DEVELOPMENT AND EVALUATION OF A DYNAMICALLY SCALED TESTBED AIRCRAFT FOR A VISUAL INERTIAL ODOMETRY DATASET

BY

FELIPE FIGUEROA

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Aerospace Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2019

Urbana, Illinois

Adviser:

Associate Professor Timothy Bretl

ABSTRACT

In this thesis we describe the design, manufacturing, and testing of a dynamically scaled aircraft, which is a scaled model of a general aviation vehicle that dynamically behaves in a similar manner as the full-scale aircraft. This scaled model (Cirrus SR22T) is to serve as a testbed for both Distributed Electric Propulsion (DEP) aircraft research and for Visual Inertial Odometry (VIO) research. The aircraft is used as a baseline to compare with the DEP aircraft, to draw conclusion regarding the effect of changing to a DEP configuration, and to provide a way to measure the effect that a DEP configuration would have on a full-scale aircraft. The aircraft is also used to collect data from various onboard sensors to provide a data set for the VIO research community to use.

ACKNOWLEDGEMENTS

First, I'd like to thank my Savior, the Lord Jesus Christ, for leading me and my family to this point and for blessing us superabundantly (see Matthew 6:33).

I also want to thank my wife, Eunice, for all the sacrifice she endured and the encouragement she provided in order for me to complete this degree. Thank you to my family as well, for encouraging me throughout my life. Thank you to the Church in Urbana, which provided me with a second family and a support network for my body, soul, and spirit.

Both professor Timothy Bretl, and professor Phillip Ansell have been integral to my success and completion of this degree. They provided the perfect environment of individual responsibility supported with advisory and technical expertise, which allowed me to grow as a researcher. Without them none of this would have been possible.

The Bretl research group, especially David Hanley, provided me with the means to overcome many technical difficulties as well as timely advice. A special thanks to all the graduate students who have helped me, especially to Aaron Perry and Kyle Pieper, whose work was the foundation to my work, and who supplied me with much support and solutions. Last but not least, thank you to all of the undergraduates who allowed me to mentor them and who were essential in the completion of many tasks along the way.

TABLE OF CONTENTS

CHAPTER 1: Introduction

Although aircraft have significant control surface redundancy, there have been over 600 deaths in the past 27 years due to complete aircraft control failure [3]. One of the most challenging control failures for a control system to deal with is a total hydraulic failure, which although rare, does occur. During the 1990's researchers investigated propulsion-controlled aircraft (PCA) as a solution to this problem [7]. PCA use the thrust from the engines to control the aircraft as a backup system and is independent of the hydraulically control system. This concept was never successfully employed on civilian transport aircraft due to numerous technical challenges [7]. One of the most significant of these challenges was the long time constant of turbofan engines, between the desired throttle input and the resulting thrust change in thrust output [8]. The time response for an emergency control system, like what NASA developed in the 1990's [7], could be long since in emergencies, performance is gladly traded for survivability. However, in order for a PAC primary aircraft control system to be able to perform as good as or better than the classic primary control systems, there is no choice but to have a short time response. Many of these challenges can potentially be addressed while maintaining the redundancy and safety standards necessary for certification by using electrical ducted fan (EDF) systems.

The EDF systems are associated with far shorter response times than turbine-driven propulsors, eliminating the time constant limitation of PCA's. Coupling the EDF system with distributed electric propulsion (DEP) aircraft lends itself to new capabilities in control systems. A DEP aircraft differs from the conventional one by having many smaller propulsors spread across the wing instead of a few bigger propulsors. This new concept improves the efficiency of the aircraft [4, 9-12] while having the potential of replacing or significantly augmenting existing

control surfaces, which will allow for large reductions in aircraft weight and fuel burn. Currently, however, there has been no research at attempting to control these DEP aircraft using propulsor thrust.

A major integration benefit of DEP comes due to the aerodynamics of placing propulsors along the leading edge of the wingspan (as a high lift device). By using DEP, the dynamic pressure over the wing is enhanced at slow speeds [14], allowing for designing an aircraft with a smaller (area) wing that is just as safe as a non-DEP aircraft during takeoff/landing. A smaller wing of course decreases the overall drag of the aircraft. A side benefit of having a smaller wing is that we are able to design the aircraft to cruise at L/D max [14], something that is difficult to achieve due to the fact that the L/D max for larger wings usually is at a lower velocity than what is needed for cruise.

An additional DEP advantage can be achieved by mounting the propulsors at each wingtip, enabling the propellers to be spun against the tip vortex [14]. This results in an increase in propulsive efficiency due to the reduction in induced drag [17], making the airplane more efficient.

Classic aircraft control has been well studied and understood. PCA has been preliminarily studied mainly in the context of emergency aircraft control [6]. The proposed research would go further by carrying out the novel study of PCA as the primary aircraft control and applying it to a state-of-the-art DEP design. In order to develop this complex control system, this study aims to advance our understanding of two multidisciplinary areas of aircraft design: the rapid and accurate control of engine thrust, and the aerodynamic coupling effects of leading edge propellers in close proximity. This knowledge could lead to more efficient aircraft design, perhaps eliminating the need of vertical and horizontal tails.

The DEP design, however, brings new challenges. With new challenges there is also higher risk. In order to mitigate this risk, there is the need for a baseline aircraft to serve as a technology proof of concept. This aircraft would test out and evaluate the subsystems that would also go on the DEP aircraft such as the flight controller and DAQ systems. This baseline aircraft was begun by our groups previous work. The end goal for this baseline aircraft is to serve as a way to quantify the effect of changing an aircraft to a DEP configuration. This done by characterizing the dynamics of the baseline to then compare with the dynamics of the DEP aircraft. This is done by performing system identification on both aircrafts, so we have two mathematical models, one for each aircraft. We can then compare, in both simulation and experimental settings, the response of the aircrafts to the same inputs at the same flight conditions.

Another contribution of this work is to relate our DEP aircraft dynamics to the full-scale general aviation aircraft. This is done by dynamically scaling the baseline aircraft, to match the dynamics of the full-scale aircraft. NASA [18] has been researching this topic since the 1930's and has developed a mature method for rigid-body dynamic models in incompressible flow that we follow. Using this method and based on the scaling factor we can compute the required mass, moment of inertia, altitude, and velocity that would cause the dynamics of our scaled baseline aircraft to behave in similar fashion as the full-scale aircraft. Once we have a mathematical model of our dynamically scaled baseline aircraft we have a way to relate our DEP aircraft dynamics to the full-scale aircraft, which would further our DEP research by providing a way to apply our findings to the general aviation sector.

Our dynamically scaled platform also combines this research work with the visual-inertial odometry (VIO) research area. VIO has been an active area of research for robotic

navigation systems for several years. This work has resulted in VIO algorithms, which are now used in commercial products such as the Google Tango. Several open source methods have also been produced for public consumption [32,46,47,48]. Several survey papers have been recently published regarding this area of work [40,56]. This approach to state estimation is generally considered valuable to aerial robot navigation and autonomous driving among many other areas of robotics. Separately from work on visual-inertial odometry, there has been significant interest in the robotics community regarding accurate descriptions of dynamic models of micro aerial vehicles [29,33]. These models have been used for several different control applications such as improved thrust control in [30], improved position control in [60], and simulation of aerial vehicles in [39]. Besides modeling work performed by the robotics community, the aerospace community has been developing models for aerial vehicles for decades [37,54,59]. In existing open sourced approaches to visual-inertial odometry, inertial measurements are either numerically integrated or preintegrated according to general rigid body kinematic equations of motion. However, using previously developed models of aerial robot dynamics, researchers have found that inertial measurement units (IMUs) can also be used to measure forces experienced by an aerial vehicle [22,38,41,42,43,52]. Using these dynamic models and IMUs to identify forces exerted on a vehicle have been used to improve the estimate of the attitude of aerial robots [26,27,31,35,36,44,57], to estimate other disturbance forces like wind [23,58], to estimate the presence of system faults [28], in relative state estimation [45], and in visual-inertial navigation [20,21,24,25,49,55]. To distinguish the use of such models to improve visual-inertial odometry from the methods used in existing open sourced approaches, we will call methods which use the IMU to measure forces on a rigid body dynamic VIO. Some researchers also use the term model-aided VIO. Researchers like [42,50] have shown that using such models in state estimators like

visual-inertial odometry improve the accuracy of attitude and velocity estimates and makes such approaches to be more robust to slower camera update rates. Several datasets already exist for testing visual-inertial odometry in aerial robotic systems. These include the EuRoC dataset [34] and Zurich Urban dataset [51]. Both datasets serve as an excellent means by which visual-inertial odometry algorithms can be compared. However, neither datasets contain an accurate dynamic model of their aerial systems. Therefore, it is currently not feasible to compare approaches to using dynamic models of aerial robots as done by [20] for example. As a result, claims regarding the use of such dynamic models for VIO have not been validated and compared to some of the more recent open source methods of visual-inertial odometry. In this work, we present the UIUC Dynamic VIO dataset. This dataset is specifically designed to enable further research on VIO methods using IMUs to measure forces experienced by a vehicle. Our dataset contains flights with a fixed wing aircraft. Ground truth, camera, and IMU data are provided for each flight. Because of this work, dynamic VIO methods can be compared against each other as well as against existing open source VIO approaches. We hope that this dataset can be used to uncover how these dynamic models can be used in the state estimation process: such as how to improve VIO with respect to different measures of robustness or relative pose error.

CHAPTER 2: Dynamically Scaled Testbed Aircraft

2.1. Model construction and sensor implementation

Based on the recent work of Pieper [61], the testbed aircraft was developed from a 21% scaled Hangar 9 Cirrus SR22T kit (Figure 2.1) [62], which was significantly modified to be dynamically scaled and to accommodate an array of sensors to make it suitable for acquiring detailed aircraft state and visual data.

Figure 2.1 – Image of Hanger 9 Cirrus SR22T

2.1.1. Model construction

The Hangar 9 kit was built according to the provided instructions, but several modifications were implemented. Due to the increase in weight to make the aircraft dynamically scaled, the landing gear was strengthened by adding an aluminum plate that joined the two aft landing gears. To increase the rigidity of the rudder and decrease the shaft play in the servo link, the push-pull cable servo link was changed to a metal push rod, fitted inside of a metallic sleeve, to avoid binding (see figure 2.2). Since there are sensors in the elevator that may require future

servicing, the elevator was changed from being a permanent installation to a removable one, by installing a set screw, that screws into the joiner rod. Stronger servo linkages were used on every control surface, and the Futaba BLS171SV servos were chosen due to their high torque to weight ratio.

Figure 2.2 – Metal push rod link for rudder

2.1.2. Sensors

In order to use this aircraft as a testbed for other research, and in order to identify the dynamics of the aircraft, the aircraft was outfitted with many sensors, while the control surface commands were logged by the avionics package. Hall effect sensors were installed in order to record the actual control surface deflections produced during flight. Hall effect sensors are used to measure a magnetic field, hence they were placed near the control surface, while a magnet was

installed on the control surface (see figure 2.3) which moves further or closer from the sensor as the control surface moves. The hall effect sensors used were the TDK HAL830UT-A.

An RCAT alpha/beta probe was also mounted on the right wing. This probe measures angle of attack and sideslip angle while being light weight. It uses a 5v nominal operation with a 0.75-degree resolution and +/- 1-degree accuracy.

To measure the in-flight airspeed, a pitot static tube attached to an All Sensors 20-CMH2O pressure sensor was integrated on the left wing. The static and total pressure tubes were routed to the sensor, and the voltage output corresponded to the dynamic pressure, which was then used to determine airspeed.

All of the mentioned sensors were connected to the data acquisition system via twisted-pair double-shielded data cabling (see figure 2.4) that helps protect against electromagnetic interference (EMI) and signal distortion. See table 2.1 for a summary of onboard sensors.

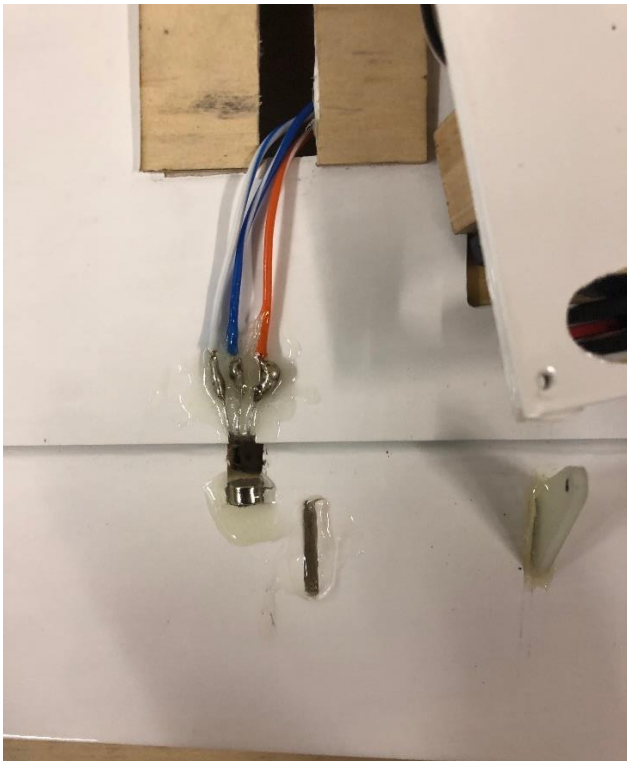

Figure 2.3 – Hall effect sensor placement

Figure 2.4 – Twisted-pair double-shielded data cabling used

Sensor	**Measuring**	**Specification**
Hall effect TDK HAL830UT-A	Aileron (x2), rudder (x1), elevator (x2) deflection	30 mT to 100 mT range
RCAT alpha beta probe	Angle of attack and side slip angle	0.75° resolution, +/- deg accuracy
All Sensors 20-CMH20 Pitot Static Tube	Dynamic pressure for velocity	-0.2 to 20 cmH2O, 5 to 50° C

Table 2.1 – summary of onboard sensors and their respective properties

2.1.3. Data acquisition system

The data acquisition system was composed of an Arduino Uno with an 8-channel 16-bit ADC shield from Iowa Scaled Engineering, an 8-channel low pass filter, and a 5V step-down voltage regulator. Figure 2.5 shows the data acquisition system and Table 2.2 summarizes the associated components and specifications. The data were acquired through the Arduino, and then recorded on the flight controller Raspberry Pi via a USB cable. There was a 2S Lipo battery connected to the voltage regulator, which powered the sensors and Arduino/ADC shield.

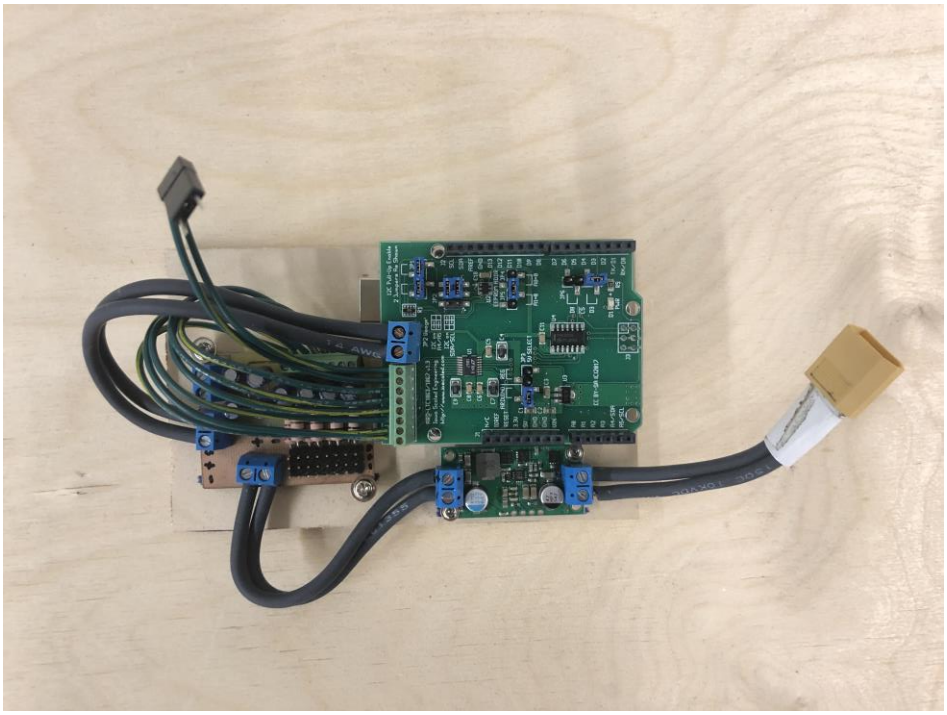

Figure 2.5 - Data acquisition system used for collecting and recording data from onboard sensors

DAQ Component	**Property**
Arduino Uno	7-12V Input Voltage, 50mA Current
LTC1867 ADC	16-bit, 8-channel, 200ksps sampling rate
Low pass filter	
Pololu voltage regulator	Constant 5V output

Table 2.2 – Data acquisition components and their respective properties

2.1.4. Propulsion and power system

Due to the increase in weight for dynamically scaling the aircraft a more powerful propulsion system was needed from that recommended for the stock Cirrus SR22T model. In the end the motor, ESC, and propeller used for this project were the Hacker A60-14L, Castle Creations Phoenix Edge 120 amps ESC, and a 19" x 12" APC thin electric composite propeller, respectively. Three 4S 12,000 mAh Lumenier LiPo batteries connected in series powered the system, and for safety an EMOCTEC safety power switch was placed between the batteries and the ESC. The ESC was also connected to the avionics (directly to the Navio2 servo rail) in order to receive the throttle commands. Figure 2.6 shows the propulsion system.

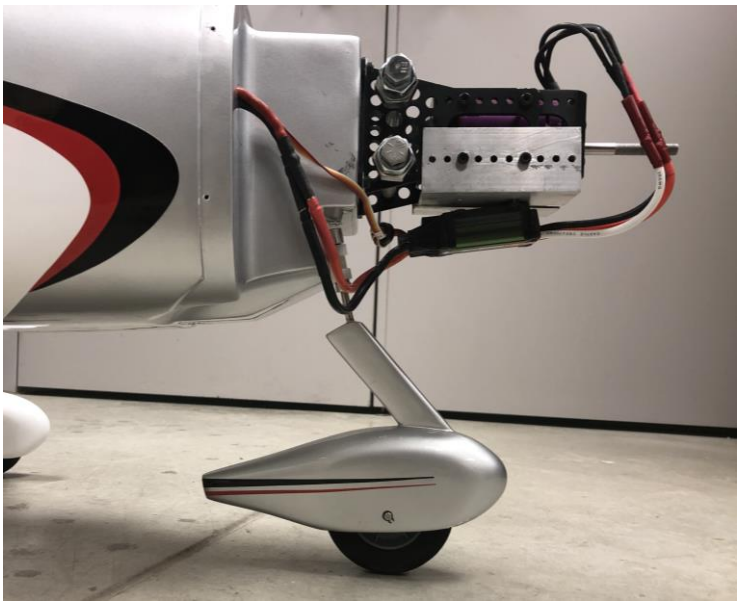

Figure 2.6 – Propulsion system

2.1.5. Flight Controller

The flight controller used for this project was a Raspberry Pi 3 with a Navio2 hat that was running a PX4 autopilot. A Futaba 14SGA 14-Channel transmitter and Futaba R7014SB 14-Channel FASSTest/FASST Receiver were used (see figure 2.7). The receiver was connected to

the Navio2 hat via the SBUS RC input on the servo rail. The PX4 code mainly passed the transmitter signals through to the control surface (via RC outputs on servo rail) but could also enter a system identification program that would overlay the transmitter signal with a predetermined signal sequence. This program was used to excite different modes of the aircraft allowing the dynamic responses (via onboard sensors) to be recorded.

The Navio2 RC outputs from the servo rail were routed to a power distribution board, a Smart-Fly PowerSystem Sport Plus which powered and routed the signal to the control surface servos. The configuration of transmitter signals and Navio2 RC outputs are summarized in Table 2.3. Figure 2.8 shows the in-depth controller wiring schematic. There was a 2S battery directly powering the Raspberry Pi/Navio2 unit via the servo rail and there were two 2S batteries connected to the power distribution board which also powered the servos.

Figure 2.7 – Futaba 14SGA transmitter and R7014SB Receiver used

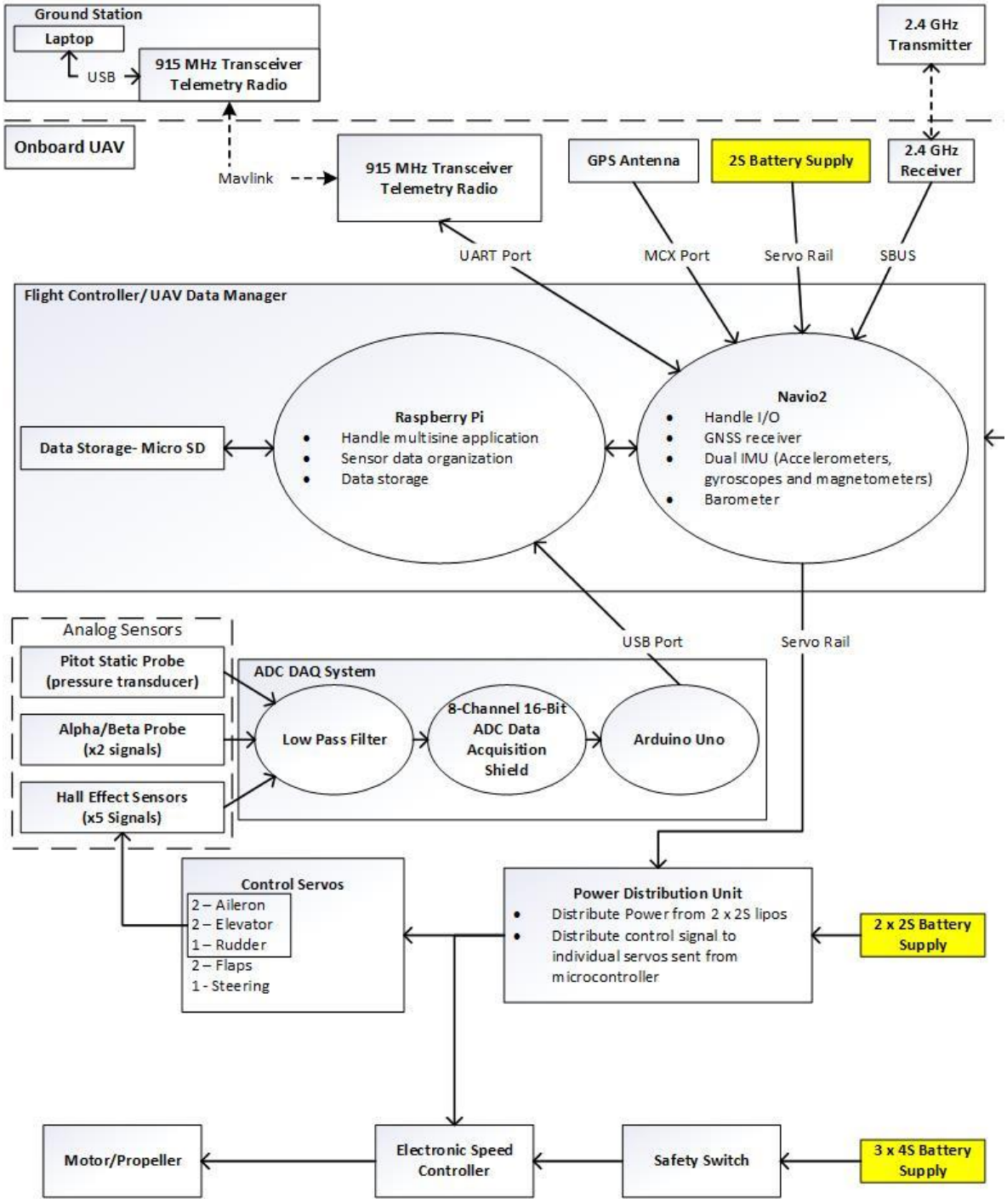

Figure 2.8 – In depth controller wiring schematic

Transmitter/Receiver	Navio2 Rail Output	Power System Sport Plus Input	Power System Sport Plus Output
Ch. 1 – Aileron	Ch. 0 – SBUS in	Ch. A – Landing Gear (from Receiver Ch. 4)	Ch. A – Front Landing Gear Servo
Ch. 2 – Elevator	Ch. 1 – Aileron	Ch. B – Aileron (from Navio2 Ch. 1)	Ch. B – Aileron Servo (x2)
Ch. 3 – Thrust	Ch. 2 - Elevator	Ch. C – Elevator (from Navio2 Ch. 2)	Ch. C – Elevator Servo (x2)
Ch. 4 – Rudder	Ch. 3 - Thrust	Ch. D – Rudder (from Navio2 Ch. 4)	Ch. D – Rudder
Ch. 5 – Flap	Ch. 4 - Rudder	Ch. E – Motor (from Navio2 Ch. 3)	Ch. E – Motor ESC
Ch. 6 - Mode		Ch. F – Flap (from Receiver Ch. 5)	Ch. F – Flap Servo (x2)
Ch. 7 - Gain			

Table 2.3 – Summary of transmitter channels and RC outputs

2.2. Dynamically scaling the aircraft

One of the goals of this study is to have a scaled version of a general aviation aircraft that not only is dimensionally proportional, but also dynamically scaled. This means that the scaled UAV will produce a scaled dynamical response relative to that of the full-scale aircraft with the same input. NASA has previously studied methodologies for dynamically scaling aircraft [18] and has come up with different relations in order to perform this process properly. Previous efforts have also been conducted at the University of Illinois at Urbana-Champaign (UIUC), GA-USTAR, a dynamically-scaled 20% scale Cessna 182 [63].

2.2.1. Calculation

Using the incompressible flow conditions scale factors for dynamic models developed by NASA [18] the scaling laws for different aircraft properties between the full and scaled aircrafts can be calculated. Table 2.4 summarizes these findings.

Linear dimension	n
Relative density ($m/\rho L^3$)	1
Froude Number (V^2/Lg)	1
Angle of attack	1
Linear Acceleration	1
Weight, mass	n^3/σ
Moment of Inertia	n^5/σ
Linear Velocity	$n^{1/2}$
Angular Velocity	$1/n^{1/2}$
Time	$n^{1/2}$
Reynolds number (VL/ν)	$n^{1/2}\nu/\nu_o$

Table 2.4 – Scale factors for rigid dynamics models tested at sea level. Multiply full-scale values by the indicated scale factors to determine model values, where n is the ratio of model-to-full-scale dimensions, σ is the ratio of air density to that at flying altitude (ρ/ρ_o), and ν is the value of kinematic viscosity.

Table 2.5 summarizes the full-scale aircraft characteristics that our scale aircraft is matching. These values were derived in the previous work on this project [61] and based on the Frasca International, Inc. Cirrus SR22-T flight simulator model.

Parameter	**Full scale aircraft**
Cruise altitude (ft)	6562
Max cruise speed (mph)	153
TOW (lbs)	2900
I_{xx} (slugs-ft^2)	2306.5
I_{yy} (slugs-ft^2)	1841.8
I_{zz} (slugs-ft^2)	3331.0

Table 2.5 – Full-scale Cirrus SR22-T aircraft characteristics.

Using the relations from table 2.4 with the full-scale aircraft characteristics from table 2.5, the following target model characteristics are produced (summarized in Table 2.6):

$$\sigma = \frac{\rho}{\rho_o} = \frac{1.007\ kg/m^3}{1.186\ kg/m^3} = 0.840 \qquad \text{Eq. 2.1}$$

$$n = 0.21$$

$$V_{max,cruise}^{model} = V_{max,cruise}^{FS} * n^{1/2} = 153\ mph * (0.21)^{1/2} = 70.11\ mph \qquad \text{Eq. 2.2}$$

$$TOW^{model} = TOW^{FS} * \frac{n^3}{\sigma} = 2900\ lbs * \frac{(0.21)^3}{0.840} = 31.97\ lbs \qquad \text{Eq. 2.3}$$

$$I_{xx}^{model} = I_{xx}^{FS} * \frac{n^5}{\sigma} = 2306.5\ (\text{slugs} * ft^2) * \frac{(0.21)^5}{0.840} = 1.121\ (\text{slugs} * ft^2) \qquad \text{Eq. 2.4}$$

$$I_{yy}^{model} = I_{yy}^{FS} * \frac{n^5}{\sigma} = 1841.8\ (\text{slugs} * ft^2) * \frac{(0.21)^5}{0.840} = 0.896\ (\text{slugs} * ft^2) \qquad \text{Eq. 2.5}$$

$$I_{zz}^{model} = I_{zz}^{FS} * \frac{n^5}{\sigma} = 3331.0\ (\text{slugs} * ft^2) * \frac{(0.21)^5}{0.840} = 1.620\ (\text{slugs} * ft^2) \qquad \text{Eq. 2.6}$$

Parameter	**Target scaled model**
Cruise altitude (ft)	1100
Max cruise speed (mph)	70.11
TOW (lbs)	31.97
I_{xx} (slugs-ft^2)	1.121
I_{yy} (slugs-ft^2)	0.896
I_{zz} (slugs-ft^2)	1.620

Table 2.6 – Scaled model aircraft characteristics

The values in Table 2.6 represent the target values for the aircraft used in the current study.

2.2.2. Inertia rig testing

The moments of inertia were measured using the inertia rig in the UIUC Aerodynamics Research Lab (ARL). Due to the size of the aircraft used in the current study, the moment of inertias had to be measured in sections and then combined using the Parallel Axis Theorem equation 2.7.

$$I = I_{cm} + md^2 \qquad \text{Eq. 2.7}$$

The measurements were performed by mounting the aircraft from a freely rotating bearing and allowing a known weight to hang from the other end (Figure 2.9). The weight created a known torque and the aircraft was allowed to freely rotate. We then record the angular moment of inertia using a gyroscope on our Navio2 and the px4 code. We record ten free rotations with four

different known weights. We then use the following energy equations to calculate the moment of inertia based on a least squares fit MATLAB function lsqlin().

$$PE = KE + Drag = AI + Bd \quad \text{Eq. 2.8}$$

Where ‘A’ and ‘B’ are coefficients based on our measurements of angular velocity. ‘I’ and ‘d’ are the parameters that are estimated, namely the moment of inertia and drag parameter due to air resistance respectively.

$$PE = \tau * \Theta = \tau \int \omega = mr \int \omega \quad \text{Eq. 2.9}$$

$$KE = \tfrac{1}{2}\omega^2 I = AI \quad \text{Eq. 2.10}$$

$$Drag = -d \int \omega^3 = Bd \quad \text{Eq. 2.11}$$

Where ω, m, and r are the measured angular velocity, hanging mass, and radius respectively.

Figure 2.9 – Scaled aircraft mounted onto the inertia rig ready to measure I_{yy}.

2.2.3. Optimization of mass placement for moment of inertia problem

With the calculated target moment of inertias that would dynamically scale the baseline aircraft and with a way to experimentally calculate our actual moment of inertias, we had the problem of adding mass in strategic locations in order to match these desired parameters as closely as possible. This problem was turned into the following optimization problem.

In order to simplify the manufacturing and the problem, three locations were chosen as potential mass mounting points. The moment of inertia equation was one of the main equations used where 'm' is the mass and r is the distance of each particle from the axis of rotation:

$$I = \sum mr^2 \qquad \text{Eq. 2.12}$$

2.2.3.1. Formal problem statement

The general form of the problem that was used was:

$$\min L(x)$$

$$st\ Ax\ \leq b \qquad \text{Eq. 2.13}$$

$$Ax = b$$

$$lower\ bound\ \leq x\ \leq upper\ bound$$

This problem form follows the MATLAB documentation in order to use the function fmincon().

2.2.3.2. *MATLAB fmincon() function*

The MATLAB function fmincon() is a nonlinear programming solver [70] that has five different algorithms to at its disposal. The default algorithm, and the one used for this problem, is the interior-point optimization algorithm which when given the problem:

$$\min_{x} f(x), subject\ to\ h(x) = 0\ and\ g(x) \leq 0 \qquad \text{Eq. 2.14}$$

Solves a sequence of equality constrained approximate problems:

$$\min_{(x,s)} f_{\mu}(x,s) = \min_{x} f(x) - \mu \sum_{i} \ln(s_i), subject\ to\ h(x) = 0\ and\ g(x) + s = 0 \qquad \text{Eq. 2.15}$$

Where s_i are the slack variables which turn the inequality constraints into equality constraints, and the right side of the cost function (with the logarithmic term) is the barrier function. Two types of steps are used at each iteration:

- Direct step in (x, s) or newton step which solves the KKT equations for the linearized approximate problem.
- Conjugate gradient step which minimizes a quadratic approximation to the approximate problem in a trust region, subject to linearized constraints.

The default step used is the direct, but if not possible, the gradient step is taken. At each iteration the algorithm decreases the following merit function:

Merit function:

$$f_{\mu}(x,s) + \nu \|h(x), g(x) + s\| \qquad \text{Eq. 2.16}$$

2.2.3.3. *Moment of inertia problem statement*

To match this architecture, our state vector *x* was chosen to have three states, namely:

$$x = \begin{bmatrix} mass\ @\ location\ \#1 \\ mass\ @\ location\ \#2 \\ mass\ @\ location\ \#3 \end{bmatrix} = \begin{bmatrix} m1 \\ m2 \\ m3 \end{bmatrix} \qquad \text{Eq. 2.17}$$

The three mass locations chosen are shown in Figure 2.10

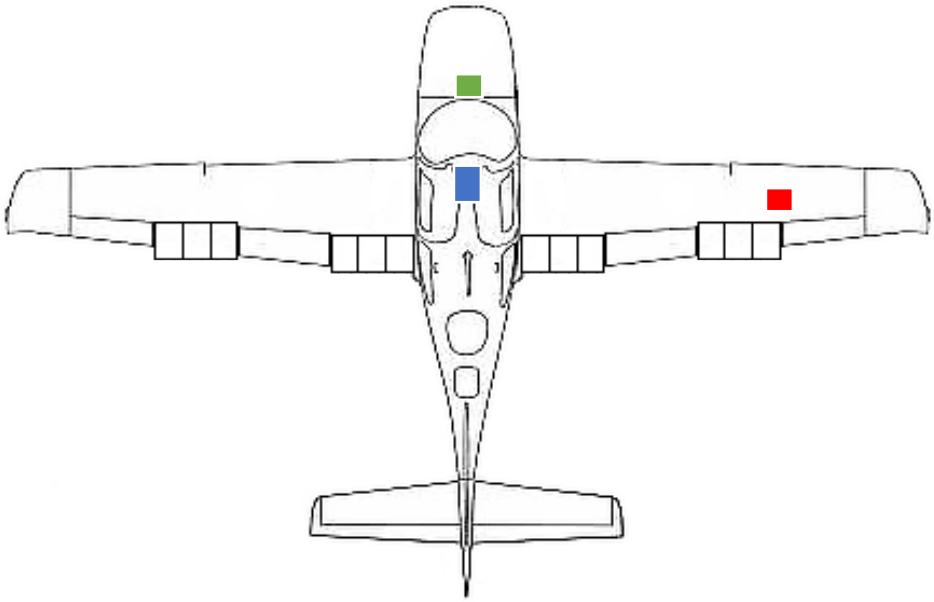

Figure 2.10 – Three locations on the aircraft chosen to add mass

The location coordinates were given the convention of subscript indicates the point location. For example, equation 1.1 shows the naming convention for the coordinates for location number 1:

$$\begin{bmatrix} x\ coordinate\ \ of\ location\ 1 \\ y\ coordinate\ \ of\ location\ 1 \\ z\ coordinate\ \ of\ location\ 1 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad \text{Eq. 2.18}$$

With the main equation used being:

$$\Delta I = (d_1)^2 m_1 + (d_2)^2 m_2 + (d_3)^2 m_3 \quad \text{Eq. 2.19}$$

Where d is the distance from the mass to the axis of rotation, e.g.

$$d_{xx} = \sqrt{(y_1)^2 + (z_1)^2} \quad \text{Eq. 2.20}$$

The constraints equations were as follows:

$$Ax \leq b$$

$$x = \begin{bmatrix} m1 \\ m2 \\ m3 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 1 & 1 \\ \left(\sqrt{(y_1)^2 + (z_1)^2}\right)^2 & \left(\sqrt{(y_2)^2 + (z_2)^2}\right)^2 & \left(\sqrt{(y_3)^2 + (z_3)^2}\right)^2 \\ \left(\sqrt{(x_1)^2 + (z_1)^2}\right)^2 & \left(\sqrt{(x_2)^2 + (z_2)^2}\right)^2 & \left(\sqrt{(x_3)^2 + (z_3)^2}\right)^2 \\ \left(\sqrt{(x_1)^2 + (y_1)^2}\right)^2 & \left(\sqrt{(x_2)^2 + (y_2)^2}\right)^2 & \left(\sqrt{(x_3)^2 + (y_3)^2}\right)^2 \end{bmatrix} \quad \text{Eq. 2.21}$$

$$b = \delta_{overshoot} * \begin{bmatrix} m_{current} - m_{desired} \\ I_{xx,current} - I_{xx,desired} \\ I_{yy,current} - I_{yy,desired} \\ I_{zz,current} - I_{zz,desired} \end{bmatrix}$$

No equality constraints were used. See conclusion section for a discussion of the constraints. The overshoot weight allows the optimal solution to overshoot the desired parameter if it reduces the overall error.

Our cost function used was a form of the Euclidian norm of the parameters calculated after adding the mass minus the desired parameters (an error) but weights were added to control which parameters have more weight hence that specific parameters' error is reduced more than the others. See the following equations:

Without weights:

$$L(x) = norm(x_{measured} + Ax - x_{desired}) \quad \text{Eq. 2.22}$$

With weights (actually used):

$$L(x)$$

$$= norm\left(\begin{matrix} w_1 * \left(\frac{m_{current} + A(1,:) * x - m_{desired}}{m_{desired}}\right)^2 + w_2 * \left(\frac{I_{xx,current} + A(2,:) * x - I_{xx,desired}}{I_{xx,desired}}\right)^2 + \\ w_3 * \left(\frac{I_{yy,current} + A(3,:) * x - I_{yy,desired}}{I_{yy,desired}}\right)^2 + w_4 * \left(\frac{I_{zz,current} + A(4,:) * x - I_{zz,desired}}{I_{zz,desired}}\right)^2 \end{matrix}\right)$$

Eq. 2.23

The bounds for our states were as follows:

$$0 \leq x \leq \infty$$

Some assumptions for this problem were:

- The mass added is added at one point. Since we used lead weights which pack more mass per volume, this was not too far off, but for the locations with higher mass, this assumption is broken in the physical airplane.
- The x, y, and z locations are assumed to be exact, when, in reality, due to physical limitations it is hard to measure exact distances to the locations.
- It was assumed we could measure the moment of inertias exactly, when, in reality, our measurements are based on experimental values that have a certain amount of errors.

2.2.3.4. *Actual values in problem statement*

The following are the actual values used in the MATLAB script:

	Location #1	Location #2	Location #3
x(ft)	1	-0.167	0
y(ft)	0	3	0
z(ft)	0.458	0	-0.833

Table 2.7 – Locations chosen on aircraft that would affect the moment of inertia strategically while making manufacturing feasible

$$\min_{x} L(x)$$

$$A = \begin{bmatrix} 1.0000 & 1.0000 & 1.0000 \\ 0.2098 & 9.0000 & 0.6944 \\ 1.2098 & 0.0279 & 0.6944 \\ 1.0000 & 9.0279 & 0.0000 \end{bmatrix} \qquad \text{Eq. 2.24}$$

$$b = \begin{bmatrix} 0.672 \\ 1.948 \\ 0.058 \\ 0.483 \end{bmatrix}$$

$$0 \leq x \leq \infty$$

The weights were changed to understand their different effects but, in the end, matching the total mass of the aircraft to the desired total mass was seen as an important parameter that was weighed higher than matching the moment of inertias.

In the end there needed to be an iterative process of adding weight in specific locations to get the actual mass and moment of inertias to match the target. Figure 2.11 shows the final weight configuration which satisfactorily achieved the target. See the results section for a summary of the dynamically scaling process.

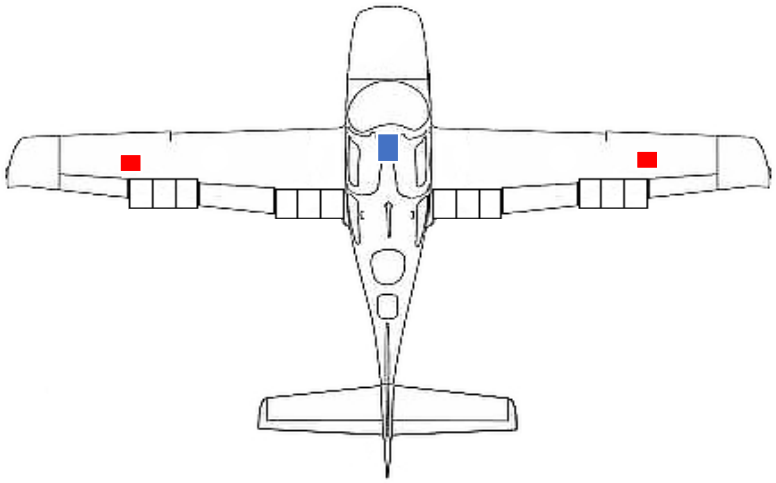

Figure 2.11 – Final weight configuration for actual dynamically scaled aircraft

CHAPTER 3: Visual Inertial Odometry Dataset

In tandem to dynamically scaling an aircraft, another use for the testbed aircraft was to collect data for a publicly available visual inertial dataset for the visual odometry community. Datasets are used by the community to test and compare new research developments in the field. Currently, the two most popular data sets in the field are the EuRoC micro aerial vehicle datasets [34], and the Zurich urban micro aerial vehicle dataset [51]. Both of these datasets were onboard a rotorcraft and at low altitudes (<50ft). The EuRoC data set was indoors (in a machine room and a Vicon room), while the Zurich urban dataset was outdoors in urban streets. While these datasets have been used, and proven, by many different groups, the addition of a dynamics model and visual-inertial dataset for a fixed wing aircraft, represents a significant contribution.

Within the vision community there has been some progress [42,50] to use the mathematical model of the system to improve the accuracy of the visual-inertia odometry algorithms. Hence, these new algorithms use the camera and IMU data, together with the model of the system to localize where in space the system is. Of course, in order to use these new algorithms, there needs to be a model of the system a-priori. As a result, the dataset is constituted to include the necessary data (stereo images, IMU measurements with RTK GPS as accurate ground truth) for VIO and the model of the system, in order for these new algorithms to be tested.

In order to collect the necessary data, two more systems were added to the testbed. The components added were an RTK GPS, a stereo camera, and a second Raspberry Pi to interface with and collect the data from these two systems. Figure 3.1 shows a schematic of the added system.

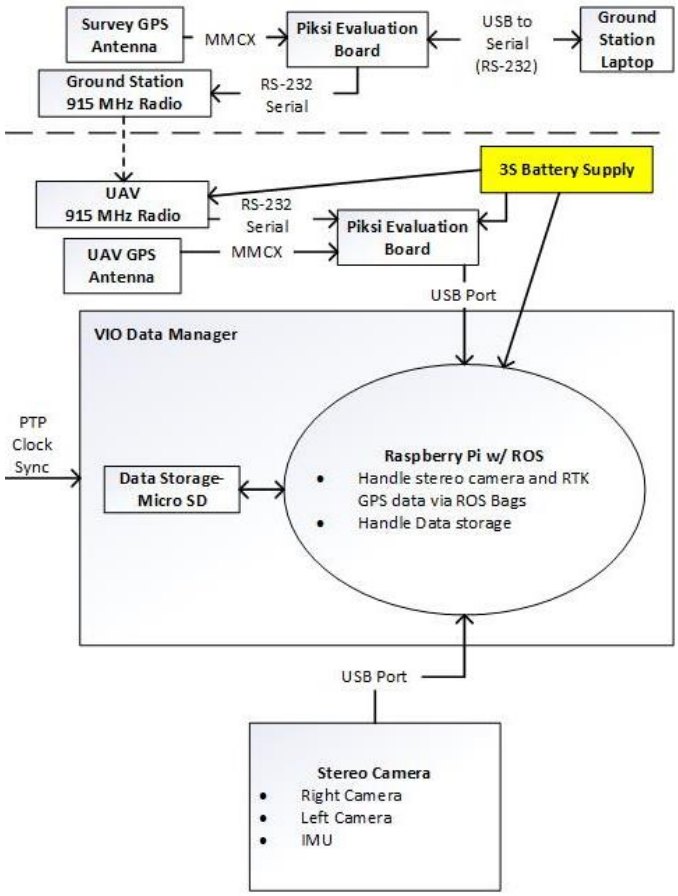

Figure 3.1 – schematic of dataset sensor system

3.1. Real-time kinematic positioning system

Ground truth is one of the most important factors for providing a good dataset. This measurement is obtained by using a real-time kinematic (RTK) positioning system. With RTK GPS, there are two GPS receivers, one on board of the UAV, the other is a ground station. Each receiver is also connected to a radio, so the ground station is able to send position corrections to the UAV receiver. With this system, position data is able to be recorded with up to 1cm + 1ppm horizontal accuracy and 1.5cm + 1ppm vertical accuracy [64].

3.1.1. Ground station

The ground station is a vital component of the RTK GPS, since it's necessary in order reduce the error due to the ionosphere and produce a more accurate positioning system. The ground station is composed of (see figure 3.1) a laptop that is connected to the Swift Navigation Piksi Multi GPS receiver with evaluation board (see figure 3.2). The receiver is also connected to a Swift Navigation GNSS mini-survey antenna and to a FreeWave 915 MHz radio that sends signals up to the UAV (see figure 3.3). Both the laptop and the receiver are powered via a wall outlet, while both the antenna, and radio are powered by the Piksi Multi.

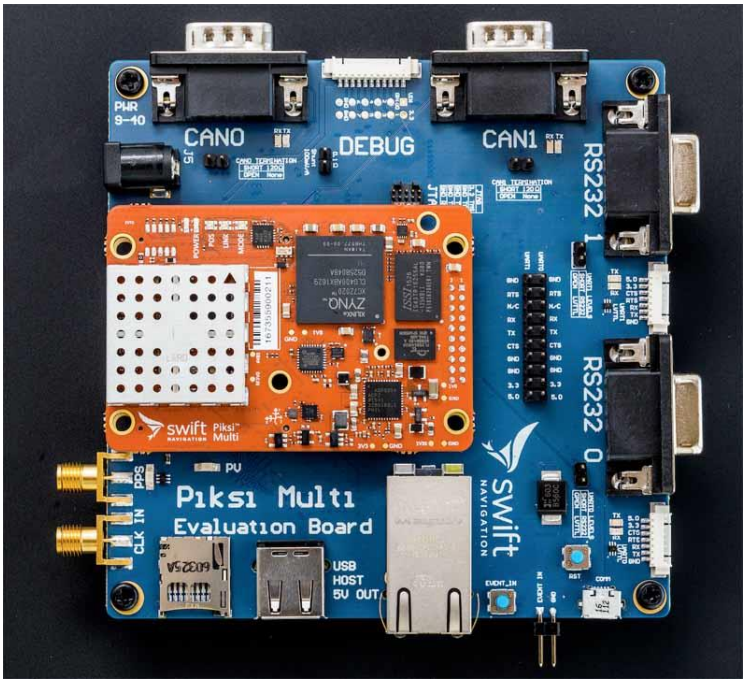

Figure 3.2 - Swift Navigation Piksi Multi GPS receiver with evaluation board

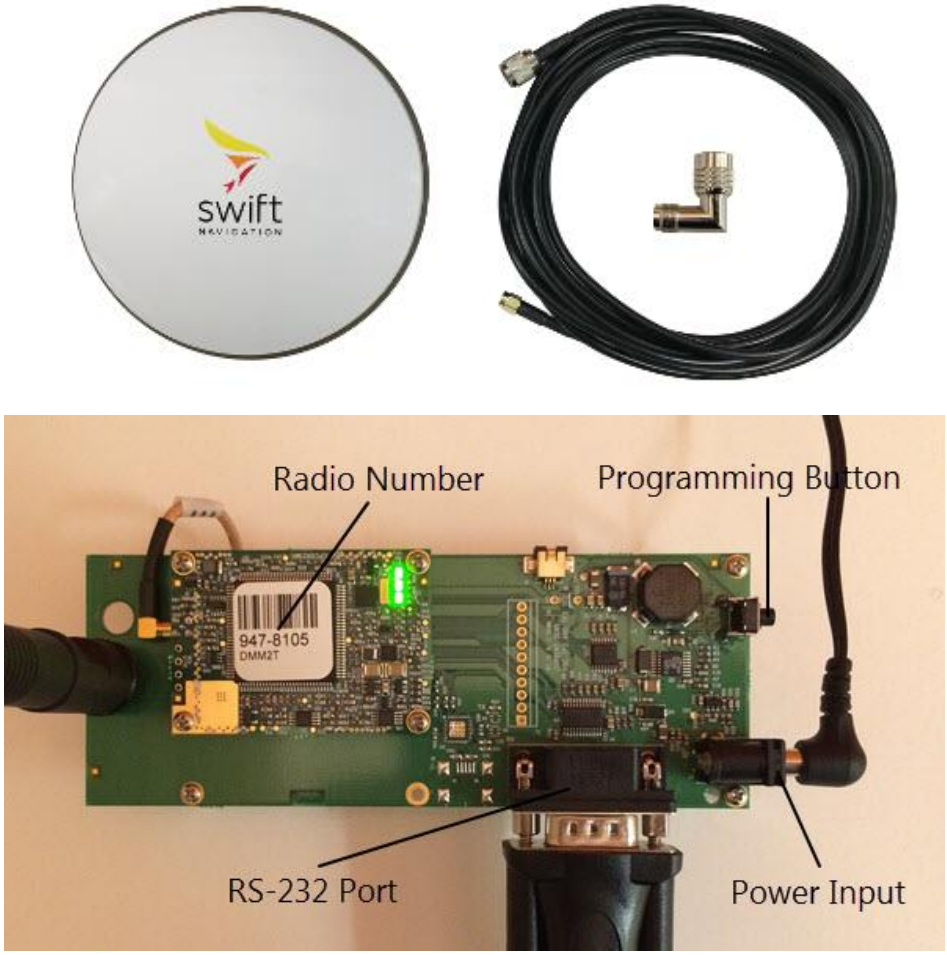

Figure 3.3 - Swift Navigation GNSS mini-survey antenna and the FreeWave 915 MHz radio

3.1.2. UAV system

Onboard of the UAV, the dataset system is very close to the ground station but with some significant differences mainly due to weight restrictions. Instead of a laptop we used a Raspberry Pi which is lighter but has less computational capabilities. Also, instead of the survey antenna we have a Maxtena M1227HCT-A2-SMA lightweight UAV GPS antenna (see figure 3.4). The UAV system has a second Swift Navigation Piksi Multi GPS receiver with evaluation board, and a second FreeWave 915 MHz radio.

Figure 3.4 - Maxtena M1227HCT-A2-SMA lightweight UAV GPS antenna

3.2. Loitor stereo camera

The Loitor Visual-Inertial stereo camera (see figure 3.5) was used for collecting image data. This camera is a common open-source unit used for VIO and simultaneous localization and mapping (SLAM). It has two complementary metal oxide semiconductor (CMOS) image sensors, and one IMU that outputs acceleration, rotational speed, and quaternions. The camera is connected to the onboard dataset Raspberry Pi via USB. All image and IMU data are sent through this USB connection for logging.

Figure 3.5 – Loiter Visual-Inertial stereo camera

3.3. VIO sensors integration

In order to record and interact with the dataset sensors a second Raspberry Pi (we refer to this unit as the dataset Raspberry Pi) was placed on board. Both the Piksi Multi and the Loitor stereo camera were plugged into the RPI via USB.

3.3.1. Raspberry Pi with Robot Operating System

Raspbian Stretch and Robot Operating System (ROS) Kinetic Kame (ROS-Comm) were onboard the Raspberry Pi. ROS was used to interact with the two sensors via open-source [65,66] ROS nodes. The data was then recorded as a rosbag and saved on the Raspberry Pi memory card. A 3S battery was used to power both the Piksi receiver and the Raspberry Pi. A full system schematic can be seen in Figure 3.1

3.3.2. Code structure

ROS provided a very convenient way to interact with the sensors on the aircraft testbed. In ROS, there are “nodes” which are “an executable that uses ROS to communicate with other nodes” [67]. The current configuration has two nodes, one for the GPS and one for the camera. A custom ROS launch code was written to start these two nodes at the same time with one command. Within ROS each node can publish to a “ROS topic”, which is an avenue for data/messages to be sent within ROS (Figure 3.6). The two nodes publish the data from each sensor over several topics. In order to record the data, the rosbag package was used, which subscribes to all the different topics which our data is in, and records them onto the our onboard memory card. These rosbags can be converted to a more useful format postprocess.

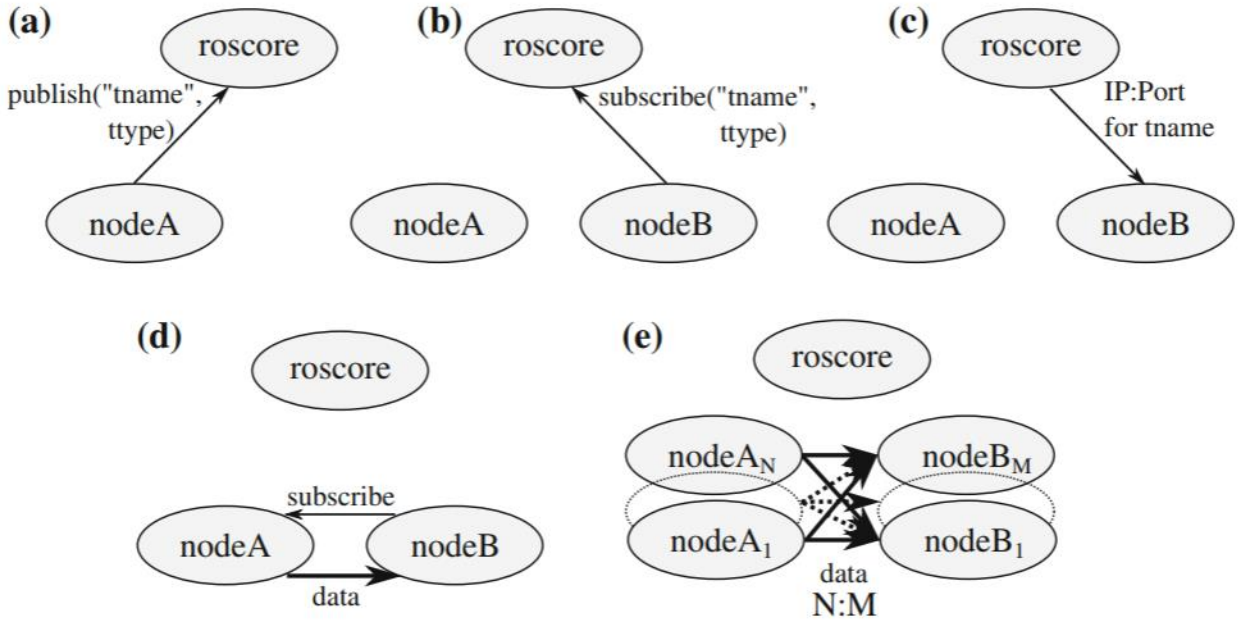

Figure 3.6 – Overview of the ROS topic mechanism. When a publisher (a) and subscriber (b) are registered to the same topic, the subscriber receives the network address and port of all publishers (c). The subscriber continues by directly contacting each publisher, which in return starts sending data directly to the subscriber (d). Many nodes can publish and subscribe to the same topic resulting in a N : M relation (e). On the network layer there are N · M connections, one for each (publisher, subscriber) tuple. The nodes can be distributed over any number of hosts within the ROS network [67].

CHAPTER 4: Results

After the testbed aircraft was built and instrumented, it was dynamically scaled. The results of this process can be found in section 4.1. Afterwards a flight test campaign was planned in order to provide the necessary data for system identification and record several flight maneuvers. The flight campaign was as follow: 1) there were a series of ground tests to prove out the systems and specially to test the safety of the electronics and the flight controller 2) two flight tests were conducted to collect the necessary data 3)the data was then processed and used in our system identification framework to get the aircraft model. Unfortunately, due to the project timeline, and weather limitations, only the first step of this flight campaign was completed. Steps two and three are left to be published at a later date.

4.1. Inertia Test

4.1.1. Angular Velocity Recorded

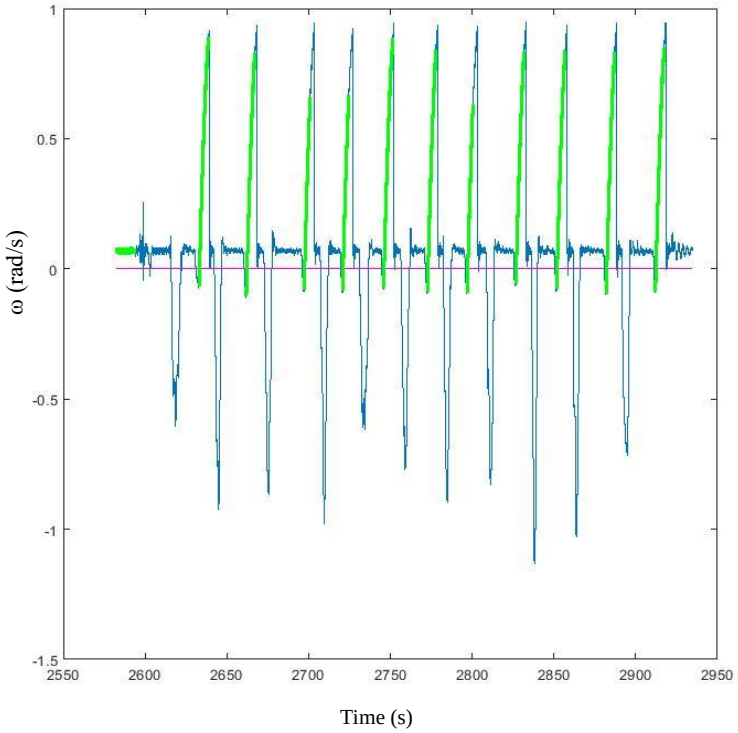

Figure 4.1 – Graph of angular velocity recorded for full aircraft roll (I_{xx}) with a hanging mass of 1lb.

4.1.2. Initial Moment of Inertia Measurement Results

Table 4.1 shows the initial scaled parameters. Notice the large percent errors.

Parameter	**Full-scale**	**Target scaled model**	**Actual scaled model**	**Percent error**
Cruise altitude (ft)	6562	1100	-	-
Max cruise speed (mph)	153	70.11	-	-
TOW (lbs)	2900	31.97	28.35	11.32%
I_{xx} (slugs-ft^2)	2306.5	1.121	0.732	34.75%
I_{yy} (slugs-ft^2)	1841.8	0.896	0.884	1.31%
I_{zz} (slugs-ft^2)	3331.0	1.620	1.523	5.96%

Table 4.1 – Summary of full-scale, target scaled model, and initial scaled model aircraft characteristics

4.1.3. Second Moment of Inertia Measurement Results

Table 4.2 shows the final scaled parameters after going through the optimization process outlined in section 2.2.3. Notice the smaller percent errors.

Parameter	**Full-scale**	**Target scaled model**	**Actual scaled model**	**Percent error**
Cruise altitude (ft)	6562	1100	-	-
Max cruise speed (mph)	153	70.11	-	-
TOW (lbs)	2900	31.97	32.67	2.19%
I_{xx} (slugs-ft^2)	2306.5	1.121	0.945	15.74%
I_{yy} (slugs-ft^2)	1841.8	0.896	0.994	11.03%
I_{zz} (slugs-ft^2)	3331.0	1.620	1.698	4.84%

Table 4.2 – Summary of full-scale, target scaled model, and final scaled model aircraft characteristics

4.2. Flight Test

4.2.1. Ground Testing

The main goal in the ground testing process was to build confidence in our system and to prove we can fly our aircraft safely. First, since the batteries for this project were from a previous project, they were visually inspected and cycled multiple times using a Tenergy balance charger to make sure they were able to keep charge and output the amperage necessary to fly our system safely for the amount of time needed.

Next, once the aircraft was fully instrumented and all the sensors were calibrated, it was put through a series of mock flights in an empty parking lot. We began by connecting and powering all the systems (avionics, data acquisition, VIO system) and taxing around the parking lot at very low throttle settings, while moving all the control surface and collecting data from all sensors. The batteries and DAQ system were checked to make sure data was being recorded and all systems still were good. Once verified, the aircraft was then put through a series of 50-75% throttle runs from 5-7 minutes to simulate flight powering conditions. The control surfaces were again being deflected during the test, including multiple multisine maneuvers at different gain values to simulate the system identification data collection. The batteries were checked again to make sure we had a large factor of safety before completely draining them. The data was postprocessed as if it were from a real test flight and checked for any anomalies. Through this rigorous testing procedure, we felt confident in our system and moved on to actual flight testing.

4.2.2. Flight test conditions

The flight-testing plan is to fly in the Eli Field in Monticello, IL which is at an altitude of 703 feet. The procedure for both tests flight would be to take the aircraft up to 400 feet while doing a couple of laps and making sure everything is functioning. Once the pilot was comfortable, the aircraft was trimmed to steady level flight and the multi-sine maneuvers would be run at low, medium, and high gains. The same procedure would be followed for testing normal flight maneuvers (e.g. doublet maneuver).

4.2.3. Aircraft flight data

The following results are from the ground testing in step one of the flight campaign. Figure 4.2-4.4 shows the multisine maneuver sent to each control surface that would be used for the system identification during data flight test. Figure 4.5-4.7 shows the recorded control surface deflection signals during a multisine maneuver during ground testing. Figure 4.8-4.10 show the IMU data during the ground test.

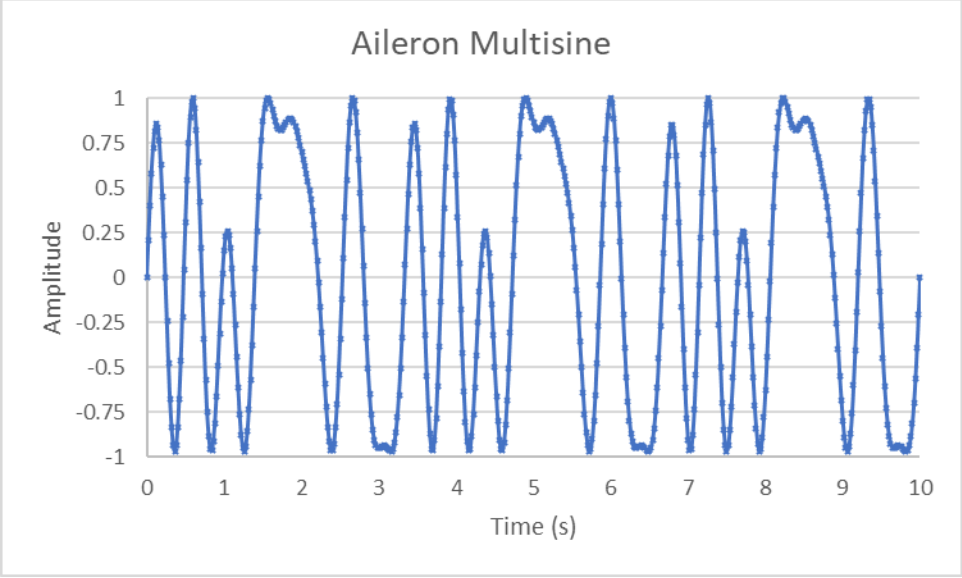

Figure 4.2 – Aileron multisine maneuver for the dynamically scaled aircraft testbed

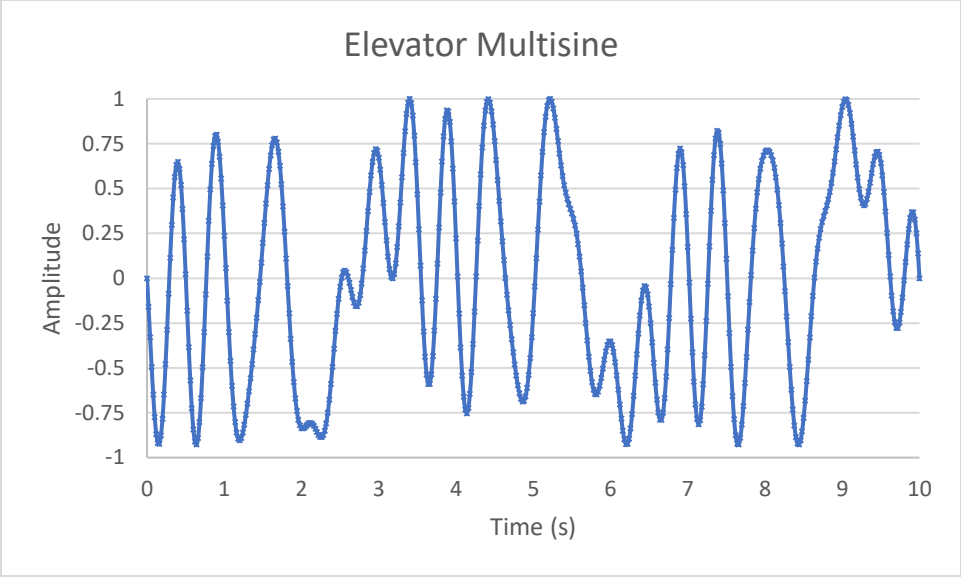

Figure 4.3 – Elevator multisine maneuver for the dynamically scaled aircraft testbed

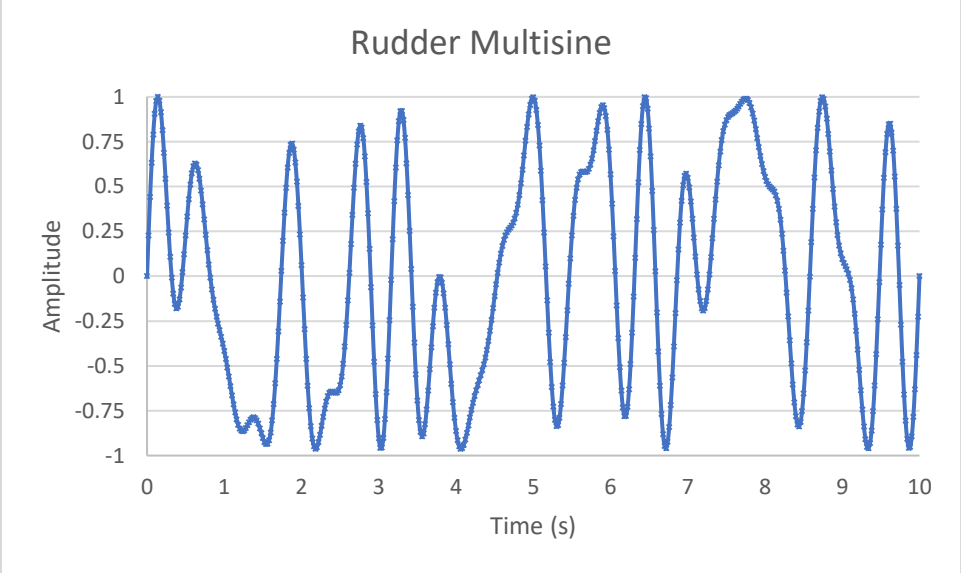

Figure 4.4 – Rudder multisine maneuver for the dynamically scaled aircraft testbed

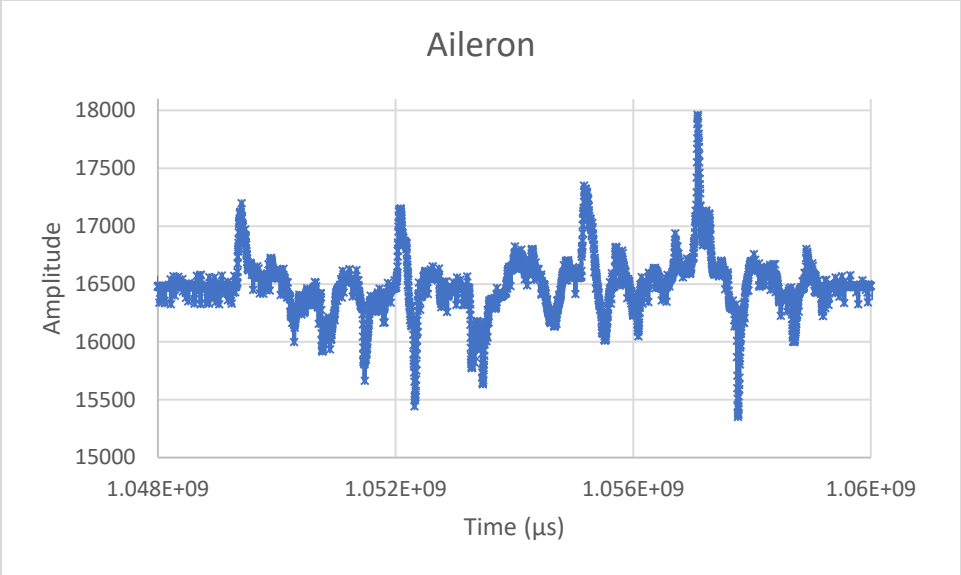

Figure 4.5 – Aileron control surface deflections during multisine maneuver

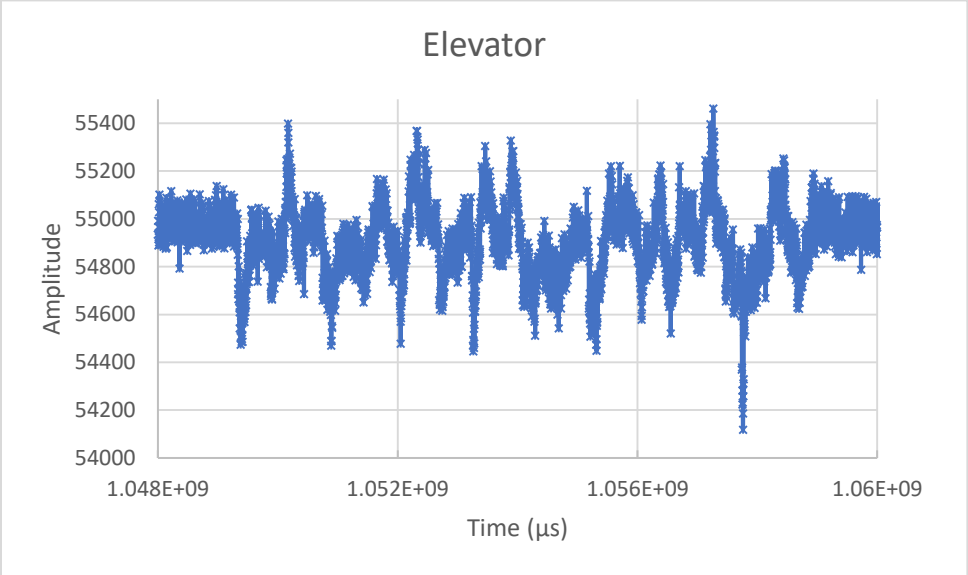

Figure 4.6 – Elevator control surface deflections during multisine maneuver

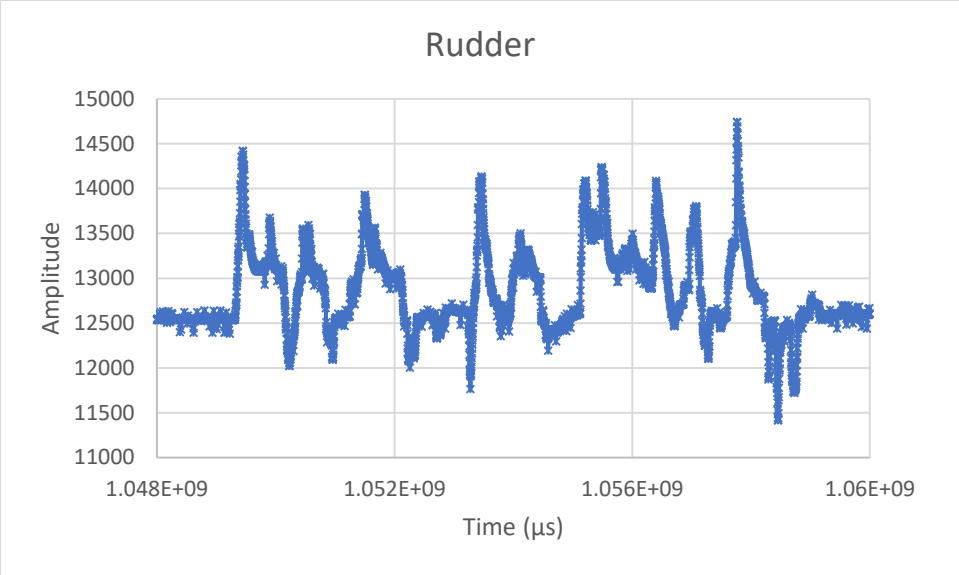

Figure 4.7 – Rudder control surface deflections during multisine maneuver

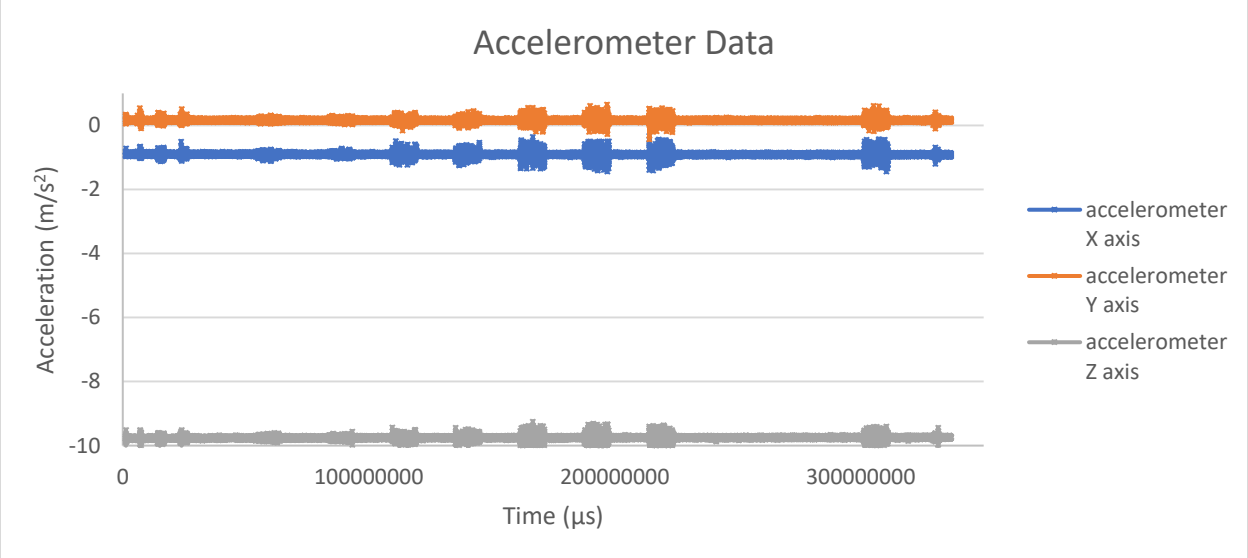

Figure 4.8 – Accelerometer data from Navio2 onboard IMU

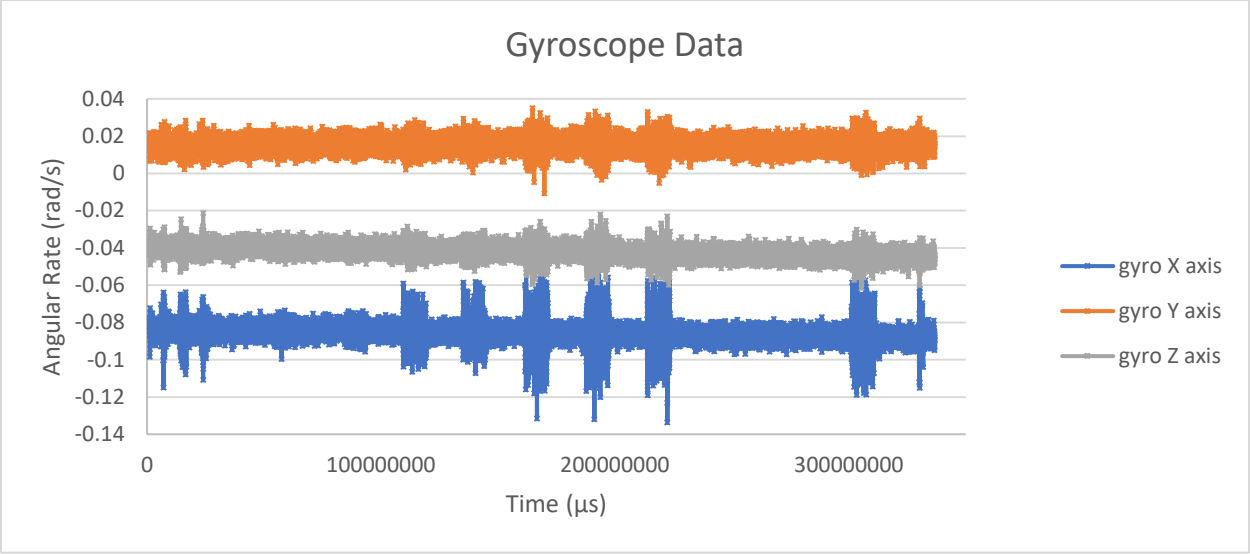

Figure 4.9 – Gyroscope data from Navio2 onboard IMU

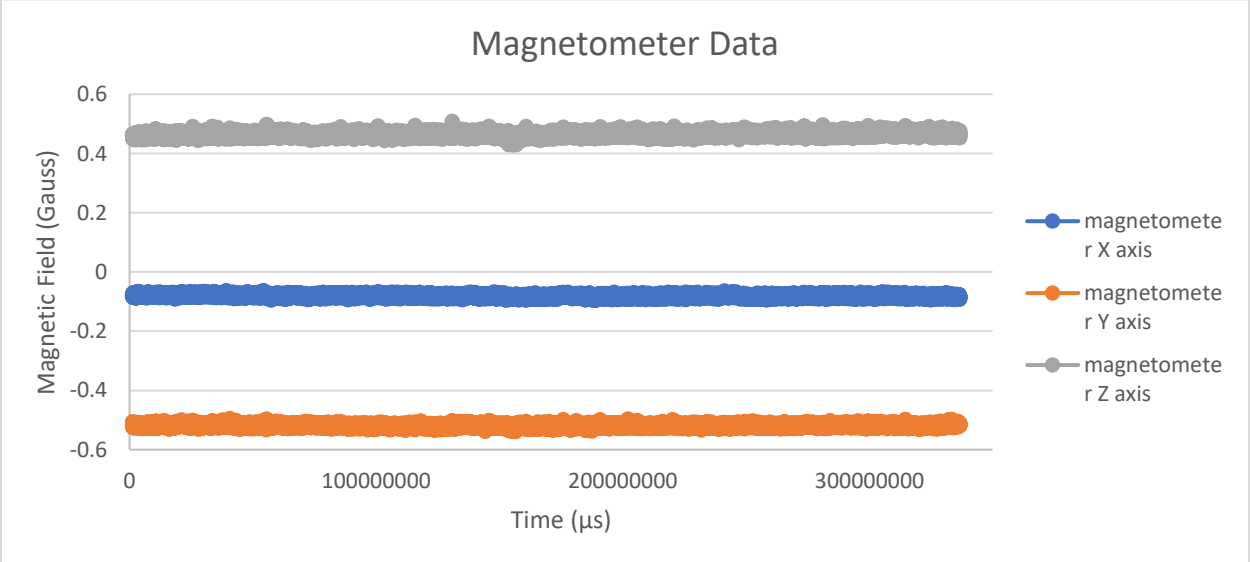

Figure 4.10 – Magnetometer data from Navio2 onboard IMU

4.2.4. VIO data

The VIO sensors (stereo camera and RTK GPS) were also collecting data during the ground test. This data was collected using ROS bags on the secondary Raspberry Pi. Once the data was removed from the Pi, it was converted to .csv and .jpg.

4.2.4.1. GPS (RTK vs normal)

Figure 4.11 shows the ground test PiksiMulti RTK GPS tracking and solution.

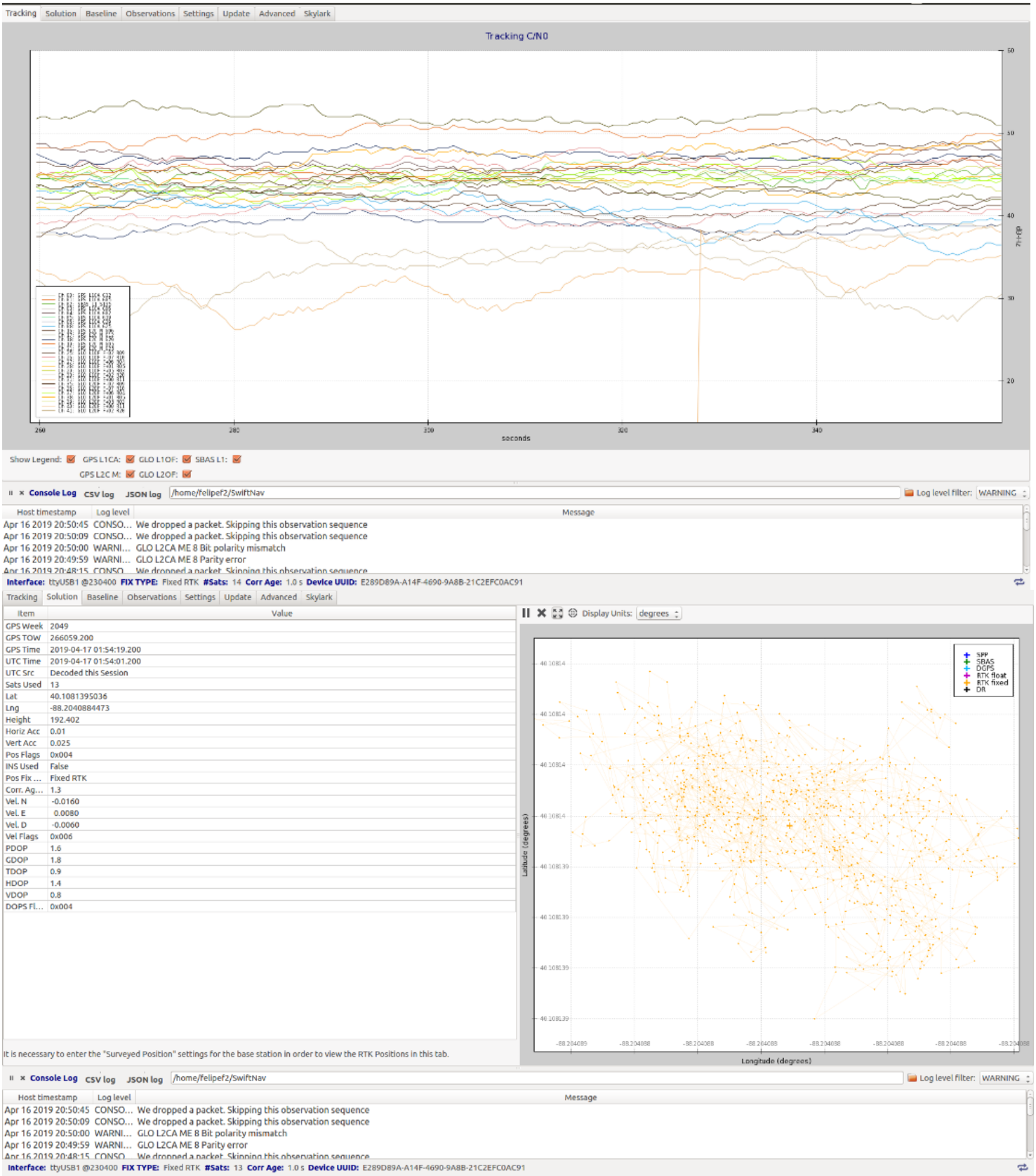

Figure 4.11 – Plot of RTK GPS solution along with satellite tracking during ground testing in the Swift Console

4.2.4.2. *Stereo camera data*

Figure 4.12 shows two sample images from the two cameras on our stereo system.

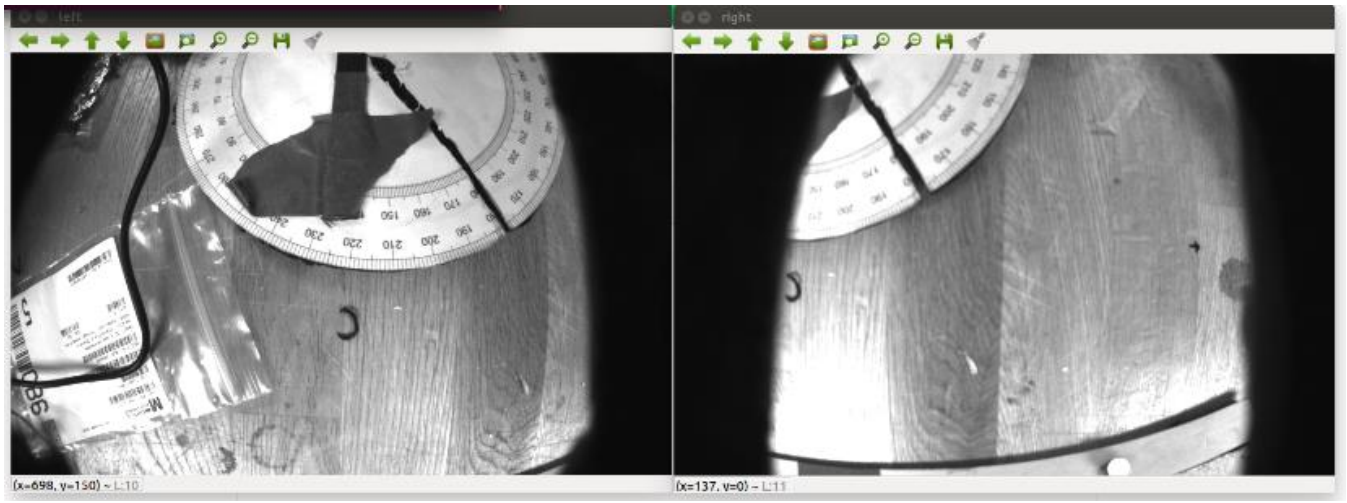

Figure 4.12 – Images from the L and R cameras in the onboard Loitor stereo camera while the aircraft was on a work bench

4.3. System Identification

The flight test data collected will be processed and used to derive the mathematical model of our baseline aircraft. The program used will be the NASA System IDentification Programs for AirCraft (SIDPAC) which is a collection of computer programs for aircraft system identification developed in MATLAB as m-file functions [68]. Because we do not have the actual inflight data, the detailed process will be forthcoming in a later publication.

CHAPTER 5: Conclusion

The following section provides a summary of the work in this thesis and some lessons learned to apply to future work.

5.1. Summary

The work in this thesis builds upon prior work in our group. The baseline aircraft was built, upgraded and instrumented. The main sensors added were hall effect sensors to measure control surface deflection, an alpha-beta probe, a pitot-static tube, and a Raspberry Pi/Navio2 flight controller with on board IMU, barometer, and GPS receiver. On the VIO side the sensors we had were a Raspberry Pi connected to a SwiftNav PiksiMulti RTK GPS receiver and Loitor stereo camera system with an onboard IMU.

After instrumentation the aircraft was dynamically scaled through an iterative optimizing process. In the end of this process the parameters of interest namely the cruise altitude, cruise speed, TOW, I_{xx}, I_{yy}, and I_{zz} were matched to the derived target values from the full-scale aircraft parameters.

Once the testbed aircraft was derived dynamically scaled, a flight test campaign was begun and will be completed at a later data in order to record sufficient data to provide in a dataset, perform system identification, and validate the derived mathematical model.

5.2. Lessons Learned

During the dynamically scaling process some lessons were learned that could help future work in this area. Although the testbed had to match a specific TOW which was fairly high (almost 6 lbs heavier than our stock airplane). After instrumentation the aircraft was a lot closer to the desired TOW, which did not leave a lot of weight to place in strategic locations to match the desired moment of inertias. Care should be taken to minimize the weight of the aircraft by integrating systems if possible.

In the optimization algorithm, the first approach taken for this problem used equality constraints instead of inequality constraints, but it quickly was found that there were no feasible solutions to this problem. To solve this, we could have added more locations to add mass, but speed and ease of manufacturing combined with the inability to exactly implement the solution on the physical aircraft caused this option to not be chosen. What was done in the end was to change the constraints to be inequalities so there would be some deviation from the desired parameters in the final solution but hopefully these deviations would be minimal. An overshoot weight value was used so the program could overshoot some parameters from the desired value, if it decreases the error in other parameters.

In the end the program was used to understand the different effects of adding mass in specific locations, and to make an informed choice of mass location that would result in a close parameter match. Ways to increase the accuracy of the solution from the problem would be to include more locations with more exact distance measurements. Another improvement would be to include non-point mass equations, where the mass is spread over a small distance range instead of one location.

5.3. Future Work

In future work, the flight campaign will be completed, including the initial system identification. This initial model would be a simple aircraft model, but in later stages the system identification process could be studied further, to see if other models could be derived that are better suited or more accurate for our DEP application. The work in this thesis will allow our group to now compare this baseline model to the DEP model and further research and learn the effects of DEP aircraft.

REFERENCES

1. M. Bangura and R. Mahony, 2017, "Thrust Control for Multirotor Aerial Vehicles," IEEE Transactions on Robotics, 33(2), pp. 390-405.
2. H. Alwi, C. Edwards, O. Stroosma, and J.A. Mulder, 2015, "Sliding-Mode Propulsion-Control Tests on a Motion Flight Simulator," Journal of Guidance, Control, and Dynamics, 38(4), pp. 671-684.
3. A.M. Stoll, J. Bevirt, M.D. Moore, W.J. Fredericks, and N.K. Borer, 2014, "Drag Reduction Through Distributed Electric Propulsion," AIAA Aviation Technology, Integration, and Operations Conference, Atlanta, Georgia, pp. 1-10.
4. J.A. Rosero, J.A. Ortega, E. Aldabas, and L.Romeral, 2007, "Moving Towards a More Electric Aircraft," IEEE Aerospace and Electronic Systems Magazine, 22(3), pp. 3-9.
5. K. Yamasaki, Y. Ichihara, H. Yasui, and S. Hiranuma, 2011, "A Thrust-only Flight-control System as a Backup for Loss of Primary Flight Controls," Mitsubishi Heavy Industries Technical Review, 48(4), pp. 1-6.
6. T. Tucker, "Touchdown - The Development of Propulsion Controlled Aircraft at NASA Dryden," NASA Monographs in Aerospace History, No. 16, Jan. 1999
7. J. Kaneshige and K. Gundy-Burlet, "Integrated Neural Flight and Propulsion Control System," AIAA paper 2001-4386, 2001
8. Felder, J.L., Kim, H., Brown, G. V., "Turboelectric Distributed Propulsion Engine Cycle Analysis for Hybrid-Wing-Body Aircraft," AIAA Paper 2009-1132, 2009.
9. Gohardani, A.S., Doulgeris, G., and Singh, R., "Challenges of Future Aircraft Propulsion: A Review of Distributed Propulsion Technology and its Potential Application for the All Electric Commercial Aircraft," Progress in Aerospace Sciences, Vol. 47, 2011, pp. 369-391.
10. Felder, J.L., Brown, G.V., Kim, H., and Chu, J., "Turboelectric Distributed Propulsion in a Hybrid Wing Body Aircraft," ISABE-2011-1340.
11. Schiltgen, B., Green, M.W., and Gibson, A.R., "Analysis of Terminal Area Operations and Short Field Performance of Hybrid Electric Distributed Propulsion," AIAA Paper 2013-4265.
12. Borer, Nicholas K., et al. "Design and Performance of the NASA SCEPTOR Distributed Electric Propulsion Flight Demonstrator." 16th AIAA Aviation Technology, Integration, and Operations Conference, Oct. 2016, doi:10.2514/6.2016-3920.
13. Moore, Mark D. "Distributed Electric Propulsion (DEP) Aircraft." NASA LANGLEY AERONAUTICS RESEARCH DIRECTORATE, aero.larc.nasa.gov/files/2012/11/Distributed-Electric-Propulsion-Aircraft.pdf.
14. Stoll, Alex M., et al. "Drag Reduction Through Distributed Electric Propulsion." 14th AIAA Aviation Technology, Integration, and Operations Conference, 2014, doi:10.2514/6.2014-2851.
15. Aerodynamic Effects of Wingtip-Mounted Propellers and Turbines
16. Clarke, Sean, et al. "NASA SCEPTOR Electric Concept Aircraft Power System: X-Plane Electric Propulsion System Design and Qualification for Crewed Flight Testing." 2016 IEEE Transportation Electrification Conference and Expo (ITEC), 2016, doi:10.1109/itec.2016.7520287.
17. Stoll, Alex M. "Comparison of CFD and Experimental Results of the LEAPTech Distributed Electric Propulsion Blown Wing." 15th AIAA Aviation Technology, Integration, and Operations Conference, 2015, doi:10.2514/6.2015-3188.
18. Chambers, R. Joseph. 2009. "*Modeling Flight - The Role of Dynamically Scaled Free-Flight Models in Support of NASA's Aerospace Programs*." [online]. Washington DC. US Government Printing Office. https://www.nasa.gov/pdf/483000main_ModelingFlight.pdf.

19. M. Bangura and R. Mahony, 2017, “Thrust Control for Multirotor Aerial Vehicles,” IEEE Transactions on Robotics, 33(2), pp. 390-405.
20. Abeywardena D and Dissanayake G (2015) Tightly-Coupled Model Aided Visual-Inertial Fusion for Quadrotor Micro Air Vehicles. Cham: Springer International Publishing, pp. 153–166.
21. Abeywardena D, Huang S, Barnes B, Dissanayake G and Kodagoda S (2016) Fast, on-board, model-aided visual-inertial odometry system for quadrotor micro aerial vehicles. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Stockholm, Sweden, pp. 1530–1537.
22. Abeywardena D, Kodagoda S, Dissanayake G and Munasinghe R (2013a) Improved State Estimation in Quadrotor MAVs: A Novel Drift-Free Velocity Estimator. IEEE Robotics Automation Magazine (2014): 32–39.
23. Abeywardena D, Wang Z, Dissanayake G, Waslander SL and Kodagoda S (2014) Model-aided state estimation for quadrotor micro air vehicles amidst wind disturbances. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 4813–4818.
24. Abeywardena D, Wang Z, Kodagoda S and Dissanayake G (2013b) Visual-inertial fusion for quadrotor Micro Air Vehicles with improved scale observability. In: IEEE International Conference on Robotics and Automation. pp. 3148–3153.
25. Abeywardena DMW (2015) Model-Aided State Estimation for Quadrotor Micro Aerial Vehicles. PhD Thesis, University of Technology Sydney, Sydney, Australia.
26. Abeywardena DMW and Munasinghe SR (2010) Performance analysis of a Kalman Filter based attitude estimator for a Quadrotor UAV. In: International Congress on Ultra Modern Telecommunications and Control Systems. pp. 466–471.
27. Allibert G, Mahony R and Bangura M (2016) Velocity aided attitude estimation for aerial robotic vehicles using latent rotation scaling. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 1538–1543.
28. Avram RC, Zhang X and Muse J (2017) Quadrotor sensor fault diagnosis with experimental results. Journal of Intelligent &Robotic Systems 86(1): 115–137.
29. Bangura M and Mahony R (2012) Nonlinear Dynamic Modeling for High Performance Control of a Quadrotor. In: Australasian Conference on Robotics and Automation. Wellington, New Zealand.
30. Bangura M and Mahony R (2017) Thrust control for multirotor aerial vehicles. IEEE Transactions on Robotics 33(2): 390–405.
31. Baranek R and Solc F (2014) Model-Based Attitude Estimation for Multicopters. Advances in Electrical and Electronic Engineering 12(5).
32. Bloesch M, Burri M, Omari S, Hutter M and Siegwart R (2017) Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback. The International Journal of Robotics Research 36(10): 1053–1072.
33. Bristeau PJ, Martin P, Salan E and Petit N (2009) The role of propeller aerodynamics in the model of a quadrotor UAV. In: European Control Conference (ECC). pp. 683–688.
34. Burri M, Nikolic J, Gohl P, Schneider T, Rehder J, Omari S, Achtelik MW and Siegwart R (2016) The EuRoC microaerial vehicle datasets. The International Journal of Robotics Research 35(10): 1157–1163.
35. Chang J, Cieslak J, Dvila J, Zhou J, Zolghadri A and Guo Z (2017) A Two-Step Approach for an Enhanced Quadrotor Attitude Estimation via IMU Data. IEEE Transactions on Control Systems Technology PP (99): 1–9.

36. Chang J, Cieslak J, Zolghadri A, Dvila J and Zhou J (2015) Design of sliding mode observers for quadrotor pitch/roll angle estimation via IMU measurements. In: Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS). pp. 393–400.
37. Dorobantu A, Murch A, Mettler B and Balas G (2013) System Identification for Small, Low-Cost, Fixed-Wing Unmanned Aircraft. Journal of Aircraft 50(4): 1117–1130.
38. Fogelberg J (2013) Navigation and Autonomous Control of a Hexacopter in Indoor Environments. Master's Thesis, Lund University, Lund, Sweden.
39. Furrer F, Burri M, Achtelik M and Siegwart R (2016) Rotors—A Modular Gazebo MAV Simulator Framework. Cham: Springer International Publishing. ISBN 978-3-319-26054-9, pp. 595–625.
40. Gui J, Gu D, Wang S and Hu H (2015) A review of visual inertial odometry from filtering and optimization perspectives. Advanced Robotics29(20): 1289–1301.
41. Hanley D and Bretl T (2016) An Improved Model-Based Observer for Inertial Navigation for Quadrotors with Low Cost IMUs. In: AIAA Guidance, Navigation, and Control Conference. San Diego, CA: AIAA SciTech, pp. 1–14.
42. Leishman R, Macdonald J, Quebe S, Ferrin J, Beard R and McLainT (2011) Utilizing an improved rotorcraft dynamic model instate estimation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 5173–5178.
43. Leishman RC, Macdonald JC, Beard RW and McLain TW (2014a)Quadrotors and Accelerometers: State Estimation with an Improved Dynamic Model. IEEE Control Systems34(1): 28–41.
44. Leishman RC and McLain TW (2015) Multiplicative Extended Kalman Filter for Relative Rotorcraft Navigation. Journal of Aerospace Information Systems12(12): 728–744.
45. Leishman RC, McLain TW and Beard RW (2014b) Relative navigation approach for vision-based aerial GPS-denied navigation. Journal of Intelligent & Robotic Systems74(1):97–111.
46. Leutenegger S, Lynen S, Bosse M, Siegwart R and Furgale P (2015) Keyframe-based visual inertial odometry using nonlinear optimization. The International Journal of RoboticsResearch34(3): 314–334.
47. Lin Y, Gao F, Qin T, Gao W, Liu T, Wu W, Yang Z and Shen S(2017) Autonomous aerial navigation using monocular visual-inertial fusion. Journal of Field Robotics: 1–29.
48. Lynen S, Achtelik M, Weiss S, Chli M and Siegwart R (2013) A robust and modular multi-sensor fusion approach applied to MAV navigation. In: Proc. of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS).
49. Lyu P, Lai J, Liu HH, Liu J and Chen W (2017) A Model-Aided Optical Flow/Inertial Sensor Fusion Method for a Quadrotor. Journal of Navigation70(2): 325–341.
50. Macdonald J, Leishman R, Beard R and McLain T (2014) Analysis of an improved imu-based observer for multirotor helicopters. Journal of Intelligent & Robotic Systems74(3): 1049–1061.
51. Majdik AL, Till C and Scaramuzza D (2017) The Zurich urban micro aerial vehicle dataset. The International Journal of Robotics Research36(3): 269–273.
52. Martin P and Salan E (2010) The true role of accelerometer feedback in quadrotor control. In: IEEE International Conference on Robotics and Automation. pp. 1623–1629.
53. Martin P and Sarras I (2016) A semi-global model-based state observer for the quadrotor using only inertial measurements. In: IEEE Conference on Decision and Control (CDC). pp.7123–7128.
54. Padfield G (2007) Helicopter Flight Dynamics. AIAA Education Series, second edition. Reston, VA: American Institute of Aeronautics and Astronautics.

55. Sa I and Corke P (2012) System identification, estimation and control for a cost effective open-source quadcopter. In: IEEE International Conference on Robotics and Automation. pp.2202–2209.
56. Santoso F, Garratt MA and Anavatti SG (2017) Visual-inertial navigation systems for aerial robotics: Sensor fusion and technology. IEEE Transactions on Automation Science and Engineering 14(1): 260–275.
57. Sanz R, Rdenas L, Garca P and Castillo P (2014) Improving attitude estimation using inertial sensors for quadrotor control systems. In: International Conference on Unmanned Aircraft Systems (ICUAS). pp. 895–901.
58. Sikkel LNC, de Croon GCHE, Wagter CD and Chu QP (2016)A novel online model-based wind estimation approach for quadrotor micro air vehicles using low cost mems imus. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2141–2146.
59. Tischler MB and Remple RK (2012)Aircraft and Rotorcraft System Identification. AIAA Education Series, second edition. Reston, VA: American Institute of Aeronautics and Astronautics.
60. Waslander SL and Wang C (2009) Wind Disturbance Estimation and Rejection for Quadrotor Position Control. In: AIAA Infotech@Aerospace Conference. Seattle, WA.
61. Pieper, Kyle Chapman, *Design, development and evaluation of a distributed electric propulsion testbed aircraft.* 2018. Master's Thesis, University of Illinois at Urbana-Champaign. http://hdl.handle.net/2142/101073.
62. "Cirrus SR22T 30cc ARF 96.8"" Horizon Hobby https://www.horizonhobby.com/product/airplanes/airplanes-14501--1/giant-scale/cirrus-sr22t-30cc-arf-han5020
63. Dantsker, O. D., et al. "GA-USTAR Phase 1: Development and Flight Testing of the Baseline Upset and Stall Research Aircraft." 35th AIAA Applied Aerodynamics Conference, 2017, p. 24p. EBSCOhost, search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-85023603096&site=eds-live&scope=site. Accessed Dec. 2018.
64. "Piksi Multi Product Summary." Swift Navigation. Version 2.2 February 7, 2019. https://www.swiftnav.com/piksi-multi-product-summary.
65. "ethz_piksi_ros." ETH Zurich – Autonomous Systems Lab. GitHub Repository, https://github.com/ethz-asl/ethz_piksi_ros/tree/master/piksi_multi_rtk_ros.
66. "vi_sensor_sdk. Loitor VIS. GitHub Repository, https://github.com/loitor-vis/vi_sensor_sdk.
67. Koubaa, Anis. Robot Operating System (ROS) The Complete Reference (Volume 1). New York: Springer, 2016.
68. Morelli, E. A. "System Identification Programs for Aircraft (SIDPAC)." AIAA Atmospheric Flight Mechanics Conference and Exhibit. EBSCOhost, search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-43049157367&site=eds-live&scope=site. Accessed Dec. 2018.
69. Kim, H., Perry, A., and Ansell, P., "A Review of Distributed Electric Propulsion Concepts for Air Vehicle Technology," AIAA EATS Conference, 11-12 July 2018.
70. "Fmincon Interior Point Algorithm." Constrained Nonlinear Optimization Algorithms - MATLAB & Simulink, www.mathworks.com/help/optim/ug/constrained-nonlinear-optimization-algorithms.html#brnpd5f.
71. H. Alwi, C. Edwards, O. Stroosma, and J.A. Mulder, 2015, "Sliding-Mode Propulsion-Control Tests on a Motion Flight Simulator," Journal of Guidance, Control, and Dynamics, 38(4), pp. 671-684.

72. A.M. Stoll, J. Bevirt, M.D. Moore, W.J. Fredericks, and N.K. Borer, 2014, “Drag Reduction Through Distributed Electric Propulsion,” AIAA Aviation Technology, Integration, and Operations Conference, Atlanta, Georgia, pp. 1-10.
73. J.A. Rosero, J.A. Ortega, E. Aldabas, and L.Romeral, 2007, “Moving Towards a More Electric Aircraft,” IEEE Aerospace and Electronic Systems Magazine, 22(3), pp. 3-9.
74. K. Yamasaki, Y. Ichihara, H. Yasui, and S. Hiranuma, 2011, “A Thrust-only Flight-control System as a Backup for Loss of Primary Flight Controls,” Mitsubishi Heavy Industries Technical Review, 48(4), pp. 1-6.
75. T. Tucker, “Touchdown - The Development of Propulsion Controlled Aircraft at NASA Dryden,” NASA Monographs in Aerospace History, No. 16, Jan. 1999
76. J. Kaneshige and K. Gundy-Burlet, “Integrated Neural Flight and Propulsion Control System,” AIAA paper 2001-4386, 2001
77. Felder, J.L., Kim, H., Brown, G. V., “Turboelectric Distributed Propulsion Engine Cycle Analysis for Hybrid-Wing-Body Aircraft,” AIAA Paper 2009-1132, 2009.
78. Gohardani, A.S., Doulgeris, G., and Singh, R., “Challenges of Future Aircraft Propulsion: A Review of Distributed Propulsion Technology and its Potential Application for the All Electric Commercial Aircraft,” Progress in Aerospace Sciences, Vol. 47, 2011, pp. 369-391.
79. Felder, J.L., Brown, G.V., Kim, H., and Chu, J., “Turboelectric Distributed Propulsion in a Hybrid Wing Body Aircraft,” ISABE-2011-1340.
80. Schiltgen, B., Green, M.W., and Gibson, A.R., “Analysis of Terminal Area Operations and Short Field Performance of Hybrid Electric Distributed Propulsion,” AIAA Paper 2013-4265.
81. Borer, Nicholas K., et al. “Design and Performance of the NASA SCEPTOR Distributed Electric Propulsion Flight Demonstrator.” 16th AIAA Aviation Technology, Integration, and Operations Conference, Oct. 2016, doi:10.2514/6.2016-3920.
82. Moore, Mark D. “Distributed Electric Propulsion (DEP) Aircraft.” NASA LANGLEY AERONAUTICS RESEARCH DIRECTORATE, aero.larc.nasa.gov/files/2012/11/Distributed-Electric-Propulsion-Aircraft.pdf.
83. Stoll, Alex M., et al. “Drag Reduction Through Distributed Electric Propulsion.” 14th AIAA Aviation Technology, Integration, and Operations Conference, 2014, doi:10.2514/6.2014-2851.
84. Aerodynamic Effects of Wingtip-Mounted Propellers and Turbines
85. Clarke, Sean, et al. “NASA SCEPTOR Electric Concept Aircraft Power System: X-Plane Electric Propulsion System Design and Qualification for Crewed Flight Testing.” 2016 IEEE Transportation Electrification Conference and Expo (ITEC), 2016, doi:10.1109/itec.2016.7520287.
86. Stoll, Alex M. “Comparison of CFD and Experimental Results of the LEAPTech Distributed Electric Propulsion Blown Wing.” 15th AIAA Aviation Technology, Integration, and Operations Conference, 2015, doi:10.2514/6.2015-3188.