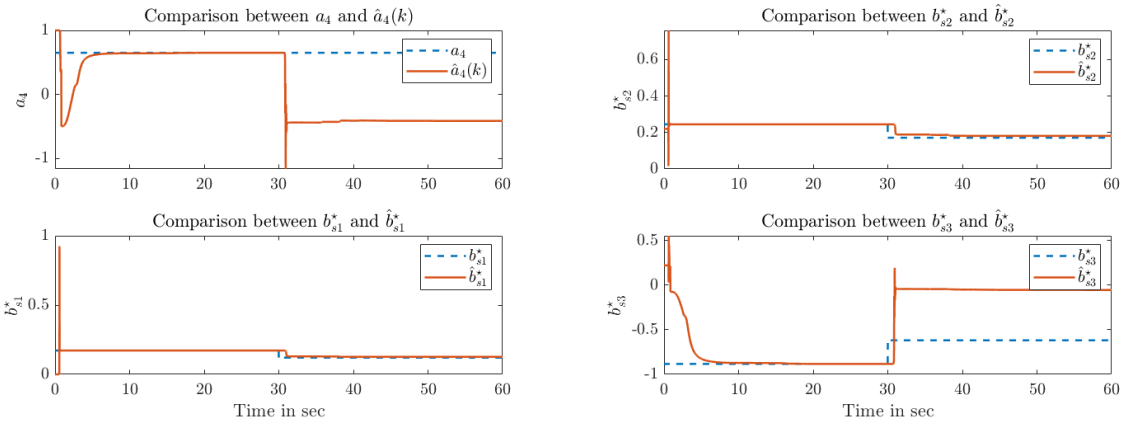
ISE integrates the square of the error over time. ISE penalizes large errors more than smaller ones (since the square of a large error will be much bigger). Control systems specified to minimise ISE will tend to eliminate large errors quickly, but will tolerate small errors persisting for a long period of time. Often this leads to fast responses, but with considerable, low amplitude, oscillation.

IAE integrates the absolute error over time. It does not add weight to any of the errors



(a) **(b)**

Figure 6.16 Control Surface Damage:Polynomial Coefficients

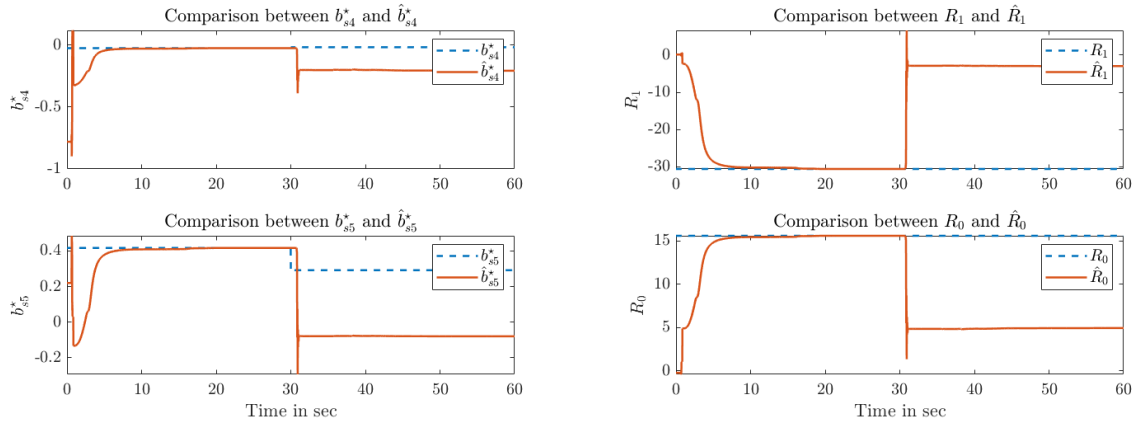


(a) **(b)**

Figure 6.17 Control Surface Damage: Polynomial and Bezout Coefficient

in a systems response. It tends to produce slower response than ISE optimal systems, but usually with less sustained oscillation.

Performance indices of Direct and Indirect for control surface damage, PIO and Icing Condition are presented in Table 6.2, 6.3 and 6.4. The bold values shows the better performance of two controllers. Indirect MRAC performs better in control surafce damage and PIO condition. This is due to the fact that Indirect MRAC explicitly estimates the plant (ARMA model coefficients) and controller parameters while in Direct MRAC only controller parameters are estimated to directly adapt such that plant tracks the reference model. However



(a) **(b)**

Figure 6.18 Control Surface Damage:Bezout Coefficient

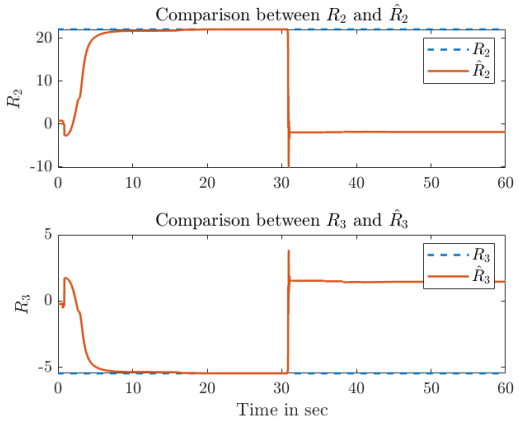


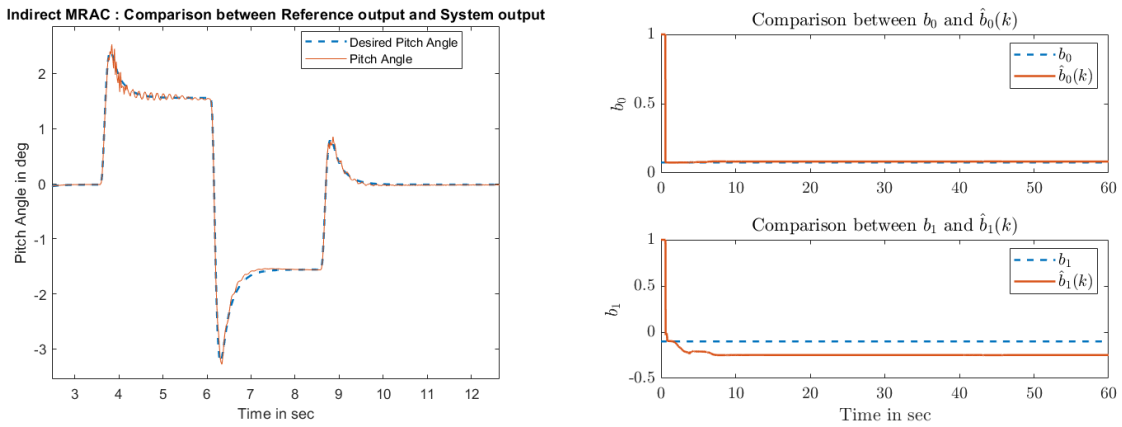
Figure 6.19 Control Surface Damage:Bezout Coefficient

for the icing condition case, Direct MRAC outperformed the Indirect MRAC.

6.5.1 Graphical Comparison

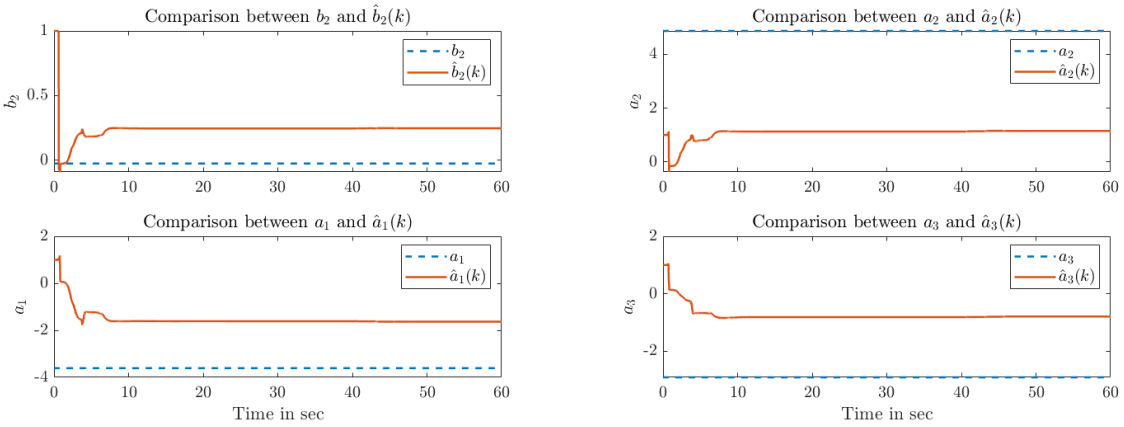
This section emphasises on the graphical analysis of both controllers at the instant of failure. Observing the figure 6.30, the partial control surface damage, Direct MRAC has a higher amplitude of oscillations than the Indirect one. However there are lesser number of oscillations in Direct controller.

For the case of PIO as it can be observed in figure 6.31b, Direct controller takes lesser time to reconfigure but with higher amplitude of oscillations whereas Indirect controller



(a) **(b)**

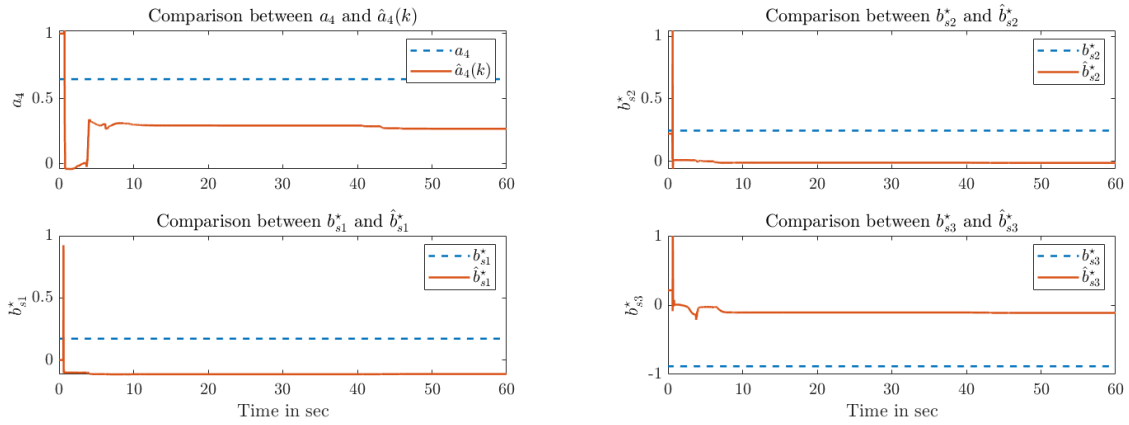
Figure 6.20 Aircraft response during PIO and Estimated Polynomial Coeff(right)



(a) **(b)**

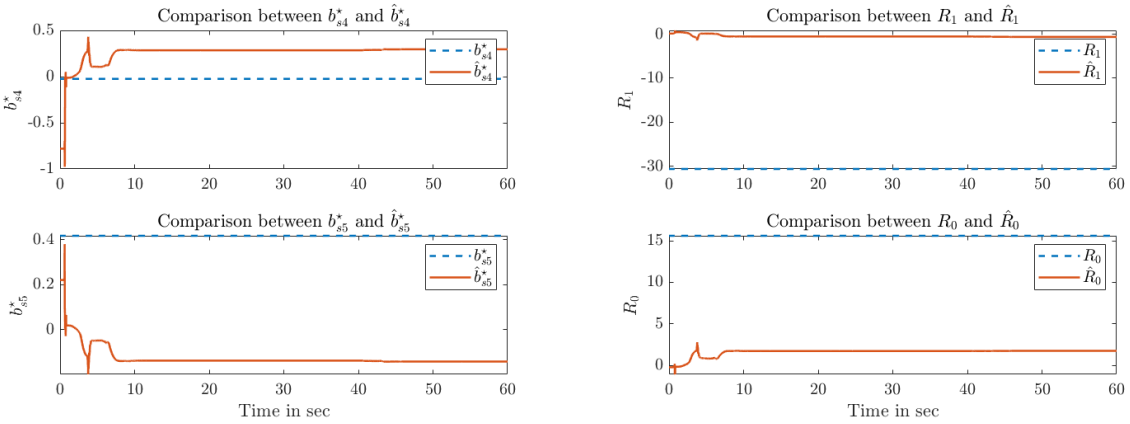
Figure 6.21 PIO: Estimated Polynomial Coeff(right)

takes longer time with lower amplitude of oscillations. An interesting observation is made for icing condition, according to Figure 6.31 , Direct MRAC controller as reconfiguration action generates lower amplitude of oscillations and takes lesser time to reconfigure when compared to Indirect. There can be several reason for this behavior, such as the uncertainty ice condition modeled in state-space is adapted from DeHavilland Twin Otter[13], hence it is not very accurate for Rascal 110. Therefore its not certain that Direct controller in all icing circumstances will outperform the Indirect MRAC controller.



(a) **(b)**

Figure 6.22 PIO: Estimated Polynomial Coeff and Bezout Coeff(right)



(a) **(b)**

Figure 6.23 PIO: Bezout Coeff

	Direct MRAC	Indirect MRAC
ISE	0.0078	**0.0050**
IAE	0.0869	**0.0653**
RMSE	0.0114	**0.00914**

Table 6.2 Performance Comparison for Control Surface Failure

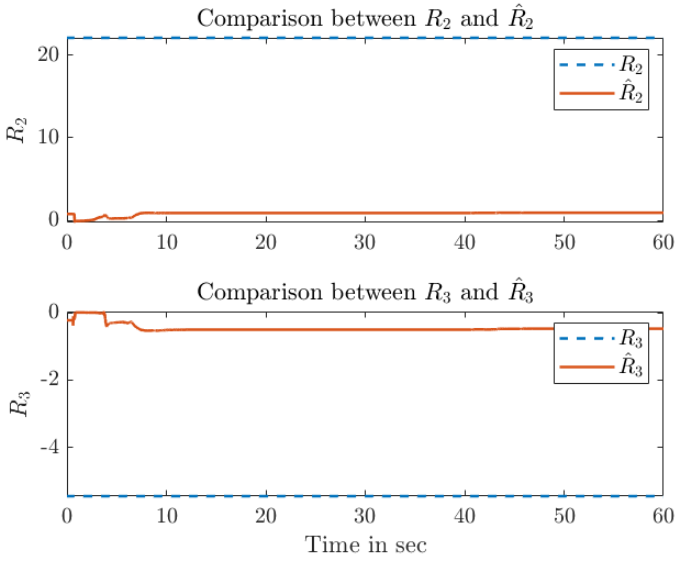
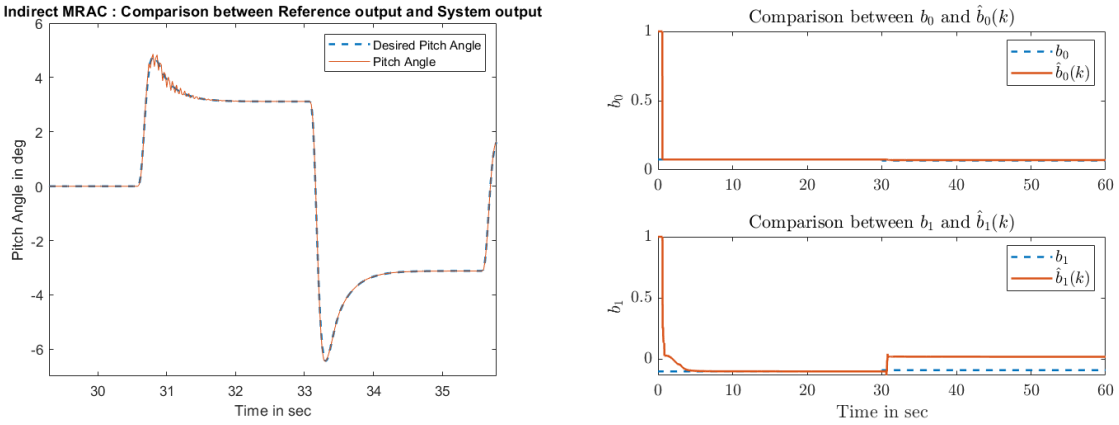


Figure 6.24 PIO: Bezout Coefficient

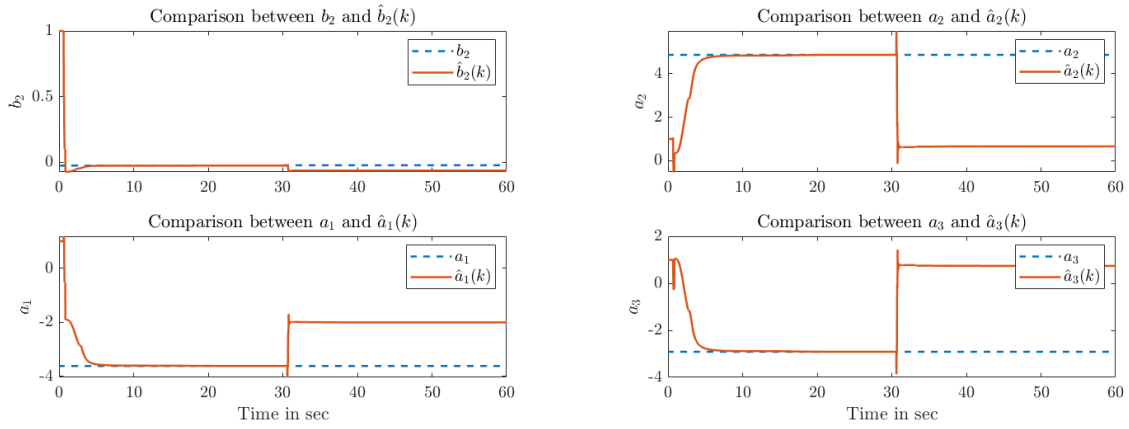


(a) **(b)**

Figure 6.25 Aircraft response during icing condition and Estimated Polynomial Coeff.(right)

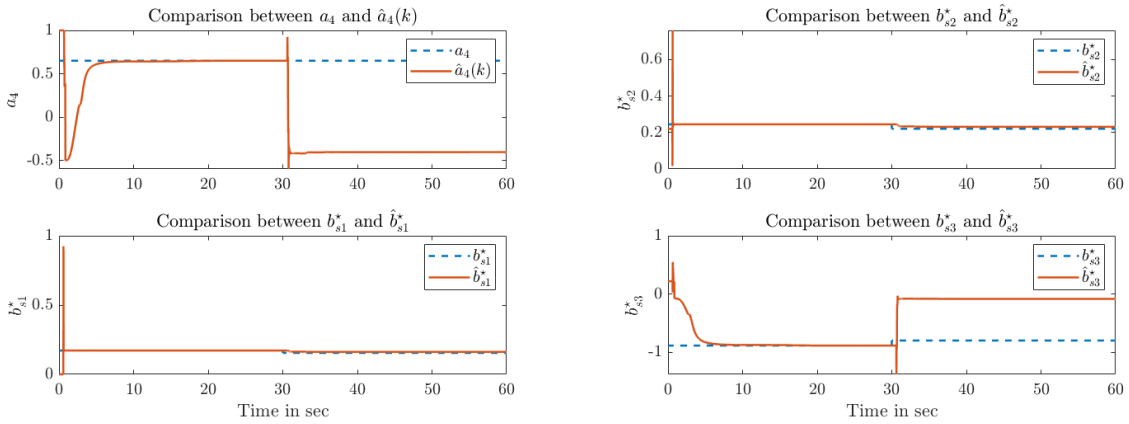
	Direct MRAC	Indirect MRAC
ISE	$2.44e^{-5}$	$\mathbf{3.46}e^{-6}$
IAE	0.0115	**0.0046**
RMSE	$6.3828e^{-4}$	$\mathbf{2.4022}e^{-4}$

Table 6.3 Performance Comparison for Pilot-in-loop Oscillation



(a) **(b)**

Figure 6.26 Icing Condition:Estimated Polynomial Coefficient

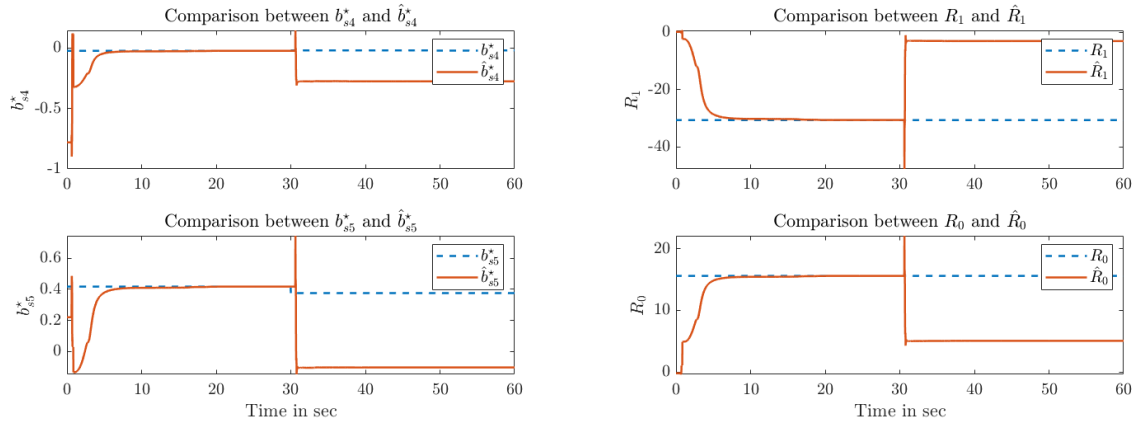


(a) **(b)**

Figure 6.27 Icing Condition:Estimated Polynomial Coefficient and Bezout Coeff.(right)

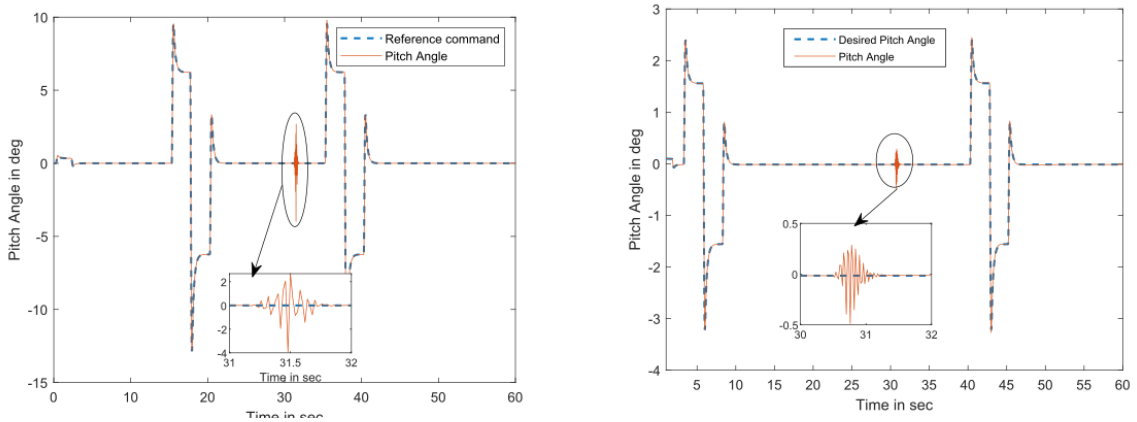
	Direct MRAC	Indirect MRAC
ISE	$\mathbf{3.6254}e^{-6}$	$2.7323e^{-5}$
IAE	**0.0017**	0.0050
RMSE	$\mathbf{2.4577}e^{-4}$	$6.7471e^{-4}$

Table 6.4 Performance Comparison for Icing Condition



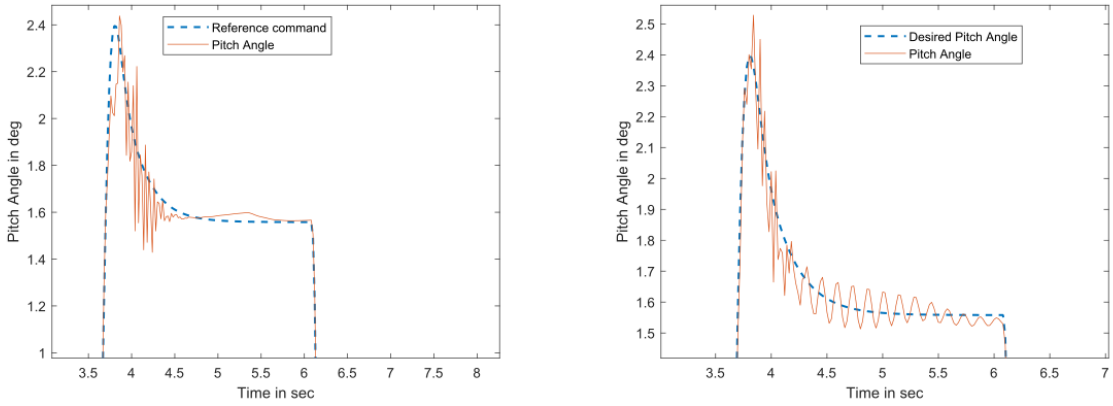
(a) **(b)**

Figure 6.28 Icing Condition:Estimated Bezout Coefficient



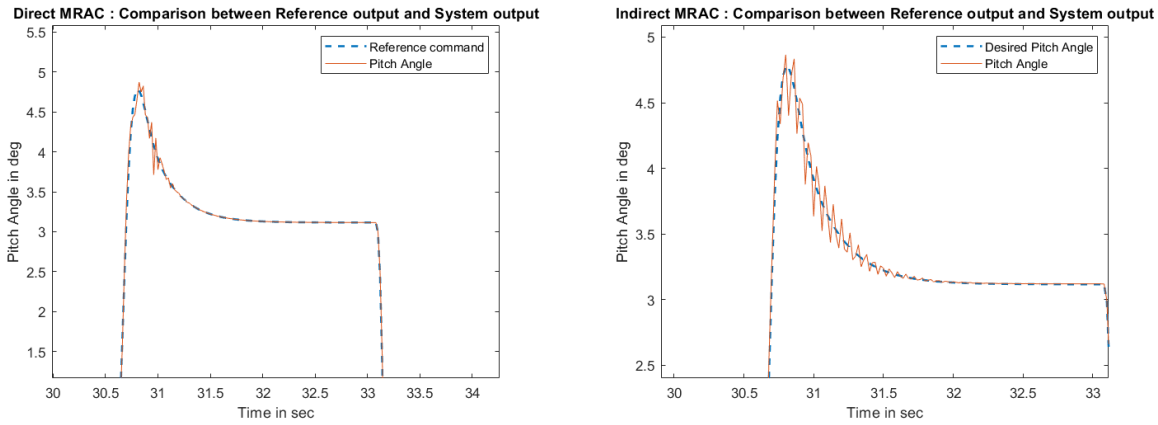
(a) Direct MRAC **(b)** Indirect MRAC

Figure 6.29 Responses during partial failure of Elevator



(a) Direct MRAC **(b)** Indirect MRAC

Figure 6.30 Output Responses during PIO



(a) Direct MRAC **(b)** Indirect MRAC

Figure 6.31 Output Responses during Icing Condition

Chapter Seven

Conclusion and Future Scope

7.1 Concluding Remark

In this research effort, a parameter identification for Rascal 110 was carried out. For an initial estimation, digital DATCOM was used that computed stability and control derivatives. A priori knowledge from initial estimation was later utilized to design flight tests for parameter identification. Data obtained from flight tests were used to compute the high fidelity model. For the state space estimation, the Matlab® System Identification Toolbox was used to evaluate the aircraft maneuvers and the aircraft system dynamic responses recorded during the flight. After evaluating the eigenvalues, damping ratios, and natural frequencies from the maneuvers, a a state space model with better fit was selected.

A high fidelity model was generated to support the design, validation and verification of adaptive discrete fault tolerance control laws. In particular, a Discrete Direct and Indirect Model Reference Adaptive Control was designed and implemented. The controllers are applied to an ARMA model that has been previously identified from flight test data.The adaptation performances of both algorithms were compared and tested in simulation for two commonly occurring failure cases: actuator delay and partial loss of elevator as well ice accretion that occurs commonly on small airplanes. The performance metrics was defined and computed.The results show the potential of Discrete Direct and Indirect MRAC to

generate compensation actions.

According to the performance metrics, Indirect controller outperfoms the direct controller for control surface damage and PIO where direct controller performs better for ice accretion case.

7.2 Future Scope

There is a lot room for future exploration on this research topic. In this thesis, only longitudinal model was identified that was used for validation of discrete adaptive control. Flight testings can be conducted to extract Lateral/Directional Model for Rascal 110. The longitudinal model obtained from the flight test data can be further optimized further to uppdate the aerodynamic coefficients generated from DATCOM.

Further in terms of identification process, coupled dynamics with individual control deflections can be obtained where the three primary control surfaces are divided into six individual components. This will improve the fidelity of the mathematical model.

Further study can be carried out for implementing discrete MRAC for non-linear systems and validate the non-linear controller with the same failures modeled in this thesis research,

Finally, a study on the stability analysis of discrete direct and indirect MRAC can be investigated.

REFERENCES

[1] “Unmanned Systems Integrated Roadmap 2009-2034”, Office of the Secretary of Defense, Washington, D.C., 2009

[2] “U.S. Army Unmanned Aircraft Systems Roadmap 2010-2035”, U.S. Army UAS Center of Excellence, Fort Rucker, AL, 2010

[3] I. Sadeghzadeh, and Y. Zhang "A Review on Fault-Tolerant Control for Unmanned Aerial" Vehicles (UAVs)", Infotech Aerospace 2011, 29 - 31 March 2011, St.Louis, Missouri

[4] Ki Seok Kim, Keum-Jin Lee, Youdan Kim" Reconfigurable Flight Control System Design Using Direct Adaptive Method," Journal of Guidance, Control, and Dynamics, vol. 26, no. July-August 2013, pp. 543-550, 2003.

[5] K. Philips, G Campa, S Gururajan, B seanor, M Napolitano, Y Gu, M.L. Fravolini "Prameter ID for application within a Fault-tolerant Flight control System", AIAA Atmospheric Flight Mechanics Conference, August 2009

[6] D.G. Ward, J.F. Monaco and M. Bodson, "Development and Flight Testing of a Parameter Identification Algorithm for Reconfigurable Control", Journal of Guidance, Navigation and Control’,vol 21., July 1998, 948-956. 10.2514/2.4329

[7] K. B. Phillips, "Aircraft Parameter Identification For Application Within A Fault-Tolerant Flight Control System", PHD Dissertation, submitted to West Virginia University, 2011 [8] J. Lee, H. S. Choi, S. Lee, E. T. Kim and D. Shin, "Time Delay Fault-tolerant Controller for Actuator Failures during Aircraft Autolanding",Japan Society of Aeronautical Space Sciences Transactions, vol 55,175-182. 10.2322

[9] A. Chamseddine , Y. Zhang, C. Rabbath, C, Fulford and J.Apkarian, "Model Refrence Adaptive Fault-tolerant Control of a Quadcopter UAV", Infotech@Aerospace, March 2011

[10] A.A. Gerner, C.L.Maurer, "Calibration of Seven-Hole Probes Suitable for High Angles in Subsonic Compressible Flows", AIAA-82-0410, 1982

[11] Y. K. Seungyong Hyung, "Reconfigurable Flight Control System Design Using Discrete Model Reference Adaptive Control," AIAA Guidance, navigation, and Control Confer-

ence, 15-18 August 2005.

[12] Moncayo H., Perhinschi M. G., Wilburn, B., Davis J., Ondrej, K., UAV Adaptive Control Laws Using Non-Linear Dynamic Inversion Augmented with an Immunity based Mechanism, AIAA Guidance, Navigation, and Control Conference, AIAA-2012-4678, 2012.

[13] K. M. Rankin, “Adaptive Control of Aircraft in Uncertain Icing Conditions”,Master’s Thesis dissertation, University of Virginia , May 2014

[14] S. O. Toole, "Development of a Remotely-Piloted Vehicle Platform to Support Implementation, Verification, and Validation of Pilot Control Systems", Masters Thesis, submitted to Embry-Riddle Aeronautical University, 2017

[15] B. Lyons, "Performance Analysis of Non-Linear Adaptive Control Laws Using Hardware in the Loop of Unmanned Aerial System", Masters Thesis, submitted to Embry-Riddle Aeronautical University, 2013

[16] C. Wang, M. Santone, and C. Cao "Pilot-Induced Oscillation Suppression by Using L1 Adaptive Control", Journal of Control Science and Engineering, Volume 2012

[17] P.D. Domans ’Control Performance Assessment: Theoretical and Industrial’, Springer, 2020

[18] Y. Yildiz and I. V. Kolmanovsky "A Control Allocation Technique to Recover From Pilot-Induced Oscillations (CAPIO) due to Actuator Rate Limiting", American Control Conference, 2010

[19] M. Bakori, H. Moncayo ’UAS Model Identification And Simulation To Support In-Flight Testing of Discrete Adaptive Fault Tolerant Control Laws’, AIAA Scitech Conference, 2020

[20] Perhinschi M. G., Moncayo H., Davis J., “Integrated Framework for Artificial Immunity-Based Aircraft Failure Detection, Identification, and Evaluation”, AIAA Journal of Aircraft, Vol. 47, No. 6, Nov.-Dec. 2010, pp. 1847-1859

[21] Moncayo H., Perhinschi M. G., Davis J., “Artificial Immune System – Based Aircraft Failure Detection and Identification Using an Integrated Hierarchical Multi-Self Strategy”,

AIAA Journal of Guidance, Control, and Dynamics, Vol. 33, No. 4, Jul.-Aug. 2010, pp. 302-320

[22] Moncayo H., Perhinschi M. G., Davis J., "Artificial Immune System – Based Aircraft Failure Detection and Identification Over an Extended Flight Envelope", The Aeronautical Journal, Vol. 115, No. 1164, Feb. 2011.

[23] Moncayo H., Perhinschi M. G., Davis J., "Artificial Immune System – Based Aircraft Failure Evaluation over Extended Flight Envelope", AIAA Journal of Guidance, Control, and Dynamics, Volume 34, No. 4, July/August 2011.

[24] Perhinschi M. G., Porter J., Moncayo H., Davis J., Wayne W. S., "Artificial Immune System-Based Detection Scheme for Aircraft Engine Failures", AIAA Journal of Guidance, Control, and Dynamics, Volume 34, No. 5, Sep/Oct 2011

[25] Moncayo H., Perhinschi M. G., "Aircraft Fault Tolerance: A Biologically Inspired Immune Framework for Sub-System Failures", VDM Verlag Dr. Muller GmbH Co. KG, VDM Publishing House ltd., Saarbrucken, Germany, 2011.

[26] Perhinschi M. G., Moncayo H., Wilburn, B., Davis J., Ondrej, K., "Testing of Immunity-Based Failure Detection Scheme with the NASA Generic Transport Model", AIAA Guidance, Navigation, and Control Conference, 2010.

[27] Klein, V, Morelli, E, "Aircraft System Identification: Theory and Practice." Reston, VA: American Institute of Aeronautics and Astronautics, 2006.

[28] K. Ogata, Discrete-Time Control Systems, Delhi: PHI Learning Private Limited, 2013.

[29] D. McLean, Automatic Flight Control Systems, UK: Prentice Hall International, 1990.

[30] B Jung, Y Kim, C Ha., "Fault-Tolerant Flight Control System design using a multiple model adaptive controllers", School of Mechanical and Aerospace Engineering, Seoul National University

[31] Y. Yildiz and I. V. Kolmanovsky, "A control allocation technique to recover from

pilot-induced oscillations (capio) due to actuator rate limiting," in American Control Conference, 2010 .

APPENDIX

.1 Raw Data recorded during Elevator Step

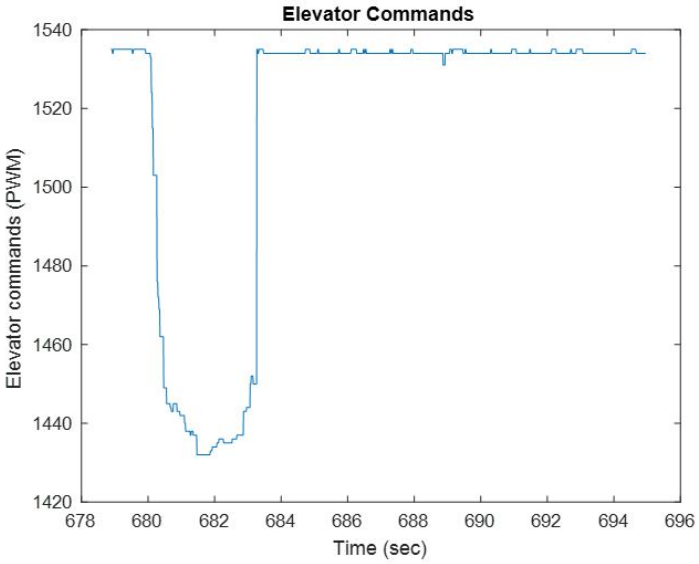


Figure 1 Step elevator command

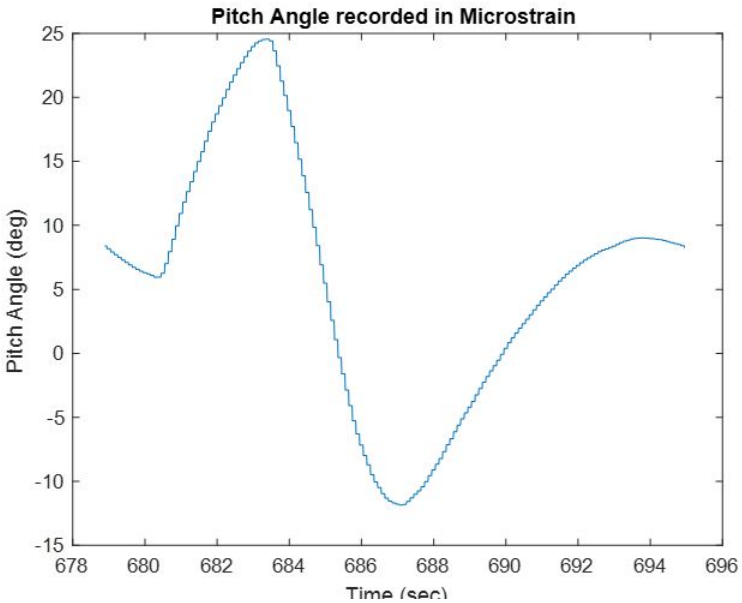


Figure 2 Pitch Angle

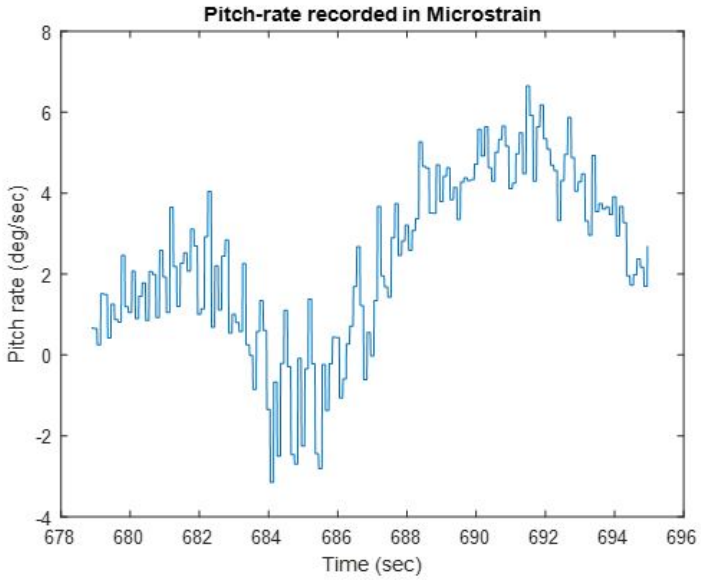


Figure 3 Pitch-rate from Microstrain

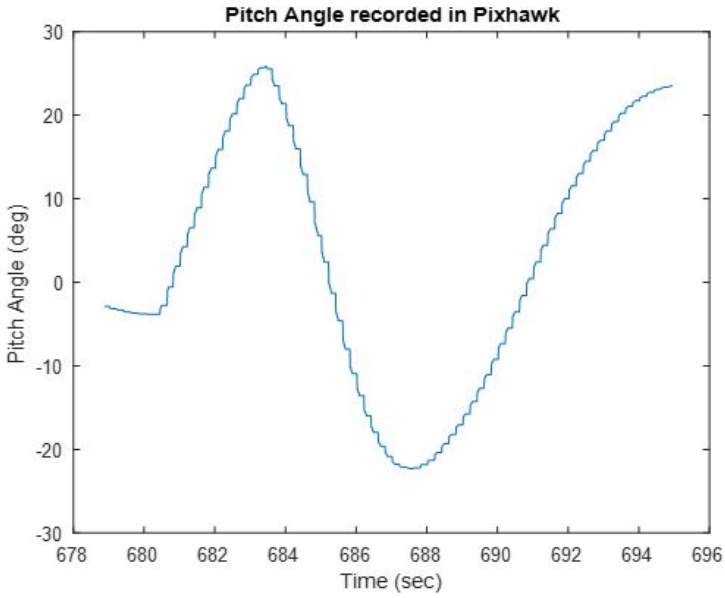


Figure 4 Pitch Angle from Pixhawk

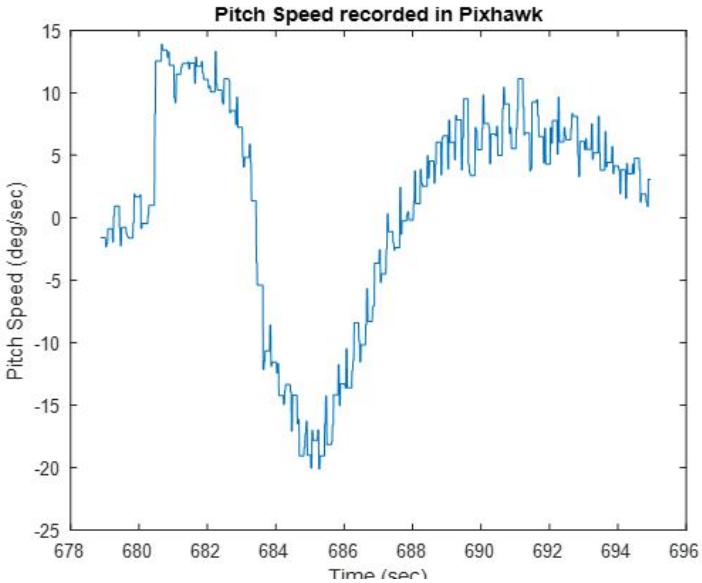


Figure 5 Pitch Speed from Pixhawk

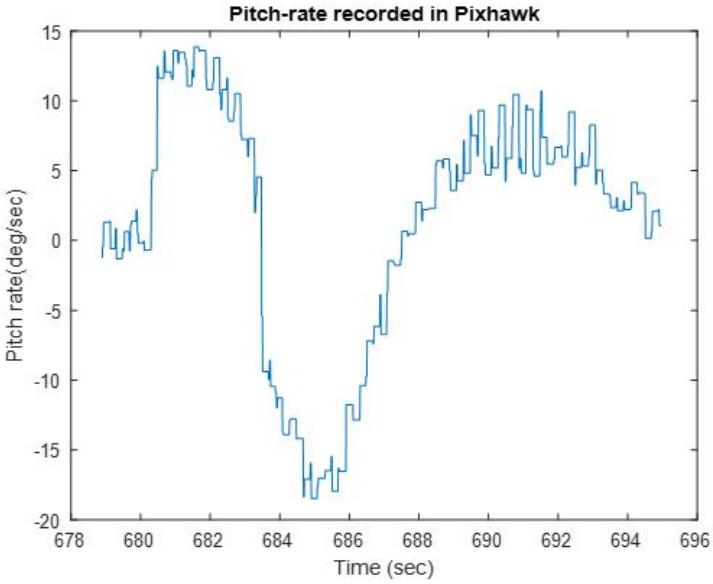


Figure 6 Pitch-rate from Pixhawk

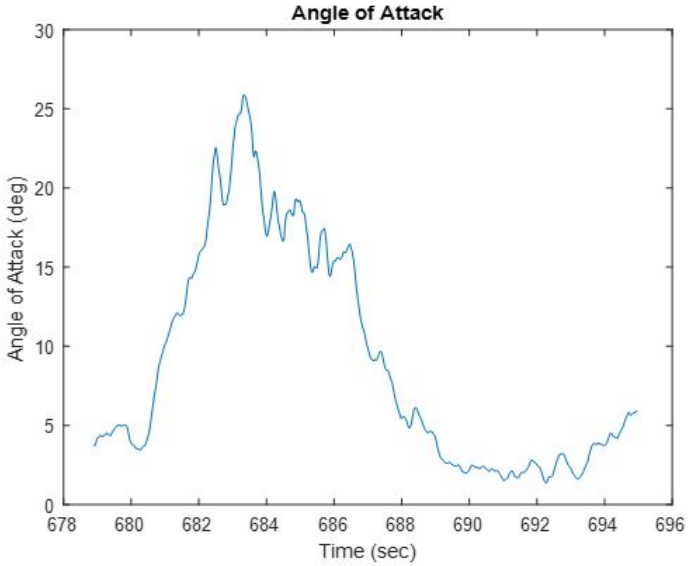


Figure 7 Angle of attack

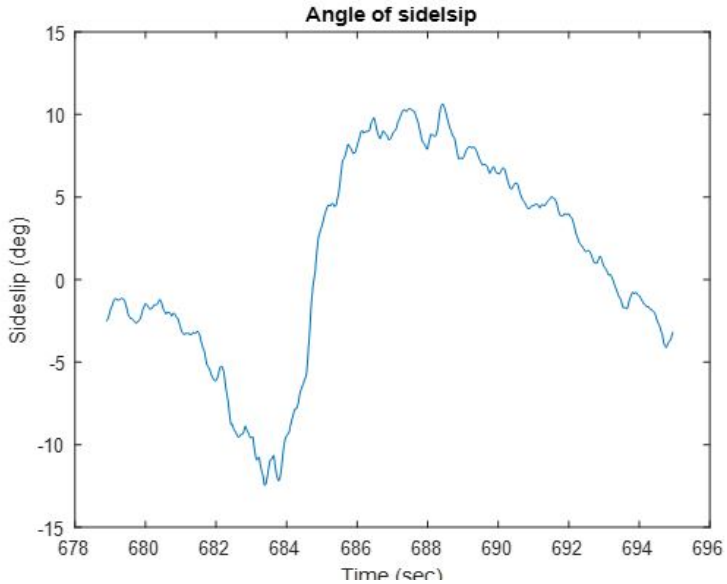


Figure 8 Angle of Sideslip