Dissertations and Theses

5-2018

Development of a Small Unmanned Aerial Vehicle Longitudinal Model for Future Flutter Testing

Said S. Hamada

DEVELOPMENT OF A SMALL UNMANNED AERIAL VEHICLE LONGITUDINAL MODEL FOR FUTURE FLUTTER TESTING

A Thesis

Submitted to the Faculty

of

Embry-Riddle Aeronautical University

by

Said S. Hamada

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Aerospace Engineering

May 2018

Embry-Riddle Aeronautical University

Daytona Beach, Florida

DEVELOPMENT OF A SMALL UNMANNED AERIAL VEHICLE LONGITUDINAL MODEL FOR FUTURE FLUTTER TESTING

by

Said S. Hamada

A Thesis prepared under the direction of the candidate's committee chairman, Dr. Claudia Moreno, Department of Aerospace Engineering, and has been approved by the members of the thesis committee. It was submitted to the School of Graduate Studies and Research and was accepted in partial fulfillment of the requirements for the degree of Master of Science in Aerospace Engineering.

THESIS COMMITTEE

Chairman, Dr. Claudia Moreno

Member, Dr. Hever Moncayo

Member, Dr. Bogdan Udrea

Graduate Program Coordinator, Dr. Magdy Attia

4.20.2018
Date

Dean of College of Engineering, Dr. Maj Mirmirani

4/20/2018
Date

Vice Chancellor, Academic Support, Dr. Christopher Grant

4/20/18
Date

ACKNOWLEDGMENTS

I would like to begin to thank the Almighty for giving me the strength and patience to complete this thesis. I would like to also thank my adviser Dr. Claudia Moreno for being an amazing adviser. She has taught me so much throughout this research and has guided me through the frustrations and accomplishments we have gone through. She has been very caring, kind and patient through the mistakes I have made. I would also like Dr. Hever Moncayo and his team, especially Cindy Nshuti, who have helped and guided me in the setup of the avionics of the UAV and for all the resources they have provided to the completion of this thesis. I would like to thank Dr. Borja Martos for guiding me through the system identification process in this research and providing resources that were helpful in the completion of this thesis. I would also like to thank Robert Moore, Eric Frantz and Ethan Jacobs for providing their skills of flying UAVs and assist me conduct the flight experiments needed for the research. Finally I would like thank my parents, Salahaddin Hamada and Khadija Agil, for their continuous love and sacrifice they have given me. Their continuous support pushed me and helped me in my achievements.

TABLE OF CONTENTS

Page

LIST OF TABLES vi

LIST OF FIGURES vii

ABBREVIATIONS ix

ABSTRACT x

1 Introduction 1

2 Background 3
2.1 State of the Art Models 4
2.1.1 Aircraft Modeling for Flutter Testing 8
2.2 Mass/Inertia Properties 9
2.2.1 Weight and CG experiment 9
2.2.2 Moments of Inertia 11
2.3 System Identification 15
2.3.1 Linear Grey-box Model 16
2.3.2 Equation-Error Parameter Estimation 17

3 Flight Test Platform 20
3.1 Airframe 20
3.2 Avionics and Sensors 21
3.3 Software 25
3.3.1 QGroundControl/Mission Planner 26
3.3.2 Pixhawk Pilot Support Package (PSP) 28

4 Flight Dynamics Model 31
4.1 Mass and Inertia Properties 32
4.1.1 Center of Gravity 32
4.1.2 Inertia Properties 34
4.2 Aerodynamic Model 34
4.3 Rigid Body Simulation Environment 37
4.3.1 Actuator Model 39

5 Longitudinal Dynamics System Identification 42
5.1 Longitudinal Dynamics 42
5.2 Methodology 45
5.2.1 MATLAB System Identification Toolbox 46

Page

5.2.2 SIDPAC and PIL 47
5.3 Validation of System Identification tools using Simulated Data 52
5.4 Flight Test Data and Results 54
6 Conclusion and Future Work 65
6.1 Conclusion 65
6.2 Future Work 65
REFERENCES 67
A Greyest Code 69

LIST OF TABLES

Table Page

3.1 Pixhawk sensors 22

3.2 Pixhawk interfaces 23

4.1 Ranger EX 757-7: center of gravity position [cm] 33

4.2 Ranger EX 757-7: moments of inertia 35

4.3 Ranger EX 757-3: stability derivatives [/rad] 36

4.4 Ranger EX 757-3: control derivatives [/rad] 36

5.1 Identified non-dimensional coefficients with simulated data 55

5.2 Short period poles using simulated data 55

5.3 Ranger EX 757-3: identified longitudinal stability derivatives [/rad] 58

5.4 Short period poles of identified models 60

5.5 Short period poles of identified longitudinal model from nonlinear simulation 64

LIST OF FIGURES

Figure Page

2.1 Simplified delta-wing model 5

2.2 The Control Configured Vehicle B-52 6

2.3 The Lockheed Martin X56-A 7

2.4 Experimental setup for weight and center of gravity location determination . 10

2.5 Experimental setup for mass moments of inertia 12

2.6 Coefficients for additional moments of inertia 15

3.1 The Ranger EX 757-3 21

3.2 The Ranger EX 757-3 top view 21

3.3 Pixhawk flight controller features 24

3.4 Data flow of avionics and sensors 26

3.5 QGroundControl sensor calibration environment 27

3.6 Mission Planner telemetry display environment 28

3.7 Simulink model uploaded on Pixhawk to perform system identification flight experiment 30

4.1 Ranger EX 757-3: center of gravity experimental setup 33

4.2 Horizontal and vertical position of center of gravity from datum point 34

4.3 Ranger EX 757-3: mass moments of inertia experimental setup 35

4.4 Aircraft nonlinear simulation environment (source: University of Minnesota) 38

4.5 Actuator dynamics: system identification experimental setup 39

4.6 Frequency response comparison between actuator model(—) and experimental data(▪▪▪) 41

4.7 Time plot comparison of model fit with experimental data 41

5.1 System identification flowchart 45

5.2 Comparison of Simulated and Estimated Angle-of-Attack 51

Figure Page

5.3 Error of Estimated Angle-of-Attack 51

5.4 Error of Estimated Angle-of-Attack in Frequency Domain 52

5.5 Comparison of the simulated and estimated pitch acceleration 53

5.6 Error of estimated pitch acceleration in time domain 53

5.7 Error of estimated pitch acceleration in frequency domain 54

5.8 Simulated frequency sweep signal generated to elevator 56

5.9 Flight test results of longitudinal time response to frequency sweep 57

5.10 Pitch rate frequency response comparison between grey-box identified model(—) and experimental data(┅) 59

5.11 Vertical accelration frequency response comparison between grey-box identified model(—) and experimental data(┅) 60

5.12 Sweep validation of the identified longitudinal models 61

5.13 Doublet validation of the identified longitudinal models 62

5.14 Doublet validation of the short period model from nonlinear simulation . . . 64

ABBREVIATIONS

ADC	Analog-to-Digital Converter
AFS	Active Flutter Suppression
CAN	Controller Area Network
CCV	Control Configured Vehicle
CG	Center of Gravity
DAST	Drones for Aerodynamic and Structural Testing
DFT	Discrete Fourier Transform
DOF	Degree of Freedom
DSM	Digital Spectrum Modulation
EPO	Expanded PolyOlefin
ESC	Electronic Speed Controller
FAA	Federal Aviation Administration
FAR	Federal Aviation Regulation
FPV	First-Person-View
GPS	Global Positioning System
HIL	Hardware-in-the-loop
IMU	Inertial Measurement Unit
I2C	Inter-Integrated Circuit
LED	Light Emitting Diode
MUTT	Multi-Utility Technology Testbed
NACA	National Advisory Committee for Aeronautics
NASA	National Aeronautics and Space Administration
PIL	Parameter Identification Library
PSP	Pilot Support Package
PVC	PolyVinyl Chloride
PWM	Pulse Width Modulation
RGB	Red Green Blue
RSSI	Received Signal Strength Indicator
SIDPAC	System Identification Programs for AirCraft
SIL	Software-in-the-Loop
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver-Transmitter
UAV	Unmanned Aerial Vehicle
USB	Universal Serial Bus
VLM	Vortex Lattice Method

ABSTRACT

Hamada, Said S. MSAE, Embry-Riddle Aeronautical University, May 2018. Development of a Small Unmanned Aerial Vehicle Longitudinal Model for Future Flutter Testing .

This thesis presents the system identification of the longitudinal motion of a small unmanned aerial vehicle with flexible characteristics. The unmanned aerial vehicle of study is an off-the-shelf aircraft with limited knowledge about the dynamic model. The first task in obtaining a model for flutter testing is determining the rigid body model. Specifically, this thesis will focus in the development of the aircraft and obtaining the longitudinal rigid body model. This involves the assembly and configuration of the avionics and sensors, performing experiments to determine mass and inertia properties and using software tools for aerodynamic analysis to get preliminary values for the static and dynamic stability derivatives. The next step was to design flight experiments for the purpose of system identification. The system identification focused on obtaining the longitudinal model and available software tools were used to get comparative results. The best model was identified using the parameter identification library which gave a 65% match in the pitch rate and 70% match in the vertical acceleration flight data from a doublet response. The short period mode was identified to have a frequency of 6.01 rad/s and a damping ratio of 0.977.

1. Introduction

The advent of Unmanned Aerial Vehicles (UAVs) has provided a means for many researchers to develop platforms on which safe, accurate and reliable testing is done. UAVs that are used as research platforms can be designed to desired configurations or can be just bought off-the-shelf. An example of a UAV that was designed to a desired configuration is the Lockheed X56A which is a flying wing designed to have structural properties ideal for flutter testing(Conner, 2014). An example of a research platform that uses an off-the-shelf UAV is one in Virginia Polytechnic and State University that uses the Telemaster by Hobby Express (Arifianto, 2013). In either case, using UAVs have lowered costs, allowed tailoring of the aircraft features and have provided safe means of testing controllers, aerodynamic surfaces, etc.

Designing UAVs for research purposes is advantageous in that specific aerodynamic, structural and performance requirements can be obtained to provide an ideal platform for the research objectives. However, the costs of obtaining such an ideal platform make this option undesirable. Therefore the resulting option is to get a low-cost, off-the-shelf UAV that might have the desirable properties or at least can be modified and still make it a suitable platform for testing. The downside of this route however is that most of the important properties, like inertia and stability derivatives, that are needed to model the rigid body model of the aircraft for simulation are not readily available.

The platform analyzed in this thesis, for future flutter testing, is an off-the-shelf UAV chosen based on the visible structural properties and customer reviews. As previously mentioned, having chosen an off-the-shelf UAV with no model available or developed, the platform can not be used for flutter testing already without obtaining a rigid body model of it. Therefore the focus of this thesis will be to obtain the longitudinal rigid body model of the aircraft. This involves performing experiments and software analyses to determine the mass,inertia and aerodynamic properties of the aircraft. Chapter 2 will give a background of dynamic aeroelasticity which is the main reason of developing this research platform. It will describe past and present state of the art models that have and are being used to test flutter. In addition, literature review on the experiments and the flight tests done in this thesis will be presented in this chapter. This will include the methodologies for obtaining the mass and inertia properties and system identification. Chapter 3 will describe the flight test platform. This will introduce the UAV that was chosen and describe the hardware/ software used and the methodology used to perform a successful flight test. Chapter 4 will describe the flight dynamics model which involves the results of the experiments and software analyses conducted to obtain preliminary results on the aircraft properties. Chapter 5 will describe the longitudinal system identification analyses that was done which involves the design of the flight experiment and the methodology used for the calculations. Lastly, Chapter 6 will provide a conclusion to the results and the future work of this research.

2. Background

Dynamic aeroelasticity is the study of interactions between aerodynamic, elastic and inertial forces and moments that govern aircraft behavior. These interactions cause undesirable and dangerous phenomena to occur and due to their destructive nature, testing must be conducted in the end of the design process to ensure none of the phenomena occur within the flight envelope of the aircraft. One of the phenomena is aeroelastic flutter which is an unstable oscillation that is excited on a structure when it travels at a critical airspeed. Naturally for such a dangerous phenomenon, measures must be taken to ensure they are not exhibited on the aircraft. Currently, the Federal Aviation Administration (FAA) stipulates under the Federal Aviation Regulation (FAR) Part 23 Section 629 passive methods like structural stiffening and mass balancing to prevent flutter. As preventative as these methods can be, they undeniably add a lot of extra weight on the aircraft and hence reduce its performance. An example to illustrate this is the NASA arrow-wing configuration supersonic cruise airplane which had to add 10,000 lbm to “increase the flutter speeds to acceptable levels” (Doggett & Townsend, 1976). These methods generally assume rigid body dynamics hence in existing control laws, rigid body dynamics and structural dynamics are typically decoupled (Chiappa, Magni, Gorrec, Kubica, & Doll, 1998). However, to improve performance objectives, engineers resort to lighter, more flexible materials such as composites. In addition, with the technological advancement of hardware and actuating systems, the possibility of designing Active Flutter Suppression (AFS) systems has resulted in several methods being developed to try suppress flutter (Livne, 2014). Examples of these methods include robust control technology (Theis, Pfifer, & Seiler, 2016), aerodynamic energy consideration (Maynard, Irving, & Gray, 1976) and method of receptances (Papatheou, Tantaroudas, & Ronch, 2013). Most of these methods have only been wind tunnel tests (Maynard et al., 1976; Papatheou et al., 2013; Doggett & Townsend, 1976) and a few have reported in-flight tests (Roger, Hodges, & Felt, 1975). With these innovative AFS systems becoming available, an avenue to faster and better performing aircraft opens up. However, due to the violent nature of flutter, it is very dangerous to test them during flight. Nevertheless, with the advent of Unmanned Aerial Vehicle (UAVs), a platform can now be established that resembles an aircraft of a conventional configuration and an in-depth study can therefore be undertaken on the nature of flutter and how it can be suppressed using active controls.

2.1 State of the Art Models

One of the early models to test AFS systems was a simplified delta-wing model tested in NASA Langley's transonic dynamics wind tunnel. This model used the method of aerodynamic energy criterion proposed by E. Nissim (Nissim, 1971) that sent commands to leading and trailing edge control surfaces using feedback signals from accelerometers located on the wing. Three different control laws were tested where two of them used both the leading and trailing edge control surfaces whilst the other only used the trailing edge control surfaces. These control surfaces were located at different span-wise locations to observe its effects on the flutter speeds. The delta-wing model was a “simplified” and “inexpensive” model that had high fineness bodies to simulate engine nacelles and a mounting plate for the fuselage. It was designed to resemble a full-scale prototype for a proposed supersonic transport wing and simulate the flutter characteristics. The wind tunnel results were “reasonably close” to the analytical results and showed a significant change in the flutter speeds with varying span-wise locations of the control surfaces (Maynard et al., 1976). A figure of the model is shown in Figure 2.1 (Maynard et al., 1976).

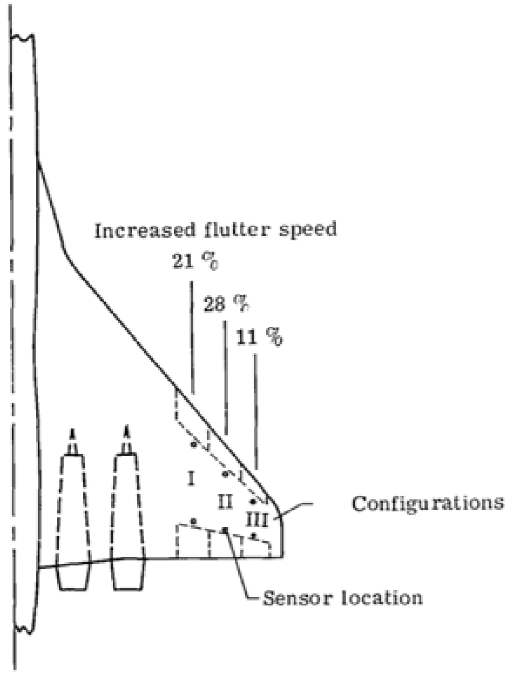


Figure 2.1: Simplified delta-wing model

In the 1970s, Boeing managed to have the first successful flight test demonstration of active flutter suppression. A Control Configured Vehicle (CCV) B-52 was used as the test model and twice it managed to fly 10 knots faster than the flutter speed “solely on an automatic control system for adequate damping" (Roger et al., 1975). The CCV B-52 used ballasts in the fuel tanks to change the structural dynamics in a way the flutter frequency was of a "mild nature" of 2.4Hz, making it slow enough for the pilots to recover in any case of failure in the AFS system. Figure 2.2 (Roger et al., 1975) shows the CCV B-52 with the modifications that were made on it.

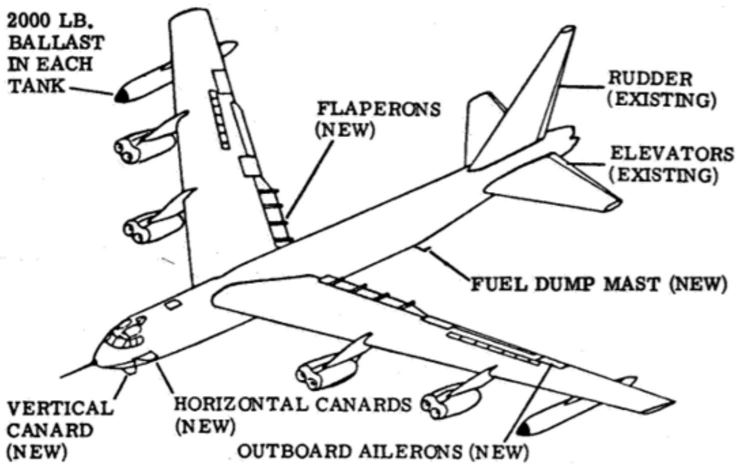


Figure 2.2: The Control Configured Vehicle B-52

With the fateful risks of testing AFS systems in manned flights, contemporary research has shifted into using UAVs as test models. The DAST program (Calzada, 2002), which was conducted in the late 70s, was one of the early models to use a UAV. Calling it the "wind tunnel of the sky", the wing was reconstructed to make it more flexible and a digital flutter suppression system was designed and integrated into its flight control system. The platform however experienced many problems which led to some mission abortions and crashes. With other programs taking priority, the DAST program was disbanded. Another more recent program was the NASA X-56A Multi-Utility Technology Testbed (MUTT)

(Figure 2.3) which uses high aspect ratio wings to collect flight test data on the flexible structural dynamics and the developed AFS system augmented in the model (Conner, 2014). The University of Minnesota Aerospace Department designed and built a mini MUTT aircraft to test the robust control technology developed for the model. The developed controllers showed promising results however, they are yet to validate their AFS in a test flight. Their mini MUTT aircraft unfortunately had a catastrophic failure when tested without AFS (Theis et al., 2016).

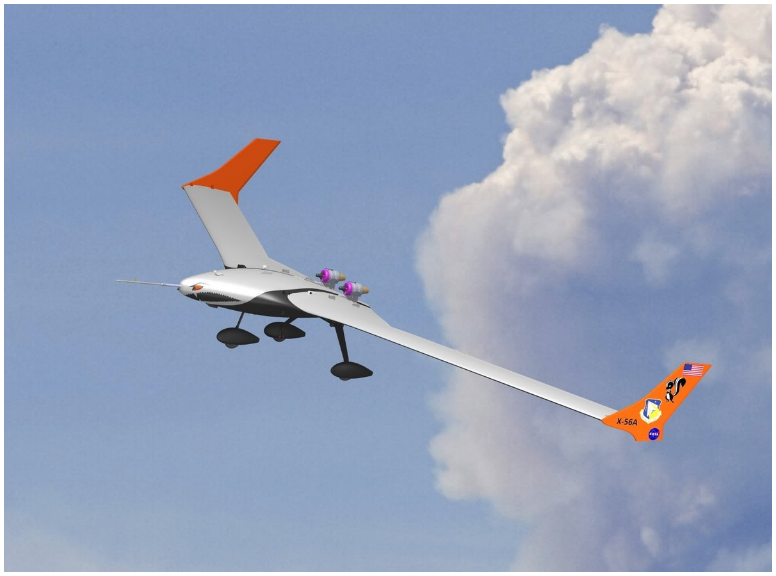


Figure 2.3: The Lockheed Martin X56-A

The ascent of UAVs has therefore provided a safe means of using a reasonable research platform that can ideally replicate the behavior of manned aircraft. With careful consideration, a research platform can be developed that does not only limit testing on only the wing, but all the other parts that are found in a conventional aircraft.

2.1.1 Aircraft Modeling for Flutter Testing

As explained earlier in the chapter, flutter occurs as a result of the interactions between the aerodynamic, inertial and flexible properties of an aircraft. Therefore modeling an aircraft for flutter testing requires a flexible model be obtained for it. The equations of motion of a flexible moving body can be decoupled to the rigid body dynamics and flexible dynamics using mean axes assumptions. This is shown in equation 2.1 (Moreno, 2015).

$$\begin{bmatrix} m_b(\dot{V}_b + \omega_b \times V_b - T_{bE}g_E \\ \\ I_b\dot{\omega}_b + \omega_b \times (I_b\omega_b) \end{bmatrix} = \Phi_b^T \mathcal{P}^c \tag{2.1}$$

$$\hat{M}_f\ddot{\eta}_f + \hat{\Xi}_f\dot{\eta}_f + \hat{K}_f\eta_f = \Phi_f^T \mathcal{P}^c$$

m_b here is the total body mass, I_b is the mass inertia, g_E is the gravitational vector, and Φ_b is the rigid body modal matrix about the center of gravity with directions customary in flight mechanics. V_b and ω_b are the velocity and angular velocity in the mean axis body frame of reference, T_{bE} transforms the gravitational vector from an earth fixed cordinate frame (E) to the body fixed coordinate frame (b). η_f is the vector of elastic modal displacements, $\hat{M}_f$ is the generalized modal mass matrix, $\hat{K}_f$ is the generalized modal stiffness matrix, $\hat{\Xi}_f$ is the generalized damping matrix, Φ_f is the flexible modal matrix and $\mathcal{P}^c$ is the vector of aerodynamic forces and moments applied to the airframe (Moreno, 2015). Therefore as seen from this equation, it is significant to determine the rigid-body properties of the aircraft before a flexible body is developed. The next sections with go through the literature review of the experiments conducted to determine the rigid-body properties/model of the UAV test platform that will be used in this research.

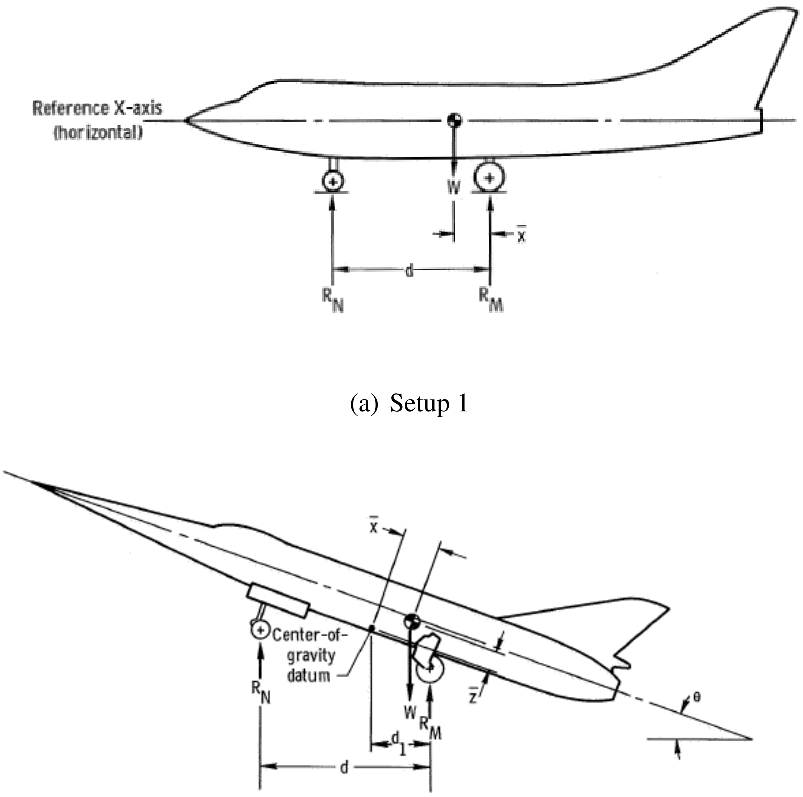
2.2 Mass/Inertia Properties

The mass and inertia properties were obtained by doing two sets of experiments: one setup to determine the mass and center of gravity (CG) of the aircraft and one to determine the moments of inertia. These include I_x, I_y, I_z and I_{xz}.

2.2.1 Weight and CG experiment

The first experiment setup to determine the mass and CG properties was designed as per the procedure laid out in the NASA technical report “Experimental Determination of Airplane Mass and Inertial Characteristics” (Wolowitz & Yancey, 1974). Two experiments were conducted under this setup. One experiments determines the weight and the horizontal position with respect to one of the landing gear. This involves placing one mass balance at the front wheel and one mass balance at the back wheel of the landing gear. Each mass balance will measure a reaction force and the sum of them will give the weight of the aircraft. Measuring the horizontal distance between the landing gear and the measurements of the reactions, summing the moment about one of the wheels will give the horizontal position of the CG. Figure 2.4(a) illustrates this experimental setup.

The second experiment setup determines the same parameters as the first setup but also includes the vertical position of the CG. This involves placing mass balances on the landing gear, picking a datum to which measurements will be taken about and repeating the experiment for several inclination angles of the plane. This means for every inclination angle, the reaction forces measured on the mass balances, the distance between the mass balances



(a) Setup 1

(b) Setup 2

Figure 2.4: Experimental setup for weight and center of gravity location determination

and the distance between the back wheel and the datum are measured. Figure 2.4(b) shows this experimental setup. As seen from figure 2.4(b), summing the forces gives the equation:

$$R_N + R_M = W \tag{2.2}$$

Summing the moments about the datum gives the equation:

$$R_N d - (R_N + R_M)(d_l - \bar{z}\sin\theta - \bar{x}\cos\theta) = 0 \tag{2.3}$$

which simplifies to:

$$\bar{x} + \bar{z}\tan\theta = (d_l - d\frac{R_N}{W})\frac{1}{\cos\theta} \tag{2.4}$$

Equation 2.4 represents an equation of a line where plotting $(d_l - d\frac{R_N}{W})\frac{1}{\cos\theta}$ against $\tan\theta$ will give a slope representing $\bar{z}$ and a y-intercept representing $\bar{x}$.

2.2.2 Moments of Inertia

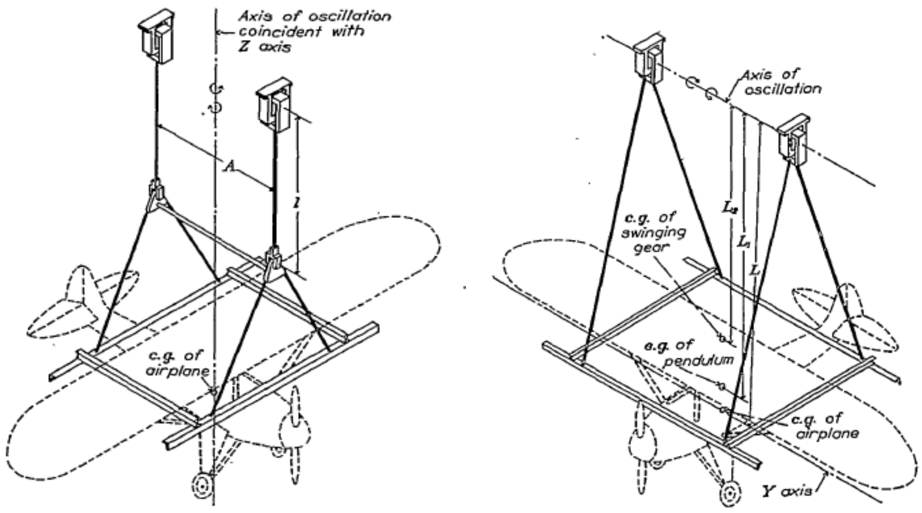
The design for the experiments to determine the moments of inertia was per the guidelines and procedure laid out in the NACA report "The Experimental Determination of the Moments of Inertia of Airplanes" (Soule & Miller, 1934). This experiment adopts the pendulum method of obtaining the moments of inertia. Assuming an undamped pendulum oscillating with small amplitude in a vacuum, the equation of motion is:

$$I\frac{d^2\theta}{dt^2} + b\theta = 0 \tag{2.5}$$

where I is the moment of oscillation about the axis of rotation, b is a constant depending on the weight and geometry of the pendulum and θ is the angular displacement. Two different pendulums are used: the bifilar pendulum used to determine the inertia about the Z (vertical) axis and the compound pendulum to determine the inertia about the remaining axes. Figures 2.5(a) and 2.5(b) show the bifilar and compound pendulum setup respectively.

For the bifilar pendulum,

$$b = \frac{WA^2}{4l} \text{ and hence } I = \frac{T^2WA^2}{16\pi^2 l} \tag{2.6}$$



(a) Bifilar pendulum setup

(b) Compound pendulum setup

Figure 2.5: Experimental setup for mass moments of inertia

where W is the weight of the pendulum. For the compound pendulum,

$$b = WL \text{ and hence } I = \frac{T^2 WL}{4\pi^2} \tag{2.7}$$

To take into account the fact that the axis of rotation is not passing through the CG for the compound pendulum, the moment of inertia passing through the CG will consequentially be,

$$I_{CG} = \frac{T^2 WL}{4\pi^2} - ML^2 \tag{2.8}$$

where M is the mass of the pendulum.

The damping in this experiment is ignored because the “observations made from the swinging experiments have shown that the decrease of amplitude during the first oscillation never exceeds one tenth the original amplitude” (Soule & Miller, 1934). The error that results from the moment of inertia by neglecting damping is below 0.02 percent.

Since the experiment is not conducted in a vacuum, the values determined for I, W and M in the previous equations are known as “virtual values”. The ambient air causes a difference in the values because of the effects of buoyancy, air trapped in the structure and “additional-mass effect.” The buoyancy and entrapped air causes the weight that is measured in air to be the virtual weight which in reality is the true weight minus the buoyancy of the structure. And so the mass that needs to be used in equation 2.8 is

$$M = \frac{M}{g} + V\rho \tag{2.9}$$

The additional mass effect is brought about by changes in the airflow momentum imparted on the body as the aircraft oscillates in the air. This additional momentum is proportional to the size and shape of the body relative to the direction of motion. Due to the relatively low mass density of the aircraft, this additional momentum effect needs to be taken into account. Even though this effect is due to the additional momentum, it is known as the additional-mass effect as this additional momentum is directly proportional to momentum of the body. And so the virtual moment of inertia (I_v) for the compound pendulum is given by:

$$I_v = \frac{T^2WL}{4\pi^2} - (\frac{W}{g} + V\rho + M_A)L^2 \tag{2.10}$$

where M_A is the additional mass considered from the additional-mass effect. This is determined by assuming the parts of the airplane that cause additional momentum in the axis of rotation are flat plates. The shapes of the plates can be considered rectangular or circular for example the fuselage can be considered rectangular and the vertical tail circular when considering the rotation about the X axis. The equation for the additional mass for a rectangular plate is:

$$M_A = \frac{\rho c^2 \pi b}{4} \tag{2.11}$$

and for the circular plate:

$$M_A = \frac{\pi D^3 \rho}{6} \tag{2.12}$$

The equations as of now have assumed the aircraft as the sole object on the pendulum, however the effects of the supporting mechanism of the aircraft need be considered. Hence the virtual inertia for the bifilar pendulum becomes:

$$I_v = \frac{T_1^2 W_1^2 A^2}{16\pi^2 l} - \frac{T_2^2 W_2^2 A^2}{16\pi^2 l} \tag{2.13}$$

and for the compund pendulum:

$$I_v = \frac{T_1^2 W_1 L_1}{4\pi^2} - \frac{T_2^2 W_2 L_2}{4\pi^2} - (\frac{W}{g} + V\rho + M_A)L^2 \tag{2.14}$$

Finally to get the actual moments of inertia, the additional moment of inertia needs to be subtracted from the virtual moment of inertia. The flat plate assumption is similarly used in the equation where the additional moment of inertia for rotation about any axis parallel to the chord as:

$$I_A = \frac{k' \rho \pi c^2 b^3}{48} + \frac{k \rho \pi c^2 b l^2}{4} \tag{2.15}$$

and for rotation about any axis parallel to the span:

$$I_A = \frac{k \rho \pi c^2 b l^2}{4} \tag{2.16}$$

where l is the distance in the plane of the plate from the center of the plane to the axis of rotation. The values k and k' are obtained from the graphs seen in figure 2.6.

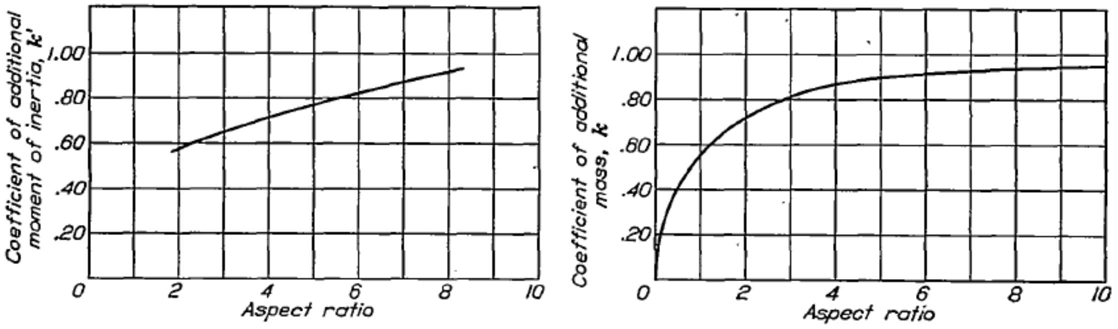


Figure 2.6: Coefficients for additional moments of inertia

2.3 System Identification

One of the important elements of determining a mathematical model of a plant is to accurately capture the dynamics of it. As will be seen in the later chapters, the first step of determining the mathematical model of the UAV used in this thesis was to use software tools to get preliminary values for the aerodynamic properties. These preliminary values form a baseline model that can be used to get an overview of how the UAV behaves. However due to assumptions and limitations of accurately representing the aircraft using the software tools, methods of obtaining accurate models using input and output flight test data can be found in literature. System identification is defined as “the determination, on the basis of observation of input and output, of a system within a specified class of systems to which the system under test is equivalent” (Klein & Morelli, 2006). This means some a priori knowledge of the system and the ability to accurately measure the inputs and outputs needed for the identification are significant.

For UAVs, system identification poses a challenge due to restrictions on the availability and quality of sensors. These restrictions come from the airframe size which limits the sensors that can be mounted on board and from the cost limitations of maintaining a low cost platform. For example regular air data booms used on aircraft to determine angle-of-attack and side-slip are too big for small UAVs and multi-hole probes that enable the estimation of angle-of-attack and side-slip are usually very expensive. Based on these limitations, a flight experiment needs to be designed based on the objectives of the identification. This design together with the results will be discussed in Chapter 5. In terms of the system identification techniques used, they depended on the tool that was used. Three main tools were used in this research: the System Identification toolbox in MATLAB, System Identification Programs for AirCraft (SIDPAC) and the Parameter Identification Library (PIL). The technique used from the System Identification toolbox in MATLAB is by estimating a linear grey-box model while SIDPAC and PIL use equation-error parameter estimation method. The next two sections provides background on each method.

2.3.1 Linear Grey-box Model

The function in MATLAB that uses this method is called “greyest”. It uses a numerical optimization method that minimizes the weighted norm of the prediction error (Ljung, 1999). The scalar output of the cost function is defined in equation 2.17.

$$V_N(G,H) = \sum_{t=1}^{N} e^2(t) \tag{2.17}$$

where $e(t)$ is the difference between the measured output and predicted output of the model. What is fed in the optimization are the transfer functions from the flight data, which are estimated using the function “tfestimate”. The background of this function is described in section 4.3.1.The linear grey-box model algorithm then predicts a model based on the defined linear equations that best fits the estimated transfer functions in the frequency domain.

2.3.2 Equation-Error Parameter Estimation

One of the techniques used in aircraft system identification is a statistical approach called regression where relationships are determined between measured variables (Klein & Morelli, 2006). One example is the model relating the non-dimensional moment coefficient to angle-of-attack and Mach number for data collected in a wind tunnel test.

$$C_m = C_{m_o} + C_{m_\alpha}\alpha + C_{m_M}M + C_{m_{\alpha M}}\alpha M + v_m \tag{2.18}$$

From this equation, α and M are the independent variables hence are assumed to be measured without error. C_m is the dependent variable and because this is a wind tunnel test, it can be directly measured by non-dimensionalizing the moment measured by a strain-gauge balance. In a flight test, this value is usually obtained using other sensor measurements of the translational and rotational motion of the aircraft. Errors from random measurement are taken into account in the term v_m. The constant model parameters that are to be identified are $C_{m_o}, C_{m_\alpha}, C_{m_M}$ and $C_{m_{\alpha M}}$. Since the equation is linearly dependent on the model parameters, this refers to linear regression. For flight test data, the modeling problem changes because the independent variables, like angle-of-attack, tend to depend on parameters like control surface deflection. However the fundamental mathematical problem does not change. Since identifying the model parameters using least-squares minimizes the error in the pitching moment equation, another name coined for this method is equation error. The general form of the regression equation used in the ordinary least squares regression is:

$$z = X\theta + v \tag{2.19}$$

where $\boldsymbol{z} = N \times 1$vector of output measurements,$\theta = n \times 1$ vector of unknown model parameters ,$X = N \times n$ matrix of regressors and $v = N \times 1$ vector of measurement errors. For least squares, the best estimator for θ comes from minimizing the sum of the squared differences between the measurements and the model:

$$J(\theta) = \frac{1}{2}(z - X\theta)^T(z - X\theta) \tag{2.20}$$

The parameter estimate $\hat{\theta}$ that minimizes the cost function $J(\theta)$ must satisfy:

$$\frac{\partial J}{\partial \theta} = -X^T z + X^T X\hat{\theta} = 0 \tag{2.21a}$$

or

$$X^T X\hat{\theta} = X^T z \tag{2.21b}$$

or

$$X^T(z - X\hat{\theta}) = 0 \tag{2.21c}$$

Therefore the solution to the ordinary least-squares estimator becomes:

$$\hat{\theta} = (X^T X)^{-1} X^T z \tag{2.22}$$

Therefore $\hat{\theta}$ will contain all the identified model parameters from the regression. In this research, the least-squares method shown is used but with the inputs and outputs in the frequency domain. This provides a physical insight on the frequency content of the system which is directly applicable for control applications. Also frequency domain methods are more efficient computationally due to the smaller number of points for parameter estimation (Klein & Morelli, 2006). The transformations for the time history data to the frequency domain together with the methodology of the system identification is explained in Chapter 5.

Before any of the experimental methods presented in this chapter are implemented, a research platform needs to be established. This includes choosing a suitable UAV and determine the avionics and sensors needed to conduct the research. The next chapter will introduce the UAV and describe all the components used in the platform.

3. Flight Test Platform

The flight test platform used in this research can be broken down into several components: the physical properties of the bare-airframe, the hardware, and the software used including the avionics and sensors. The following sections give a detailed description of each of these components.

3.1 Airframe

The UAV used in this research is a low-cost, off-the-shelf aircraft called the Ranger EX 757-3 made by Volantex. It is a light, long range vehicle that is used for First-Person-View (FPV) flying and has a high wing with a wing span of about 2 meters. The body of the aircraft is made of PVC making it durable and crash resistant while the lifting surfaces are all made of EPO foam. The wing has an aluminum rod within it to help support the aerodynamic loads exhibited on it. With the long wing span and the foam making it light and providing structural elasticity, it is perceived that the Ranger EX 757-3 is an ideal platform for flutter testing. In addition, customer reviews suggest experiencing wing flexing under high loads (Cole, 2014). The control surfaces of the aircraft include simple inboard flaps, ailerons, elevators and rudder. Figure 3.1 shows an image and figure 3.2 shows the top view and some dimensions of the UAV.

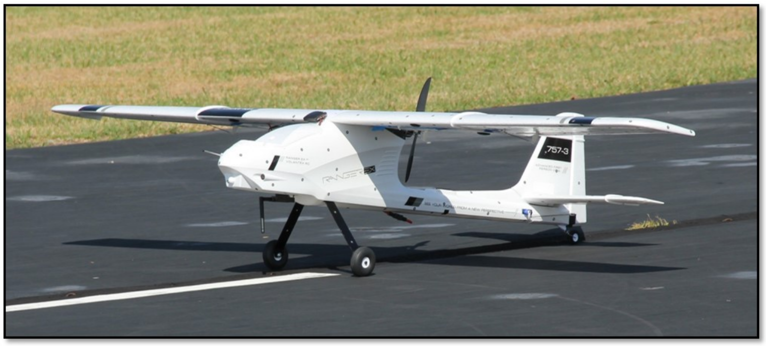


Figure 3.1: The Ranger EX 757-3

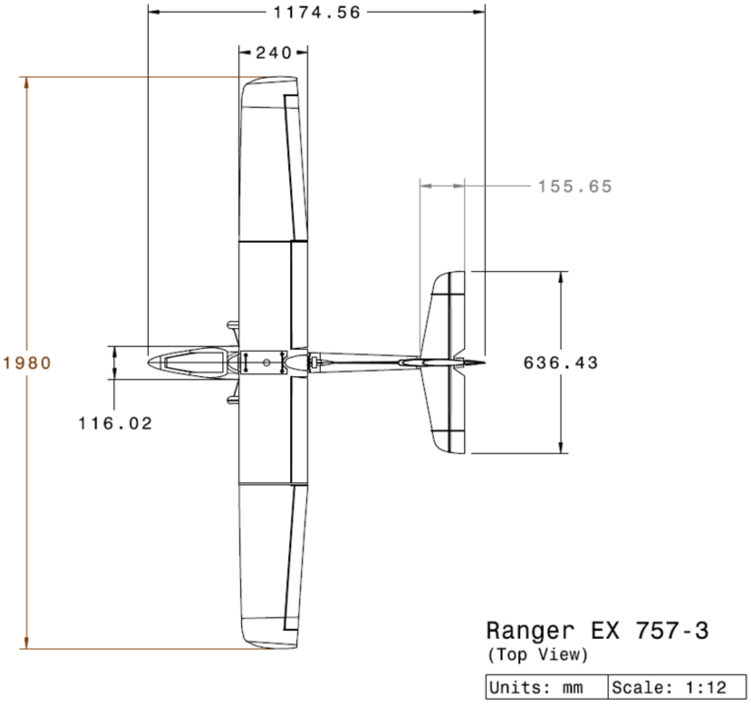


Figure 3.2: The Ranger EX 757-3 top view

3.2 Avionics and Sensors

The avionics and sensors used are all off-the-shelf and relatively inexpensive to offer a low-cost platform for flutter testing. The hub of the avionics is the flight computer Pixhawk autopilot by 3D Robotics. All servos and external sensors are connected to the Pixhawk. All autopilots and controls are programmed on it, and with the SD card housing available, all measurements are logged using it. The Pixhawk houses a processor, an Inertial Measurement Unit (IMU) and interfaces to communicate with other electronic devices. The processor specifications are 32 bit STM32F2427 Cortex M4 core with FPU, 168 MHz, 256 KB RAM, 2 MB Flash and 32 bit STM32F103 fail-safe co-processor. The sensors present in the IMU together with the specifications are presented in table 3.1. The available interfaces on the Pixhawk are described in table 3.2 (Pixhawk, n.d.). Figure 3.3 (Pixhawk, n.d.) shows the Pixhawk with all the features and ports available on it.

Table 3.1: Pixhawk sensors

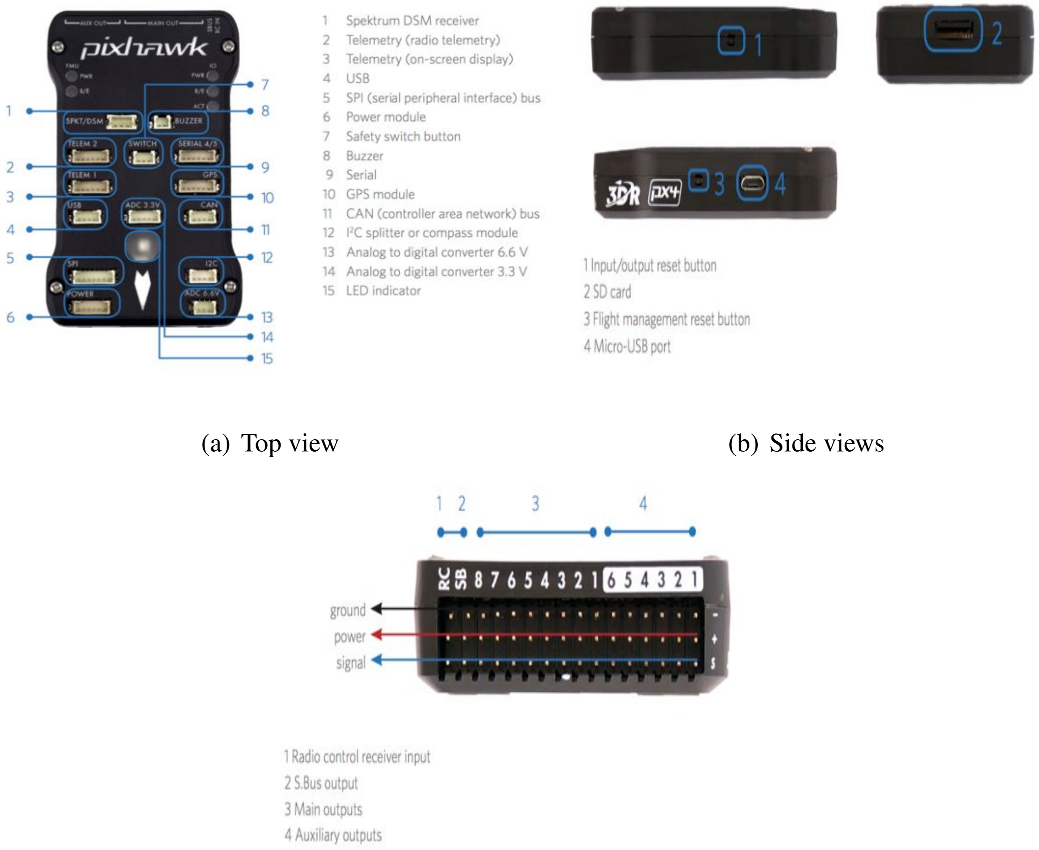
Sensor	Specification	Use
Gyroscope	ST Micro L3GD20H 16 bit	Measure body angular rates
Integrated accelerometer/-magnetometer	ST Micro LSM303D 14 bit	Measure body accelerations/heading
3-axis accelerometer/gyroscope	Invensense MPU 6000	Measure body accelerations and angular rates
Barometer	MEAS MS5611	Measure altitude

The control surfaces are deflected using MG90s 9g Micro Tower Pro servos which receive signals from the Pixhawk in Pulse Width Modulation (PWM). They are all connected

Table 3.2: Pixhawk interfaces

Interface	Abbreviation	Use
Universal Asynchronous Receiver-Transmitter ports	UART	Configure data formats and transmission speeds for serial communication
Controller Area Network bus	CAN	Allow communication with other devices
Spektrum/ Digital Spectrum Modulation port	DSM	Communicate with transmitters
Received Signal Strength Indicator	RSSI	Measurement of power for received signals
Inter-Integrated Circuit	I^2C	Protocol to communicate with devices
Serial Peripheral Interface bus	SPI	Serial communication interface
Analog-to-Digital Converter	ADC	Converts analog signals to digital
micro-Univeral Serial Bus ports	micro-USB	Standard connector to communicate with computer

on the main output channels seen in figure 3.3(c) with the aileron servos connected to channel 1, elevator servo to channel 2 and rudder servo to channel 4.


pixhawk
1 Spektrum DSM receiver
2 Telemetry (radio telemetry)
3 Telemetry (on-screen display)
4 USB
5 SPI (serial peripheral interface) bus
6 Power module
7 Safety switch button
8 Buzzer
9 Serial
10 GPS module
11 CAN (controller area network) bus
12 I²C splitter or compass module
13 Analog to digital converter 6.6 V
14 Analog to digital converter 3.3 V
15 LED indicator
1 Input/output reset button
2 SD card
3 Flight management reset button
4 Micro-USB port
ground
power
signal
1 Radio control receiver input
2 S.Bus output
3 Main outputs
4 Auxiliary outputs

(a) Top view

(b) Side views

(c) Front view

Figure 3.3: Pixhawk flight controller features

For controlling throttle, commands were sent from channel 3 to a 40A Electronic Speed Controller (ESC) which was connected to a 1000kV brushless motor. The motor was powered by a 5400 mAh 14.8V 4 cell LiPo battery through an APM power module which has a cable running to the ESC and another one to power the Pixhawk. For a livestream of data recorded in-flight, a 915MHz 3DR telemetry radio transmitter is attached at the lower exterior surface of the fuselage and is connected to the TELEM1 port shown in figure 3.3(a). The data is streamed at a baud rate of 57600 and is picked up by the receiver on the ground station which then displays the desired aircraft states to the user. The software that is used on the ground station to provide a real-time graphic display of the aircraft states is an open-source platform called Mission Planner by Michael Oborne. This software communicates with the Pixhawk using MAVLINK messages and can be used to upload firmware, calibrate sensors, create waypoint navigation, monitor aircraft properties, record telemetry logs and also analyze desired variables.

The external sensors connected to the Pixhawk include a pitot tube recommended by the Pixhawk developers to measure airspeed, a Ublox GPS + compass module to monitor location and 4 triple axes accelerometers attached on the wing to measure in-flight vibrations. The pitot tube was located at the nose of the aircraft to avoid turbulent interference on the airspeed measurement and was connected to the Pixhawk through the I2C port together with the magnetometer on the GPS module. The accelerometers, to avoid over-loading of the Pixhawk and due to available drivers, were configured on a MEGA2560 arduino board and their data logged on it. The data flow between the avionics is seen in figure 3.4.

3.3 Software

Three main software tools were used to configure the UAV for a flight: QGroundControl, Mission Planner and the MATLAB Pixhawk Pilot Support Package (PSP) for MATLAB R2015b. These softwares serve the purposes of calibrating the sensors attached or within the Pixhawk, view telemetry data and configuring commands/controls that are to be implemented in-flight. The following subsections describe the purposes of each of the software.

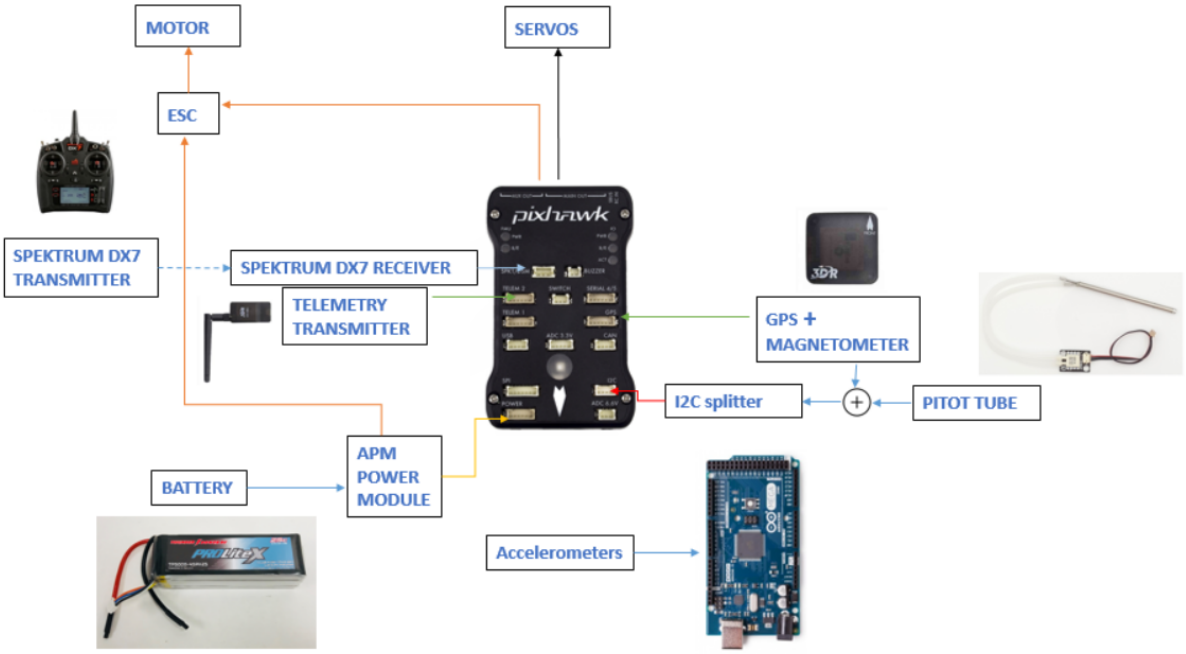
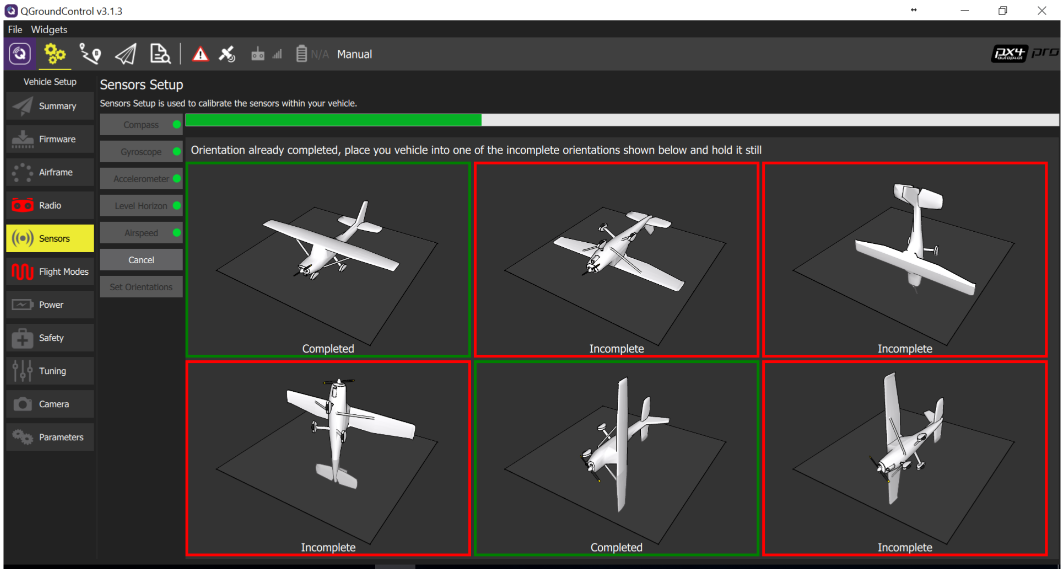


Figure 3.4: Data flow of avionics and sensors

3.3.1 QGroundControl/Mission Planner

These two software have been put together in this section as they are similar in their functions. They are both open source software that can be used to calibrate sensors, view/log telemetry data, create waypoint navigation and configure flight patterns. They both use MAVLINK to communicate with the Pixhawk and can be used to download and load available firmware onto the Pixhawk. The difference and the reason why both of them are used is mainly the methodology of calibrating the sensors. QGroundControl offers a simpler way for calibrating the compass, accelerometers and gyros and also offers calibration of the airspeed sensor which could not be found on Mission Planner.

The process of calibration starts by first uploading the QGroundControl firmware onto the Pixhawk. To calibrate the compass the aircraft is set at different configurations and rotated about the different axes. The accelerometers were calibrated by simply setting and holding the aircraft at different configurations and the airspeed was calibrated by first ensuring the pitot tube was not measuring wind and then blowing across it after. Figure 3.5 shows an example of the graphic instructions QGroundControl shows for the calibration process.



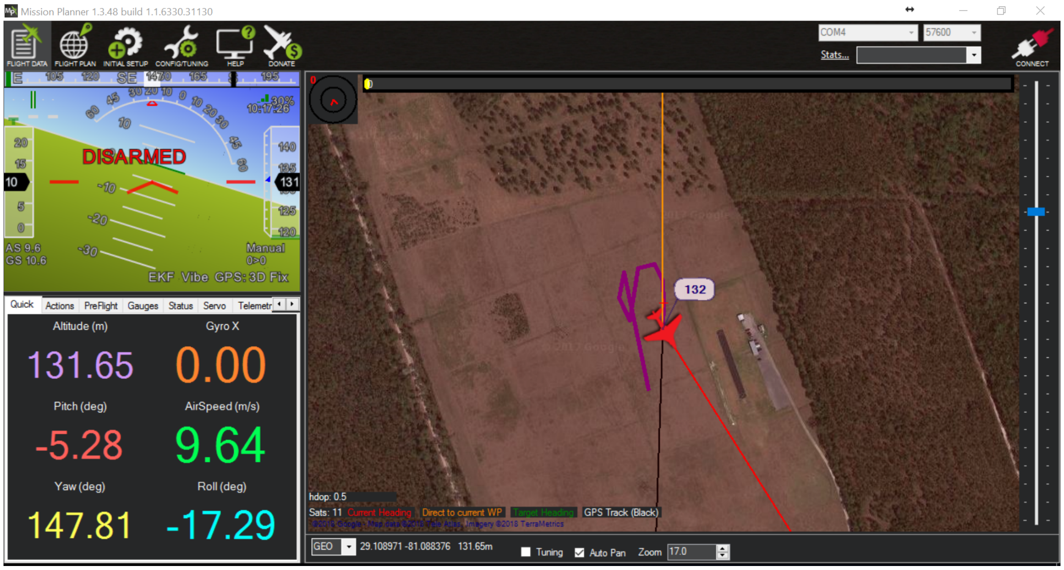
Sensors Setup

Sensors Setup is used to calibrate the sensors within your vehicle.

Orientation already completed, place you vehicle into one of the incomplete orientations shown below and hold it still

Figure 3.5: QGroundControl sensor calibration environment

As mentioned in the previous chapter, Mission Planner is just used to display desired parameters like airspeed and altitude during flight and use the telemetry data, that it automatically logs on the computer, for analysis. Unlike QGroundControl that saves the telemetry logs in binary and needs a Python code to convert the codes to csv files, Mission Planner enables the user to convert the telemetry logs to MATLAB files which is convenient as most analyses of the flight data was done using MATLAB. Figure 3.6 shows the dashboard of Mission Planner where real-time data is displayed.



Altitude (m)	Gyro X
131.65	0.00
Pitch (deg)	AirSpeed (m/s)
-5.28	9.64
Yaw (deg)	Roll (deg)
147.81	-17.29

Figure 3.6: Mission Planner telemetry display environment

3.3.2 Pixhawk Pilot Support Package (PSP)

The Pixhawk PSP enables Pixhawk users to develop custom firmware on MATLAB and Simulink. Instructions to installing the PSP are available in the user manual (Kuznicki & Lee, 2016). The way the PSP works is that desired models are created in Simulink, and once completed, the PSP compiles and generates a C/C++ code that is then uploaded to Pixhawk. Therefore the Simulink blocks that need to be used should be ones for code generation, which means they are blocks that can be compiled and converted to C code. This can be determined by typing the command “showblockdatatypetable” which displays a table with all the blocks and some of their properties. The PSP also comes with S-function blocks that enable the user control certain components on the Pixhawk. For example the transmitter block allows the extraction of the signals coming from the transmitter and manipulate them instead of sending them directly to the actuators. The LED block, another example, controls the RGB LED on the Pixhawk. More details on the blocks and their functions can be found in the manual (Kuznicki & Lee, 2016). In addition, there a blocks that can be used to extract raw, processed or even estimated data from sensors. The blocks use algorithms written in C code known as “topics” and can be called using the uORB Function/Call Trigger and include high/low pass filters and estimators. Measurements of true airspeed, altitude and many more can be extracted using the topics called from this block. The support package also allows the user to create their own S-function block for their own purposes. One thing to note is that since the Pixhawk has an operating system itself and inbuilt applications, a command needs to tell the Pixhawk to use the custom firmware instead of the default. This is done (see manual for more details) by inserting a text file called “rc.txt” that lists all the applications that need to be started up when the Pixhawk turns on. In this file a command is included to use the simulink model uploaded and other commands for instance to open serial ports and allow MAVLINK messages be sent and received. The example of the MAVLINK command allows one to use Mission Planner without necessary using the firmware downloaded from Mission Planner. Therefore even though custom firmware is being used on the Pixhawk, it can still communicate with Mission Planner and enable the user to view whatever parameters they choose to monitor. The support package also comes with a file where the user can enter all the offsets that the sensors have after calibration. After calibrating the sensors on QGroundControl, the offsets can then be viewed on QGroundControl and manually changed in the file “sensor param”. This file will then be compiled together with the model. An overview of one of the models used is shown in figure 3.7.

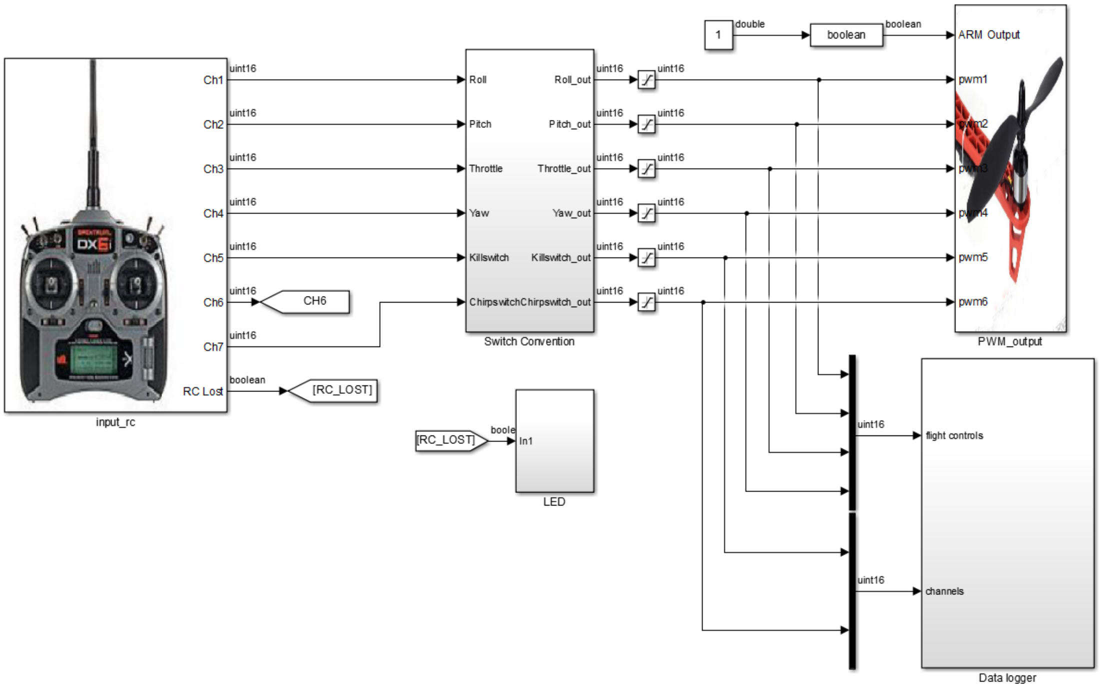


Figure 3.7: Simulink model uploaded on Pixhawk to perform system identification flight experiment

4. Flight Dynamics Model

In Chapter 2, it was seen that modeling an aircraft for flutter testing requires the structural dynamics be integrated to the rigid body model to get an accurate representation of the aircraft in the aeroelastic regime. Therefore as a starting point of the modeling process, the rigid body model needs to be determined. The generalized 6-DOF equations of motion are used of a rigid airframe with constant mass distribution. These equations (Cook, 2007) describe the motion of the aircraft at the center of gravity and can be summarized as:

$$\begin{aligned} m\left(\dot{u}-rv+qw\right) &= X_a+X_g+X_c+X_p+X_d \\ m\left(\dot{v}-pw+ru\right) &= Y_a+Y_g+Y_c+Y_p+Y_d \\ m\left(\dot{w}-qu+pv\right) &= Z_a+Z_g+Z_c+Z_p+Z_d \end{aligned} \tag{4.1}$$

$$\begin{aligned} I_x\dot{p}-\left(I_y-I_z\right)qr-I_{xz}\left(pq+\dot{r}\right) &= L_a+L_g+L_c+L_p+L_d \\ I_y\dot{q}-\left(I_x-I_z\right)pr+I_{xz}\left(p^2-r^2\right) &= M_a+M_g+M_c+M_p+M_d \\ I_z\dot{r}-\left(I_x-I_y\right)pq+I_{xz}\left(qr-\dot{p}\right) &= N_a+N_g+N_c+N_p+N_d \end{aligned} \tag{4.2}$$

where m is the total mass of the aircraft, I_x, I_y, I_z, I_{xz} are the moments and product of inertia about the body axis, u, v, w are the body axis velocities, and p, q, r are the body angular rates. Here, X, Y, Z represent the total forces in the body axis, and L, M, N are the total moments in the body axis. The subscripts a, g, c, p, d correspond to the contributions to the forces and moments from aerodynamic, gravitational, control surface, power and atmospheric disturbance effects. From these equations, it can be seen that the mass and inertia properties and the aerodynamic model of the Ranger EX need to be determined. The following sections describe the experiments/analyses conducted to determine these properties.

4.1 Mass and Inertia Properties

The mass and inertia properties of the Ranger EX 757-3 involves determining the total mass, the center of gravity (CG) location for stability calculations and the moments of inertia. The total mass, m, of the aircraft involves by simply resting the plane on two mass balances, one placed at the front wheels and one at the tail wheel. This yielded a result of 2.41 kg with all the electrical components on-board. The experiments to determine the CG and inertia are described in the following subsections.

4.1.1 Center of Gravity

The horizontal and vertical locations of the CG are determined using the same setup used to measure the total mass (Wolowitz & Yancey, 1974). Figure 4.1 displays this setup. A datum is picked on the UAV and the reaction measurements, R_N, R_M,together with the distances between the reactions (d) and between the datum and the reaction at tail wheel (R_M) are measured for several tilt angles (θ) along the longitudinal plane. These tilt angles were measured using the IMU in the Pixhawk to get accurate measurements. With these measurements, the following equilibrium equations were used to determine the horizontal location ($\bar{x}$) and the vertical location ($\bar{z}$) of the CG:

$$R_M + R_N = W \tag{4.3}$$

$$\bar{x} + \bar{z}\tan\theta = \left(d_1 - d\frac{R_N}{W}\right)\frac{1}{\cos\theta} \tag{4.4}$$

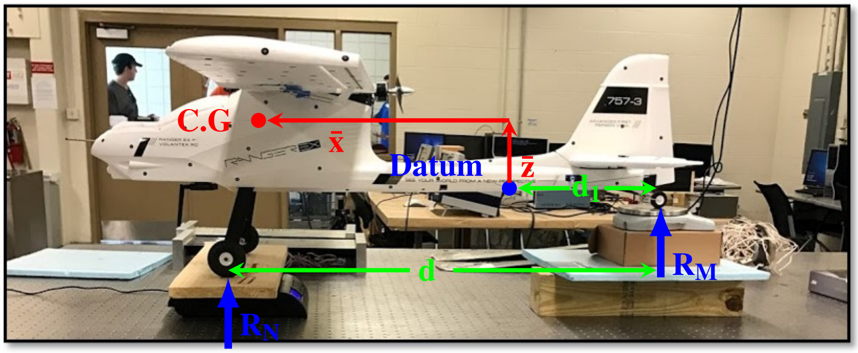


Figure 4.1: Ranger EX 757-3: center of gravity experimental setup

The plot in figure 4.2 shows the resulting CG locations with respect to the datum and using a priori geometrical dimensions, table 4.1 shows the locations from useful reference points of the aircraft in the stability calculations.

Table 4.1: Ranger EX 757-7: center of gravity position [cm]

Reference Point	x_{cg}	z_{cg}
Test Datum	-43.46	11.70
Body Nose	39.80	8.30
Wing Leading Edge	9.80	-5.40

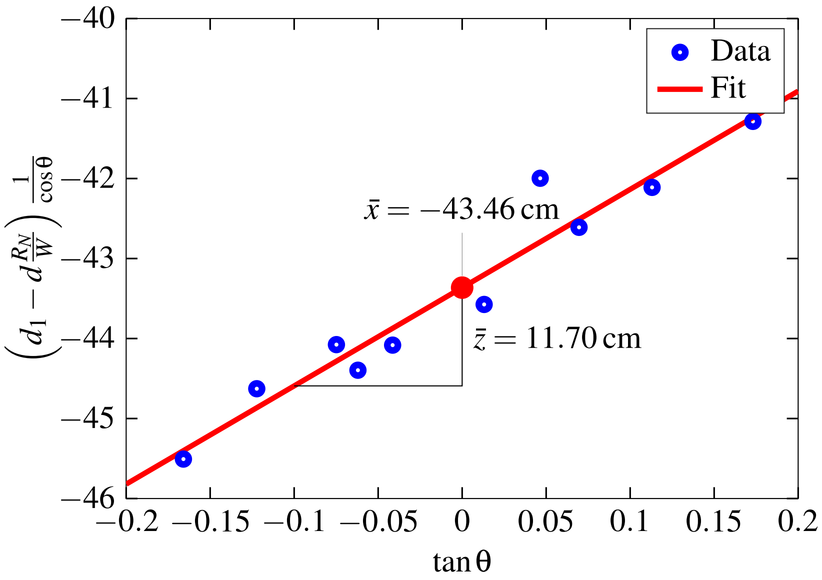


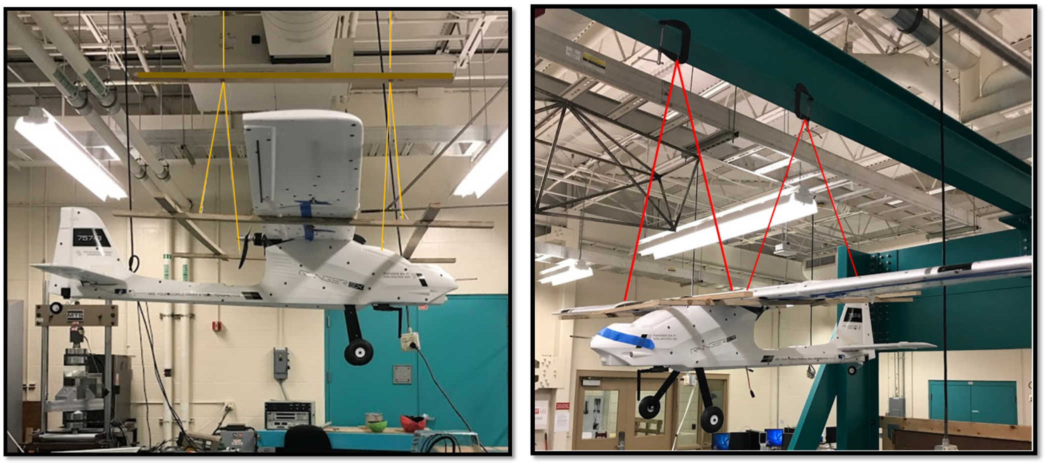
Figure 4.2: Horizontal and vertical position of center of gravity from datum point

4.1.2 Inertia Properties

The moments of inertia for the Ranger EX were obtained using the compound and bifilar pendulum setup (Soule & Miller, 1934) shown in figures 4.3(b) and 4.3(a). The compound pendulum was used to determine the moments of inertia about the x and y axes and the bifilar pendulum to determine the moment of inertia about the z axis. The period of oscillation about each axis was measured using the Pixhawk for accurate measurements. The values for the calculated inertia are displayed in table 4.2

4.2 Aerodynamic Model

The aerodynamic properties of the Ranger EX 757-3 were obtained using an open source software called XFLR5. This software is used to analyze airfoils,wings and planes at



(a) Bifilar pendulum setup (b) Compound pendulum setup

Figure 4.3: Ranger EX 757-3: mass moments of inertia experimental setup

Table 4.2: Ranger EX 757-7: moments of inertia

Moment of Inertia		[kg-m^2]
Roll	I_x	0.2529
Pitch	I_y	0.1965
Yaw	I_z	0.4758
Product	I_{xz}	0.01629

low Reynolds Numbers and determine the stability and control derivatives using the Vortex Lattice Method (VLM). This method assumes inviscid flow. The analysis was done using a velocity of 15 m/s and an altitude of 135m. The results for the stability and dynamic derivatives are shown in table 4.3 and the control derivatives are shown in table 4.4. Note that the lateral static stability derivative is unstable however the magnitude is small and can easily be recoverable by a pilot or a controller.

Table 4.3: Ranger EX 757-3: stability derivatives [/rad]

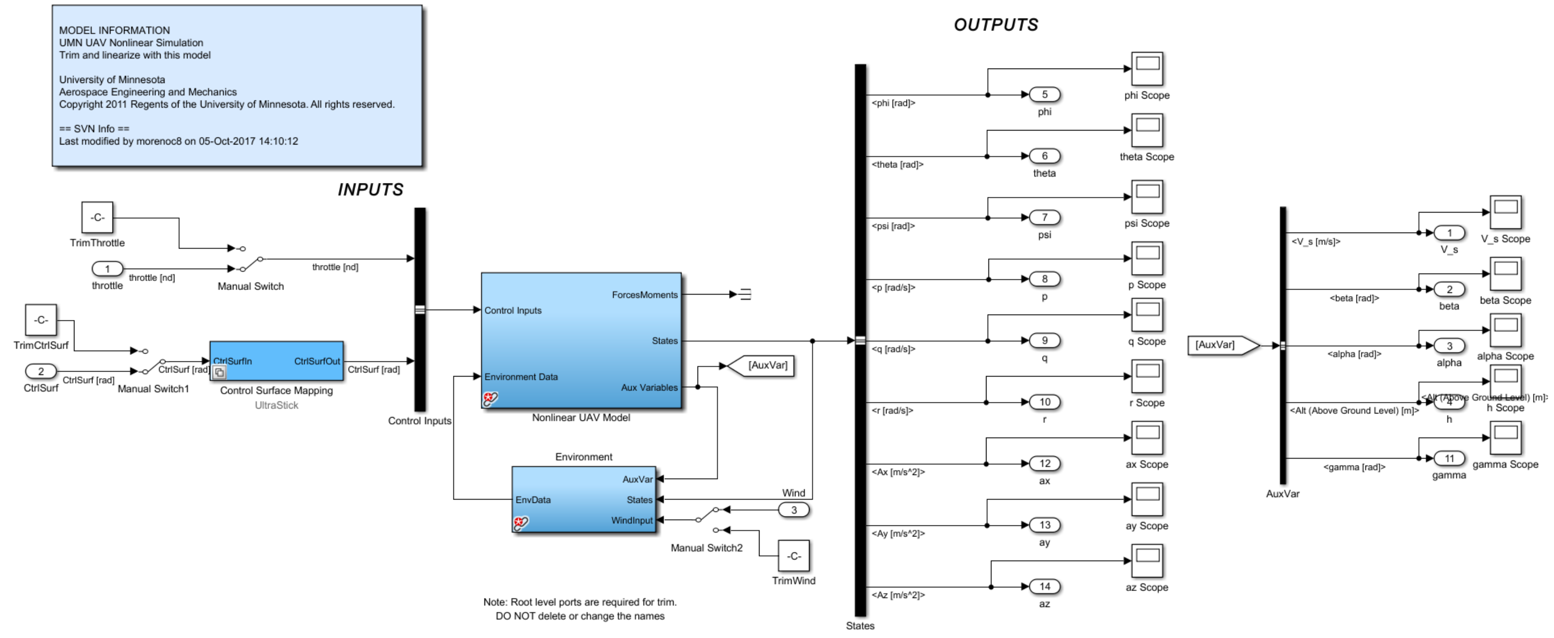
Longitudinal		Directional		Lateral	
$C_{D\alpha}$	0.007	C_{Y_β}	-0.1891	C_{Y_β}	-0.1891
$C_{L\alpha}$	5.4226	C_{Y_r}	0.1268	C_{Y_p}	-0.02292
$C_{m\alpha}$	-0.70812	C_{n_β}	0.05606	C_{l_β}	0.01369
C_{L_q}	7.5153	C_{n_p}	0.02643	C_{l_p}	-0.55118
C_{m_q}	-11.262	C_{n_r}	-0.03789	C_{l_r}	0.05847

Table 4.4: Ranger EX 757-3: control derivatives [/rad]

Elevators		Rudder		Ailerons		Flaps	
$C_{D_{\delta_e}}$	.00195	$C_{Y_{\delta_r}}$	0.18886	$C_{Y_{\delta_a}}$	0.0000	$C_{D_{\delta_f}}$	.0305
$C_{L_{\delta_e}}$	0.44587	$C_{L_{\delta_r}}$	0.0000	$C_{L_{\delta_a}}$	0.0000	$C_{L_{\delta_f}}$	1.0868
$C_{m_{\delta_e}}$	-1.1355	$C_{m_{\delta_r}}$	0.0000	$C_{m_{\delta_a}}$	0.0000	$C_{m_{\delta_f}}$	0.26711
$C_{l_{\delta_e}}$	0.0000	$C_{l_{\delta_r}}$	0.029	$C_{l_{\delta_a}}$	-0.06709	$C_{l_{\delta_f}}$	0.2723
$C_{n_{\delta_e}}$	0.0000	$C_{n_{\delta_r}}$	-0.03131	$C_{n_{\delta_a}}$	0.0651	$C_{n_{\delta_f}}$	0.0000

4.3 Rigid Body Simulation Environment

The simulation software package used in this research was developed by the Department of Aerospace Engineering and Mechanics at the University of Minnesota. This software is freely available at the research group website http://www.uav.aem.umn.edu. The nonlinear simulation implemented in Simulink represents a conventional fixed-wing rigid aircraft with 6 degrees-of-freedom (DOF). Here, several submodels constitute the numerical simulation, including airframe mass and inertia properties, propulsion dynamics, aerodynamics, servo-actuator dynamics, sensor noise properties, computational time delays, and environmental effects. In addition, a software-in-the-loop (SIL) and hardware-in-the-loop (HIL) are available for test and validation of flight controllers prior to flight testing. The software package also comes with examples of other UAS test beds used at University of Minnesota like mini-MUTT and the UltraStick 25e. Figure 4.4 shows the Simulink model and the main blocks that constitute the nonlinear simulation. The block with all the aerodynamic/mass and inertial properties of the UAV are being used in the block “Nonlinear UAV model” to calculate the forces, moments and states of the aircraft. The block “Environment” models the gravitational and atmospheric disturbances . The “Control surface mapping” block simply ensures the control surface commands have the right signs especially for the ailerons where a positive aileron command would mean a negative command to the left aileron and positive command to the right aileron. The inputs for the simulation include throttle, control surface commands and wind disturbances. The outputs of the simulation including the aircraft states are scoped on the right. Since the simulation



MODEL INFORMATION
UMN UAV Nonlinear Simulation
Trim and linearize with this model

University of Minnesota
Aerospace Engineering and Mechanics
Copyright 2011 Regents of the University of Minnesota. All rights reserved.

== SVN Info ==
Last modified by morenoc8 on 05-Oct-2017 14:10:12

Note: Root level ports are required for trim.
DO NOT delete or change the names

Figure 4.4: Aircraft nonlinear simulation environment (source: University of Minnesota)

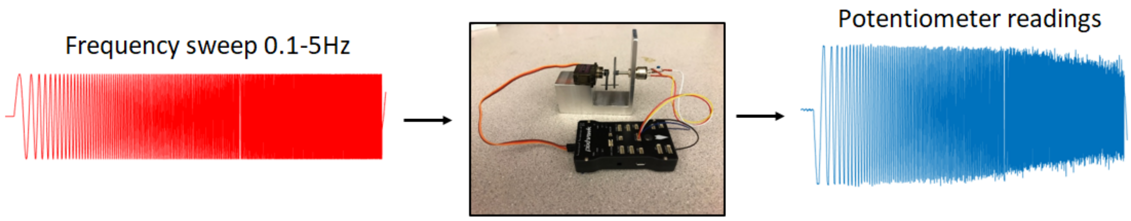


Figure 4.5: Actuator dynamics: system identification experimental setup

includes servo-actuator dynamics, an experiment to determine the actuator model of the servos used on the Ranger was conducted. The procedure and results are explained in the next section.

4.3.1 Actuator Model

The process of determining the actuator model is in fact system identification with a procedure similar to that one described in the next chapter. The procedure involves sending a known input to the actuator, measuring the output and determining a transfer function that best fits the model. The input that was sent to the actuator was a frequency sweep of 0.1-5Hz. The output in this case will be the deflection of the servo which was measured using a potentiometer. Figure 4.5 shows the experiment setup. As it can be seen in figure 4.5, the frequency sweep input is generated by the Pixhawk. Using the input and output data, a transfer function is estimated using the “tfestimate” function in MATLAB. This function uses a Welch’s average periodogram method where the estimated transfer function is the quotient of the cross power spectral density of the input and output, P_{xy} and the power spectral density of the input P_{xx}.

$$T_{xy} = \frac{P_{xy}}{P_{xx}} \tag{4.5}$$

The input and output are evaluated in different sections, windowed with a Hamming window and a specified % overlap. Once the transfer function has been estimated, an optimization tool called “fitmagfrd” is used to determine a transfer function model that best fits the experimental data. Using this function, the user can specify the order and relative order of degree of the transfer function desired for the fit. In this experiment it was found that a third order transfer function with a relative difference of 3 best fits the experimental data. The transfer function is shown in equation 4.6. The fit of the transfer function in the frequency domain is shown in the bode plot in figure 4.6. When the experimental data was tested again in the time domain with the model fit, about a 92% match was found in the data. Figure 4.7 shows the comparison. It was noted that with the identified transfer function model for the actuator, the bandwidth was determined to be about 3.5 Hz. This is too small for future control applications for flutter suppression however for the objective of this thesis, it is a sufficient actuator to conduct flight experiments.

$$G_a = \frac{26230}{s^3 + 80.13s^2 + 2139s + 27110} \tag{4.6}$$

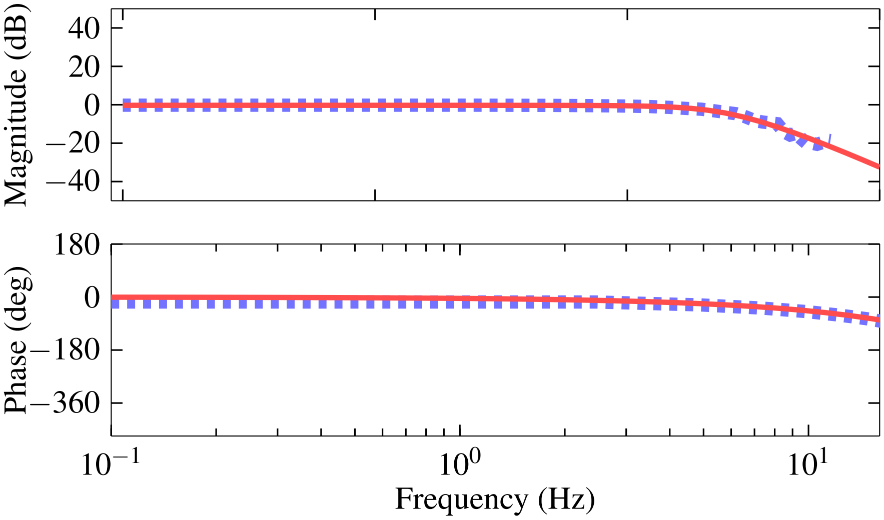


Figure 4.6: Frequency response comparison between actuator model(—) and experimental data(▪▪▪)

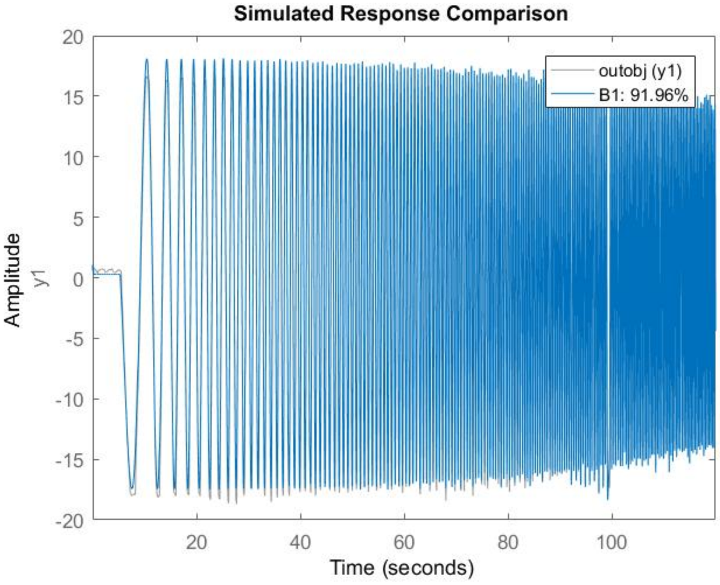


Figure 4.7: Time plot comparison of model fit with experimental data

5. Longitudinal Dynamics System Identification

In the previous chapter, the process of obtaining the baseline model of the Ranger EX 757-3 using software tools and ground experiments was discussed. This gave an initial estimate or impression of how the aircraft behaves. However for control design and future development of the UAV, the aircraft dynamics have to be well represented in the model. As presented in Chapter 2, system identification involves obtaining measurements for the inputs and outputs and a model that maps one to the other. Therefore the first step is to determine the longitudinal model that will be used. This will be described is section 5.1. Section 5.2 will describe the methodology used for system identification and section **??** will describe the flight test data and results.

5.1 Longitudinal Dynamics

The longitudinal dynamics are determined by starting with the aircraft rigid body equations of motion which include the force equations along the X and Z axis and the pitching moment equation.

$$m\left(\dot{u} - rv + qw\right) = X \tag{5.1}$$

$$m\left(\dot{w} - qu + pv\right) = Z \tag{5.2}$$

$$I_y\dot{q} - \left(I_x - I_z\right)pr + I_{xz}\left(p^2 - r^2\right) = M \tag{5.3}$$

These equations are then linearized about a particular trim airspeed and altitude using small disturbance theory and the linear model is then represented using the pertubed states. The longitudinal dynamics are then represented using the states $[u, w, q, \theta]^T$. The input that is assumed to only affect the longitudinal dynamics is the elevator deflection δ_e. Therefore the longitudinal model can be represented with the following state-space system:

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_w & X_q - W_e & -g\cos\theta_e \\ Z_u & Z_w & Z_q + U_e & -g\sin\theta_e \\ M_u & M_w & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} X_{\delta_e} \\ Z_{\delta_e} \\ M_{\delta_e} \\ 0 \end{bmatrix} \delta_e \qquad (5.4)$$

The terms U_e, W_e and θ_e represent the trim values of the velocities and pitch angle. The rest of the terms represent the dimensional aerodynamic derivatives that need to be identified. The coefficients of the states represent the stability derivatives and the coefficients of the input represent the control derivatives. Each of these dimensional derivatives can be written in terms of the non-dimensional derivatives in the following way:

$$\begin{array}{llll} X_u = \dfrac{C_{X_u} QS}{mU_e} & X_w = \dfrac{C_{X_\alpha} QS}{mU_e} & X_q = \dfrac{C_{X_{\hat{q}}} QS\bar{c}}{2mU_e} & X_{\delta_e} = \dfrac{C_{X_{\delta_e}} QS}{m} \\ Z_u = \dfrac{C_{Z_u} QS}{mU_e} & Z_w = \dfrac{C_{Z_\alpha} QS}{mU_e} & Z_q = \dfrac{C_{Z_{\hat{q}}} QS\bar{c}}{2mU_e} & Z_{\delta_e} = \dfrac{C_{Z_{\delta_e}} QS}{m} \\ M_u = \dfrac{C_{M_u} QS\bar{c}}{I_y U_e} & M_w = \dfrac{C_{M_\alpha} QS\bar{c}}{I_y U_e} & M_q = \dfrac{C_{M_{\hat{q}}} QS\bar{c}^2}{2I_y U_e} & M_{\delta_e} = \dfrac{C_{M_{\delta_e}} QS\bar{c}}{I_y} \end{array} \qquad (5.5)$$

The coefficients are the C's where the subscript represents the stability or control derivative. As it can be seen in equation 5.5, the non-dimensional dynamic derivatives for the pitch rate are with respect to $\hat{q}$. This is the non-dimensional term for the pitch rate where $\hat{q} = \frac{q\bar{c}}{2U_e}$.

The terms Q and S are the dynamic pressure and reference area of the wing. The linearized acceleration measurements are:

$$a_x = \dot{u} + qW_e + g\cos\theta_e\theta + g\sin\theta_e \tag{5.6}$$

$$a_z = \dot{w} - qU_e + g\sin\theta_e\theta - g\cos\theta_e \tag{5.7}$$

This longitudinal model provides the two dynamic modes: phugoid and short period. The phugoid mode is a slow, lightly damped mode that is predominantly in the responses of u, θ and a_x. The short period mode is a faster, more damped mode and is predominantly in the responses of w, q and a_z. For manned aircraft, performing flight experiments to identify both modes is relatively easy compared to UAV applications. The flight experiment to identify the phugoid mode requires long durations which is not applicable to UAVs due to the limited range of visual line of sight. However since this is a slow mode, it is not significant to control applications and hence can be ignored. It is a mode easily recoverable by a pilot or a feedback control. And so the longitudinal system identification will focus on identifying the short period mode which can be decoupled from the longitudinal model shown earlier by using the state vector $[w, q]^T$ which is shown in equation 5.8. The outputs based on the sensors available are the pitch rate (q) and the vertical acceleration (a_z) and are related to the states shown in equation 5.9.

$$\begin{bmatrix} \dot{w} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} Z_w & Z_q + U_e \\ M_w & M_q \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} + \begin{bmatrix} Z_{\delta_e} \\ M_{\delta_e} \end{bmatrix} \delta_e \tag{5.8}$$

$$\begin{bmatrix} q \\ a_z \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ Z_w & Z_q - U_e \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ Z_{\delta_e} \end{bmatrix} \delta_e \tag{5.9}$$

5.2 Methodology

The general method used for system identification in this thesis is creating frequency responses for the experimental input and output data and finding a model that fits these responses. The reason of performing the identification in the frequency domain is because it easier to see the system behavior about different frequencies. In addition, it helps in focusing the identification about a specific frequency range that accurately represents the dynamics. The system identification process from model postulation to validation is shown in the flowchart in figure 5.1 (Klein & Morelli, 2006). Since the main objective of this chapter is to identify the longitudinal dynamics of the Ranger EX 757-3, the model pos-

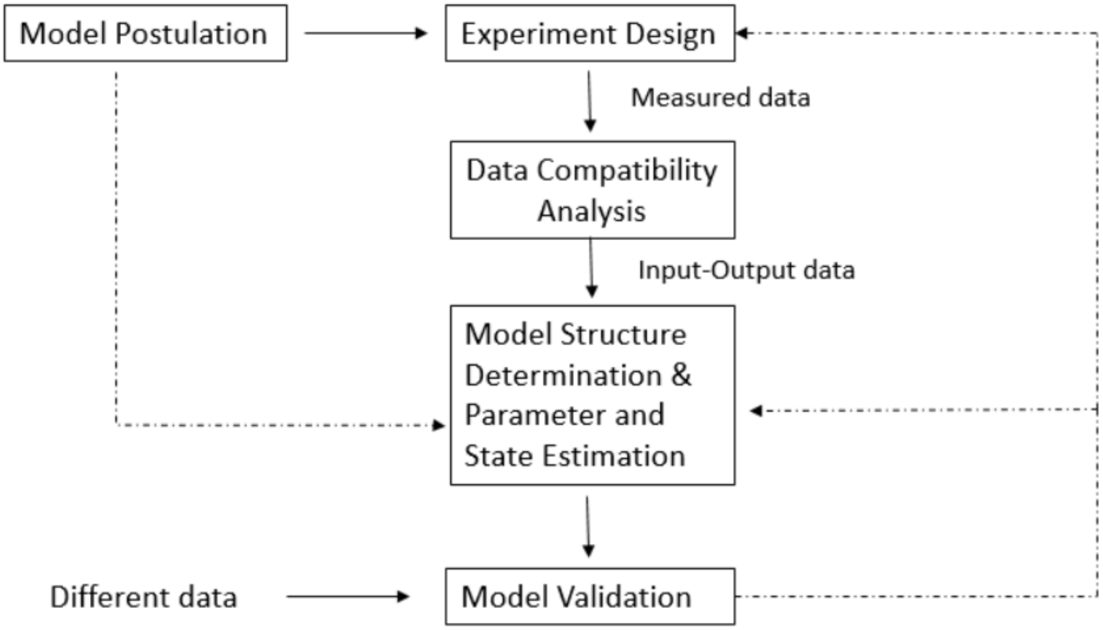


Figure 5.1: System identification flowchart

tulation will involve a priori knowledge of the longitudinal dynamics of the aircraft. This will then dictate how the flight experiment be designed. However for UAVs, there are limitations to how the flight experiment can be conducted. And so this will in turn modify the model postulation. After performing the flight experiment, the measured data is analysed to check compatibility in the kinematic relationships where after are organized and fed through the algorithms of the different tools for parameter estimation. Once the model parameters have been identified a different set of data for example a doublet implemented in-flight is tested with the identified model and compared to the flight data to see how well the model captures the dynamics. If the validation is not good enough, parameters like the frequency range are varied until a good model is obtained. If a good model is unobtainable, this might mean the flight data didn't capture the dynamics required and hence might result in a repeat of the flight experiment.

As mentioned in Chapter 2, three main tools were chosen for system identification in this research: the System Identification toolbox in MATLAB, SIDPAC (System Identification Programs for AirCraft) and PIL (Parameter Identification Library). SIDPAC can be obtained by purchasing the book "Aircraft System Identification Theory and Practice" by Klein and Morelli while PIL can be obtained freely on the MATLAB File Exchange at the site " https://www.mathworks.com/matlabcentral/fileexchange/1013-pil".

5.2.1 MATLAB System Identification Toolbox

The first method involved writing a script in MATLAB that uses the function "greyest" to estimate a linear grey-box model. The way this is implemented is by defining the linear equations shown in equations 5.8 and 5.9. These equations are in terms of the dimensional coefficients. So to determine the non-dimensional coefficients, their relationship to the dimensional coefficients were included in the algorithm. The non-dimensional coefficients were set to be the model parameters that were to be identified by the algorithm and the initial guesses for these values were from the baseline model determined in Chapter 4. The algorithm allows a range of values be specified for each model parameter, and so values that reflect the physical properties of aircraft are specified. The model parameters can also be specified whether they should be fixed to the initial values or be free parameters to be identified. Once the model equations are defined, transfer functions of the experimental data, using the function “tfestimate”, are determined and fed into the algorithm to be used for the model fitting. The advantage of this method is that the identification is done using only the transfer functions that are given to the algorithm. This avoids the need to estimate parameters that are not available like angle-of-attack. And so the transfer functions that are provided to the algorithm are the pitch rate to elevator deflection ($\frac{q}{\delta_e}$) and vertical acceleration to elevator deflection ($\frac{a_z}{\delta_e}$). The code used can be viewed in Appendix A.

5.2.2 SIDPAC and PIL

SIDPAC and PIL have several system identification techniques one can use depending on the application. The technique, as highlighted in Chapter 2, used in this research is the Least Squares Regression in the Frequency domain. This means the input and output data are transformed from the time domain to the frequency domain and a least squares regression is then done to determine the model parameters. The main difference between the two tools is the method they use to transform the flight data from the time domain to the frequency domain. SIDPAC uses the chirp z-transform while PIL does a real time discrete Fourier transform (DFT). The chirp z-transform is a generalization of the DFT such that it samples the z plane along a logarithmic spiral contour that correspond to straight lines in the s-plane. The DFT samples the z-plane at uniformly spaced points along the unit circle (Shilling, 1972). The chirp z-transform is defined as:

$$X_k = \sum_{n=0}^{N-1} x(n) z_k^{-n} \tag{5.10}$$

$$z_k = A \cdot W^{-k}, k = 0, 1, ..., M-1$$

where A is the complex starting point, W the complex ratio between points and M the number of points to calculate (Shilling, 1972). The DFT is defined as:

$$X_k = \sum_{n=0}^{N-1} x(n) \cdot e^{-\frac{2\pi i}{N}kn} \tag{5.11}$$

The model equation used in SIDPAC comes from the short period model and acceleration measurement equation mentioned earlier. To determine the force coefficients, the term $\dot{w}$ in equation 5.8 is substituted into equation 5.7 to give the equation:

$$a_z + g\cos\theta_e = Z_w w + Z_q q + Z_{\delta_e}\delta_e \tag{5.12}$$

Here the term $g\sin\theta_e\theta$ is small compared to the other terms hence neglected. The vertical velocity w can be expressed in terms of angle-of-attack α using small angle approximation ($\alpha \approx \frac{w}{U_e}$) hence the model equation for the vertical force can expressed as:

$$a_z + g\cos\theta_e = Z_\alpha \alpha + Z_q q + Z_{\delta_e}\delta_e \tag{5.13}$$

where $Z_\alpha = U_e Z_w$. Using the same small angle approximation, the moment equation can be expressed as:

$$\dot{q} = M_\alpha \alpha + M_q q + M_{\delta_e} \delta_e \tag{5.14}$$

where $M_\alpha = U_e M_w$. Once the dimensional coefficients are identified, the non-dimensional coefficients are calculated which are to be used for the nonlinear simulation. The model equations in PIL are essentially the same as the one for SIDPAC but due to the setup of the tool, they were expressed in terms of the non-dimensional coefficients. This means the model equations can be written in the form:

$$C_Z = C_{Z_\alpha} \alpha + C_{Z_{\hat{q}}} \frac{q\bar{c}}{2U_e} + C_{Z_{\delta_e}} \delta_e \tag{5.15}$$

$$C_m = C_{m_\alpha} \alpha + C_{m_{\hat{q}}} \frac{q\bar{c}}{2U_e} + C_{m_{\delta_e}} \delta_e \tag{5.16}$$

As seen from the model equations used in SIDPAC and PIL, the input measurements needed are angle-of-attack, pitch rate and elevator deflection. The output measurements needed are the vertical acceleration and pitch acceleration. The pitch rate and vertical acceleration can be retrieved from the accelerometer and gyro readings and the elevator deflection can be estimated from the command that was sent from the Pixhawk. The angle-of-attack and pitch acceleration are not available hence need to be estimated.

The angle-of-attack was estimated using equations derived from the acceleration equations assuming small angle approximation during trim condition (Morelli, 2012) and are as follows:

$$\dot{\beta} \approx p\alpha - r + \frac{1}{U_e}(g\cos\theta\sin\phi + a_y) \tag{5.17a}$$

$$\dot{\alpha} \approx q - p\beta + \frac{1}{U_e}(g\cos\theta\cos\theta + a_z) \tag{5.17b}$$

After estimating α, it is passed by a third order high pass Butterworth filter to remove bias and drift errors. This method was validated using simulated data to determine the error of this method. This means that the α values produced from simulation were compared to α values that were estimated using simulated values for the parameters seen in equation 5.17. Figure 5.2 shows the difference between the plots of the simulated α and the estimated α. Figure 5.3 and 5.4 show the error between the simulated and estimated values in the time and frequency domain respectively. It can be seen that initially the error is big due to the loss of the low frequency content after passing through the high pass filter on the estimated α. However the error then reduces to acceptable levels after a few seconds. Looking at the error in the frequency domain, it helps determine in what frequency range that model identification needs to be done to yield accurate results. In this case it can be seen that is within 1-5Hz. The model to determine the pitching moment derivatives is:

$$\dot{q} = M_\alpha\alpha + M_q q + M_{\delta_e}\delta_e \tag{5.18}$$

To determine the pitch acceleration $\dot{q}$, two possible methods can be used. The first method is to perform numerical differentiation on the data collected on q in the time domain and feed in the results together with the other inputs through the least squares in the frequency

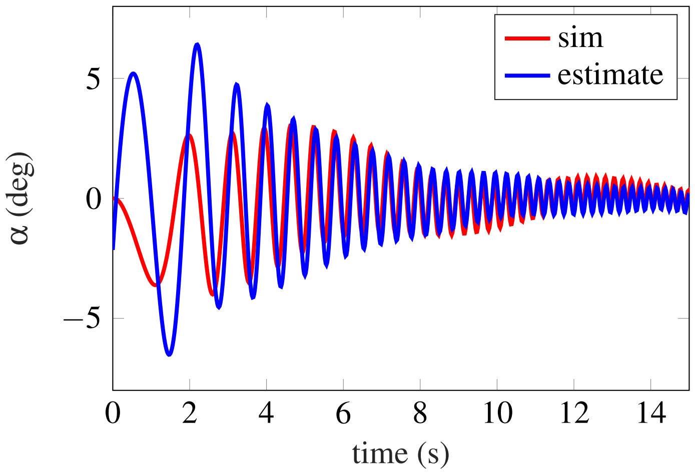


Figure 5.2: Comparison of Simulated and Estimated Angle-of-Attack

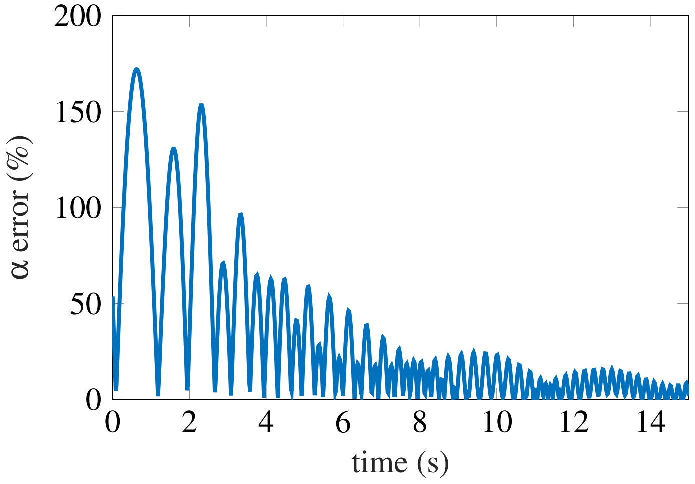


Figure 5.3: Error of Estimated Angle-of-Attack

domain algorithm. The second method is to differentiate q in the frequency domain by simply multiply $j\omega$, where $j = \sqrt{-1}$ and ω being the frequency range, to the transformed values of q in the frequency domain. From analysis and validation using simulation data, better results were found by performing the numerical differentiation in the time domain.

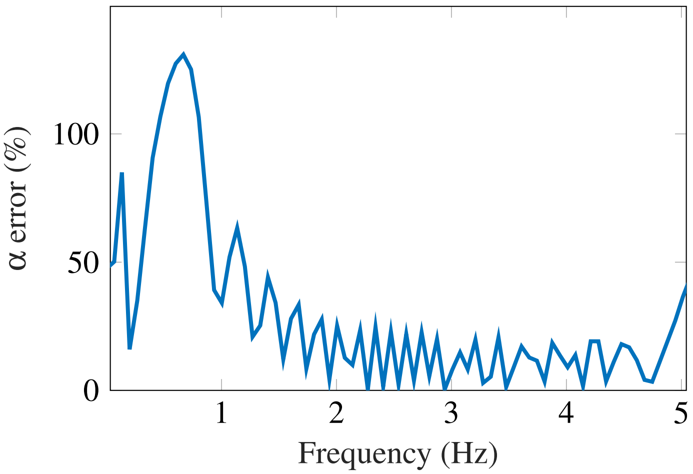


Figure 5.4: Error of Estimated Angle-of-Attack in Frequency Domain

A plot to compare the $\dot{q}$ from the simulated data to the estimated $\dot{q}$ from numerical differentiation is shown in figure 5.5. Figure 5.6 and 5.7 show the error of the estimated $\dot{q}$ in the time and frequency domain respectively. As seen from figure 5.6, the error increases with time due to the limitations of the numerical differentiation at higher frequencies. The rapid changes at these frequencies cannot be accurately differentiated with a fixed step size. However the errors still remain within satisfactory levels. The error in the frequency domain shows that a good frequency range to perform the model fitting should be around 0.5 to 5Hz.

5.3 Validation of System Identification tools using Simulated Data

A verification of the estimation methods and system identification tools was done by using simulated data. The simulated data was produced using known values for the non-

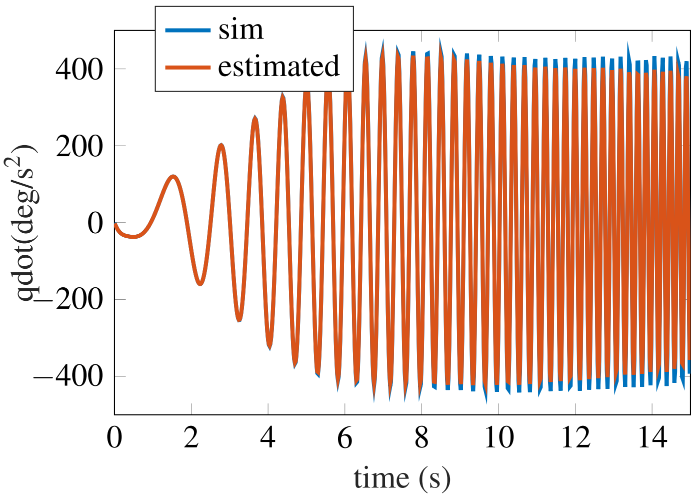


Figure 5.5: Comparison of the simulated and estimated pitch acceleration

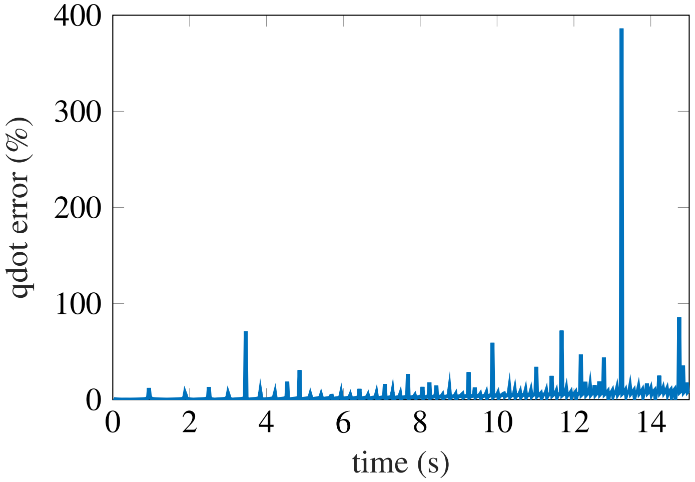


Figure 5.6: Error of estimated pitch acceleration in time domain

dimensional coefficients and a comparison was done with the outputs of the identification tools. The simulated parameters that were fed into each tool were manipulated and restructured the same way the flight data was going to be fed into the algorithms. Table 5.1 shows

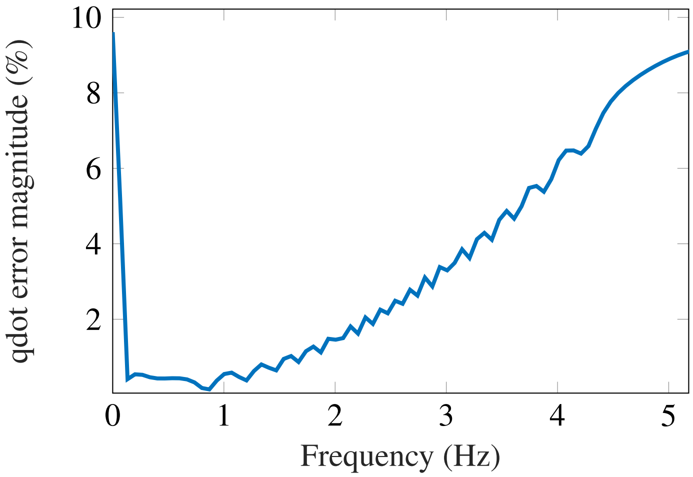


Figure 5.7: Error of estimated pitch acceleration in frequency domain

the results and errors from each tool. As seen from the results, the algorithms of each tool have limitations and it comes to show that if with perfect simulated data there are slight errors in the results, there should be errors expected when working with flight data which contains noise and biases. Having determined appropriate model equations,outputs and inputs, a flight experiment can now be designed to collect data for system identification.

5.4 Flight Test Data and Results

The main objective of the flight experiment is to trim the Ranger EX 757-3 to a particular velocity and altitude and send a frequency sweep that is intended to excite the short period mode to the elevator and measure the responses. For a UAV with limited range of sight, the pilot can fly the Ranger in circuit patterns that allows a 20-30 second time window to conduct the experiment (Dorobantu, 2013). The intention of conducting the

Table 5.1: Identified non-dimensional coefficients with simulated data

Simulation		Greyest		SIDPAC		PIL	
Coefficient	Value	Value	Error(%)	Value	Error (%)	Value	Error(%)
C_{L_α}	5.5630	5.8111	4.4596	5.3983	2.9612	5.5398	0.4178
$C_{L_{\hat{q}}}$	8.6925	8.4305	0.376	10.1583	16.8635	8.5425	0.2152
$C_{L_{\delta_e}}$	0.5675	0.4773	15.9005	0.6074	7.0299	0.5387	5.0737
C_{m_α}	-1.3797	-1.1974	13.2101	-1.3855	0.4247	-1.4832	7.5025
$C_{m_{\hat{q}}}$	-13.5724	-16.4189	20.9725	-12.6289	6.9521	-9.7125	28.4394
$C_{m_{\delta_e}}$	-1.5715	-1.6530	5.183	-1.4901	5.1818	-1.4485	7.8308

Table 5.2: Short period poles using simulated data

Tool	Pole	Damping ratio	Frequency (rad/s)
Baseline model	$-7.19 \pm 8.01j$	0.668	11.08
Greyest	$-8.03 \pm 7.51j$	0.73	11.00
SIDPAC	$-6.85 \pm 7.93j$	0.654	10.5
PIL	$-6.27 \pm 8.18j$	0.608	10.3

experiment in trim condition is to make the Ranger to behave as “linearly” as possible. In this experiment, the Ranger was trimmed at a velocity of 12 m/s and an altitude of 150 m.

Once the Ranger was trimmed, the pilot engages the flight computer to send a computer generated frequency sweep that goes from 0.1 Hz to 5 Hz and varies linearly in 10 seconds. The frequency sweep follows the following function:

$$y(t) = \sin(2\pi(f_0 t + \frac{f_1 - f_0}{2T} t^2)) \tag{5.19}$$

where f_0 is the initial frequency, f_1 is the final frequency and T is the time it takes to sweep from f_0 to f_1. Figure 5.8 shows the sweep signal that is generated with the function. When this signal is sent to the elevator, the aileron, rudder and throttle command are fixed to the trim values to ensure decoupling in the aircraft motion. The pilot will not have command authority on these control surfaces except for the elevator where the pilot commands get augmented to the computer generated signal. This is because the frequency sweep input in the elevator can trip the Ranger out of its trim condition and hence the pilot can adjust the

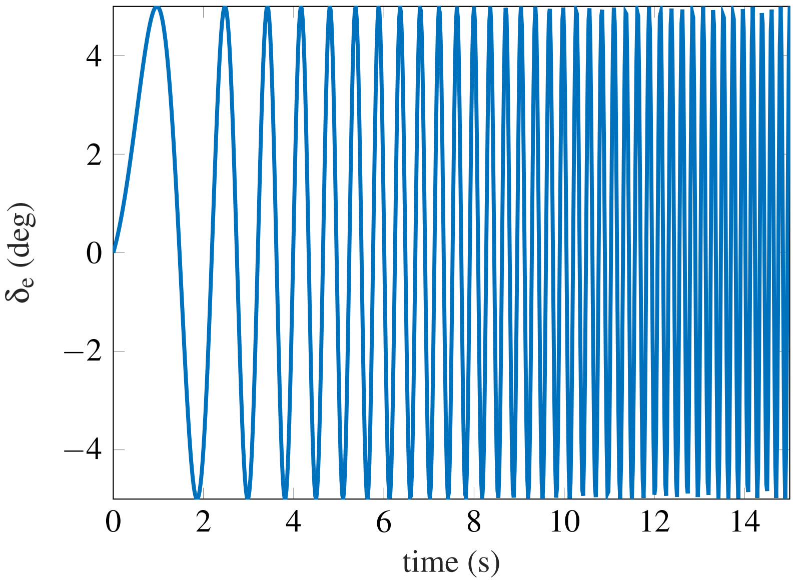


Figure 5.8: Simulated frequency sweep signal generated to elevator

elevator to bring it back to trim. Once the frequency sweep is engaged, the responses were measured using the accelerometers,gyros, altitude and airspeed sensor. After successfully conducting the experiment, the experiment was repeated at the same conditions however with a pilot generated doublet that will then be used for the validation process of the system identification.

The data that was gathered from the flight data of the inputs and outputs to the model equations are shown in figure 5.9. The augmentation of the pilot inputs to the computer generated sweep can be clearly seen in the graph. The α shown here is estimated.

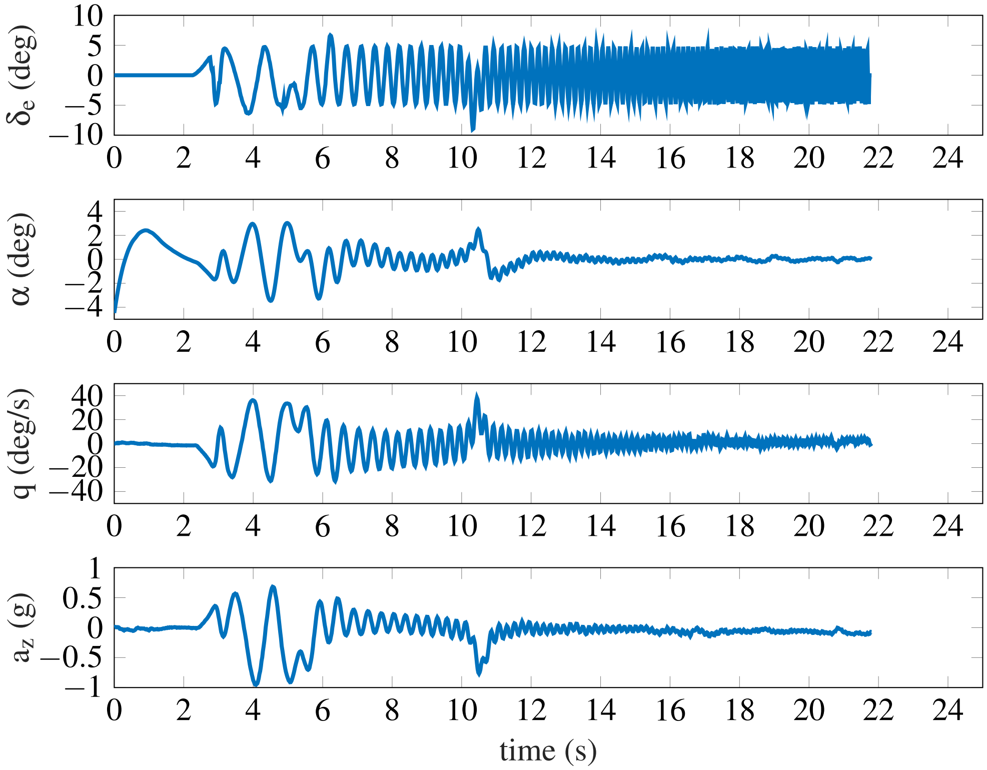


Figure 5.9: Flight test results of longitudinal time response to frequency sweep

As mentioned before, three software tools were used for system identification. Each tool had the same flight data used for analysis. The model equations varied slightly for each tool. For SIDPAC, the model equations are exactly equation 5.13 and 5.18 to determine the Z and M coefficients. PIL uses the same equations as SIDPAC but expressed in terms of the non-dimensional coefficients. The setup of the PIL Simulink file made it easier to express it that way. A valid assumption that is used to determine the lift coefficients is that for small angles, the force in the z direction is predominantly the lift. Therefore $Z \approx -L$ which means all the Z derivatives are the negatives of the lift derivatives. For the “greyest” function in MATLAB, the model equations were the ones found in the short period model. The only difference is that the coefficient C_{Z_w} does not only consider the lift component, C_{L_α}, but also has the drag component C_{D1}. This can seen in appendix A. Table 5.3 displays the non-dimensional coefficients identified by each tool. The greyest function was the first

Table 5.3: Ranger EX 757-3: identified longitudinal stability derivatives [/rad]

Greyest		SIDPAC		PIL	
C_{L_α}	5.415	C_{L_α}	5.8994	C_{L_α}	5.6074
$C_{L_{\hat{q}}}$	5.000	$C_{L_{\hat{q}}}$	15.5984	$C_{L_{\hat{q}}}$	8.5896
$C_{L_{\delta_e}}$	0.1178	$C_{L_{\delta_e}}$	0.8643	$C_{L_{\delta_e}}$	0.6859
C_{m_α}	-0.1000	C_{m_α}	-0.3994	C_{m_α}	-0.1092
$C_{m_{\hat{q}}}$	-53.22	$C_{m_{\hat{q}}}$	-5.3471	$C_{m_{\hat{q}}}$	-9.9047
$C_{m_{\delta_e}}$	-2.328	$C_{m_{\delta_e}}$	-1.0126	$C_{m_{\delta_e}}$	-1.0448

tool that was used during the research. For determining the transfer function estimate of the flight data, a Hamming window of 109 and 108 overlap was used in the MATLAB function “tfestimate”. Figures 5.10 and 5.11 show the comparison in the frequency response between the identified model and experimental data for the pitch rate and vertical acceleration respectively. Even though there is a good fit in the frequency domain, as seen from the results in table 5.3, the magnitude of $C_{m_{\hat{q}}}$ is higher than most general aviation. It is noted that a lower magnitude of $C_{m_{\hat{q}}}$ is found if the inertia is set as a free parameter. When this is done, the $C_{m_{\hat{q}}}$ value lowers to -11.5 however the $C_{L_{\hat{q}}}$ value becomes slightly lower than C_{L_α}. Again, this is unconventional for general aviation. However, the coefficients identified from SIDPAC and PIL are within similar ranges to each other and is the range that is expected for conventional aircraft. Using these non-dimensional coefficients and plugging them in

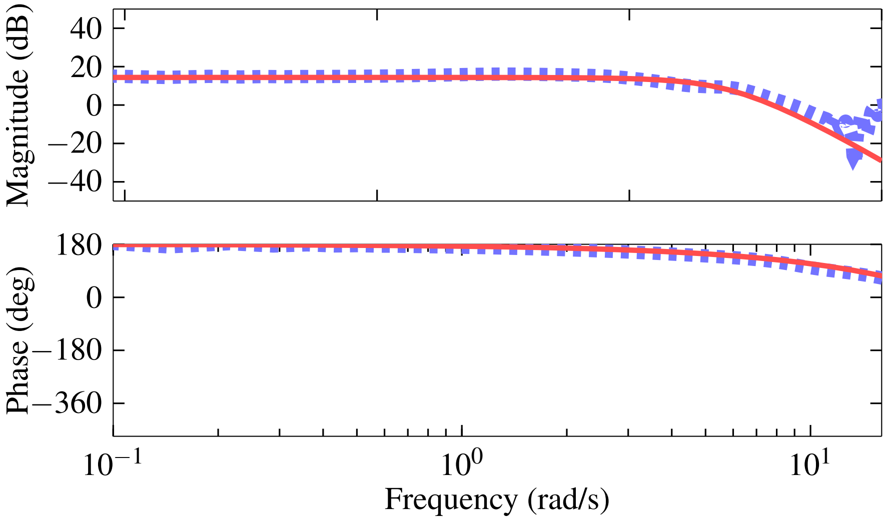


Figure 5.10: Pitch rate frequency response comparison between grey-box identified model(—) and experimental data(▮▮▮)

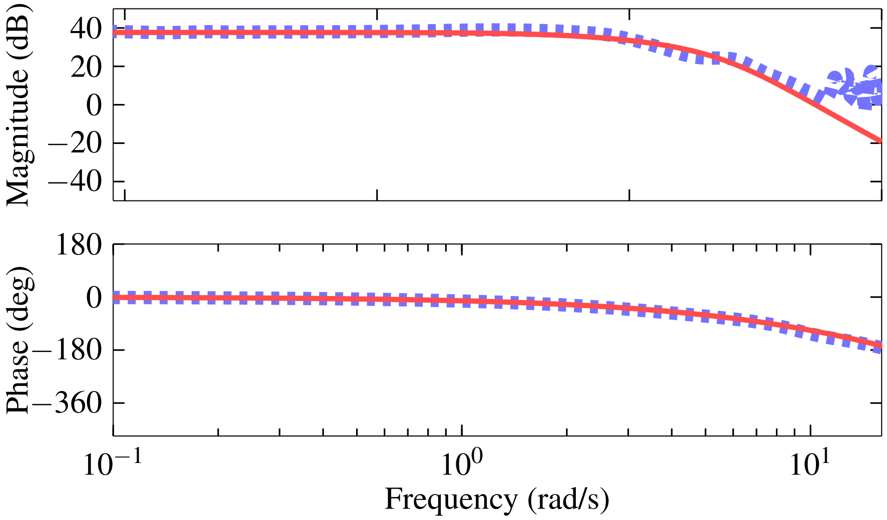


Figure 5.11: Vertical accelration frequency response comparison between grey-box identified model(—) and experimental data(▪▪▪)

equation 5.5 to get the dimensional coefficients, the state-space of the short period model can be made and the short period poles be determined. Table 5.4 show the poles, damping and frequency of the short period mode for the three identified models. The greyest estimation resulted in a short period mode that is over-damped which is again unconventional.

Table 5.4: Short period poles of identified models

Tool	Pole	Damping ratio	Frequency (rad/s)
Greyest	$-9.04, -27.4$	1.1581	9.04, 27.4
SIDPAC	$-5.0 \pm 2.59j$	0.891	5.71
PIL	$-5.87 \pm 1.36j$	0.974	6.03

The SIDPAC and PIL results however, are under-damped as expected. The identified short period mode for PIL is faster and more damped than the identified mode from SIDPAC. These models were first validated using the sweep data that was used for the identification. Figure 5.12 shows the comparison between the flight data and the three identified models for the pitch rate and vertical acceleration. As seen from the figure, the identified models from all three tools capture the dynamics pretty well. The "compare" function in MATLAB, which uses a normalized root mean square algorithm, was used to determine the % match of each identified model with the flight data. The greyest model provided the greatest % match with 68.91% for the pitch rate and 67.49% for the vertical acceleration. The

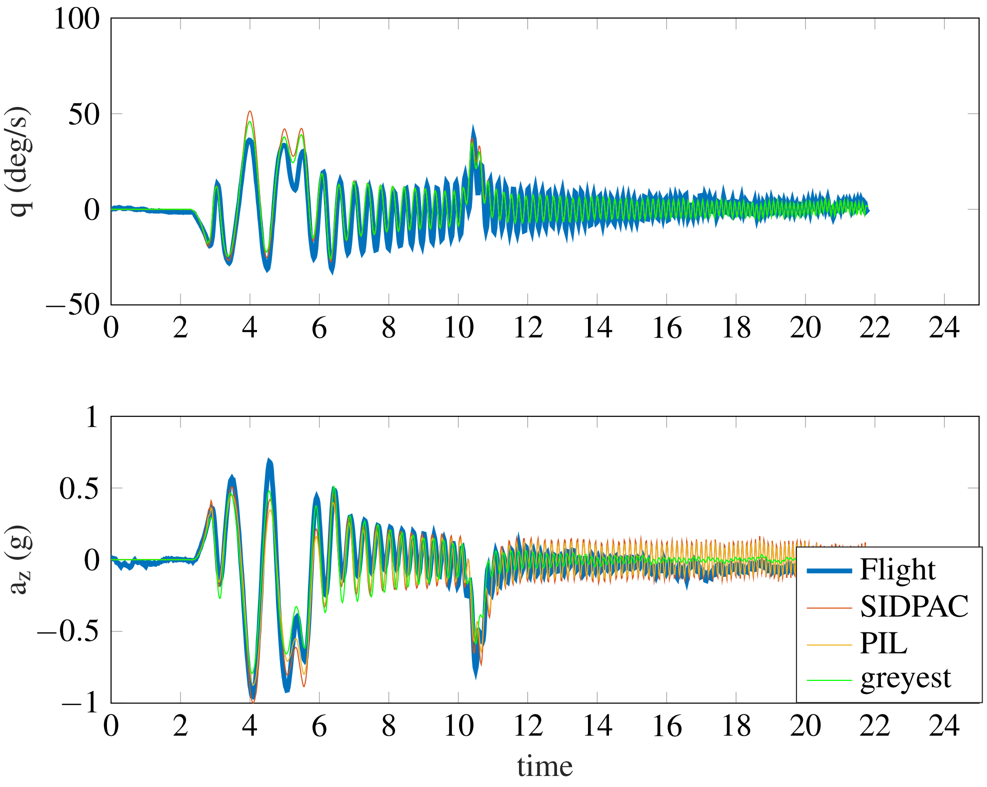


Figure 5.12: Sweep validation of the identified longitudinal models

SIDPAC results gave a 34.1% match for the pitch rate and a 42.1% match for the vertical acceleration. The PIL results gave a 35.34% match and a 47.34% match for the vertical acceleration. Therefore even with the irregularities in the greyest results, it provides a better match for the sweep data. Part of system identification process however, and to get a better comparison, is to validate the identified model using a different set of flight data. Therefore using a pilot induced doublet during the flight test, the identified models are compared to the flight data and are shown in figure 5.13. Using the compare function for the doublet responses, the greyest model had a 55.55% match for the pitch rate and a 61.44% match with the vertical acceleration. SIDPAC had a match of 50.84% for the pitch rate and a

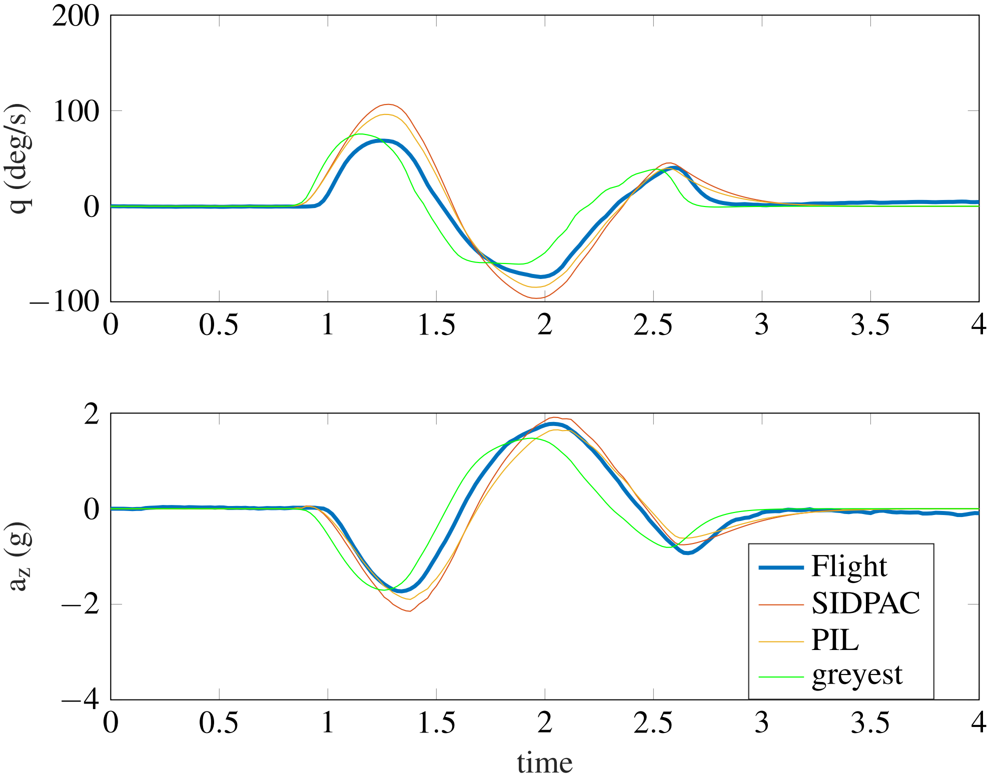


Figure 5.13: Doublet validation of the identified longitudinal models

match of 68.13% for the vertical acceleration. PIL had the highest match compared to both tools with a 65% match for the pitch rate and a 70% match for the vertical acceleration. As it can be seen from figure 5.13, there are trade-offs from the different models. The greyest model seems to have a phase difference with the flight data but has a magnitude closer to it. The other two models seem to be in phase with the flight data but overshoots the magnitude especially in the first peak. But judging by the best match to the flight data, PIL seems to be the best model that represents the longitudinal dynamics. And not only does it have the best match, the stability and control derivatives are within a reasonable range for general aviation, unlike the greyest model. Therefore it is safe to choose the PIL model as the one that can be used for future control applications of the Ranger EX 757-3.

A final validation done was to use the identified non-dimensional coefficients of PIL in the aircraft nonlinear simulation presented in the previous chapter and compare the short period model that is produced by the linearization algorithms to the ones presented earlier. The short period model that is produced by the nonlinear simulation is shown in equation 5.20. The modes were found to be very close to the one identified earlier and the compare plots showed the same percent matches of 65% for the pitch rate and 70% for the vertical acceleration. Table 5.5 and figure 5.14 show the modes and the responses from the doublet input of the short period model produced from the nonlinear simulation.

$$\begin{bmatrix} \dot{w} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -7.485 & 20.68 \\ -0.4007 & -4.25 \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} + \begin{bmatrix} -10.89 \\ -46.78 \end{bmatrix} \delta_e \tag{5.20}$$

Table 5.5: Short period poles of identified longitudinal model from nonlinear simulation

Pole	Damping ratio	Frequency (rad/s)
$-5.87 \pm 1.29j$	0.977	6.01

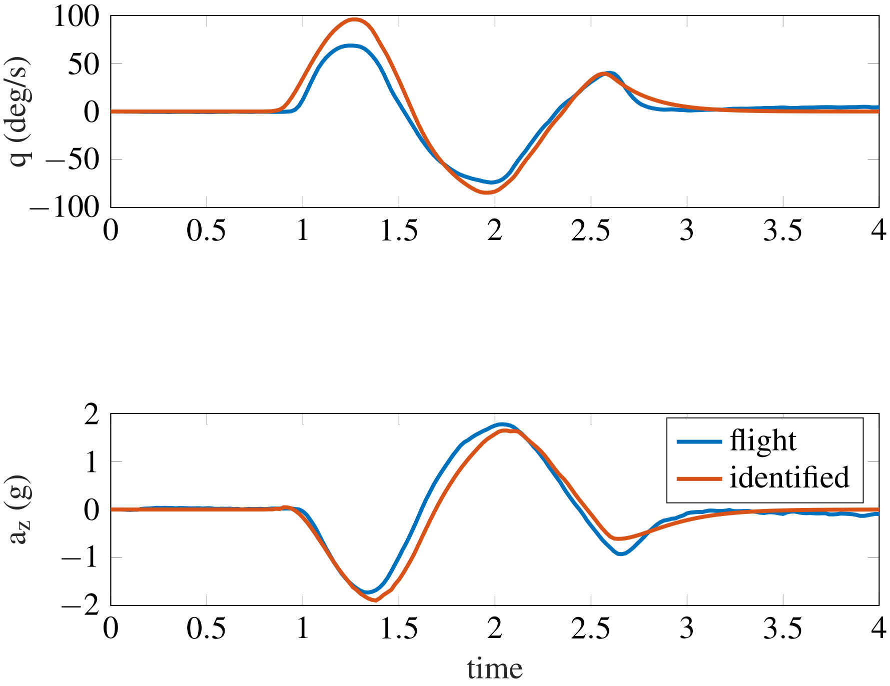


Figure 5.14: Doublet validation of the short period model from nonlinear simulation

6. Conclusion and Future Work

6.1 Conclusion

In the chapter of system identification, it was found that the results of PIL gave an accurate model and the best, compared to the other two tools, to represent the longitudinal dynamics of the Ranger EX 757-3. A 65% match between the identified model and doublet flight data for the pitch rate and 70% match for the vertical acceleration are satisfactory and suitable for future control applications. The experimental values determined for the inertia, mass and CG helped with the system identification process as they reduced the number of free parameters in the model equations and were necessary to determine the non-dimensional coefficients. The identified non-dimensional coefficients in turn has provided with a longitudinal model that can now be used to calibrate the nonlinear simulation which will provide a platform to test on the ground before taking the Ranger up in flight.

6.2 Future Work

The longitudinal model is just one piece of the model needed to represent the aircraft dynamics of the Ranger EX 757-3. The remaining models that need to be identified are the lateral and directional dynamics. This is a much tougher process as there is coupling between the two configurations. Therefore the next important task for this research is to identify the lateral and directional models which will complete the rigid body model of the Ranger EX 757-3. After this, structural analysis needs to be conducted and a flexible model be identified so it can be integrated with the rigid body dynamics and create a suitable model for flutter testing. During the structural testing, it will be determined whether the Ranger is a suitable platform for flutter testing, and if not, what modifications need to be done.

REFERENCES

Arifianto, O. (2013). *A low-cost unmanned aerial vehicle research platform: Development, modeling, and advanced control implementation* (Unpublished doctoral dissertation). Department of Aerospace Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA.

Calzada, R. (2002). *Firebee drone, dast.* Retrieved 2017-11-15, from `https://www.nasa.gov/centers/dryden/multimedia/imagegallery/DAST/DAST_proj_desc.html`

Chiappa, C., Magni, J.-F., Gorrec, Y. L., Kubica, F., & Doll, C. (1998). Flexible structure control by modal multi-model approach: Application to a flexible aircraft. In *21st congress of international council of the aeronautical sciences.* Melbourne, Australia.

Cole, J. (2014). *Volantex rc ranger ex review.* Retrieved from `https://www.rcgroups.com/forums/showthread.php?2187587-Volantex-RC-Ranger-EX-Review`

Conner, M. (2014). *X-56a multi-utility technology testbed.* Retrieved from `https://www.nasa.gov/centers/armstrong/research/X-56/index.html`

Cook, M. V. (2007). *Flight dynamics principles*. Oxford, UK: Elsevier Ltd.

Doggett, R. V. J., & Townsend, J. L. (1976). Flutter suppression by active control and its benefits. In *Proceedings of the supersonic cruise aircraft research conference.* Hampton, VA.

Dorobantu, A. (2013). *Test platforms for model-based flight research* (Unpublished doctoral dissertation). Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN.

Klein, V., & Morelli, E. (2006). *Aircraft system identification theory and practice*. Reston, Virginia: American Institute of Aeronautics and Astronautics, Inc.

Kuznicki, S., & Lee, D. (2016). Pixhawk pilot support package user guide [Computer software manual].

Livne, E. (2014). Active flutter suppression-plans for assessing the technology's state of the art. In *Joint advanced materials and structures center of excellence technical review.* Seattle, WA.

Ljung, L. (1999). *System identification: Theory for the user*. Linkoping University,Sweden: Prentice Hall PTR.

Maynard, M. S., Irving, A., & Gray, D. L. (1976). *Development and demonstration of a flutter-suppression system using active controls* (Technical report). Hampton, VA: National Aeronautics and Space Administration, NASA Langley Research.

Morelli, E. A. (2012). Real-time aerodynamic parameter estimation without air flow angle measurements. *Journal of Aircraft*, *49*(4), 1064-1074.

Moreno, C. P. (2015). *Linear, parameter-varying control of aeroservoelastic systems* (Unpublished doctoral dissertation). Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN.

Nissim, E. (1971). *Flutter suppression using active controls based on the concept of aerodynamic energy* (Technical report). Hampton, VA: National Aeronautics and Space Administration, NASA Langley Research.

Papatheou, E., Tantaroudas, N. D., & Ronch, A. (2013). Active control fro flutter suppression: An experimental investigation. In *International forum on aeroelasticity and structural dynamics.* Bristol, UK.

Pixhawk. (n.d.). *px4 autopilot.* Retrieved from `https://pixhawk.org/modules/pixhawk`

Roger, K. L., Hodges, G. E., & Felt, L. (1975). Active flutter suppression-a flight test demonstration. *Journal of Aircraft*, *12*(6), 551-556.

Shilling, S. A. (1972). *A study of the chirp z-transform and its applications* (Unpublished doctoral dissertation). Kansas State University, Manhattan, KS.

Soule, H. A., & Miller, M. P. (1934). *The experimental determination of the moments of inertia of airplanes* (Technical report). Langley Field, VA: National Advisory Committee for Aeronautics, NACA Langley Aeronautical Lab.

Theis, J., Pfifer, H., & Seiler, P. (2016). Robust control design for active flutter suppression. In *Aiaa atmospheric flight mechanics conference.* San Diego, CA.

Wolowitz, C. H., & Yancey, B. R. (1974). *Experimental determination of airplane mass and inertial characteristics* (Technical report). Edwards, CA: National Aeronautics and Space Administration, Flight Research Center.

A. Greyest Code

```
%% System ID Rigid Body Dynamics
clear all, close all, clc
%% Flight data

load('data_nov_ch3.mat') % flight data

Act = tf(26230,[1 80.13 2139 27110]); % Actuator model
inp = de-de(1); %rad
outq = detrend(q_flight-q_flight(1)); %rad/s
outa = detrend(az_flight-az_flight(1))*9.81; %m/s2

nt = length(de);
nfft = pow2(nextpow2(nt/2));
[Txy,F] = tfestimate(inp,outq,nt/10,nt/10-1,12*nt,50);
qde_exp = frd(Txy,2*pi*F);
nfft = pow2(nextpow2(nt/2));
[Txy,F] = tfestimate(inp,outa,nt/10,nt/10-1,12*nt,50);
azde_exp = frd(Txy,2*pi*F);
```

```

sys_exp = [qde_exp;azde_exp];
figure
bode(sys_exp,{0.1 100})

%% Longitudinal Dynamics: Short period approximation
% Elevator to Pitch rate:
% q/de = Mde*(s-Za/U1)/(s^2-(Mq+Za/U1)*s+(Mq*Za/U1-Ma))

odefun = 'detosp_2';
V = 13;
h = 150;
m = 2.41;
Iyy = 0.1965;
Cmde = -0.8487;
Cmq = -11.44;
Cma = -0.5;
CLa = 5.156;
CD1 = 0.381;
CLde = 0.7731;
CLq = 7.5153;
deda = 0.3;
```

```
parameters = {'m',m;'Iyy',Iyy;'CD1',CD1;'CLa',...
    CLa;'CLq',CLq;'CLde',CLde;'Cma',Cma;'Cmq',...
    Cmq;'Cmde',Cmde;'V',V;'h',h};

fcn_type = 'c';

sysp = idgrey(odefun,parameters,fcn_type);
sysp.Structure.Parameters(1).Minimum = 2;
sysp.Structure.Parameters(1).Maximum = 3;
sysp.Structure.Parameters(2).Minimum = 0;
sysp.Structure.Parameters(2).Maximum = 0.4;
sysp.Structure.Parameters(3).Minimum = 0;
sysp.Structure.Parameters(3).Maximum = 0.05;
sysp.Structure.Parameters(4).Minimum = 4;
sysp.Structure.Parameters(4).Maximum = 2*pi;
sysp.Structure.Parameters(5).Minimum = 5;
sysp.Structure.Parameters(6).Minimum = 0;
sysp.Structure.Parameters(7).Maximum = -0.1;
sysp.Structure.Parameters(8).Maximum = 0;
sysp.Structure.Parameters(9).Maximum = 0;
sysp.Structure.Parameters(10).Minimum = 0;
sysp.Structure.Parameters(11).Minimum = 0;
```

```

%
sysp.Structure.Parameters(1).Free = false;
sysp.Structure.Parameters(2).Free = false;
% sysp.Structure.Parameters(3).Free = false;
% sysp.Structure.Parameters(4).Free = false;
% sysp.Structure.Parameters(5).Free = false;
% sysp.Structure.Parameters(6).Free = false;
% sysp.Structure.Parameters(7).Free = false;
% sysp.Structure.Parameters(8).Free = false;
% sysp.Structure.Parameters(9).Free = false;
sysp.Structure.Parameters(10).Free = false;

sysest = greyest(sys_exp,sysp)
T_e = sysest.Report.Parameters.ParVector;


%% Verification
m = T_e(1);
Iy = T_e(2);
CD1 = T_e(3);
CLa = T_e(4);
```

```
CLq = T_e(5);
CLde = T_e(6);
Cma = T_e(7);
Cmq = T_e(8);
Cmde = T_e(9);
U1 = T_e(10);
h = T_e(11);

cbar = 0.23;
[~,~,~,rho] = atmosisa(h);
q1 = 0.5*rho*U1^2;
S = 0.44; % m2

Zw = -CD1-CLa;
Zq = -CLq;
Zn = -CLde;
Mw = Cma;
Mq = Cmq;
Mn = Cmde;

zwc = Zw*q1*S/(m*U1);
zqc = Zq*q1*S*cbar*U1/(2*m*U1^2)+U1;
```

```
znc = Zn*U1*q1*S/(m*U1);
mwc = Mw*q1*S*cbar/(U1*Iy);
mqc = Mq*q1*S*cbar^2/(2*U1*Iy);
mnc = Mn*q1*S*cbar/Iy;

Ax = [zwc zqc; mwc mqc];
Bx = [znc; mnc];
Cx = [0 1; zwc zqc-U1];
Dx = [0;znc];

tsim = (0:0.02:(length(de)-1)/50)';
%
% sp_exp = ss(Ax,Bx,Cx,Dx);
% zsim = lsim(sp_exp*Act,de-de(1),tsim);

% figure
% subplot(211); plot(tsim,outq,tsim,zsim(:,1))
% subplot(212); plot(tsim,outa,tsim,zsim(:,2))

load('Validation_doublet.mat')
t_db = [0:0.02:(length(de_db)-1)*0.02]';
de_db = de_db-de_db(1);
```

```
q_db = q_db-q_db(1);
az_db = az_db-az_db(1);
Act = tf(26230,[1 80.13 2139 27110]);
sp_exp = ss(Ax,Bx,Cx,Dx);
zsim = lsim(sp_exp*Act,de_db*pi/180,t_db);
figure
subplot(211); plot(t_db,q_db*pi/180,t_db,zsim(:,1))
ylabel('q')
subplot(212); plot(t_db,az_db,t_db,zsim(:,2))
ylabel('az')
xlabel('time')
```

```
function [A,B,C,D] = detosp_2(m,Iy,CD1,CLa,CLq,CLde,Cma,Cmq,Cmde,U1,h,
Act = tf(26230,[1 80.13 2139 27110]);
[Ac,Bc,Cc,Dc] = ssdata(Act);
cbar = 0.23;
[~,~,~,rho] = atmosisa(h);
q1 = 0.5*rho*U1^2;
S = 0.44; % m2

Zw = -CD1-CLa;
Zq = -CLq;
```

```
Zn = -CLde;
Mw = Cma;
Mq = Cmq;
Mn = Cmde;

% CZa = zwc*m*Vo/(Q*S);
% CZq = (zqc-Vo)/Vo*2*m*Vo^2/(Q*S*cbar);
% CZde = znc/Vo*m*Vo/(Q*S);
% CMa = mwc*Vo*Iy/(Q*S*cbar);
% CMq = mqc*2*Vo*Iy/(Q*S*cbar^2);
% CMde = mn*Iy/(Q*S*cbar);

zwc = Zw*q1*S/(m*U1);
zqc = Zq*q1*S*cbar*U1/(2*m*U1^2)+U1;
znc = Zn*U1*q1*S/(m*U1);
mwc = Mw*q1*S*cbar/(U1*Iy);
mqc = Mq*q1*S*cbar^2/(2*U1*Iy);
mnc = Mn*q1*S*cbar/Iy;

Ax = [zwc zqc; mwc mqc];
Bx = [znc; mnc];
Cx = [0 1; zwc zqc-U1];
```

```
Dx = [0;znc];

A = [Ax Bx*Cc; zeros(3,2) Ac];
B = [Bx*Dc;Bc];
C = [Cx Dx*Cc];
D = Dx*Dc;
end
```