ACTIVE CONTROL OF A MORPHING WING AIRCRAFT AND FAILURE ANALYSIS FOR SYSTEM RELIABILTIY

THESIS

Madison J. Montgomery, Captain, USAF

AFIT-ENG-MS-19-M-045

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-19-M-045

ACTIVE CONTROL OF A MORPHING WING AIRCRAFT AND FAILURE ANALYSIS FOR SYSTEM RELIABILTIY

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Madison J. Montgomery, BS

Captain, USAF

March 2019

DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENG-MS-19-M-045

ACTIVE CONTROL OF A MORPHING WING AIRCRAFT AND FAILURE ANALYSIS FOR SYSTEM RELIABILTIY

Madison J. Montgomery, BS

Captain, USAF

Committee Membership:

Dr. Robert Leishman
Chair

Dr. David Jacques
Member

Dr. James Joo
Member

AFIT-ENG-MS-19-M-045

Abstract

Most modern aircraft have rigid wings and control surfaces that are optimized for a single phase of flight. A wing that can change its shape based on flight conditions would improve performance but a lack of adequate materials and complicated control requirements made it largely infeasible. With the advent of new materials and improvements in flight control technology, increasing research and development (R&D) has been undertaken to find a morphing alternative to standard wings and control surfaces. A morphing wing aircraft could benefit from a longer range, decrease in fuel consumption, lower wing strength and weight, lower radar signature, and reduced noise.

The Air Force Research Laboratories Advanced Structural Concepts Team (AFRL/RQVS) along with the Utah State University (USU) Aerodynamics Lab have been working to develop a camber morphing aircraft that is optimized for different flight conditions. They have constructed a wing capable of changing its shape in six different locations spanwise along its wing surface and have a goal to flight test the wing on an unmanned aerial vehicle (UAV).

This thesis presents the research on the design, construction, and implementation of a control system capable of controlling the AFRL morphing wing using a set of optimized equations developed by USU. A method of using system identification on flight test data was developed to quantify the capabilities of the wing. A flight test plan was

created to safely test the prototype aircraft and collect the necessary data and a flight test of the morphing wing was performed.

The aircraft crashed shortly after takeoff due to a malfunction of the morphing wings surface structure and the control system shutting down prior to takeoff. There was not enough flight test data to characterize the morphing wing nor to verify the effectiveness of the control system. A failure mode analysis study and follow-on testing was performed on the control system to determine the cause of the flight test failure and improve the reliability of the control system. After the study, improvements were made to the controller to decrease the chance of failure and reduce the effects of any remaining potential failures.

Acknowledgments

I would like to express my sincere appreciation to my faculty advisor, Dr. Rob Leishman, for his guidance and support throughout the course of this thesis effort. His insight and experience were invaluable. More thanks go to the ANT UAV Lab technicians who were critical to building the control system and supporting flying operations. I would also like to thank my mom for editing this thesis and all my family for their support and encouragement.

Madison J. Montgomery

Table of Contents

Page

Abstract iv
Table of Contents vii
List of Figures ix
List of Tables xi
Chapter I: Introduction 1
1.1 Overview 1
1.2 Relevant Research 2
1.3 Research Objectives 4
1.4 Limitations and Assumptions 5
Chapter II: Literature Review 6
2.1 Morphing Wing Technology 6
2.2 Characterization of Aircraft Through Flight Data 13
2.3 System Reliability Design 18
Chapter III: Methodology 21
3.1 Background 21
3.2 System Design: 28
Chapter IV: Results and Analysis 52
4.1 Integration and Flight Testing 52
4.2 System Reliability Analysis and Testing 59
Chapter V: Conclusions 84
5.1 Failure Mode Analysis 85
5.2 Future Work 87

Appendix 89
Bibliography 90

List of Figures

Page

Figure 1: Types of morphing wings. 10

Figure 2: AFRL Variable Camber Compliant Wing prototype. 11

Figure 3: Depiction of piral divergence. 17

Figure 4: Depiction of an oscillatory Dutch roll mode. 18

Figure 5: Side view of morphing wing design 22

Figure 6: View of morphing wing from the rear 22

Figure 7: Base aircraft hooked up to standard wings. 25

Figure 8: Morphing wing control setup 27

Figure 9: Circuit diagram for the morphing wing control system. 31

Figure 10: Result of using a low-pass filter on a PWM signal. 33

Figure 11: Simple Low-Pass Filter Design 33

Figure 12: Passing a 50Hz servo PWM signal through low-pass filters 35

Figure 13: Passing a 500Hz PWM signal through two different low-pass filters 36

Figure 14: Controller mounted inside of morphing wing fuselage. 37

Figure 15: Simple comparison of the three different flight modes 39

Figure 16: Control system code flowchart. 41

Figure 17: Power supply setup 43

Figure 18: Morphing wing communication setup. 45

Figure 19: Morphing wing aircraft prior to testing 47

Figure 20: Aftermath of the morphing wing flight test crash. 53

Figure 21: Demonstration of the right wing in three different flight modes. 54
Figure 22: Flight path of failed flight test. 56
Figure 23: Left wing of aircraft prior to takeoff. 57
Figure 24: Left wing of aircraft experiencing wing expansion after takeoff. 57
Figure 25: Pixhawk and Raspberry Pi aileron wing deflection comparison. 58
Figure 26: Failure mode analysis breakdown 60
Figure 27: Filtered x-axis acceleration data in time (left) and frequency domain (right). Comment in here about the dots and their significance… 64
Figure 28: Filtered y-axis acceleration data in time (left) and frequency domain (right). 64
Figure 29: Filtered z-axis acceleration data in time (left) and frequency domain (right). 65
Figure 30: Vibration table testing of controller in the y-axis... 67
Figure 31: Vibration table testing of the controller in the z-axis. 68
Figure 32: Morphing wing controller endurance test setup. 70
Figure 33: Motion capture test setup and coordinate frame for x, y and z motion. 74
Figure 34: Wing movement response time comparison using direct RC connection 75
Figure 35: Standard wing movement compared with RC transmitter stick movement with Pixhawk feedthrough. 76
Figure 36: Morphing wing response time with Pixhawk feedthrough. At the 45 second mark the control gain is switched from 30% to 60%. 76
Figure 37: Wing position error and standard deviation comparison of standard, morphing, and morphing wing with the low-pass filter coefficient halved 79

List of Tables

Page

Table 1: Flight Test Plan 47

Table 2: Flight Test Maneuvers 50

Table 3: Vibration Table Frequencies and Acceleration 65

Table 4: Approximate Delay from Pilot Input to Wing Movement.................................. 77

Table 5: Control System Setup Manual 83

ACTIVE CONTROL OF A MORPHING WING AIRCRAFT AND FAILURE ANALYSIS FOR SYSTEM RELIABILTIY

Chapter I: Introduction

1.1 Overview

This document presents the development and flight testing of a control system for a morphing wing on an unmanned aerial vehicle (UAV). The controller was required to read pilot input commands and flight state information and output a command in degrees of deflection to a set of six servos controlling the morphing wing. The morphing wing is actively controlled by changing its shape based on flight conditions. A Pixhawk 2 autopilot system provided the flight data and a RC receiver provided the pilot input commands. A Raspberry Pi was used to handle the input data, and using algorithms derived from USU, output PWM signals to the servo controllers in the wing.

A flight test plan was developed and implemented to determine the effectiveness of the morphing wing and the controller. Tragically, the aircraft crashed shortly after takeoff due a pair of failures: a structural failure of the morphing wing and a control system failure. Not enough flight data was recorded to characterize the morphing wing and as a result, the later stages of research were re-focused on increasing the reliability of the system using failure mode analysis. Improving reliability included performing tests to verify the ability of the system to withstand the rigors of flight testing.

This document is organized as follows: the remainder of Chapter 1 provides a short overview of the project including the required research, the problem background, and the objectives of this research. In addition, the limitations and assumptions are described.

Chapter 2 discusses all the research literature used to complete this project, as well as information required to understand the following chapters. Chapter 3 explains the methods used to design the aircraft control system and the flight test plan. Chapter 4 details the results of the flight test and the failure mode analysis performed to increase the reliability of the system. Finally, Chapter 5 discusses the conclusion of this research and provides insights into possible future research to improve upon what has been completed.

1.2 Relevant Research

Morphing wing technology research and development has been on the rise due to advancements in materials and the potential benefits that morphing wings could provide. The main benefit of morphing wings is their ability to adapt to certain flying conditions to improve efficiency[1]. Morphing wings may also provide the benefit of reducing structural loads on an aircraft and could replace traditional control surfaces including rudders, flaps, and ailerons[2][3]. There are challenges with morphing wings, primarily associated with the increase in complexity. Current research is working to overcome these issues [4][1].

There are many different types of morphing wings that could improve flight performance. Some of these designs have already been implemented on production aircraft, such as the F-14 [1]. This research focuses on a style of morphing called *camber morphing*, which involves changing the curve, or camber, of the wing surface. Several studies have been performed on the development of camber morphing aircraft and have shown theoretical and analytical improvements over standard wing designs [5][6][7]. Unlike most types of morphing, camber morphing can potentially replace traditional aircraft controls such as ailerons, flaps and rudder, all with only the revised wing control surfaces [8].

AFRL has performed research and development on the structure and mechanics of camber morphing wings to produce the morphing wing used in this study. AFRL has created several prototype wings and performed simulations and wind tunnel testing to prove the concept and verify the efficiency benefits. In AFRLs design, the internal structure can adapt to change the camber of the wing at six different locations along the wingspan, creating a twisting surface for roll control [9][10][11].

A key aspect of morphing wing development is determining how to optimize the shape for different flight conditions. Members of USU Aero Lab have performed research involving lifting-line theory to show what the optimal camber at different points of the wing should be to obtain the minimum induced drag [12][13]. Also, USU has developed methods of optimizing the camber during a rolling motion [3].

Another area of research required for this study was the ability to characterize a complex aircraft system using flight test data. A method of aircraft characterization was developed using system identification to fill in the state space matrices of the linearized lateral equations of motion [14][15][16]. With an estimated state space equation of the lateral equation of motion, one can determine some of the aspects of an aircraft flight performance and stability by analyzing the eigenvalues of the matrices [17].

Flight testing a prototype aircraft involves a lot of risk and therefore it can be beneficial to take a systems engineering approach to the aircrafts design. A technique called failure mode effects and criticality analysis (FMEC) allows one to look at the potential causes of failures, the likelihood of those failures and the severity of the failures to improve the reliability of a system [18].

1.3 Research Objectives

The main goal of this research was to design and implement a controller for a morphing wing prototype capable of performing a flight test and to then determine the capabilities of the morphing wing and the control system utilizing that flight test data. After the flight test crash, a comprehensive failure analysis was performed with the objective of improving reliability. The research goals are designing a morphing wing flight controller, verifying the morphing wing capabilities, and improving the reliability. The goals and requirements are:

- Design flight controller
    - Input pilot commands and aircraft state
    - Perform calculations based on input data
    - Output control signal to six points of morphing wing
    - Ability to access program remotely
    - Fit inside small UAV
- Verify the capabilities of the morphing wing design
    - Develop method of determining aircrafts capabilities from flight test data
    - Analyze data from standard wing base aircraft for comparison
    - Develop a flight test plan to obtain the necessary data from flight test
    - Flight test morphing wing
- Improve system reliability
    - Analyze flight crash data
    - Perform failure mode analysis
    - Test and improve controller design

1.4 Limitations and Assumptions

The control system relies upon mostly commercial-off-the-shelf (COTS) equipment and in several cases the devices are not used in the way they were designed. Some flight stability problems could be the result of the COTS equipment instead of the wing design or aerodynamic equations. The linear lateral equations derived from state space equations are only valid around slight perturbations to steady equilibrium flight conditions. The equations cannot show how the system will respond to large aerobatic maneuvers. The flight testing used to produce the equations can contain maneuvers that fall outside of the linear range and lead to inconsistent data.

Flight test data used for the system identification is from the Pixhawk autopilot system. The accuracy of this data is not perfect and will not provide the same level of accuracy that one might find in a wind tunnel test. One method of determining the effectiveness of the morphing wing is to compare flight test data of the same base aircraft and to switch between the standard wing and the morphing wing. Although the flight test plan for both aircraft is similar, one cannot mimic the exact parameters of the flight tests. This may cause inconsistencies within the comparison data.

An increase in efficiency is one of the main benefits to a morphing wing. This study will not focus on proving efficiency gains. We did compare the characteristics of the base aircraft with standard wings and the morphing wing; however, efficiency cannot be accurately drawn out of the data collected. One could compare the amount of energy each aircraft used from the batteries but it would be very difficult to perform the exact same flight conditions and flight paths for an accurate comparison.

An important aspect to the morphing wing controller design was to keep the delay between pilot input and wing deflection at a minimum. Many UAVs are designed to not use a pilot for their operations. In this case some of the control delay can be mitigated through software. However, even when being controlled by an autopilot keeping the delay low increases the ability of the aircraft to fly.

Chapter II: Literature Review

The following chapter consist of the background research information used for the morphing wing control system design and testing. It is organized as follows: Section 2.1 presents morphing wing technology and concepts including benefits, developments, and the aerodynamics. Section 2.2 discusses how to characterize a system through system identification and how one uses the results to help determine the overall stability of an aircraft. Section 2.3 discusses how failure analysis is used to increase the reliability of a system.

2.1 Morphing Wing Technology

The use of morphing wings to control an aircraft is not a new concept and has been used ever since the first aircraft flown by the Wright-Brothers. The Wright Flyer used twisting wings for roll control[19]. Morphing wings enable an aircraft to adapt to flight conditions for better performance over standard wing surfaces. This section presents some of the different types of morphing wings as well as some of the research being done. Research by AFRL and USU has led to building the morphing wing prototype and the aerodynamic equations used for this study. For a more in-depth review of general morphing aircraft see [1].

2.1.1 Benefits of Morphing

The reason for the research and development of morphing wings is due to the many potential benefits that they provide. Standard wings are optimized for a small range of flight conditions as they are rigid structures. The idea of morphing wings is to change the shape of the wing based on flight conditions for optimal lift and drag characteristics. All morphing wing designs presented in this thesis are based on this principal. While flaps do provide some improvements to the aerodynamics during events such as takeoff, they are inefficient due to their single hinge movement which creates gaps and irregularities [1].

Another potential benefit of morphing wings is the ability to manipulate pressure distribution along the wingspan. Different flight conditions and maneuvers can produce a wide range of forces on an aircraft. A standard wing must be able to withstand all the stresses throughout flight at every point along the wing. A morphing wing could be optimized for maneuvers to create less stress on the aircraft support structure. Lower stress requirements could lead to a decrease in the overall structural weight of the aircraft and a decrease in weight increases efficiency[2].

Some morphing wing designs can replace ailerons for roll control. Aircraft ailerons are inefficient due to the less than optimal shape and gaps created in the surface during roll and steady flight. The Wright-brother’s original aircraft used a twisting of the wing surface to produce a rolling motion. A morphing wing can also create this twisting moment, increasing the efficiency of the aircraft[3].

Morphing wings can also benefit from a reduction or elimination of adverse yaw. Adverse yaw is when an unwanted yaw moment is produced from standard aileron surfaces when the aileron deflections produce uneven drag. The yaw moment is in the opposite direction from what one would desire when rolling. Morphing wings can eliminate this adverse effect and provide a pure rolling motion. Also, the morphing wing can be used to purposely produce a yaw moment to replace some functions of the rudder, permitting the use of a smaller vertical stabilizer or potentially removing the requirement entirely[2].

Another capability of morphing wings is the ability to replace the flaps of an aircraft. Several morphing wing designs change the camber of the wing by flexing the wing surface. This can perform the same function as flaps while being more efficient. Altogether the morphing wing could replace the flaps, ailerons, and rudder of an aircraft, all of which can reduce drag and improve range[2][3].

Most of the benefits to morphing wings are ones which increase the aircrafts efficiency. A more efficient aircraft will consume less fuel and have an increased range. Less fuel consumption can create significant cost savings to aircraft operators. The increase in range is critical for missions requiring long flight times, such as the Northrop Grumman RQ-4 Global Hawk[1].

Another potential benefit to morphing wing which is mostly relevant to military application is the reduction of the aircrafts radar cross section. A morphing wing can eliminate gaps in the surface from traditional control surfaces and, with yaw control, eliminate the need of a vertical stabilizer. Both improvements reduce an aircraft’s radar reflection. Also, a morphing wing can help reduce the noise caused by the control surfaces[8].

Although morphing wings can provide many potential benefits, there are also many difficulties that must be overcome. One key issue is that many morphing wing designs add weight to an aircraft that eliminate any efficiency gains. Also, the mechanisms required for movement may use up space in the wing that many aircraft require for fuel. There are also some reliability concerns with the increased number of moving parts and complexity[4][1].

2.1.2 Morphing Wing Types

The “morphing wing” is a very broad term which characterizes many different types of wings. A morphing wing can mean any wing which alters its shape during flight. One example of a morphing wing is what is known as variable swept wing. In this design, the aircrafts wings will swing back and forth based on flight conditions. The F-14 tomcat is an example of the swept design as shown in Figure 1 [1]. This morphing design has probably seen the most success with implementation on several production aircraft.

Another type of morphing is folding wings, which were tested on the XB-70 prototype bomber [1]. Several other recent studies have also examined this type[20][21][22].

Yet another style of morphing is often referred to as telescopic wings, where the wings extend outward. This is shown on the right-hand side of Figure 1. This design has been researched and shown potential but has seen little development outside of preliminary R&D[23]. All three of these wing styles where designed to optimize the aircrafts performance during the different stages of flight by effecting the lift and drag characteristics.

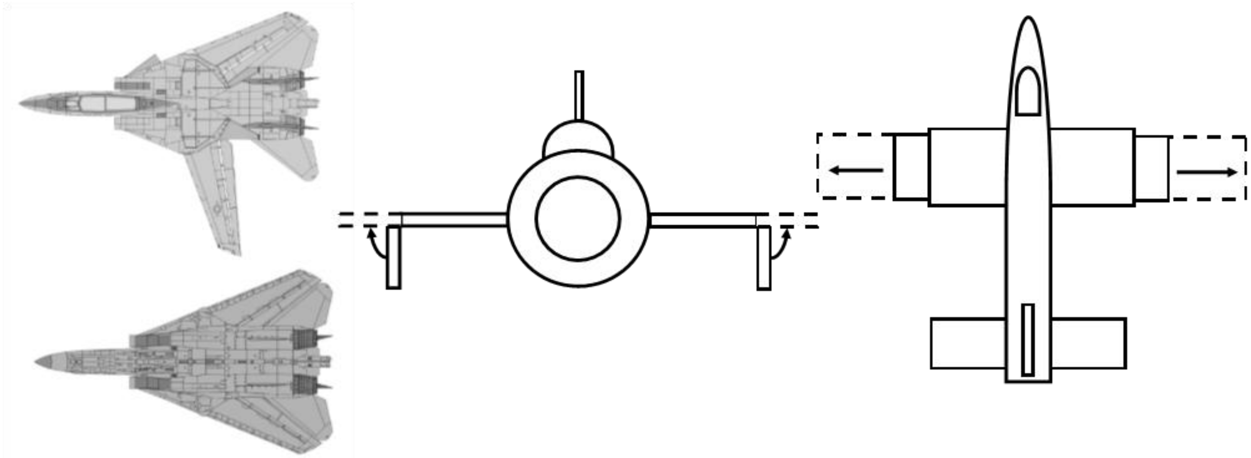

Figure 1: An example of sweepback wing morphing of the F-14 (left), folding wing morphing (middle) and telescopic morphing (right).

2.1.3 Camber Morphing Development

With the development of better materials, another style of morphing that has seen more recent research is camber morphing. Camber morphing creates a similar effect as flaps and ailerons on standard aircraft designs. A morphing of camber can change the lift to drag ratio, pitching moment, rolling moment and stall characteristics of an aircraft[8]. One of the first examples of camber morphing was an effort funded by the Air Force to add morphing capability to the F-111A called the Mission Adaptive Wing (MAW) [5]. This study showed that the morphing wing can have many of the potential benefits outlined above, however, there were issues with the weight and complexity of the system. Later DARPA initiated the “smart wing” camber morphing program that used smart material actuators and flexible materials [6]. A more recent study on morphing wing capabilities was done by FlexSys called the “Mission Adaptive Compliant Wing” [7]. This design was tested by hanging a prototype wing below a test aircraft and showed a significant increase in fuel savings. The mission adaptive compliant wing had separate morphing actuators for the leading and trailing edges of the wing surface [7].

2.1.3.1 AFRL Camber Morphing Wing

The morphing wing design used in this study was developed from the Variable Camber Compliant Wing (VCCW) created by AFRL CITE. This system was designed to be a wing structure completely contained within single piece composite skin. Figure 2 shows the prototype VCCW. In this design, the leading and trailing edge both move together through the means of a compliant-mechanism rib and require a single actuator per segment. The wing is capable of increasing the camber by 6% [9]. Simulation results showed that the morphing wing is theoretically capable of a 4% lift improvement on take-off and a 15% drag reduction at cruise [8]. This design was tested in a wind tunnel multiple times and showed it capable of morphing during up to 56 mph aerodynamic loads at different angles of attack [10] [11].

Figure 2: AFRL Variable Camber Compliant Wing prototype in undeformed, deformed 6%, and twist configurations.

2.1.3.2 Other Camber Morphing

In recent years some more research has been done on camber morphing wings. Of the camber morphing wing research few have flight tested controlling an aircraft with morphing wings. Research done by Virginia Tech designed, built, and flight tested a camber morphing wing UAV using micro fiber composite (MFC). Their morphing wing depended solely on pilot input for control. Wind tunnel testing showed the ability of the wing to produce sufficient roll control authority. During flight testing the UAV was difficult to control due to control lag between the pilot and the morphing of the wing[24][25]. There is other examples of research into using MFCs to enable the camber morphing[26][27][28].

2.1.4 Morphing Aerodynamics and Control

A key aspect of a morphing wing is the ability to optimize the wing for different stages of flight. This presents a complex optimization problem, which is wing specific. The algorithm in this study that equates current aircraft conditions to optimal wing configurations was based on Prandtl's lifting-line theory [12]. This provides the sectional lift characteristics across a wing surface. Modern development of Prandtl's work has led to equations which break down the desired wing position into an asymmetric portion and symmetric portion[3] [13]. The asymmetric portion is what enables the aircraft to roll and can be thought of as a morphing aileron. The symmetric portion enables the optimization of lift and drag during flight, replacing the function of flaps. The two components are both derived by minimizing the induced drag during roll and flight. The equation which relates the wing position to the state of the aircraft is shown in (1), from [29]. Here θ denotes the spanwise location of the wing surface. A value of $\theta = 0$ references the left wingtip and $\theta = \pi$ denotes the right wingtip. In the equation b is the wingspan, c is the chord length, R_A is the wing aspect ratio, δ_L is the lift control variable, $C_{\ell,\bar{p}}$ is the roll-damping derivative, C_{ℓ,δ_ℓ} is the roll-control derivative, δ_ℓ is the asymmetric control magnitude, and $\tilde{C}_{L,\alpha}$ is the airfoil section lift slope coefficient. For a wing with six different segments there will be a total of six equations. For more information of the derivation of the equation see [29].

$$f(\theta) = \underbrace{\frac{4b\delta_L}{\pi R_A \tilde{C}_{L,\alpha} c_{\text{root}}}\left[\frac{sin(\theta)}{c(\theta)/c_{\text{root}}} - 1\right]}_{\textbf{Symmetric Portion}} + \underbrace{\frac{C_{\ell,\delta_\ell}}{C_{\ell,\bar{p}}}\delta_\ell \, cos(\theta)}_{\textbf{Asymmetric Portion}} \tag{1}$$

2.1.5 UAV Control Latency

An aircraft's control system must respond quickly to pilot input commands to insure safety and stability of the aircraft. High delay can make it difficult for a pilot to control the aircraft as the operator will tend to overcorrect undesired movement which can lead to divergent oscillations [30]. In the case of operating a UAV there is an increase in delay due to the pilot input commands passing through a RC transmitter and receiver. A morphing wing control system that relies upon pilot inputs can further increase this delay. Research using flight test simulations on pilots showed that at one second of delay the aircraft becomes uncontrollable. At a quarter second delay the aircraft was controllable but there was a slight impact on performance [30].

2.2 Characterization of Aircraft Through Flight Data

An important aspect of this study is to determine the flight effectiveness of the morphing wing and the controller. This section details how one can take flight data and use system identification to create linearized state space equations that characterize an aircraft. The linearized state space equations can be used to help determine the overall stability of the aircraft.

2.2.1 Aircraft Linearized System of Equations

To characterize an aircraft using system identification one must know which inputs and outputs on a flight test to utilize and what general format that the equations will take. The general form used for this study are state space equations of the linearized lateral equations of motion. We are only interested in the lateral equations because the morphing wing control will mostly effect roll and yaw. The linearized equation used is in the aircraft body reference frame and is shown in (2), from [14]. The equation simply states that given the changes in aileron input ($\Delta\delta_a$) and rudder input ($\Delta\delta_r$) and current changes in y-axis velocity, often referred to as sideslip, (Δv), rolling rate (Δp), and yawing rate Δr one can determine the new states of the aircraft. The values within the matrices are predetermined based on the aircraft properties and are the forces (F), moments (M), and inertia (I) in the different axis. It is these values that one hopes to determine through system identification. These equations are highly simplified and are only relevant when an aircraft is confined to small deviations from steady equilibrium flight conditions. However, the equations will provide valuable information on the aircraft's stability and performance characteristics. For more information on the derivation of this equation see [14].

$$\begin{bmatrix} W/g & 0 & 0 \\ 0 & I_{xx_b} & -I_{xz_b} \\ 0 & -I_{xz_b} & I_{zz_b} \end{bmatrix} \begin{Bmatrix} \Delta\dot{v} \\ \Delta\dot{p} \\ \Delta\dot{r} \end{Bmatrix} = \begin{bmatrix} F_{y_b,\delta_a} & F_{y_b,\delta_r} \\ M_{x_b,\delta_a} & M_{x_b,\delta_r} \\ M_{z_b,\delta_a} & M_{z_b,\delta_r} \end{bmatrix} \begin{Bmatrix} \Delta\delta_a \\ \Delta\delta_r \end{Bmatrix} + \begin{bmatrix} F_{y_b,v} & F_{y_b,p} & F_{y_b,r} - V_0 W/g \\ M_{x_b,v} & M_{x_b,p} & M_{x_b,r} \\ M_{z_b,v} & M_{z_b,p} & M_{z_b,r} \end{bmatrix} \begin{Bmatrix} \Delta v \\ \Delta p \\ \Delta r \end{Bmatrix} \quad (2)$$

2.2.2 System Identification

An aircraft is a complex system and characterization is difficult. System identification is one effective method of characterizing a system in which there are unknown elements within the system model. System identification looks at known inputs and observed outputs of a complex system to create identifying systems of equations. The equations can be written in the form of (3). Here x(t) is the state vector, A is the state matrix, u(t) is the input vector, and B is the input matrix [15]. To identify the aircraft system, one must determine the values for the A and B matrices. Equation (2) can be turned into this format by taking the inverse of the first matrix and multiplying it by the other two matrices. Thus, A is a 3x3 matrix and B is a 3x2 matrix. A more accurate model can be created by partially filling in known values from (2). One goal of this study is to determine these values for the morphing wing aircraft.

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{3}$$

MATLAB has a useful tool for creating models based on input and output data called the system identification toolbox [16]. Using this tool one can determine the values of the A and B matrices by recording the pilot input commands of a flight test and the responding y-axis velocity, roll rate, and yaw rate of the aircraft throughout the flight. The toolbox works by first importing the raw input and output data into the tool. Then one can filter this data to remove noise and remove the mean. A state-space estimation is then created from a piece of the input/output signal. A separate portion of the input/output signal is then used to verify the estimation by sending the input data through the estimated state-space equations and comparing the output to the flight test output data.

2.2.3 Stability Characterization

The estimated linearized lateral equations of motion can tell about how an aircraft system behaves. After using system identification to determine the values, one will have a quantitative way of looking at the stability of an aircraft based on flight test results. [17]. To obtain the desired stability characteristics one must first add the change in bank angle ($\Delta\phi$) to the 3x3 A matrix in (3) and turn it into a 4x4 matrix. At level flight at zero elevation angle the result is shown in (4).

$$\begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} & 1 \\ A_{2,1} & A_{2,2} & A_{2,3} & 0 \\ A_{3,1} & A_{3,2} & A_{3,3} & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \Delta v \\ \Delta p \\ \Delta r \\ \Delta\phi \end{Bmatrix} \tag{4}$$

The eigenvalues of (4) are associated with the stability of the aircraft in flight and how the aircraft will respond to perturbations. There are four eigenvalues; two real roots and one complex pair. The two real values are associated with the spiral mode and a roll mode, and the complex pair is associated with a Dutch roll mode. The roll mode value is associated with how fast the roll rate becomes a constant after a roll command is given. The spiral mode is associated with the aircraft's response to a yawing motion. The Dutch mode is a combination of yawing, rolling, and sideslip. The magnitudes of the real components determine how quickly the mode becomes stable or unstable after a perturbation. The sign of the real values determines if the mode is convergent or divergent. For a typical aircraft, the roll mode is highly damped and convergent, the spiral mode is lightly damped or lightly divergent and the Dutch roll can be slightly to moderately damped [17].

For the spiral mode, a slight divergence is typically not an issue if the value is small. A divergent spiral mode means that over time the aircraft will spiral laterally inward as shown in Figure 3. If the divergence is large, the aircraft could quickly spiral out of control.

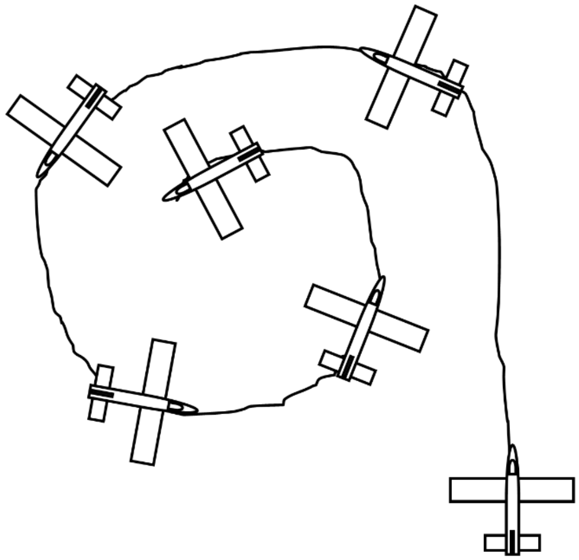

Figure 3: Depiction of spiral divergence.

The Dutch roll has a real and complex component. The real component determines the amount of damping and the complex determines the natural frequency of the oscillations. Figure 4 shows a slightly damped Dutch Roll mode. If this mode is divergent or only slightly damped, then the aircraft can be more difficult to control.

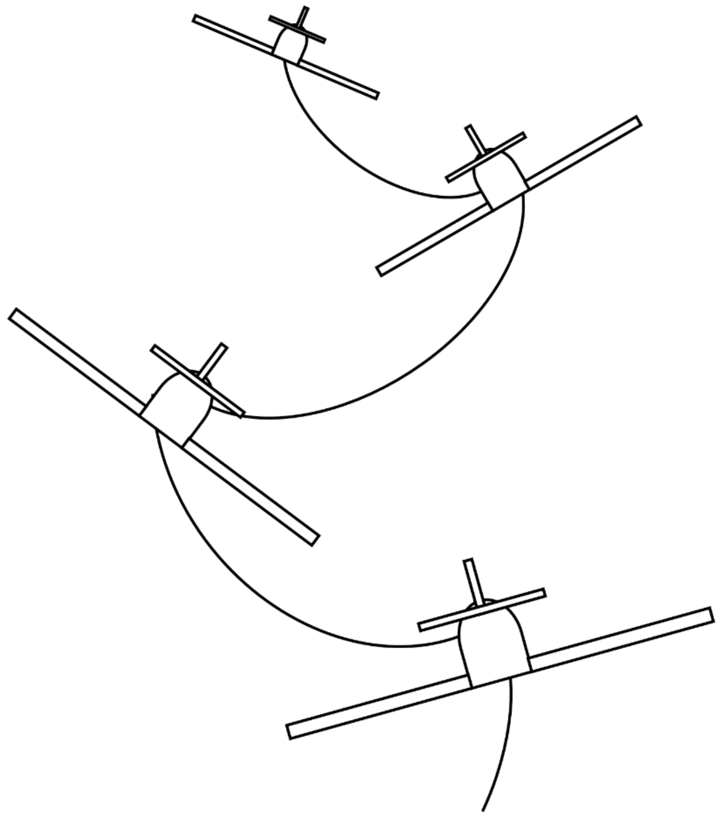

Figure 4: Depiction of an oscillatory Dutch roll mode.

While these values will not provide one with the full characterization of the aircraft's stability during flight, it does provide a way to compare a baseline aircraft with standard wings to one with morphing wings. It also provides a general depiction of how the aircraft will respond to perturbations. If any of the eigenvalues are far away from their typical values then this can provide further evidence that the aircraft is unstable. For more information of the dynamics of lateral modes see [17].

2.3 System Reliability Design

An important aspect of system design is reliability, especially when dealing with high risk prototypes. Morphing wing technology is still undergoing development and there are ample unknown aspects increasing the odds of failure. When dealing with aircraft flight testing, the impact of these failures can be high. Aircraft development is expensive and a failure of a control system can easily lead to a crash. An effective method of increasing the reliability of a system is failure mode effects and criticality analysis (FMECA) [18].

FMECA has multiple steps for identifying and avoiding failures. After system requirements and functionality have been established, the first step is to identify failure modes. A failure mode is how a component of a system can fail. An example can be a control surface fails to respond to a pilot input. The next step is to determine possible causes of failures. Following the earlier example, the control surface may fail due to a bug in the code that causes the control surface to stop responding. After finding all the possible causes of failure, one then determines the effects of failure. In the previous example, a failure of a control surface would have a heavy effect in that it could possibly cause a crash. The final step in finding failures is determining if there are any ways to detect a failure beforehand. There are many ways in which one could detect a failure such as a sensor detecting an increase in temperature.

The next four steps are related to identifying the criticality of the failure. Criticality of failure is a combination of a failure's severity, frequency, and the probability that the failure can be detected before a major failure, such as a plane crash. These three components are rated from1 through 10, where 1 is the least severe and 10 the most severe. When the failure has a high impact on the system it is rated in the 7-10 range. If a failure is very likely to occur, it is assigned a number in the 7-10 range. Finally, if a failure is unlikely to be detected then it has a high value as well. The combination of these three elements are called a risk priority number (RPN).

The last step in whole process is finding methods of reducing the RPNs of the failures with high values. The key here is looking at what can be done to reduce the severity and frequency of a failure, while increasing the probability of detection. An example of decreasing the severity of a loss of an aircraft control element is having redundant systems or the ability to rely on other control methods. An example of decreasing failure likelihood would be to run diagnostics on an aircraft before flight. To increase the ability to detect a failure one could implement feedback from the control program to the pilot when errors occur. For more information on reliability analysis of a system see [18].

Chapter III: Methodology

This chapter details the setup and design of the morphing wing project. Section 3.1 details the work done by other groups towards the overall goal of flying a morphing wing aircraft, including building the wing and deriving the necessary control equations. This section also describes the requirements of the overall controller and the regular flight testing that was required prior to the design. Section 3.2 describes the work done to build the necessary components to control and flight test the warping wing aircraft. The section additionally includes how the controller was built, coded, integrated into the aircraft, and how the flight test plan was prepared.

3.1 Background

Researchers with the Air Force Research Lab (AFRL/RQVS) designed and created a morphing wing capable of replacing the ailerons and flaps of an aircraft. Each wing, right and left, was designed to morph using three flexible ribs. Researchers with the USU Aero Lab derived the control equations for the morphing wing design. AFIT Researchers integrated the morphing wing onto an off-the-shelf UAV and designed and built a controller to be able to fly and compare the new wing to a standard wing.

3.1.1 Morphing Wing Design

Central to the morphing wing is a hollow spar made out of a single piece of rectangular carbon fiber. Several flexible ribs attach to the spar, to form the base of the wing. Three of the ribs in each wing are a compliant mechanism design made from 3D printed plastic. Servo motors push off the rigid spar to move the compliant ribs, which force the wings surface to flex up and down. Each servo motor moves independently, enabling the wing surface to flex in multiple directions, as illustrated in Figure 5 and Figure 6. The skin is a combination of thin carbon fiber sheets covered by nylon. Most of the design of this wing will not be discussed in this thesis as it is being submitted for a patent.

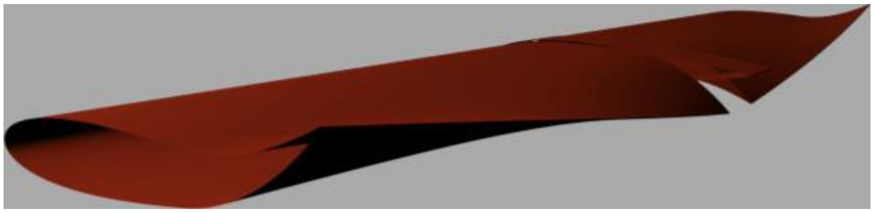

Figure 5: Side view of morphing wing design

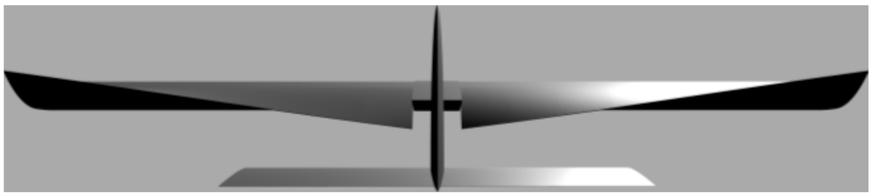

Figure 6: View of morphing wing from the rear

3.1.2 Morphing Wing Control Equations

Researchers with the USU Aero Lab derived the equations required for the morphing wing using lifting theory combined with numerical optimization [29]. The equations determine the required angle of deflection of the three morphing wing segments to achieve a specified lift coefficient and roll rate. The angle of deflection in each segment is determined by two distinct polynomial equations: an asymmetric portion, which determines the roll component; and a symmetric portion, which determines the flap levels. Thus, there are six polynomial equations for the left and right half of the wing. Each equation requires two variables; aircrafts lift coefficient (C_L), and desired roll rate ($\bar{p}$), combined with pre-determined polynomial coefficients. C_L is determined according to (5).

$$\mathrm{C_L} = \frac{\mathrm{W}\cos(\gamma)}{\frac{1}{2}\rho \mathrm{V}^2 \mathrm{S}\cos(\phi)} \tag{5}$$

Equation (5) requires the climb angle (γ), airspeed (V), and bank angle (ϕ), all of which can be obtained from the aircraft autopilot. The other two parameters, the weight W and the frontal area, S, are measured constants. $\bar{p}$ is determined from the pilot aileron command.

The polynomial coefficients expressed in (6)-(11) below were determined by running the morphing wing parameters through an optimization algorithm. This algorithm ran through a range of C_L of 0 to 1.2 and $\bar{p}$ of 0 to 0.24. The resulting data was curve fit as a two-dimensional polynomial. δ denotes the degrees of deflection of the wing surface on the inner, middle, and outer section of the wing.

$$\delta_{si}(C_L, \bar{p}_{\text{bounded}}) = \begin{cases} 16.405C_L + 52.998C_L\bar{p}_{\text{bounded}}^2 - 43.114C_L\bar{p}_{\text{bounded}}^4 - 32275C_L\bar{p}_{\text{bounded}}^6 \\ -49.353C_L^3 - 1349.6C_L^3\bar{p}_{\text{bounded}}^2 + 69014C_L^3\bar{p}_{\text{bounded}}^4 - 406620C_L^3\bar{p}_{\text{bounded}}^6 \\ +100.61C_L^5 + 4319.9C_L^5\bar{p}_{\text{bounded}}^2 - 220190C_L^5\bar{p}_{\text{bounded}}^4 + 716520C_L^5\bar{p}_{\text{bounded}}^6 \\ -83.321C_L^7 - 4596.6C_L^7\bar{p}_{\text{bounded}}^2 + 215010C_L^7\bar{p}_{\text{bounded}}^4 + 867790C_L^7\bar{p}_{\text{bounded}}^6 \\ +24.905C_L^9 + 1562.5C_L^9\bar{p}_{\text{bounded}}^2 - 59264C_L^9\bar{p}_{\text{bounded}}^4 - 1339800C_L^9\bar{p}_{\text{bounded}}^6 \end{cases} \tag{6}$$

$$\delta_{sm}(C_L, \bar{p}_{\text{bounded}}) = \begin{cases} 3.731C_L - 68.77C_L\bar{p}_{\text{bounded}}^2 + 5569.3C_L\bar{p}_{\text{bounded}}^4 - 72028C_L\bar{p}_{\text{bounded}}^6 \\ -1.8741C_L^3 - 72.884C_L^3\bar{p}_{\text{bounded}}^2 - 7707.1C_L^3\bar{p}_{\text{bounded}}^4 + 182520C_L^3\bar{p}_{\text{bounded}}^6 \\ +5.8634C_L^5 + 210.63C_L^5\bar{p}_{\text{bounded}}^2 + 11787C_L^5\bar{p}_{\text{bounded}}^4 - 369340C_L^5\bar{p}_{\text{bounded}}^6 \\ -1.9585C_L^7 - 86.905C_L^7\bar{p}_{\text{bounded}}^2 - 9104.5C_L^7\bar{p}_{\text{bounded}}^4 + 245280C_L^7\bar{p}_{\text{bounded}}^6 \end{cases} \tag{7}$$

$$\delta_{so}(C_L,\bar{p}_{\text{bounded}})=\begin{cases}-3.3943C_L+33.659C_L\bar{p}_{\text{bounded}}^2+4749.8C_L\bar{p}_{\text{bounded}}^4-81459C_L\bar{p}_{\text{bounded}}^6\\-6.4754C_L^3-184.46C_L^3\bar{p}_{\text{bounded}}^2-5722.6C_L^3\bar{p}_{\text{bounded}}^4+176220C_L^3\bar{p}_{\text{bounded}}^6\\+10.418C_L^5+336.28C_L^5\bar{p}_{\text{bounded}}^2+16367C_L^5\bar{p}_{\text{bounded}}^4-480800C_L^5\bar{p}_{\text{bounded}}^6\\-3.9714C_L^7-118.12C_L^7\bar{p}_{\text{bounded}}^2-15990C_L^7\bar{p}_{\text{bounded}}^4+373890C_L^7\bar{p}_{\text{bounded}}^6\end{cases}\tag{8}$$

$$\delta_{ai}(C_{L,\text{ bounded}},\bar{p})=\begin{cases}3.1947\bar{p}-33.635\bar{p}^3+366.26\bar{p}^5\\+4.2276C_{L,\text{ bounded}}^2\bar{p}+267.35C_{L,\text{ bounded}}^2\bar{p}^3-173.66C_{L,\text{ bounded}}^2\bar{p}^5\\-18.021C_{L,\text{ bounded}}^4\bar{p}-490.23C_{L,\text{ bounded}}^4\bar{p}^3-10289C_{L,\text{ bounded}}^4\bar{p}^5\\+21.11C_{L,\text{ bounded}}^6\bar{p}+36.807C_{L,\text{ bounded}}^6\bar{p}^3+36392C_{L,\text{ bounded}}^6\bar{p}^5\\-8.2921C_{L,\text{ bounded}}^8\bar{p}+172.28C_{L,\text{ bounded}}^8\bar{p}^3-26283C_{L,\text{ bounded}}^8\bar{p}^5\end{cases}\tag{9}$$

$$\delta_{am}(C_L,\bar{p})=37.084\bar{p}+63.499\bar{p}^3+0.16024C_L^2\bar{p}+0.54527C_L^2\bar{p}^3\tag{10}$$

$$\delta_{ao}(C_L,\bar{p})=66.161\bar{p}+2.9897C_L^2\bar{p}\tag{11}$$

At high levels of C_L and $\bar{p}$ the polynomial equation can become unrealistic and require bounded levels. The limit of $C_{L,\text{bounded}}$ is determined by (12) and the limit of $\bar{p}_{\text{bounded}}$ is determined by (13).

$$\bar{p}_{\text{limit}}=-1.25|C_L^3|+2.625C_L^2-1.9375|C_L|+0.70875\tag{12}$$

$$C_{L,\text{ limit}}=29.61715925|\bar{p}^3|-11.71436588\bar{p}^2-0.54643659|\bar{p}|+1.2\tag{13}$$

3.1.3 Base Aircraft

The aircraft chosen to fly the morphing wing was a COTS Big Stick XL UAV, as shown in Figure 7. The aircraft included its own set of wings with standard ailerons and flaps. The morphing wing was designed to roughly be equal in size to the standard wing. The base aircraft is capable of supporting up to a 15lb wing. A custom designed wing box allowed the morphing wing to interface with the base aircraft and be fully detachable.

Figure 7: Base aircraft hooked up to standard wings.

3.1.4 Standard Wing Flight Testing

The base aircraft was flown with a standard set of wings before the control system was designed and hooked up to the warping wing. The standard wing employed typical aircraft aileron and flap control surfaces allowing for a comparison of performance. Weights were added to the standard wing to match the estimated weight of the future warping wing. The data gathered from the flight test was recorded by a Pixhawk 2 autopilot system and includes many flight parameters, such as position and velocity data. This data can then be developed into the linear lateral equations of motion using system identification as described in section 2.2.2**.**

3.1.5 Controller Requirements

A controller was designed to use AFRLs wing combined with USU's equations as shown in Figure 8. The controller needed to have the capability to read pilot commands and the required data to calculate C_L from eq (5), and to output servo commands to all six servos. The aileron commands were supplied from a RC transmitter to a RC receiver onboard the aircraft, and the flight data was supplied by a Pixhawk 2 autopilot system through a serial port. The controller must be light weight, fit inside the aircraft, integrate with the autopilot system and RC receiver, run off a battery power supply, and enable remote access to the control program for diagnostics.

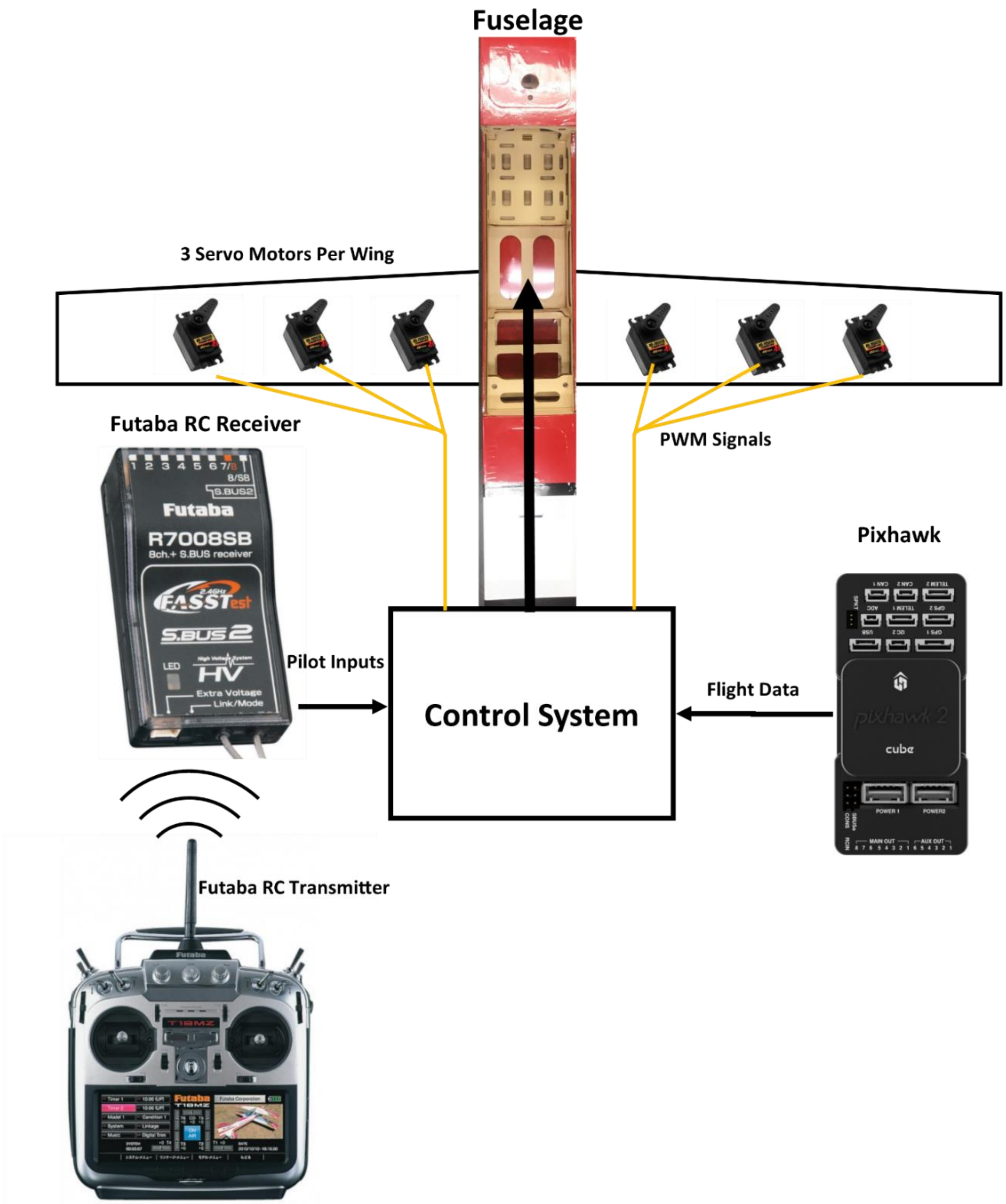

Figure 8: Morphing wing control setup. The wing holds six actuators to control the morphing behavior. The Futaba transmitter sends pilot commands to the RC receiver. The Pixhawk autopilot provides vehicle state information from the onboard sensors, and the controller board computes and relays the desired positions to all six actuators.

3.2 System Design:

This section details how the controller was designed and built to fulfill the requirements and to be able to flight test the morphing wing aircraft. The section first describes how the controller was designed, including handling the required inputs and outputs, and how the controller was built and mounted inside the base aircraft's fuselage. Next the controller code design is discussed, along with the controller's integration into the aircrafts power supply, and how the user interfaces with the controller while it is within the aircraft. Finally, the flight test plan necessary to safely fly the aircraft and collect the information for analyzing the morphing wing performance is described.

3.2.1 Control System

The control system block illustrated in Figure 8 must be housed within a small computer or microprocessor board to be able to compute the required control commands. The board must be able to take several inputs to compute the positions of the six actuator inputs. The required inputs are: desired roll, vertical velocity, airspeed, and bank angle. The desired roll is obtained from the RC receiver, which outputs the pilot commands in the form of a pulse width modulated (PWM) signal. A Pixhawk 2 autopilot system provides the other required inputs through a serial communication port. The Pixhawk uses an accelerometer, gyroscope, GPS, barometer and a compass to determine the UAV's current state. The Pixhawk is capable of sending out vehicle state information, such as the velocity, airspeed and bank angle through the serial port at a rate of 4 Hz. Thus, the controller board must be able to read PWM inputs from the receiver, communicate with the Pixhawk via serial, and output PWM commands to the six servos.

The first controller was designed to use an Arduino Uno microcontroller. The Arduino was chosen as it is relatively fast, the availability of open source code, and can output PWM commands. It was found, however, that the Arduino was difficult to interface with the Pixhawk system as the Arduino required a good understanding of serial communication and there was little open source code available for Pixhawk to Arduino interfacing.

The next controller chosen was an ODROID-XU4. The ODROID has the ability to integrate with the Pixhawk autopilot, as students at AFIT have accomplished this task before. However, it runs a Linux operating system in the background, which reduces the speed of the response. It was found, however, that the ODROID did not have a good way of outputting and inputting a PWM signal as the I/O interface operated at a maximum of 1.8V.

The last controller chosen was the Raspberry Pi. Although this controller was slower than the ODROID, it had a much better I/O interface with 3.3V, which allows for general purpose input output (GPIO). Also, the Raspberry Pi is able to interface with the Pixhawk using DroneKit software, which connects to the Pixhawk with python commands and a USB to serial cable. Although the Raspberry Pi can output PWM, it is not capable of outputting the required 6 PWM signals. The Raspberry Pi, however, has many third-party units which integrate with it to expand its capabilities. In this case we used an Adafruit servo driver and software allowing up to 12 PWM outputs. The PWM input requirement also required the addition of third-party boards as described in 3.2.1.1.

3.2.1.1 Pilot RC Input Design

The most important inputs are the pilot aileron commands to effectively control the aircraft. It is imperative that these values are read into the controller quickly as latency can make the aircraft unflyable. The Pixhawk can output the pilot input commands through the use of DroneKit, though unfortunately this data has a large delay and only updates at a frequency of 4Hz. At this data rate the aircraft would be unflyable.

The team decided to bring in the RC signal via a multiple-step conversion process, to provide inputs at a sufficiently fast rate and to not over tax the computation capability. First, the PWM signal was converted to an analog voltage using a KNACRO PWM-to-voltage module. Because the Raspberry Pi cannot read an analog signal, the analog signal was converted to an inter-integrated circuit (I2C) serial comm using an analog-to-digital (ADC) converter and software from Adafruit. Then the I2C serial communication could be read into the controller program to provide a fast, low-latency communication of the pilot roll commands.

The control system hardware design was created as shown in Figure 9. Both the ADC and servo controller were hooked up to the Raspberry Pi through the Pi’s SDA/SCL ports. All grounds were hooked up to the same rail as the servo power and the whole control system runs off of a 5V regulator connected into the Raspberry Pi.

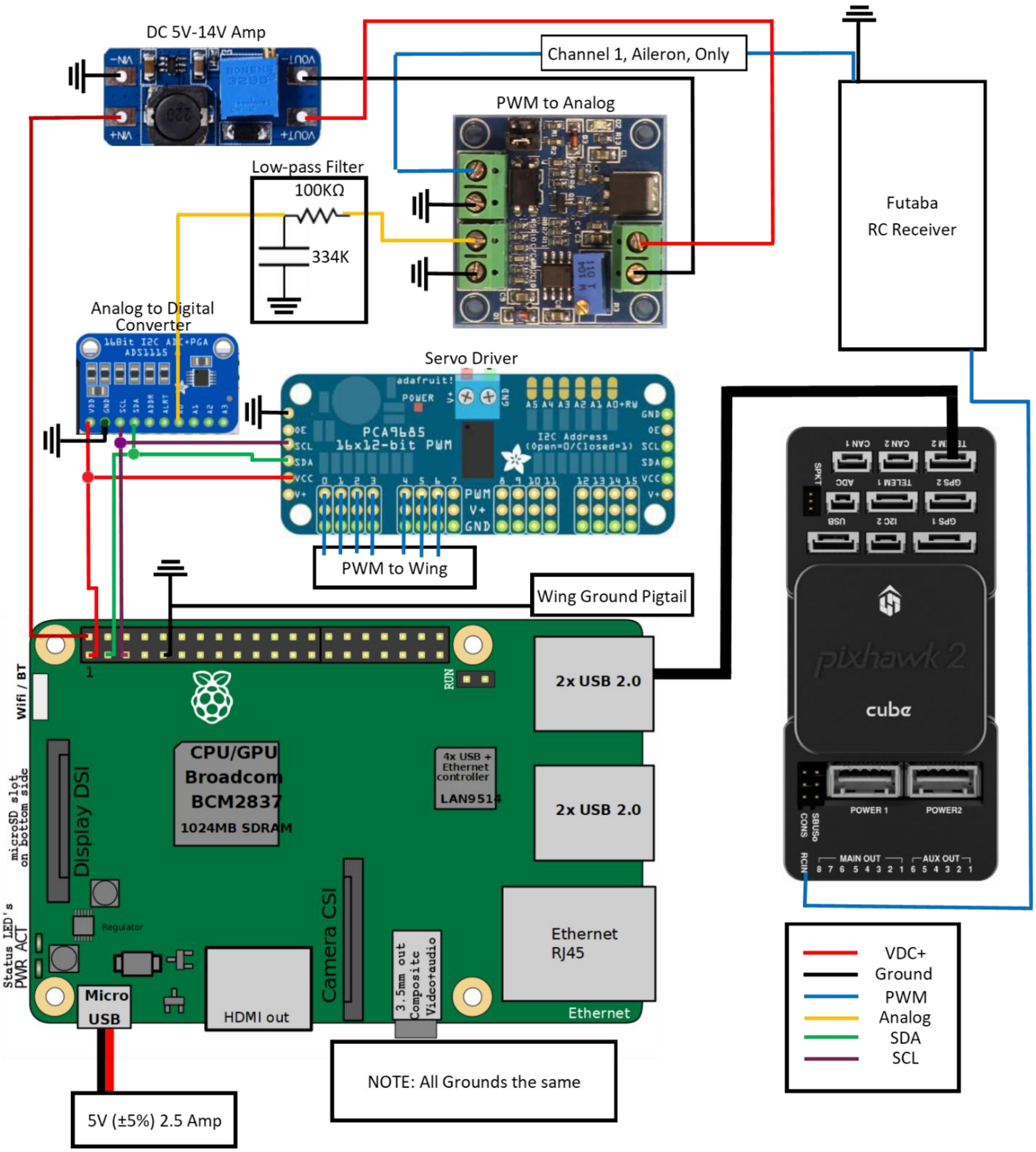

Figure 9: Circuit diagram for the morphing wing control system. The Pixhawk is connected to the Raspberry Pi through a serial USB port and the Pi receives PWM commands from the RC receiver by passing through a PWM-to-analog device, low-pass filter, and a ADC. The Pi outputs the servo motor commands through a servo driver.

The PWM-to-analog signal contained a lot of noise due to the low frequency of the RC receiver signal, which is 50 Hz. The PWM-to-analog device was designed to use frequencies of 3kHz and above. This is likely due to the fact that the lower the PWM frequency when converting to an analog signal, the more difficult it becomes – as shown below [31]. The effects of using a PWM-to-analog filter on a higher frequency can be seen below in Figure 13.

To clean up the analog signal, a simple low-pass filter was designed and utilized in between the PWM-to-analog device and the analog-to–digital converter. A low-pass filter converts a PWM to analog signal by taking the average of the voltage over time. The higher the percent of duty cycle, the higher the voltage. Figure 10 demonstrates the effect of the filter on two different PWM signals. One can also see the sawtooth error created by using this method, which is similar to the output of the PWM-to-analog output. The filter was effective at reducing, but not eliminating the 50Hz sawtooth component. The filter consisted of a capacitor and resistor combination as seen in Figure 11. The values of the capacitor and resistor were determined through experimentation.

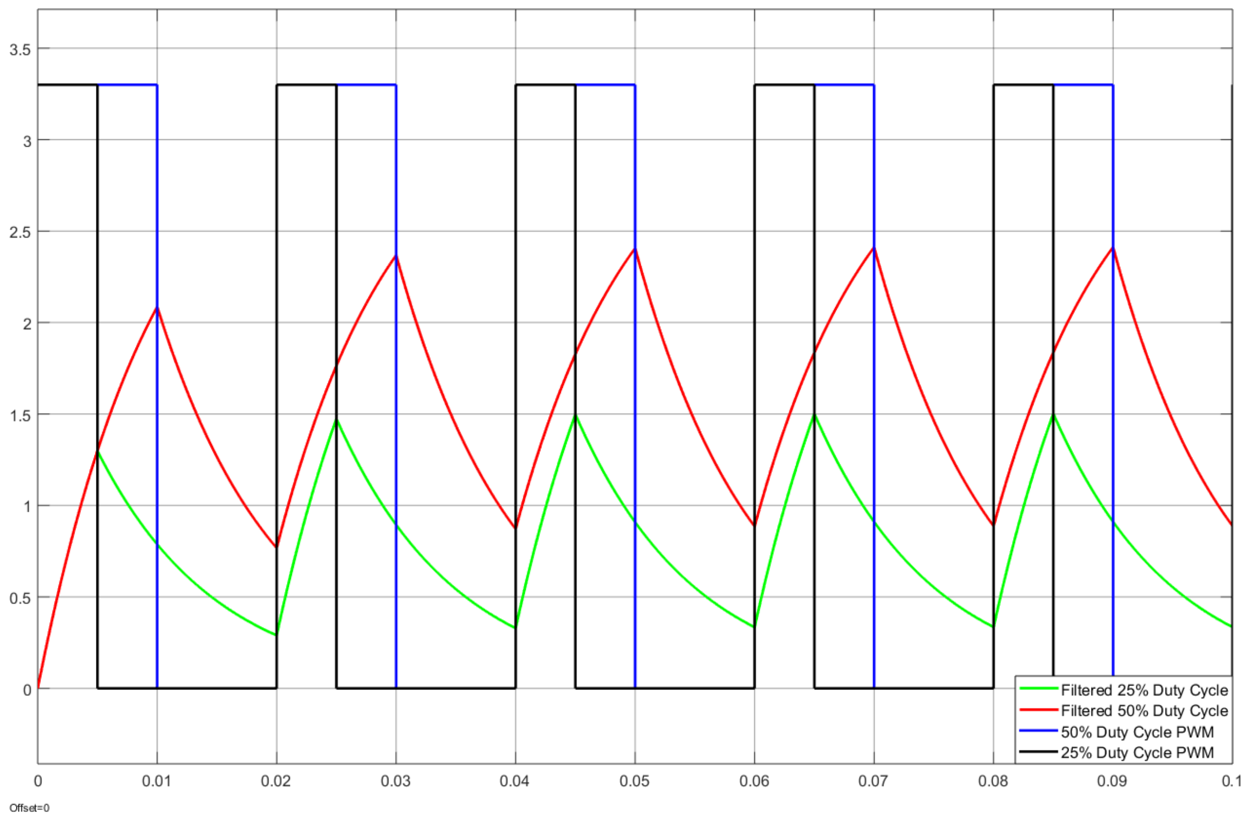

Figure 10: Result of using a low-pass filter on a PWM signal. The higher the percent of the duty cycle the higher the resulting voltage.

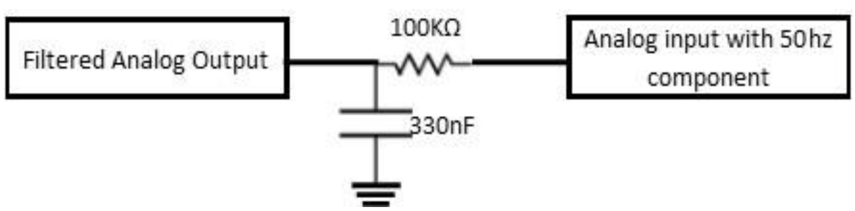

Figure 11: Simple Low-Pass Filter Design

The strength of the low-pass filter is proportional to the product of the resistor and capacitor, as shown in (14). Here, as the value of the resistor and capacitor increase the cut-off frequency decreases. A small capacitor-resistor combination reduces only a small portion of the sawtooth component while a large resistor-capacitor combination results in a more analog signal but at the expense of a slow reaction time.

$$f_c = \frac{1}{2\pi RC} \tag{14}$$

Figure 12 demonstrates the difference between using different filter strengths. Here the PWM signal mimics the same signal that one would see sent to a servo motor and would command a servo to go to the zero-degree position. The strong filter is five times the strength of the weaker and has one fifth the error. However, the weak filter takes significantly less time to stabilize. Thus, one must find a balance between error in the signal and reaction time. As explained in chapter two, in order to be flyable, the reaction time between pilot input and wing deflection must be fast[30]. Keeping the total delay under 0.25 seconds was a design requirement for the filter. The filter chosen for the controller was in-between the strong and week filter designs. The filter used a 100KΩ resistor and 330nf capacitor.

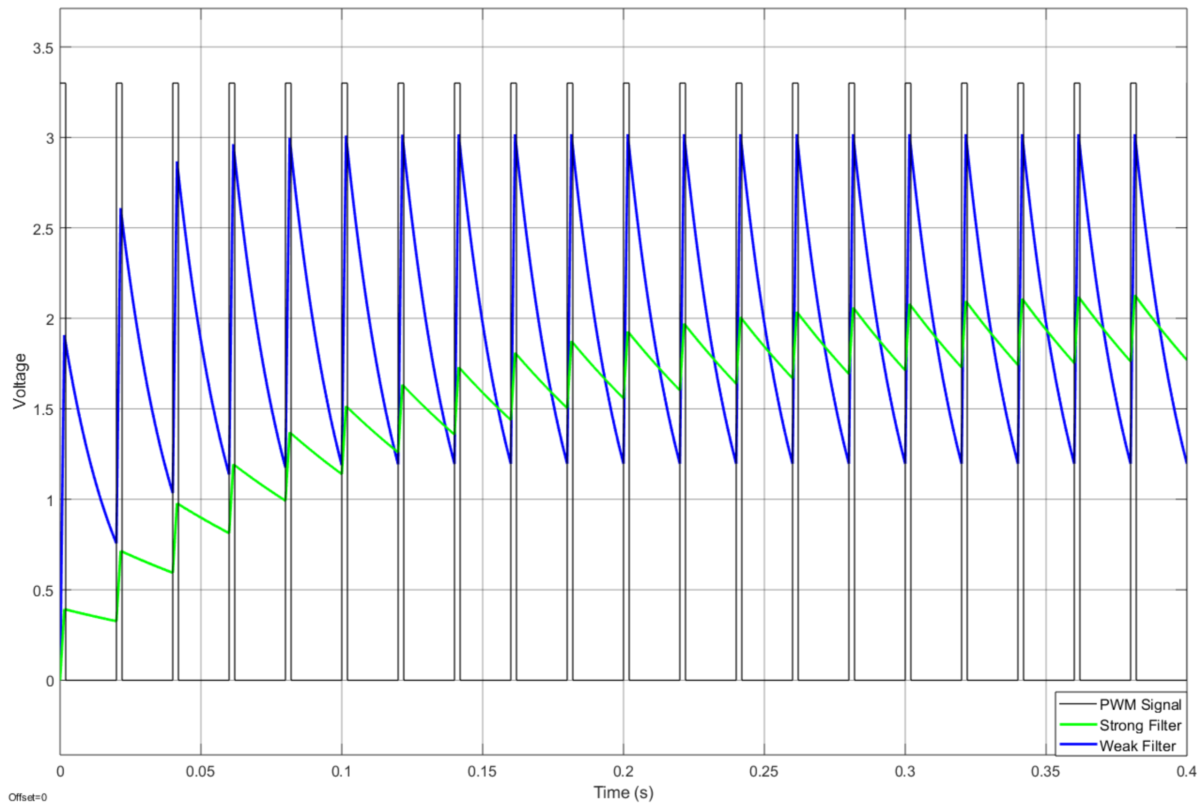

Figure 12: Passing a 50Hz servo PWM signal through low-pass filters of different strengths.

Figure 13 demonstrates the effect of using the same PWM signal as shown in Figure 12 but with a frequency of 500Hz instead of 50Hz. At a low frequency of 50Hz the PWM signal is difficult to cleanly and quickly convert to an analog signal. A higher frequency provides the filter with more data points and results in less error.

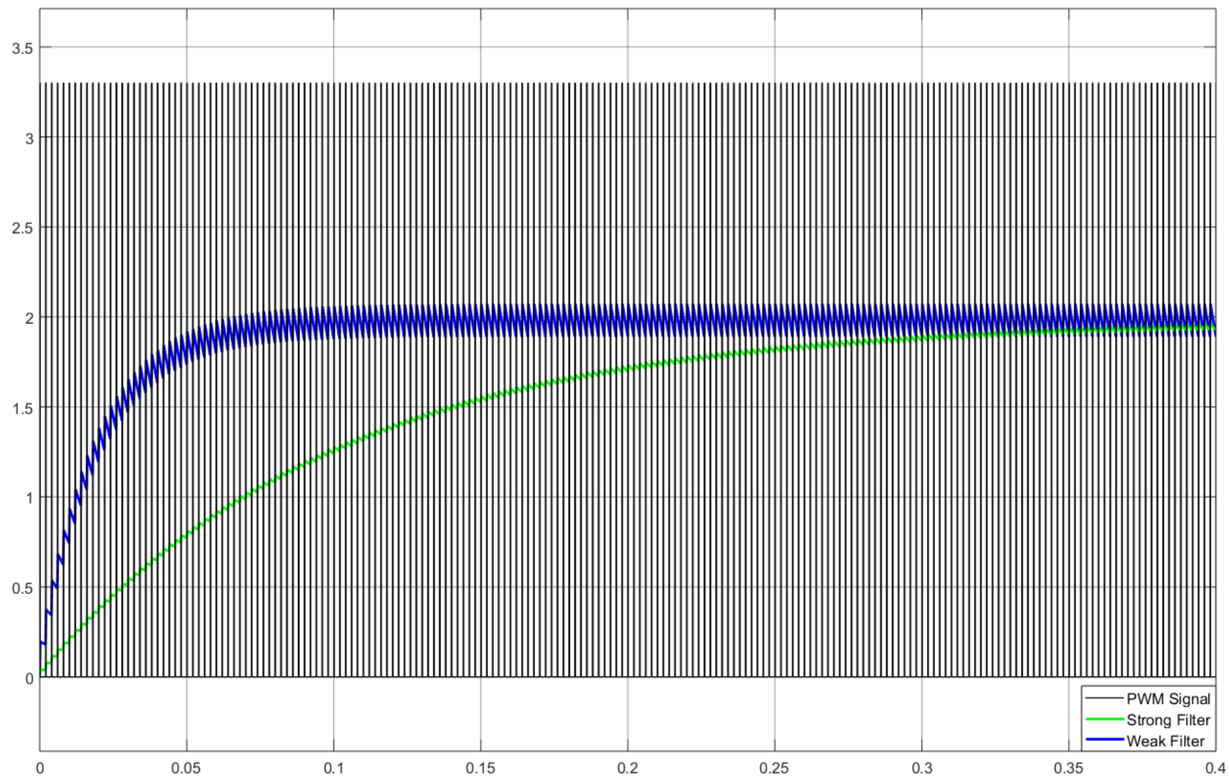

Figure 13: Passing a 500Hz PWM signal through two different low-pass filters

3.2.1.2 Controller Mounting

The controller was built using the circuit design in Figure 9 and is shown in Figure 14. The controller components were mounted to plywood with plastic standoffs to avoid potential short circuiting and provide clearance between the wood plate and the components. All board connections were soldered to avoid any connections separating during flight. The plywood board was attached to the center of the fuselage via Velcro pads.

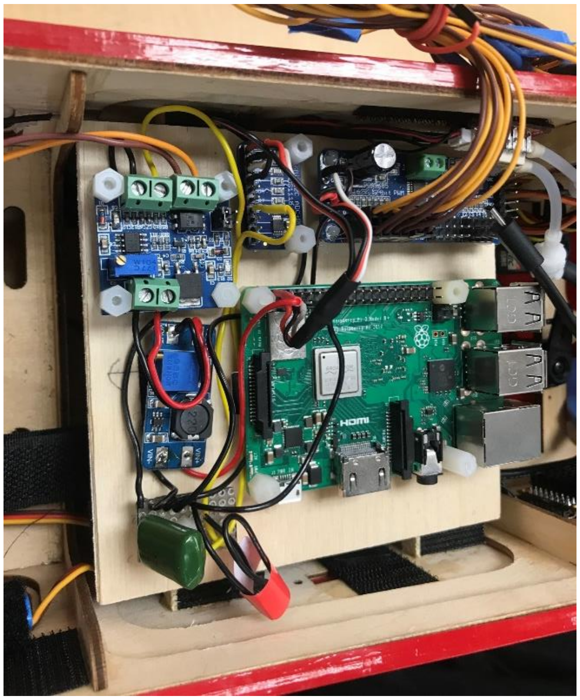

Figure 14: Controller mounted inside of morphing wing fuselage.

3.2.2 Controller Code Design

The Raspberry Pi and the attached components provide the compute capability, but some software must be programmed to control the aircraft. Python was the coding language selected, as code required to run the servo driver and read from the ADC were available in Python from Adafruit[32][33]. DroneKit, which also uses python, was used to communicate with the Pixhawk over the serial connection[34].

The controller software needed to be designed in such a way as to be able to incrementally increase the strength of the warping wing and active flaps. The incremental increase in capability was necessary as there were several attributes about the aircraft that were unknown. The first was the available roll control authority of the warping wing. Another was the amount of lift and drag produced by the active lift control that simulates the functions of a flap. Finally, it was not known if the wing would produce undesired motions, such as a yaw or pitch, in conjunction with the desired roll. The controller was designed to have three separate modes of flight operation. A highly simplified representation of the flight modes is shown in Figure 15.

The first flight mode was a safety mode where all actuators respond directly to pilot input without any optimization equations and without flight data input. This mode would provide the maximum amount of roll control authority in case of an emergency.

The second mode would utilize the asymmetric optimization equations that relay pilot input to actuator position. In this mode, the outer actuators move the most and the inner actuators move the least.

The last mode would utilize the asymmetric and symmetric optimization equations. Here the controller would take flight data from the Pixhawk and pilot input to determine the position of the actuators according to the optimized parameters described above. In this mode the pilot would have the least amount of control over the wing. The symmetric optimized expression would cause the inner most actuators to deform and act like flaps. - active flaps

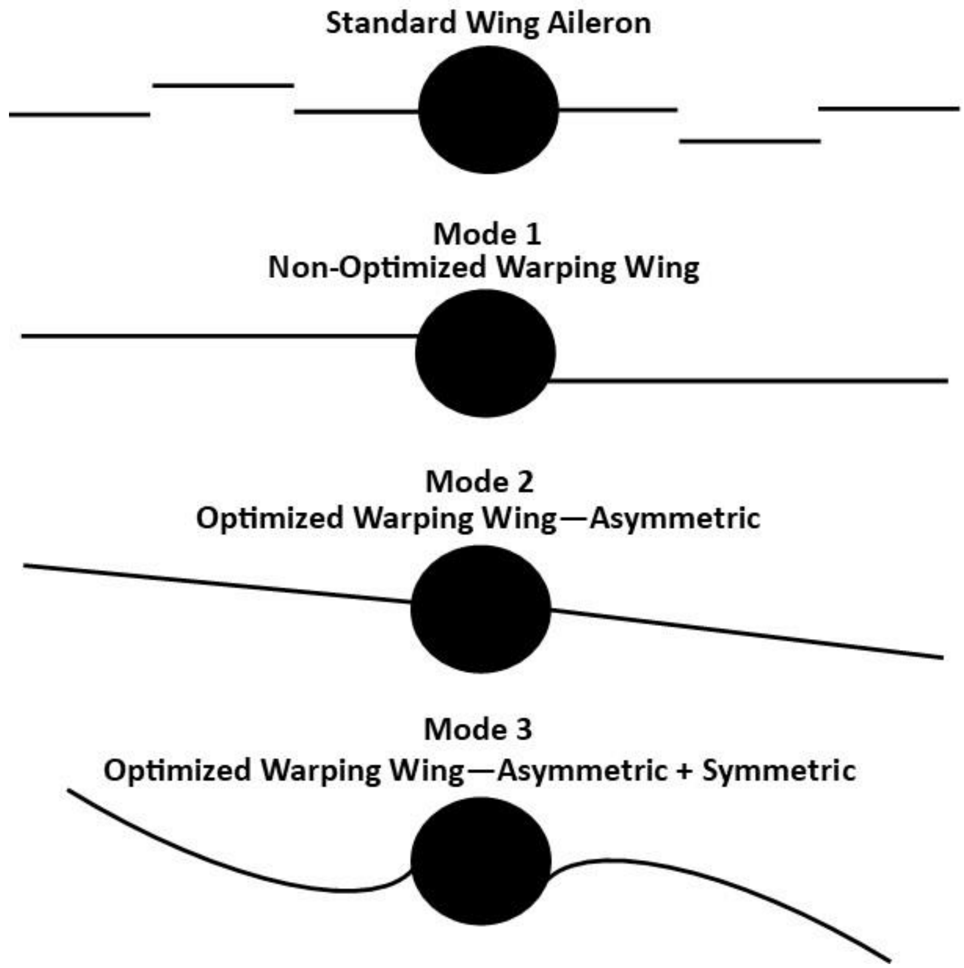

Figure 15: Simple comparison of the three different flight modes used for the morphing wing testing to a standard aileron used on a traditional wing.

All three modes were designed to be interchangeable during flight so that the pilot could switch to Mode 1 at any time. Also, manual flaps were implemented into all the modes so that the pilot had lift control at all times. The manual flaps were designed to move all six actuators the same amount and be variable from zero to five degrees.

Another control design is the strength of the pilot commands to wing deflection. Without the knowledge of the amount of roll control authority provided in the different flight modes, it was necessary for the pilot to be able to manually adjust this value while in the air. This control gain was set to affect aileron deflection in modes one and two, as well as active flap deflection in mode 3.

With all these implementations the pilot was able to control many aspects of the aircraft during flight. In total, the controller needed to read pilot roll control commands, flap levels, aileron deflection strength, active flap defection strength, and mode changes. Only the roll control command has the strict speed requirement and for all other pilot commands the Pixhawk, with its larger delay and 4Hz data stream, was more than sufficient. The RC transmitter can output flaps, modes, and roll control authority, and the Futaba receiver outputs the roll directly to the controller and all the other values to the Pixhawk.

A block diagram of the controller code is shown in Figure 16. When the control loop is activated, the Raspberry Pi will collect data from the Pixhawk and the RC value from the ADC. Since the Pixhawk data only comes in at 4Hz, we utilize a sample and hold architecture so that the Raspberry Pi does not have to wait a quarter second to obtain the data and continue to the next step of the program.

After obtaining the data, the controller will enter one of the three modes based on pilot input. In mode one, the program will take the RC aileron value, multiply it by the control gain, and then add in the flaps. For example, if the pilot pulls the stick all the way to the right, signifying a right roll, this will be read as a positive 20^o deflection command by the right aileron. If the control gain is set to 0.5, the 20^o becomes 10^o. If the flaps are set to 5^o, then the final output of the controller will be a 5^o positive deflection.

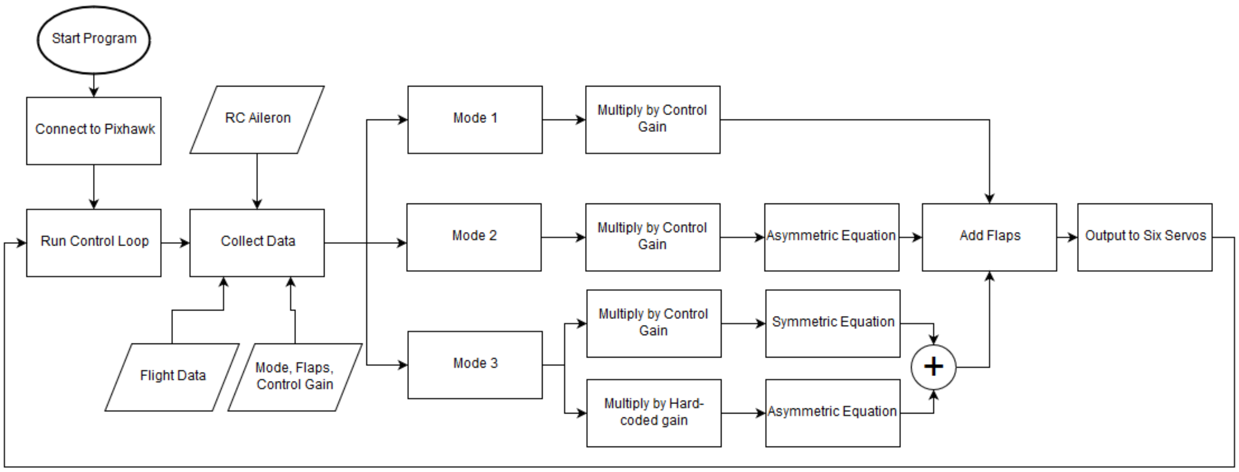

Figure 16: Control system code flowchart. After collecting data, the program selects one of three flight modes based on input commands from the pilot.

In the case of mode two, the aileron deflection command from the pilot is multiplied by the control gain, goes through the asymmetric equation, and then added with flaps. In mode three, the pilot commands go through the asymmetric equations and the symmetric equations, which are then added together. When in mode three, the symmetric portion is multiplied by the control gain, while the asymmetric values are multiplied by a predetermined value which is hardcoded into the program. The asymmetric control gain is determined based upon testing performed in mode two. The value is hardcoded because having the pilot control both gains at the same time would be overdemanding of the safety pilot.

3.2.3 Aircraft Power Supply

A key requirement for the controller design is the ability to integrate with the batteries on-board the aircraft. The aircraft had several different power supplies to provide the various voltages required by the system. Two 10 Ah batteries in series providing 50V, one 2.2 Ah 12V battery, and one 3.2 Ah 6.6V battery. The power supplies were set up as shown in Figure 17. The main battery supply was the two 10 Ah batteries that supplied power to the motor and the wings. Each wing was powered through a voltage regulator to ensure stable power to the servos. The Pixhawk and Raspberry Pi were supplied by the 2.2 Ah battery through two 5V voltage regulators. All of the components in the control system used the voltage regulated 5V supply. The RC receiver has its own power supply which also powered the rudder and elevator servos. If anything happened to the main battery supply, the aircraft would still have elevator and rudder control. The grounds of the wing voltage regulators, the RC receiver, and the control system were all tied together.

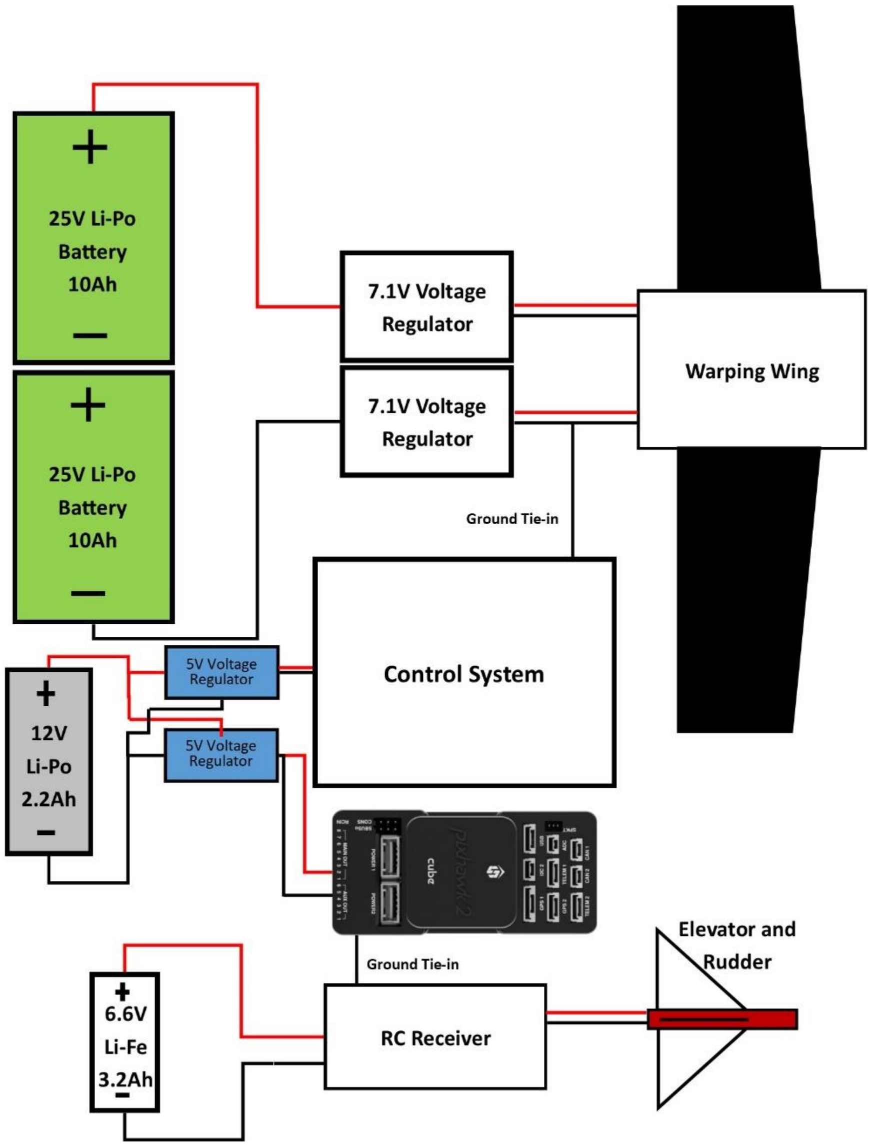

Figure 17: Power supply setup. The morphing wing had three different battery supplies.

3.2.4 System Communication

Another key aspect of the controller design was designing a method of communicating with the Raspberry Pi without having to plug into the board directly. Having a connection to the Pi is essential; it provides one the ability to start and stop the program, to feedback real time data to the operator while the program is running, and to make edits to the code. Additionally, it is difficult to access the controller while it is in the aircraft, as the wing must be removed. The Raspberry Pi 3 has a wireless antenna on-board which allows it to access Wi-Fi. To connect with the Pi the operating laptop was connected to a wireless router through ethernet, and the Pi through its Wi-Fi.

the operating laptop can access the Pi using Secure Shell (SSH), enabling access to the control program. The program streams data to the SSH terminal about the current status of the wing to help assess if the program is operating properly. In a flight test the program must operate in the background as once the connection between the Raspberry Pi and operating laptop is severed the program must still run. To run the program in the background the “nohup” Linux command is utilized.

In addition to sending flight data to the Raspberry Pi, the Pixhawk also can stream flight data down to a ground station laptop. Two antennas, one connected to the laptop and the other the Pixhawk, transmit the data over large distances. This allows for the ground station to get real time flight data when the aircraft is flying. The “Mission Planner” program is used to connect to the Pixhawk and view the data in real time. Although having this connection is not critical to flight testing, it can be valuable for diagnosing potential problems. Figure 18 shows the communication setup between the two laptops and the morphing wing.

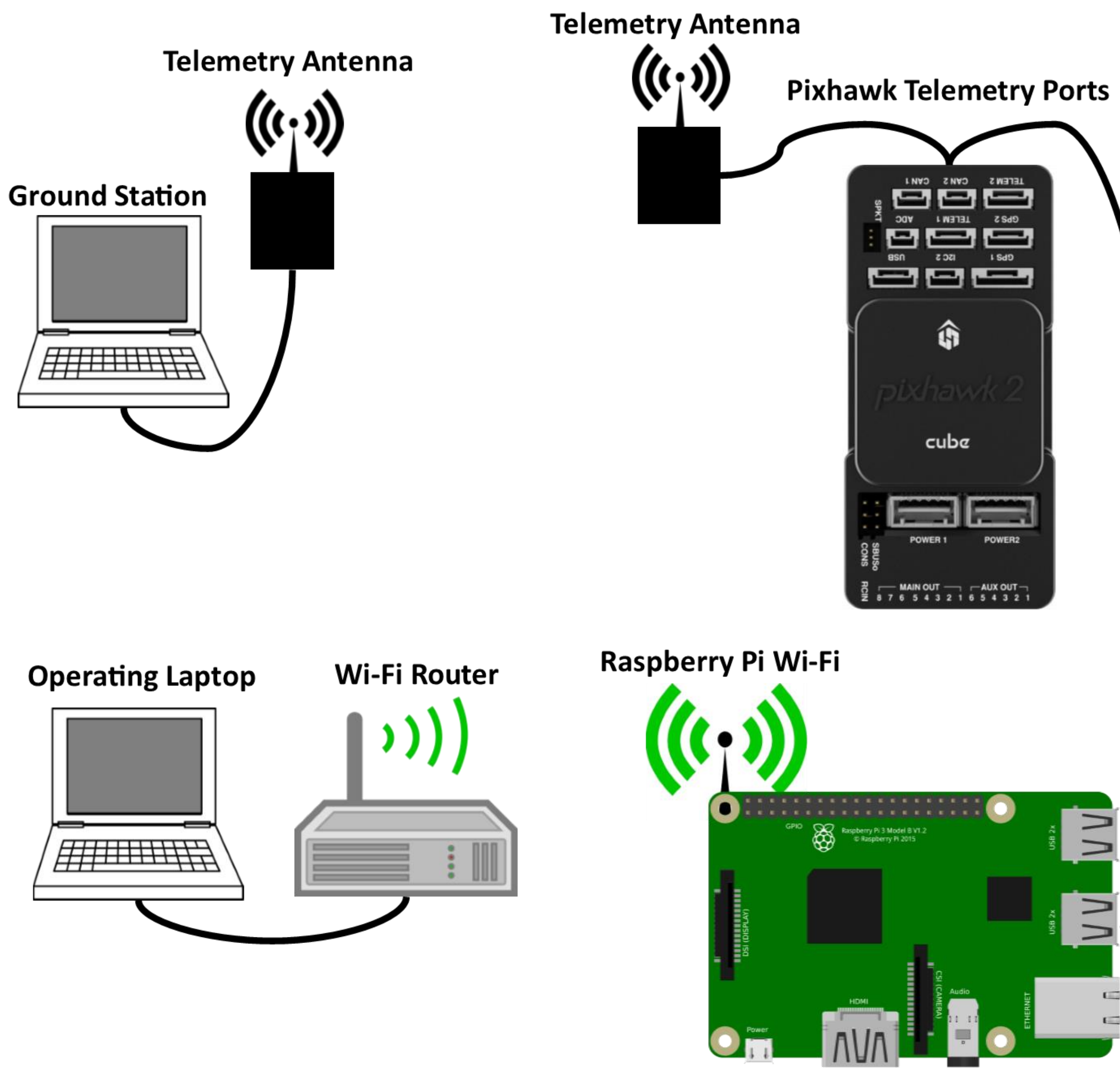

Figure 18: Morphing wing communication setup. One laptop is used to communicate with the Pixhawk telemetry, and the other laptop is used to communicate with the Raspberry Pi.

3.2.5 Flight Test Plan

After designing the controller, the next step was to prepare a flight test plan. There were several unknowns that a flight test would enable us to determine and these values will be necessary in determining the effectiveness of the wing and its control system. They unknown values are:

- The amount of roll control authority generated by the movement of the entire wing surface in mode 1.
- The amount of roll control authority generated from the asymmetric optimization equations as in mode 2.
- The effects of the symmetric equations from mode 3 on the aircrafts lift and drag.
- The stability of the aircraft during all three flight modes.

A flight test plan was created to determine the unknown characteristics and to fly safely using a prototype wing without previous wind tunnel tests. A figure of the prototype aircraft to be tested is shown in Figure 19. The plan was created to incrementally test the aircraft and slowly increase the amount of complexity in the system. Table 1 shows the flight test plan.

The first objective was to perform taxi testing with the aircraft going up-and-down the runway with increasingly higher speeds. The taxi tests would be performed while in mode 1 and are necessary to make sure that the aircraft frame can handle the weight of the aircraft, and if the aircraft is properly balanced.

Figure 19: Morphing wing aircraft prior to testing

Table 1: Flight Test Plan			
Test	**Objective**	**Collection Required**	**Flight Plan**
1. Taxi testing	Stability of aircraft on ground	Visually observe wing drop	Aircraft will go up-and-down runway at increasing speeds
2. Gain testing, non-optimized wing	Check flight stability and determine proper control gains	Flight test data from Pi and Pixhawk, visual data from on-board cameras, and feedback from safety pilot	Aircraft will fly a race track pattern around runway land and repeat until satisfied with gains and stability

3. Gain testing optimized wing asymmetric equations only	Check flight stability and determine proper control gains	Flight test data from Pi and Pixhawk, visual data from on-board cameras, and feedback from safety pilot	Aircraft will fly a race track pattern around runway, land, and repeat until satisfied with gains and stability
4. Gain testing optimized wing asymmetric and symmetric equations	Check flight stability and determine proper control gains	Flight test data from Pi and Pixhawk, visual data from on-board cameras, and feedback from safety pilot	Aircraft will fly a race track pattern around runway land and repeat until satisfied with gains and stability
5. System identification and performance data collect	Perform various maneuvers with aircraft to gather data required to properly analyze flight characteristics	Flight test data from Pi and Pixhawk, visual data from on-board cameras	Aircraft will perform crab maneuvers, vertical and horizontal doublets, in all three flight modes

The second round of tests involve flying the aircraft in a simple racetrack, oval, pattern. This test uses only mode 1 to decrease risk of failure. The objective of this test is to determine the amount of roll authority produced from the movement of the entire wing. Also, this test will determine the effect of the manual flaps. Flight test data will be recorded on the Pixhawk and the Raspberry Pi to characterize the aircrafts flight capabilities. In between each flight the aircraft will be inspected to make sure everything is still functioning properly. The test will be concluded once the control gain has been tuned to provide the adequate roll control authority. Because the roll utilizes the entire wing surface the amount of roll control authority will be higher than in other modes. Thus, the amount of control gain will be set low, at 30% of maximum, to prevent the rolling rate from being too sensitive and potentially damaging the aircraft.

The third round of tests is similar to the second but requires running the aircraft in mode 2. In mode 2, the aircraft will fly using the asymmetric equations. The objective of this flight test is to determine the roll control authority produced by the wing surface and to determine the overall stability in mode 2. The control gain will be set to be higher than the predetermined safe level for mode 1. At anytime during these flight tests the pilot has the capability to switch to mode 1. Multiple flights will be performed until a proper control gain is determined.

In the fourth round of tests the third mode is tested to determine the correct control values to set for the symmetric equations. The same process as the previous two flight tests is employed. The control value for the asymmetric portion of the equations is set from the previous flight test and is hard coded into the program.

The final series of tests are to preform specific flight maneuvers in all three modes to collect data for aircraft characterization. The maneuvers to be performed are shown in Table 2. The maneuvers are primarily designed to exercise the ailerons and see how the plane responds. A “crab” maneuver is one in which the aircraft maintains a non-zero yaw while maintaining a constant flight direction. A “doublet” is where the pilot will move the aileron stick in one direction and then back in the opposite direction. These maneuvers are performed with low and high roll rates wherein the pilot will slowly and then quickly slowly move the stick. When using rudder mixing the safety pilots RC transmitter will automatically add in rudder commands at the same time as aileron commands, order to assist the pilot in performing turns.

Table 2: Flight Test Maneuvers					
Maneuver	Rudder Mixing	Speed (ft/s)	Roll Rate	Attitude	Note
2	Low	Mid	None	Horizontal	Crab w/ no bank
3	Low	High	None	Horizontal	Crab w/ no bank
8	Low	Mid	Low	Horizontal	Doublet
9	Low	Mid	Low	Vertical	
10	Low	Mid	High	Horizontal	Doublet
11	Low	Mid	High	Vertical	
12	Low	High	Low	Horizontal	Doublet
13	Low	High	Low	Vertical	
14	Low	High	High	Horizontal	Doublet
15	Low	High	High	Vertical	
20	Off	Mid	Low	Horizontal	Doublet
21	Off	Mid	Low	Vertical	
22	Off	Mid	High	Horizontal	Doublet
23	Off	Mid	High	Vertical	
24	Off	High	Low	Horizontal	Doublet
25	Off	High	Low	Vertical	
26	Off	High	High	Horizontal	Doublet
27	Off	High	High	Vertical	
32	High	Mid	Low	Horizontal	Doublet
33	High	Mid	Low	Vertical	
34	High	Mid	High	Horizontal	Doublet
35	High	Mid	High	Vertical	
36	High	High	Low	Horizontal	Doublet
37	High	High	Low	Vertical	
38	High	High	High	Horizontal	Doublet
39	High	High	High	Vertical	

3.2.5.1 Data Capture

During all of the flight tests the Pixhawk and Raspberry Pi will collect flight data on the aircraft. Using sensors and filters, the Pixhawk records the aircrafts position, velocity, acceleration in the x, y, and z navigation coordinate frame and the attitude. The

Pixhawk also records the airspeed, wind speed and direction, and the pilot's yaw, pitch, roll, and throttle commands. The Raspberry Pi was programed to record the commanded position of all six servos in degrees. The Pi also records the pilot set control gain, flaps in degrees, $\bar{p}$, and C_L. Lastly, the Pi collects time data from the Pixhawk and records it with the data allowing the Pixhawk and Pi data to be time synchronized.

The data can be used following the flights to determine flight characteristics of the aircraft by taking the pilot input data and examining the response of the aircraft. By using system identification, one can take these inputs and outputs to create a state space model of the lateral aircraft equations of motion. The state space model allows for analysis of the stability aspects of the aircraft. Another way to determine the overall stability of the aircraft is to compare data from the standard wing testing to see if the pilot had to use more of the other control surfaces to perform the same maneuvers. This will allow for us to see if the roll also produced high amounts of yaw or pitch.

In addition to flight data, cameras are set up on the wing of the aircraft to record the position of the aircrafts wing surfaces during flight. Two cameras, looking at left and right wing, record video and time on flash drives. The videos will provide visual feedback of the wing's reaction to pilot commands and current flight conditions. The video could be important for determining possible sources of error due to structural issues and controller output problems. Possible controller output problems include the wings performing unexpected maneuvers during flight, such as rapid movements, and vibrations.

Chapter IV: Results and Analysis

This chapter details the results of the tests performed on the morphing wing and the controller. Section 4.1 describes the results of the flight test, following the flight test plan as described in section 3.2.5. Section 4.2 discusses the analysis and subsequent tests performed to increase the reliability of the control system for future flight testing.

4.1 Integration and Flight Testing

Before flight testing, integration tests were performed with the morphing wing and control system. Then, following the flight test plan, the first round of tests involved taxi testing the morphing wing. The taxis were completed successfully, however the results were inconclusive and did not provide the required data to answer questions about the wings flyability. The aircraft was then pushed towards a first flight. The aircraft was in a unresponsive roll during the whole flight, the safety pilot was unable to recover the aircraft and it crashed. An image of the crash aftermath is shown in Figure 20. Flight data and video were analyzed to find the causes of the crash, and two primary problems were found: the aircrafts wings lost their shape and the controller stopped working prior to takeoff.

Figure 20: Aftermath of the morphing wing flight test crash.

4.1.1 Integration Testing

Before a flight test occurred, the controller was connected to the prototype morphing wing and tested. The controller was able to perform all the design requirements including the ability to switch between modes using the RC transmitter. Figure 21 shows right morphing wing performing a right and left roll in all three flight modes.

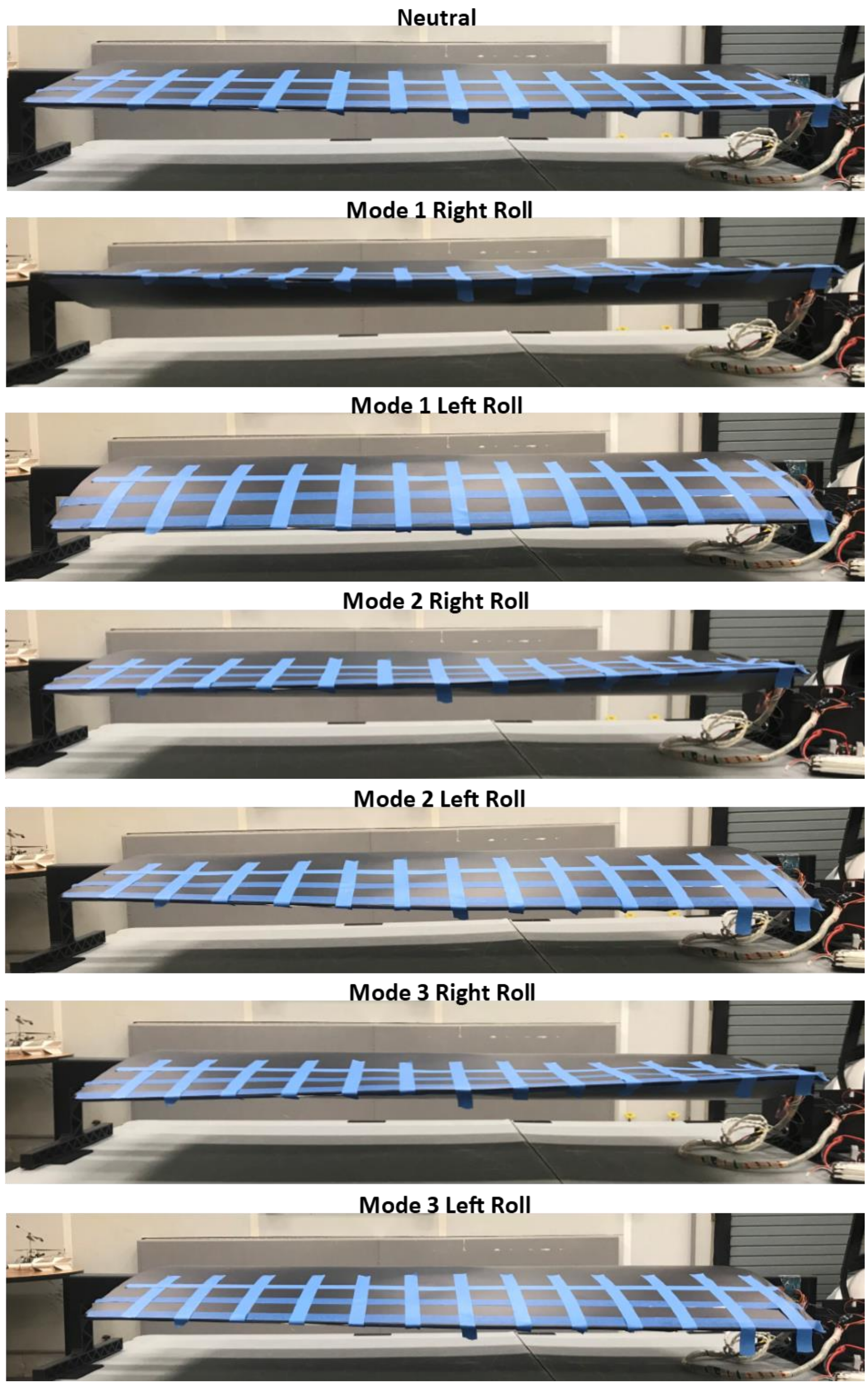

Figure 21: Demonstration of the right wing in three different flight modes. In each mode the wing is set to go full right and then full left roll.

4.1.2 Taxi Testing

The first round of testing of the morphing wing aircraft involved taxi testing the aircraft up and down the runway at speeds just under the take-off speed. These tests were successful in that the aircraft was able to maneuver at multiple speeds and to properly turn without the wing edges dipping low to the ground. The aircrafts frame also supported the weight of the aircraft without any impact to the structure of the aircraft. There was a little instability at higher speeds, however this was likely due to a high crosswind gust of up to 18 knots.

4.1.3 Flight Test

After the taxi testing, but before the wings were taken off per the test plan, the aircraft was positioned at the end of the runway for takeoff. The wind had been heavy all day and there was a short window for a possible first flight. Shortly after takeoff, the aircraft entered into a left roll and began to turn left. The pilot allowed the aircraft to turn with the intent of leveling it off after it turned 180^{o}. The pilot was unable to pull the aircraft out of the left turn as he lacked any roll control authority. The aircraft then spiraled down in a left turn until it impacted with the ground at 24 m/s. The flight path of the failed test can be seen in Figure 22. The back-and-forth pattern on the runway is the taxi testing performed prior to the flight.

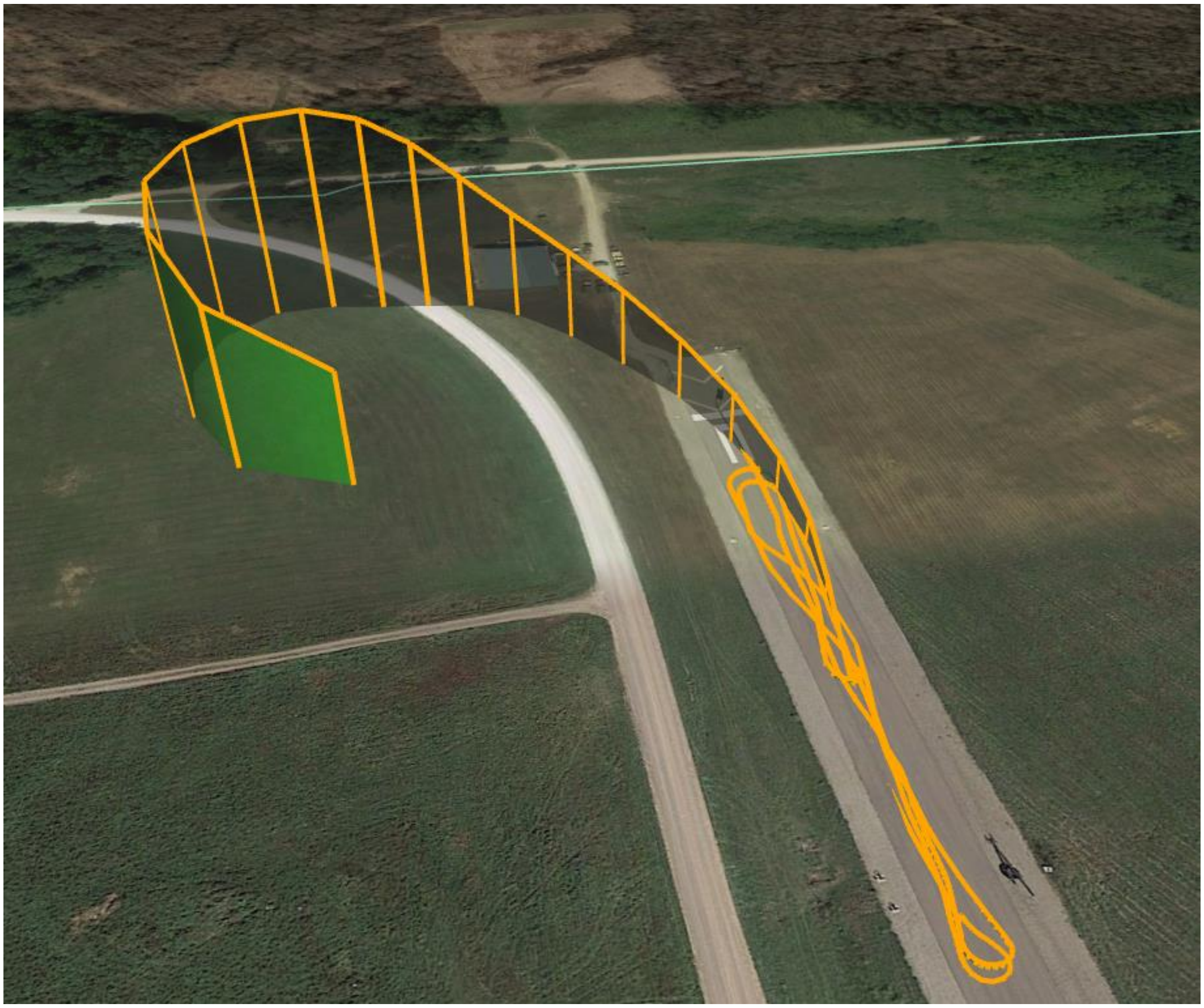

Figure 22: Flight path of failed flight test. Note the multiple loops of the taxi tests up and down the runway.

Video imagery of the two cameras on the aircraft showed that the wing surface did not hold up to the aerodynamic forces produced after it reached a speed of 20 m/s. The aircrafts wings bulged out to a point where the wings did not provide the required lift and control to fly the aircraft. A figure of the aircraft wing surface before and after takeoff can be seen in Figure 23 and Figure 24. The wing expanded to up to twice its size during flight.

Figure 23: Left wing of aircraft prior to takeoff.

Figure 24: Left wing of aircraft experiencing wing expansion after takeoff.

The second issue observed through video analysis was that the wing actuators were never seen to move during the flight. The data sent to the ground station by the Pixhawk and the controller log files were compared to validate this claim. By lining up the times on the data recorded by the Pixhawk and the data recorded by the Raspberry Pi, as shown in Figure 25, one can see that at approximately 15:05, the Raspberry Pi stops recording data while the Pixhawk continues. The ground station was connected to the Pixhawk through the datalink during the entire flight and showed no anomalies. Comparing the video of the wing deflection and data recorded by the Raspberry Pi and Pixhawk, one can see that at no time after 15:05, when the Pi stops recording, do the wings deflect. Note that the whole first flight is contained in the data after 15:12, seven minutes after the controller board stops working. After 15:05, according to the Pixhawk data, there are two incidents were the pilot issued a roll command but the video showed no wing deflection. This tells us that not only did the Raspberry Pi stop recording data but the controller code stopped working. The pilot had roll control during most of the taxi testing but not during the actual flight.

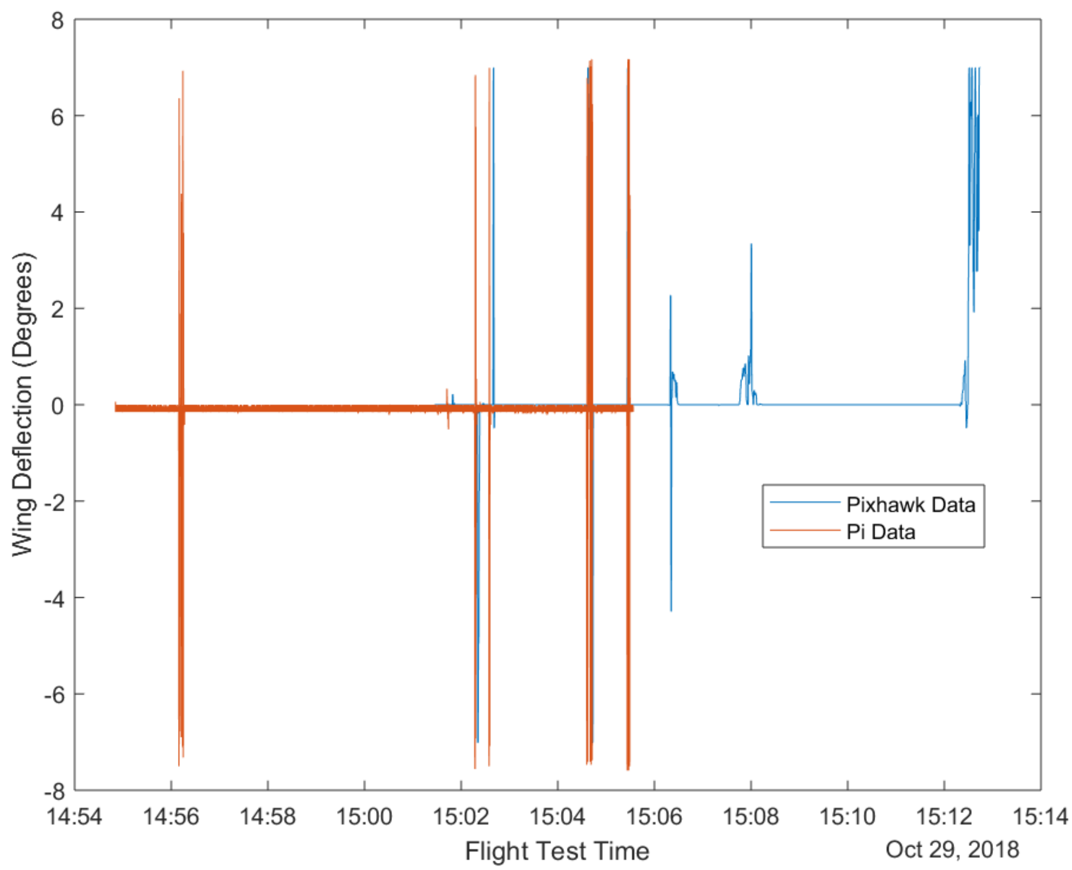

Figure 25: Pixhawk and Raspberry Pi aileron recorded flight data wing deflection comparison.

The two-fold failures, the wing structure and the control code, completely doomed the flight. However, even if only one failure had occurred, either one or the other, the result would have likely been the same. The aircraft with the standard wing was capable (with an extremely skilled pilot) of landing without roll control, the aircraft with the morphing wing attached would have been more difficult to land. The inflated wing would likely have washed out any control surface deflection, even if they had been able to move. The focus after the failed flight test was to first ascertain why the errors occurred and then to increase the reliability of the system through further testing so as to decrease the risk of failure for future flights.

4.2 System Reliability Analysis and Testing

A failure mode analysis was performed on the controller to determine sources of problems and to increase the reliability of the system. Based on this analysis, tests were performed to determine the validity of the potential failures and find possible other sources of error. Adjustments were made in order to decrease the likelihood of failures and decrease the effect of any failures should one occur.

4.2.1 Failure Analysis

Failure analysis is used to determine all the ways for a system to fail as a way of helping to prevent future problems and mitigate their impact. This analysis was performed after the flight test crash to help find the cause of the controller failure as well as to improve the design for future testing. Figure 26 shows a failure cause analysis tree for the morphing-wing controller system. The failure tree that was created to understand the root cause behind the wing expansion, completed by AFRL, is shown in the appendix. The cause of failure was broken down into five categories: Software, Hardware, Testing Procedure, Communication, and Electrical. Some of the sources of error, such as vibration under hardware, can cause problems in other categories, such as vibration causing a short circuit.

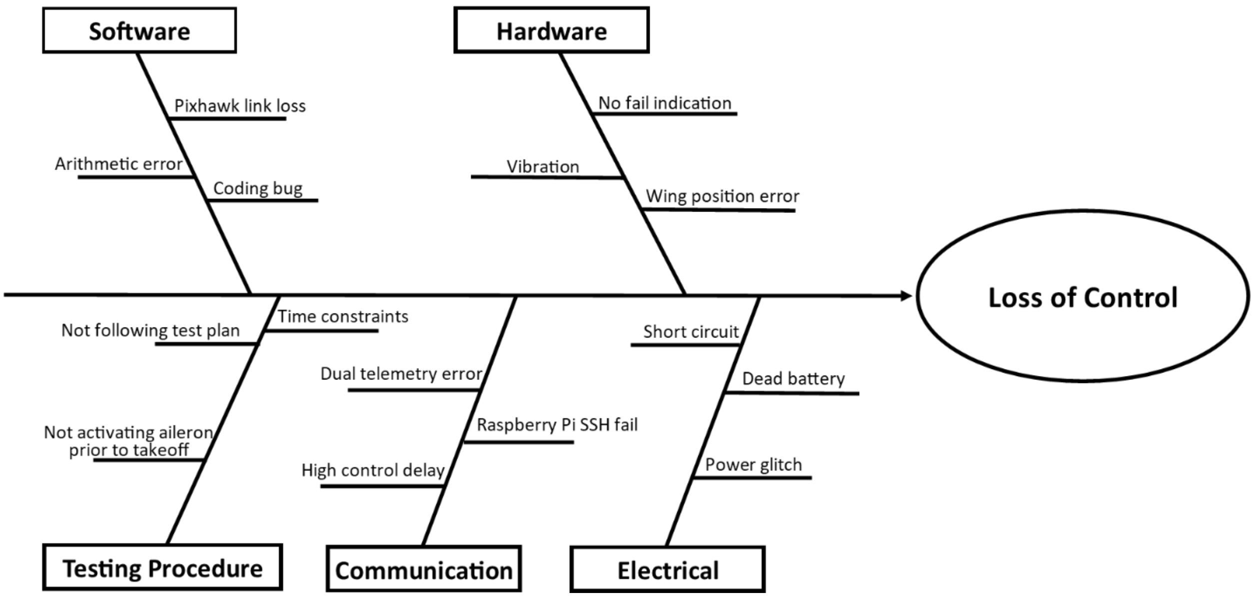

Figure 26: Failure mode analysis breakdown

4.2.1.1 Software

The controller code runs a loop where the program collects data from the Pixhawk and pilot, runs an algorithm, and outputs PWM values to the six servos. If at any time in this loop there is an error, then the program will exit the loop and the aircraft will no longer be controllable. Possible sources of error might be algorithm failures, such as dividing by a zero, Pixhawk communication errors, or any other coding bugs that can occur.

4.2.1.2 Hardware

Under hardware there are three causes of error: vibration, wing position errors, and the lack of failure indication. Most of the components within the controller were not designed to withstand vibrations associated with flying in a UAV. Vibration could cause several problems during flight. One is that the power cable could become loose and cause a power glitch, forcing the Pixhawk and Raspberry Pi to restart. The cable could also become entirely detached and shut down all power. In both of these scenarios the aircraft would lose control. Also, any of the connections sending the pilot input commands to the receiver could break loose and create a loss of signal. A loss of pilot input commands would also cause a loss of control. With vibrations, the Raspberry PI might become compromised, causing errors in the program. Finally, the cable connecting the Pixhawk and Raspberry Pi, which is a USB connection, could become loose or dislodged. A disconnection or loose connection could also cause errors in the program.

Another potential problem can be caused by error in the wings position. This can occur from the PWM-to-analog conversion, error in the RC receiver signal, and from vibrations inherent in the servos. The wing position error can cause a loss of control if the vibrations drown out the actual position commands from the pilot. Loss of control after takeoff can also occur as a result of the pilot not knowing that the controller is not operating properly.

4.2.1.3 Testing Procedure

The testing procedure category includes everything that is planned out in the flight test plan. Although the plan was intended to incrementally increase the complexity of the tests to avoid potential issues, it was not perfect. Adding in the requirement for the pilot to always activate and visually confirm the aircrafts aileron movement prior to takeoff could avoid a failure in the future. During the flight tests some of the procedures were not followed as written. The flight test plan required the aircraft to be taken back and inspected after performing the taxi test. Bad weather caused a delay in the flight test, which created a sense of urgency, resulting in the aircraft not being inspected after taxiing.

4.2.1.4 Communication

The controller is underneath the morphing wing and within the fuselage during flight testing. The controller is connected to an operating laptop through SSH allowing for the user to start and observe the controller Because of the limited access. The control program must be set to run in the background of the Raspberry Pi so that after disconnecting from the operating laptop the controller continues to run. A potential failure can occur if this process is completed incorrectly. A communication loss between the operating laptop and the Raspberry Pi can cause potential issues. Another potential failure mode is connecting to the Pixhawk with the Raspberry Pi and Pixhawk at the same time. With two simultaneous connections, one might interfere with the other; this could cause the program to fail or to receive incorrect data back from the Pixhawk.

Another failure potential is when the delay between pilot input and wing movement becomes too great. If the delay is over a quarter of a second , the pilot might have problems controlling the aircraft[30]. The delay can be expended by the PWM-to-analog converter, the Raspberry Pi program, and the servo motors.

4.2.1.5 Electrical

The last failure category is the electrical system. The controller is powered by a lithium battery onboard the aircraft, which also supplies power to the Pixhawk. If this battery fails, then the pilot will only have elevator and rudder control and not roll control. If the power at any time drops below the required 5v for the Raspberry Pi, the Raspberry Pi will shutdown. The control program is not set to run on system startup. The program is initiated through an SSH command from the operating laptop. Thus, even if power is restored to the Raspberry Pi the aircraft will not regain control. Even if the voltage returns to normal the Raspberry Pi will not run the program.

4.2.2 Reliability Testing

The next step was running the control system through tests based on the failure mode analysis. A vibration test was done to test the vibration, short circuit, and power glitch possible failure modes. Endurance testing demonstrated the dead battery, arithmetic error, and coding bug modes. Wing position testing was for the potential high control delay, and wing position error. Finally, a communication test was performed to verify the Raspberry Pi SSH fail, dual telemetry error, and Pixhawk link loss.

4.2.2.1 Vibration Testing

Testing the controller's ability to deal with vibrations required first determining what vibrations one can expect in a flight test. The flight test data from the standard wing tests conducted in April 2018 provided the data to characterize the vibrations. The Pixhawk contains an accelerometer and records the accelerations in the x, y, and z direction at 960Hz. This raw data, however, contains error, as well as non-vibration movement. The data was filtered to determine the necessary frequencies and magnitude of the vibration. The data was put through a Fourier transform and a 1Hz high-pass filter to better characterize the vibrations. The magnitude of the Fourier transform was normalized to determine the amount of acceleration at the different frequencies. The x, y, and z accelerations where analyzed as shown in Figure 27, Figure 28, and Figure 29.

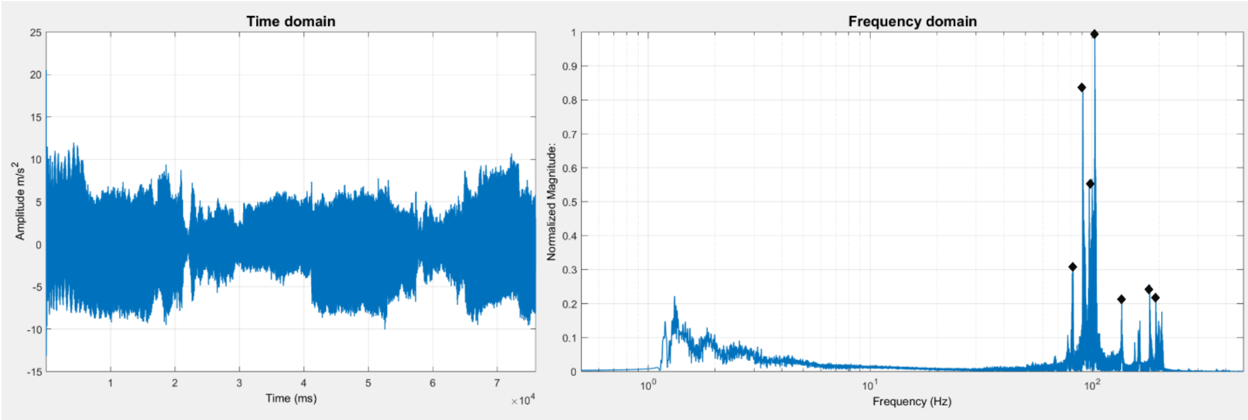

Figure 27: Filtered x-axis acceleration data in time (left) and frequency domain (right). The black markers in the frequency domain are the points chosen for the vibration testing.

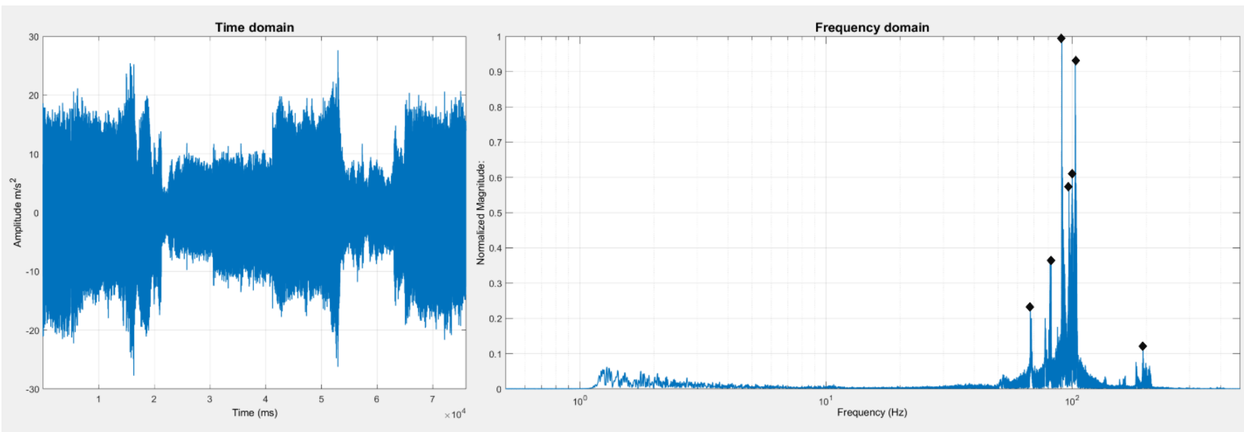

Figure 28: Filtered y-axis acceleration data in time (left) and frequency domain (right). The black markers in the frequency domain are the points chosen for the vibration testing.

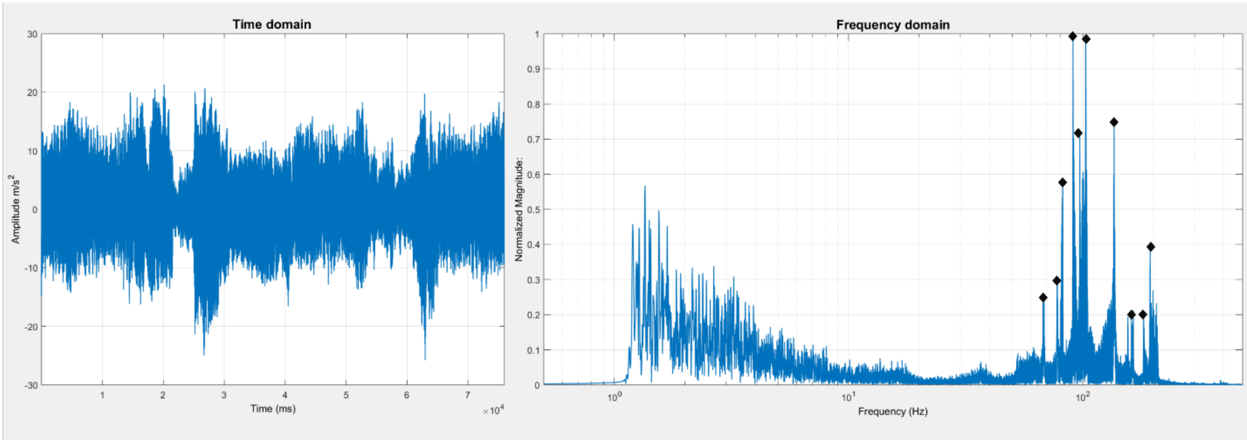

Figure 29: Filtered z-axis acceleration data in time (left) and frequency domain (right). The black markers in the frequency domain are the points chosen for the vibration testing.

The largest and most prominent peaks in the frequency domain were used to determine what frequencies and accelerations to use for the vibration testing. Table 3 shows the chosen frequencies and accelerations. The system was tested with 100% and 150% of the acceleration from the data. The 150% acceleration allowed for the possibility of higher vibrations, error in the recorded data, and not using every single amplitude and frequency on the actual flight. The used frequencies are also labeled in the three graphics shown previously. The vibration table was unable to perform the accelerations experienced at the lower frequencies, less than 5Hz. Although the lower frequencies showed some acceleration, these are most likely caused by flight movements and not vibrations.

Table 3: Vibration Table Frequencies and Acceleration

X-Axis			Y-Axis			Z-Axis		
Frequency (Hz)	100% m/s^2	150% m/s^2	Frequency (hZ)	100% m/s^2	150% m/s^2	Frequency (hZ)	100% m/s^2	150% m/s^2
82	3	4.5	67.6	5.8	8.7	67.6	5	7.5
90.8	8.5	12.75	81.4	9.15	13.725	77.8	6	9
97	5.42	8.13	90.9	25	37.5	82	11.6	17.4
102	10	15	97.1	14.2	21.3	90.8	20	30
136	2	3	100.1	15.3	23	97	14.5	21.75

182	2.5	3.75	102.9	23.2	34.8	102.9	19.6	29.4
193.6	2.2	3.3	193.7	2.85	4.275	136	15.2	22.8
						164.2	4.2	6.3
						181.8	4.2	6.3
						194	8	12

The next step was to set up the controller to run the warping wing control while it was being vibrated. Since we were not interested in the morphing wings ability to withstand vibrations, the wing was not attached to the controller and instead the PWM signal was fed out to a PWM reader and all data was recorded on the Raspberry Pi. The controller was attached to a wooden board via velcro and was hooked up to the same power supply as during the flight test. A Pixhawk autopilot was also included and run off of the same power supply. A RC receiver was linked to the system providing aileron commands, flight data, control gain settings, and mode settings, in the same manner as during a flight test.

4.2.2.1.1 No Vibrational Issues

The controller was strapped down to a vibration table and vibrated in the three axes one at a time, due to the capability limitations of the table. A picture of the setup for the y and z direction are shown in Figure 30 and Figure 31, respectively. The controller was vibrated at 100% and 150% acceleration in each axis for fifteen minutes. The controller performed without issue and no wires or connections came loose or dislodged. All three flight modes were executed during the vibration testing and the RC transmitter provided the necessary roll commands.

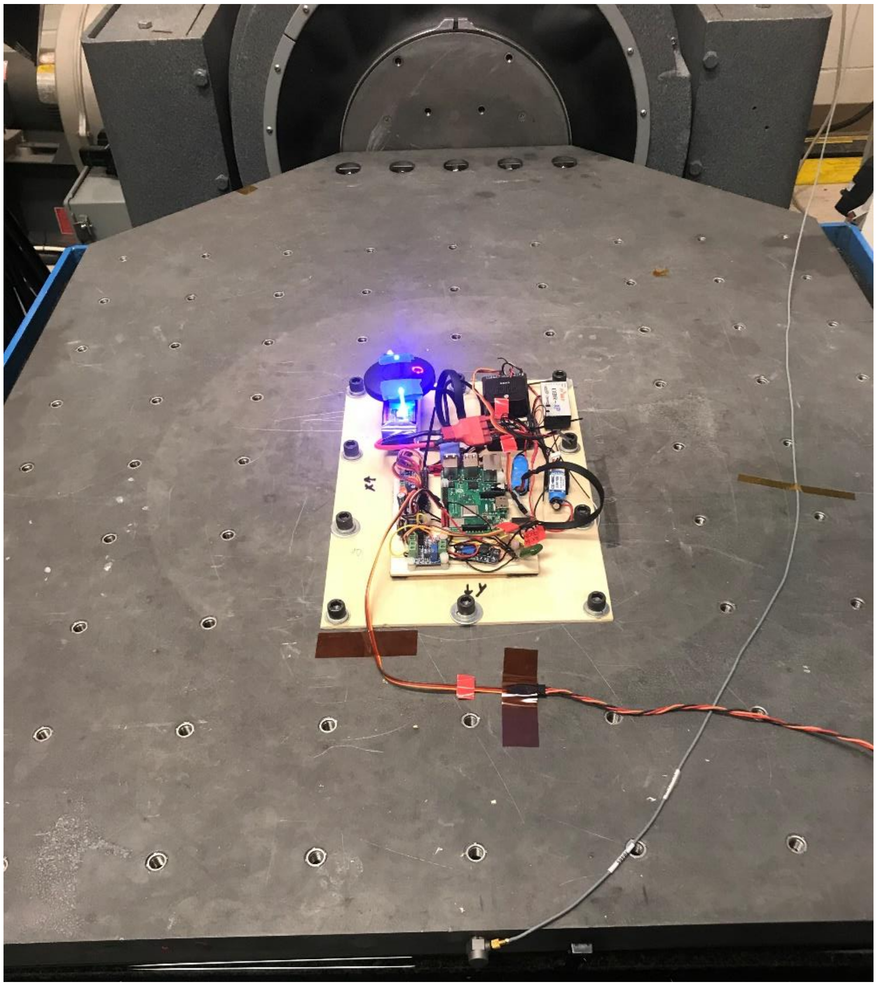

Figure 30: Vibration table testing of controller in the y-axis.

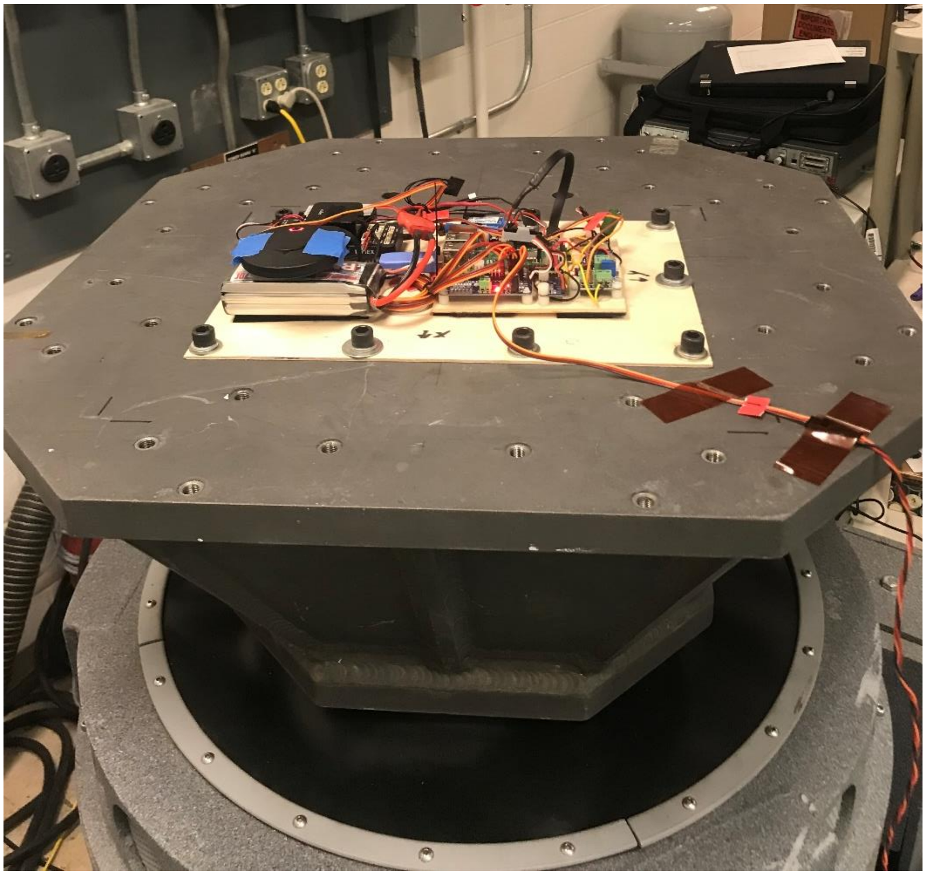

Figure 31: Vibration table testing of the controller in the z-axis.

4.2.2.2 Endurance Testing

The endurance testing was split up into two different types of endurance. One was the ability of the power supply to support both the controller and Pixhawk for the duration of a flight test. This, however, was proved during the vibration testing where, for over an hour and a half, the controller and Pixhawk ran off of the same battery supply that was in the flight test. As most flights run a maximum of 15 minutes, the power supply is more than sufficient. The other type of endurance tested was the ability of the controller to run through an entire flight test, receiving pilot inputs, flight data, and outputting servo commands, without issues.

The controller was set up so that it could run through simulated flights to test the control systems endurance. Test data from the standard wing from six different flights conducted during the April 2018 flight tests were used and run through the Raspberry Pis control program to simulate six flights. The April flights were around ten minutes each and were programed to loop through twice, allowing the wing to be tested for six twenty-minute intervals.

The Raspberry Pi was programed to only pull flight data at the same data rates as it would in a real flight: 4Hz for flight data and 50Hz for pilot inputs. Since in the standard wing testing the pilot inputs for mode changes, flaps, and control gain did not exist, a RC controller and receiver provided these inputs to the Raspberry Pi's program from an actual Pixhawk connection, the same as was designed for actual flight testing. A prototype warping wing was conncected to the controller during the endurance test. Having the actual wings in use during the testing provided valuable visual feedback on any potential performance issues. A picture of the endurance testing setup is shown in Figure 32..

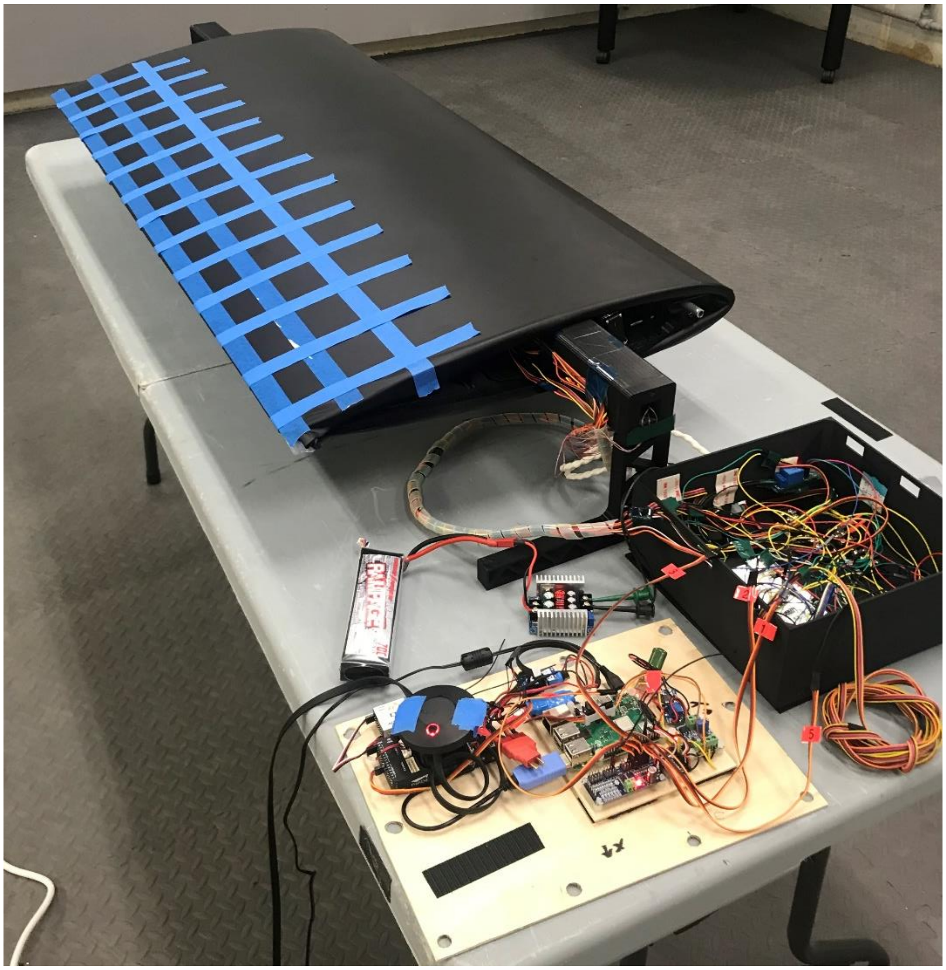

Figure 32: Morphing wing controller endurance test setup.

4.2.2.2.1 Dividing by Zero

The program encountered an error and shutdown, eliminating roll control for the wing during one of the endurance tests. It was found that under certain flight conditions the equations which determined the position of the active flaps in mode 3 would go to infinity and crash the program. the climb angle needed to solve for the lift coefficient in (5), takes the inverse sine of the vertical velocity over the airspeed, see (16). If the vertical velocity, V_z, is greater than the airspeed, V, then (16) goes to infinity and fails.

$$C_L = \frac{W\cos(\gamma)}{\frac{1}{2}\rho V^2 S\cos(\phi)} \tag{15}$$

$$\gamma = \sin^{-1}\left(\frac{Vz}{V}\right) \tag{16}$$

Even though (2) can cause the controller to fail, this condition was not the root cause of the failure during the flight test crash. At the time of the failure the airspeed was less than 10 m/s and the code was designed to not use (16) until airspeed was above 10 m/s as the equation is only needed while in flight. The potential for failure was fixed, however, by prohibiting the use of (2) in the code, when the vertical velocity is higher than the airspeed. Instead at $V_Z > V$ C_L is set to zero because at $V_Z = V$, $C_L = 0$.

4.2.2.2.2 Acrobatic Maneuvers

After fixing this source of potential failure, there were no program ending issues while running the Endurance tests. However, it was found that the active flaps would sometimes change rapidly while in mode 3 during some portions of some of the simulated flights. Quick movements, such as those displayed during the simulation, might cause instability and make it difficult to control the aircraft. It was determined that these rapid active-flap changes occurred due to acrobatic maneuvers, including barrel rolls, that were performed during some of the standard wing flight tests. Because the Pixhawk autopilot system only transfers flight data at 4Hz, the control system is unable to keep up with rapid maneuvers while in mode 3. While simulating maneuvers that would occur during a standard flight, there were no incidents with the active flaps. We conclude that acrobatic maneuvers should not be conducted while utilizing Mode 3 of the morphing wing controller.

4.2.2.2.3 Serial Connection Constraints

Also included in endurance testing was an evaluation of the connections of the ground station and Raspberry Pi to the Pixhawk. The Pixhawk has two ports for transmitting data. One of these ports is connected to the Raspberry Pi, and the other is hooked up to a modem, which broadcasts telemetry data to the ground station. The Raspberry Pi encounters issues when two different devices are connected to these ports. Anytime a device is connected to the Pixhawk, a series of heartbeats are transmitted back and forth. The Pixhawk will send out requested heartbeats on both ports, thus the two devices will receive requested responses from each other. DroneKit on the Raspberry Pi gets confused by these additional heartbeats and constantly sends error messages out to the user. The endurance testing showed that these errors do not cause the control program to fail but it could cause issues in the future. Another problem is that if the telemetry link on the ground is set up first the heartbeat error messages prevent DroneKit from connecting to the Pixhawk. When using the ground station for a telemetry link, it must be connected after the control program has started.

4.2.2.3 System Reaction Time Testing

An important aspect of designing the control system was reducing the reaction time between pilot input and deflection of the wing. When operating a UAV, the reaction time of the control surfaces is not instantaneous. The control signal must be transported through a RC transmitter and receiver before the receiver outputs a PWM signal to the servos controlling the wing. The amount of delay in this process is important to keep below a quarter of a second so that the aircraft can be controlled effectively[30]. A test was performed to determine all the sources of latency within the system. A motion capturing VICON chamber was utilized to measure the differences between pilot inputs and the movement of the wing. The VICON chamber provides 3D position coordinates of small reflective markers over time.

In the motion capture test, a reflective marker was placed on the RC transmitter's aileron stick, three trackers where placed on the warping wing, and three trackers where placed on the standard wing. A picture of the motion capture test setup is shown in Figure 33. The RC transmitter, standard wing, and morphing wing were all secured down to avoid any extra movement causing error in the position data. The RC receiver was connected to both the Morphing wing control system and the servo controlling the standard wing. This setup allowed a direct comparison of the two systems' reaction times and for determining all the latencies of the two systems.

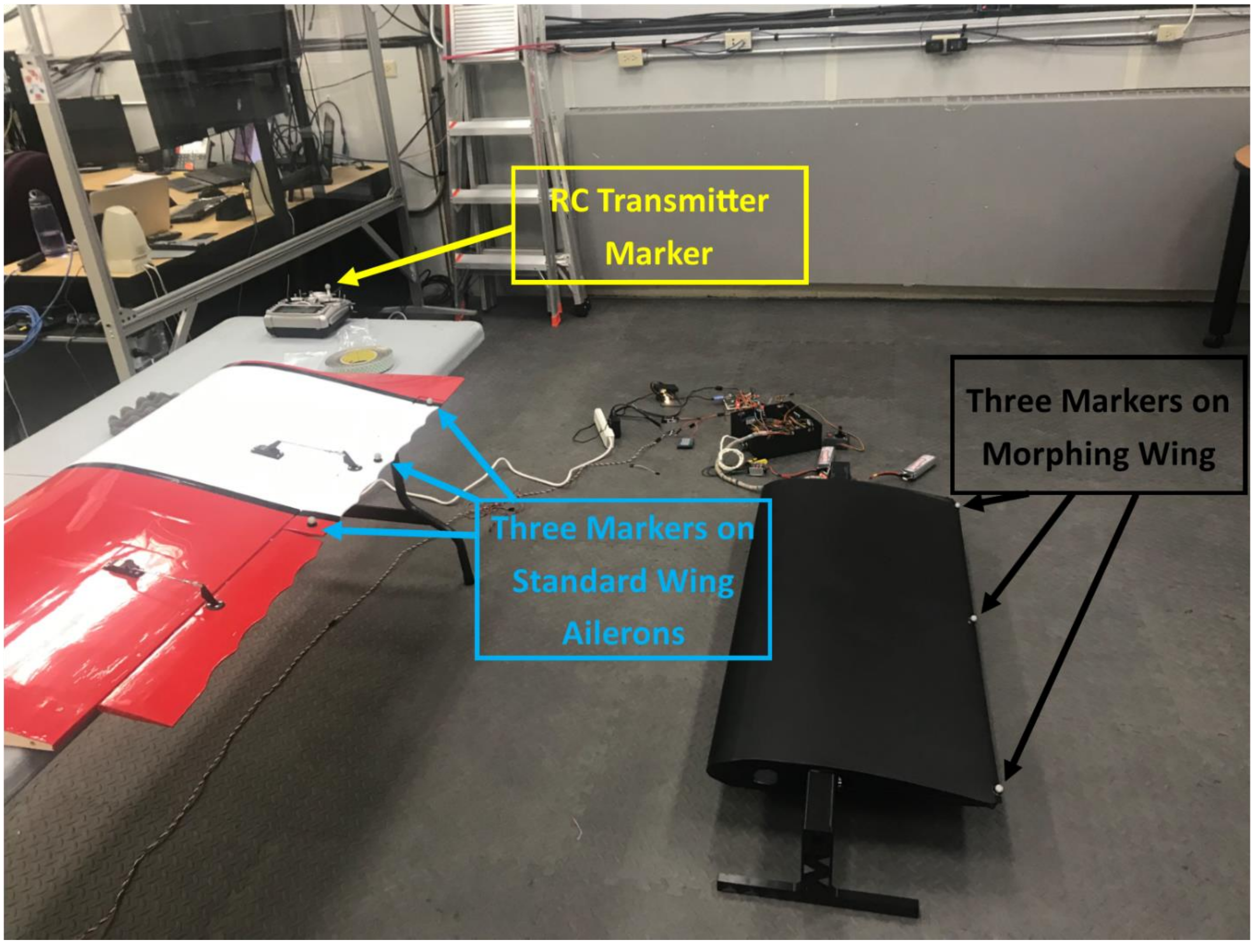

Figure 33: Motion capture test setup and coordinate frame for x, y and z motion.

4.2.2.3.1 Reaction Test – No Pixhawk

The motion capture system recorded the 3D position in x, y, and z of all the markers with the Standard wing and Morphing wing receiving the same RC input. For simplification, only the y-axis was used for the RC transmitter, and only the z-axis was used for the wing movement. While using a single axis captures most of the movement, there is a slight z-axis movement in the stick and a slight y-axis movement in the wing surfaces. The results of the test are shown in Figure 34. Here, one can see the difference in delay between the stick movement and the deflection of the morphing wing and the standard wing. These movements were normalized to the stick movement to more easily visualize the differences in response times. In this test the control gain on the morphing wing was set to 30%. You need to quantify the time differences observed in this figure here.

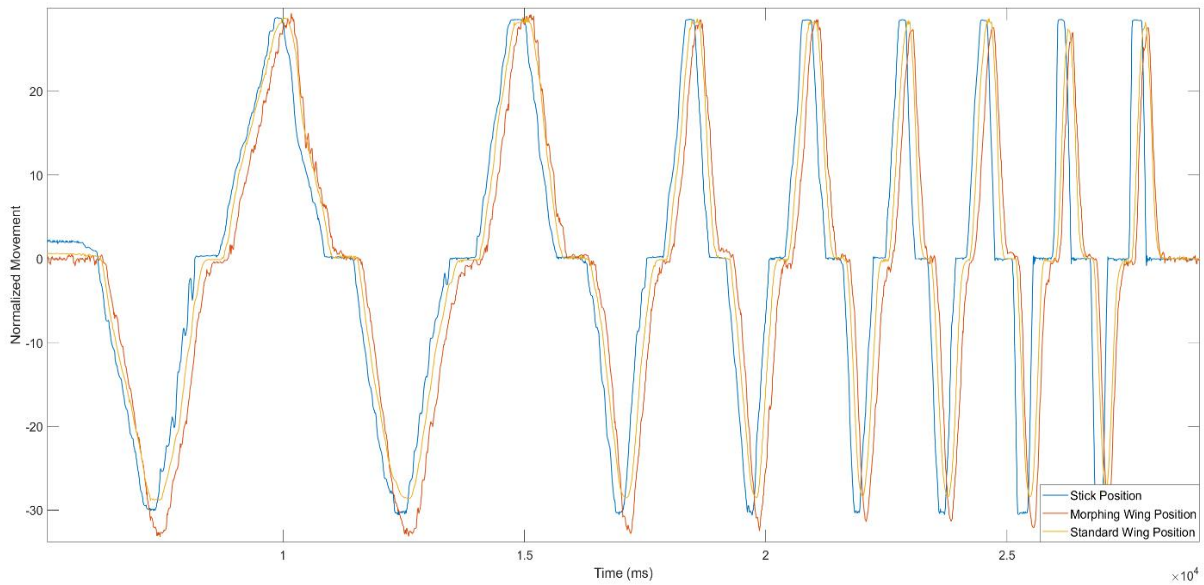

Figure 34: Wing movement response time comparison using direct RC connection

4.2.2.3.2 Reaction Test – Including Pixhawk

Tests were also performed to measure the delay when using the standard and morphing wings with the aileron commands first going into the Pixhawk. This data is useful because of the future goal of running the wing using the Pixhawk autopilot system to control the aircraft. Results of the Pixhawk feedthrough is shown in Figure 35 and Figure 36. Also, in the case of the Morphing wing motion test in Figure 36, at approximately 45 seconds into the test, the control gain was switched to 60%. In Figure 36, when the wing was moved towards the ground in the z direction, the motion capture system lost track of the wing, causing holes in the data at the 45-65 second mark above 30mm in movement.

In the two graphs below, the movements are not normalized and are actual distance values.

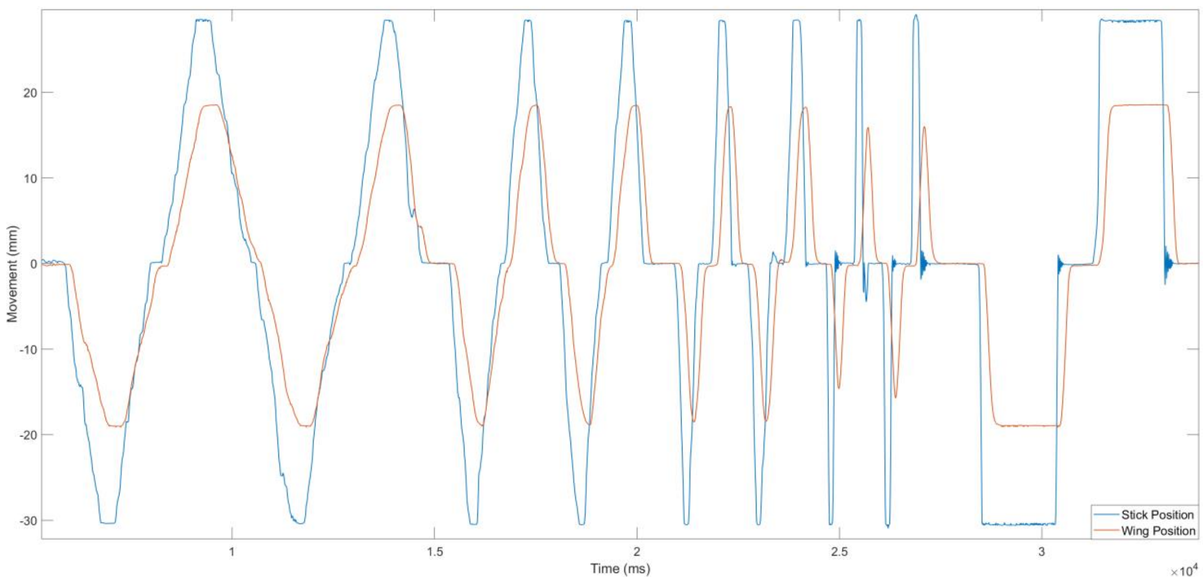

Figure 35: Standard wing movement compared with RC transmitter stick movement with Pixhawk feedthrough.

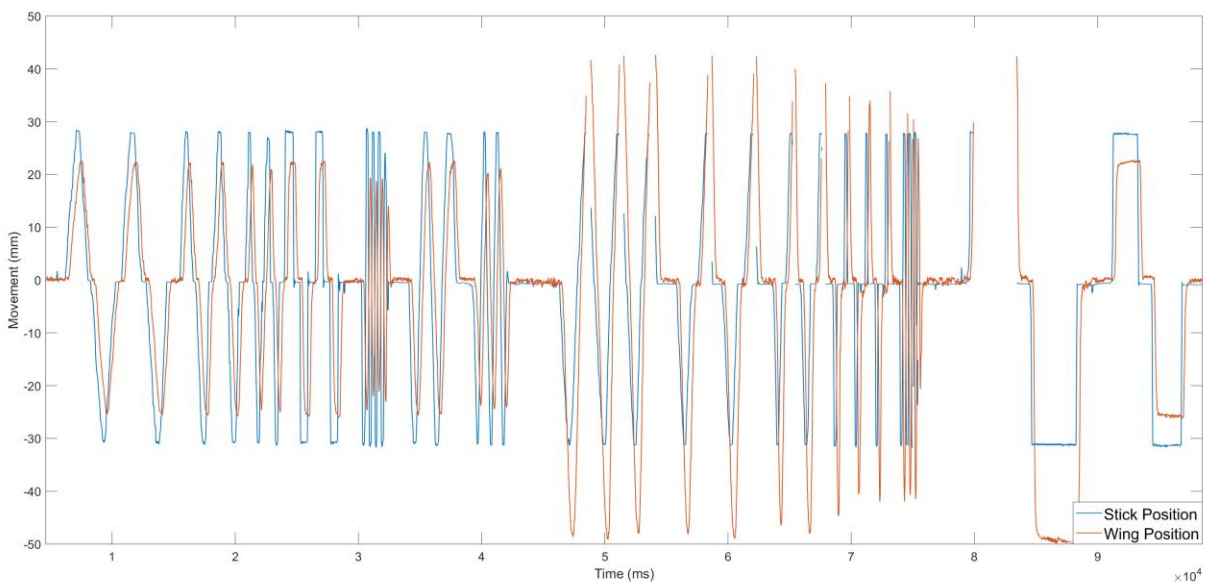

Figure 36: Morphing wing response time with Pixhawk feedthrough. At the 45 second mark the control gain is switched from 30% to 60%.

4.2.2.3.3 Overall Delay Results

The amount of delay was calculated by taking the difference in time between when the stick first moves and when the wing first moves using data from all the tests. The average of five different points was taken and the results are displayed in Table 4. Additionally, this table displays the delay caused by just the Pixhawk, the morphing wing controller, and the wing movement. The delay due to the Pixhawk was calculated by taking the difference in delay between the morphing wing and the standard wing, using the Pixhawk and the RC receiver. Also calculated was the amount of delay caused by the movement of the wing surface. This value was calculated by measuring the delay time from when the stick reaches its maximum point to when the aileron reaches its maximum point. Because the morphing wing surface requires more force to move, it was thought that this might generate a larger delay than on the standard wing. However, it was found that the movement delay was similar when the morphing wing control was set to a 30% gain.

Table 4: Approximate Delay from Pilot Input to Wing Movement

Method	Delay (ms)
Standard Wing Directly through RC Receiver	68
Standard Wing Pixhawk Passthrough	138
Morphing Wing Controller RC Receiver Input	155
Morphing Wing Controller Pixhawk Passthrough	238
Morphing Wing ½ Low-pass Filter Pixhawk Passthrough	223
Estimated Delay Due to Pixhawk	70-85
Estimated Delay Due to Morphing Wing Controller	87
Estimated Delay Due to Morphing Wing Movement (30% gain)	50
Estimated Delay Due to Standard Wing Movement (full motion)	65

It was found that feeding the pilot aileron command through the Pixhawk generates an extra amount of delay; this can be seen in the difference between the first two rows of Table 2 (and on row 6). When using the standard wing the added delay due to the Pixhawks is still well under the quarter-second threshold. However, when using the morphing wing controller, the addition of the Pixhawk causes the delay to become very close to the quarter second threshold, per row 4. At 0.238 seconds of delay, the pilot might begin to have a hard time controlling the aircraft. When using the morphing wing controller with direct RC, the amount of delay is at a controllable level.

4.2.2.4 Wing Position Vibration

In addition to capturing the wings movement delay, the motion capture testing also exposed the error created by using the PWM-to-analog conversion. Figure 37 shows the amount of erroneous movement in the wings surface while the wing is set to neutral. The figure compares the amount of error created by the standard wing, the morphing wing, and the morphing wing with the low-pass filter strength halved. The weaker low-pass filter was tested to see how much control delay could be saved. As previously shown in Table 4, halving the filter only saved 15 milliseconds. However, using the half filter produces a large increase in the amount of error produced. Included in Figure 37 are the standard deviations of the movement errors. The error produced in the standard wing is the smallest, with a standard deviation of 0.19 mm, as the RC receiver PWM signal is directly fed to the servo. Using the morphing wing with a full filter at 100% control gain the standard deviation was 0.73 mm. Using half of the filter produces a standard deviation of 1.46 mm. Thus, halving the filter only decreases delay by 15 milliseconds while doubling the amount of error in the wing position. Either way the amount of error produced is small compared to the 50mm of deflection that occurs at full aileron at 60% control gain.

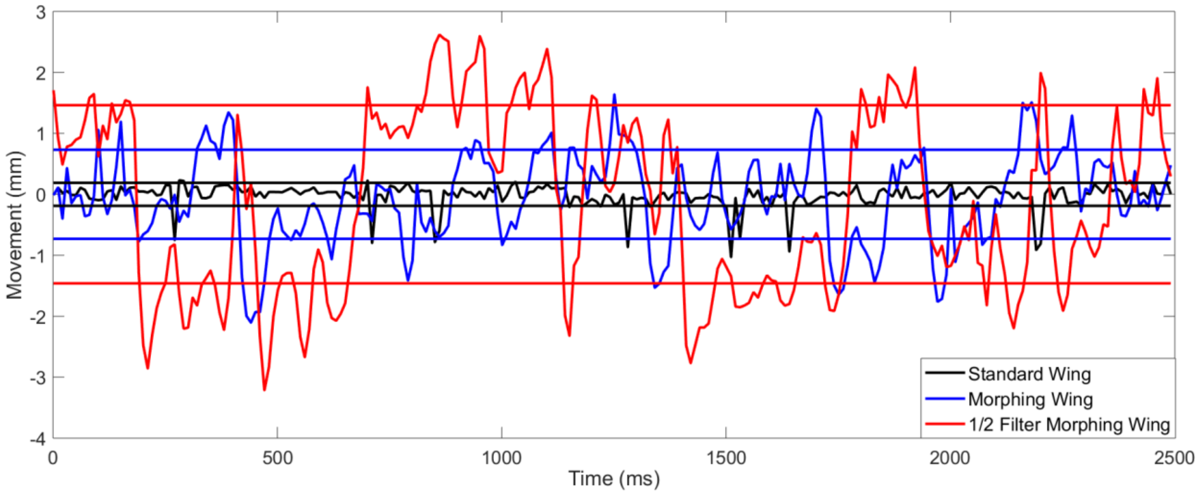

Figure 37: Wing position error and standard deviation comparison of standard, morphing, and morphing wing with the low-pass filter coefficient halved

4.2.2.5 Raspberry Pi Connection and Program Initiation Testing

Another possible source for failure was the initial connection required between the operating laptop and the Raspberry Pi to start the controller program. In order to initiate the control program the operating laptop has to be connected to the Raspberry Pi through Wi-Fi and controlled through SSH. After a connection is established the “nohup” command must be used to run the control program in the background. Testing was performed by repeatedly connecting and disconnecting to laptop and the Raspberry Pi, in order to rule out sources of errors. During the testing it was found that if the Linux “nohup” command was entered improperly, there was no feedback supplied to the operating laptop that an error occurred. Furthermore, if the disconnection between the Raspberry Pi and the operating laptop was initiated by physically disconnecting the laptop from the wireless router, then the Raspberry Pi would continue to operate the control program as long as a wireless signal maintained connection. When the laptop is disconnected from the Pi properly by using the logout command the control program will only run if nohup was used correctly. Thus, if one types in the nohup command wrong, initiates the program, and then physically disconnects the router, the program will not be running in the background and will continue to run until the Raspberry Pi loses Wi-Fi connection.

4.2.2.5.1 Root Cause of Flight Test Control Failure

The connection problem described was found to be the root cause of the failure of the control system on the flight test crash. In preparation for the flight test, the nohup command was entered incorrectly by typing the slashes in the wrong direction. In windows a file location looks like this: C:\home\documents while in linux it is C:/home/documents. Then the control program was run, and the laptop cable to the router was physically disconnected. The warping wing control program continued to operate until several minutes later when the aircraft was far down the runway and the Raspberry Pi lost Wi-Fi connection. In-between the failure and the flight test, there was no indication that the control program was not running, as the aileron was never activated.

4.2.3 Failure Mitigation

Based on the failure mode analysis and testing, changes were made to the system to help mitigate risk and decrease the effects of potential failures, not just correcting the root cause.

4.2.3.1 Vibration Failure Mitigation

Although the controller was tested through potential vibrations and had no issues, one cannot eliminate all possible movements that could occur on a real flight test. Soldered connections should be used where possible and all other connections should be glued to increase the controller's ability to withstand vibrations and forces due to flight maneuvers. Also, the Raspberry Pi was set to run mode 1 on start up without a Pixhawk connection, therefore decreasing the amount of time it takes to start the program in the event of a power glitch, which would stop the program. The Raspberry Pi would also automatically run the program in the event of a restart. However, it was found that restarting in this setup still requires thirty seconds, which limits the benefit, as It would be difficult for the aircraft to stay in the air for the required time.

4.2.3.2 Code Failure Mitigation

To increase the reliability of the system within the code, several layers of coding were added to enable the program to fail but continue by running a simpler program. Three separate coding loops were created; the first loop was the standard program which ran first. The second loop is a safety mode used if the first loop failed. The third loop just sends the wings to the neutral position. If the main program encountered an error, the second loop would initiate causing the controller to operate in mode 1 and never connect to the Pixhawk. The control gain must be hardcoded and the pilot cannot change modes or use flaps when the Raspberry Pi is not connected to the Pixhawk. Without the need for the Pixhawk and using no complex algorithms to determine wing position, this mode would have a lower chance for error. If the aircraft enters this control loop, the pilot will still have control of the aircraft and will be able to land the aircraft safely.

If the second control loop fails then the aircraft would enter the third loop in which the wing surfaces would return to the neutral point. This ensures that if the code fails mid maneuver the ailerons will not be stuck in the last position that they were commanded. Even with the rudder and elevator functioning properly, if the ailerons are stuck in a full right or left roll, the aircraft will be uncontrollable. While this final mode would mean loss of all roll control, having the ailerons stuck in neutral position would at least increase the chances of landing the aircraft safely.

4.2.3.3 Failure Detection

Another problem that occurred with the failed flight test was that even though the program had ended, there was no indication to the pilot that they did not have roll control of the aircraft prior to takeoff. Had the pilot knew there was a problem with the controller, a takeoff would never had occurred. To mitigate this problem an indication light was added in order to provide visual feedback that the program is operating properly. If the program fails during flight testing this light will turn off and signal to the pilot that there is an issue. If the indicator light turns off while on the ground, a takeoff will not happen, and if in the air it will let the pilot know they have lost roll control and to land the aircraft as soon as possible. The sooner the pilot knowns that there is an issue with control, the more likely they will be able to land the aircraft safely.

4.2.3.4 Communication Failure Mitigation

A system startup manual was created to decrease the chance of a communication issue causing a failure. Before a flight test, strict compliance with this startup will ensure that the code is implemented properly and avoid the problem which caused the failure during the flight test. Table 5 shows the steps required to connect to the Raspberry Pi and run the program. This list will help future users run the controller. Using a computer to run mission planner and connect to the Pixhawk and stream telemetry down is listed as optional because it is not required to operate the aircraft. Another method of ensuring the command is used correctly would be to create a script to run the required commands.

Table 5: Control System Setup Manual	
Step	Description
1.) Connect to Raspberry Pi	Using computer which is directly hooked up to the router, use SSH to connect to the Raspberry Pi.
2.) Run Control Program	In SSH terminal type: “nohup python /home/pi/WarpW4.py”
3.) Telemetry Connection to Pixhawk (optional)	Using another computer, connect to the Pixhawk telemetry by using mission planner.
4.) Logout of Raspberry Pi	When the program has entered the control loop, type “logout” in the SSH terminal.
5.) Test Controller	After logout. make sure program is working in background by having safety pilot activate the aileron command.
6.) Re-connect to Raspberry PI	After completion of test, reconnect to the aircraft using SSH.
7.) End Control Program	The control program will still be running in the background of the Pi. To disconnect, type “sudo PS -A” into the SSH terminal. Then find the python process in the list, and type “sudo kill <python process number>” .

Chapter V: Conclusions

In this research a device capable of controlling a morphing wing on a UAV was developed. The controller was able to meet the design requirements of taking pilot inputs and flight data and outputting a control signal to the six actuators in the wing. The controller was made sufficiently small and light enough to fit within the space of a UAV fuselage and integrate with the autopilot system and existing battery power supplies. The control system was also able to communicate with the ground station and record relevant flight data.

The flight test plan and follow-on data analysis was not able to be fully implemented because of a crash shortly after takeoff on the first flight. A successful taxi test did occur and demonstrated the ability of the aircraft to reliably move on the ground. Post-flight test analysis showed that the failure was due to the combination of a failure of the morphing wing skin as well as the control system shutting down before takeoff. The controller was working during taxi testing and was able to record data.

The root-cause of the controller failure was found during the post-test analysis. The control system was designed to be accessed remotely through Wi-Fi to initiate the control program and to be able to diagnose any issues. Post-crash connection testing found that during the flight test the initiation command for the software was used incorrectly, causing a confusing and misleading state. The control program acted as though everything was working fine, however the program was not actually running in the background. The program would only remain functional while the aircraft was in Wi-Fi range of the base station. The program did not fail until the team was in the middle of taxi testing, and the failure was not detected by the ground crew or the safety pilot.

5.1 Failure Mode Analysis

An extensive failure mode analysis was performed to determine possible causes of failures to make the system more robust and less-likely to crash during the next flight test event. Possible sources of error were broken down into software, hardware, flight testing procedures, communication, and electrical. Several tests were performed to determine if the possible errors where viable and find addition sources of error.

A vibration test was carried out to determine if the control system could withstand the many sources of vibration involved during a flight. The results showed that the control system was not susceptible to vibration related failures, although adding glue to non-soldered connections would further increase reliability.

Another test was performed to verify the ability of the operating laptop to communicate with the controller and initiate the program. This test was the one to identify the root cause and the required steps needed to avoid the issue in the future.

Two system endurance tests were performed to verify the power supply and the ability of the control system to work throughout a long flight. The power system showed that it was capable of supplying power for over an hour and a half, far more than required for the 15-minute flights. The results of the control system endurance test showed that there was a potential cause of failure when the vertical velocity was higher than the airspeed, a scenario that indeed took place during some acrobatic flights with the standard wing. An arithmetic error in this situation would end the program. After this issue was fixed no other problems occurred and the control system was able to operate for two continuous hours by playing back data from six older flight tests.

A motion capture test was performed to determine if delay between pilot input and morphing wing movement and to see if the delay was high enough to cause instability. The morphing wing and standard wing were both connected to the control system and their positions tracked while performing various maneuvers. The testing showed that there is an estimated 87ms delay due to the morphing wing control system. It was shown, however, that the total delay was still within a safe and controllable range. Using the Pixhawk to feedthrough pilot input commands did add enough delay that the Pixhawk and controller likely cannot be used in combination. With the Pixhawk feedthrough, the total delay was 238ms, which is high enough to potentially cause problems. This is important to note because future testing may involve using the Pixhawk autopilot to control the aircraft instead of a safety pilot.

The motion capture testing also showed that the morphing wing control system adds noise to the output signal and causes the morphing wing to vibrate. This error is unlikely to cause issues with controlling the aircraft as it has a standard deviation of less than a millimeter.

After the failure mode testing several changes were added to the control system to improve the reliability of the controller. One change was to have the controller program run upon startup if the program restarted unexpectedly during flight. Another change was to add an indicator light to the controller that would be visible to the safety pilot if the controller stops working properly. Also, in the case of a coding failure the program will try to run a simplified version of the code before stopping.

To decrease the likelihood of a failure from initiation of the program using the operating laptop, a control system setup manual was created for future testing. If followed correctly, the failure that occurred during the flight test will not occur again.

5.2 Future Work

It was unfortunate that the flight test was a failure and the data required for characterizing the morphing wing was not obtained. However, there are plans for AFRL, AFIT, and USU to continue testing the morphing wing including another flight test in the future. Another morphing wing is being developed by AFRL to fix the issue which caused the wing to expand after takeoff. A test will be performed with the morphing wing attached to a truck and driven down a runway at speed to verify that the expansion problem has been solved.

A future goal is adding yaw control to the morphing wing, thus replacing the function of the rudder. This would require inputting pilot rudder commands in addition to aileron commands and adjustments would need to be made to the aerodynamic equations. Another future test could be to use the Pixhawk autopilot system to control the aircraft instead of a pilot.

Further reliability testing of the control system would help reduce the likelihood of another failure. One such test could be to fly the morphing wing control system in the standard wing base aircraft. The control system would not be controlling the aircraft but could still be running and collecting data as if it were. This would be a low risk way of verifying that the control system can run a full flight test.

There are several potential improvements that could be done to the control system. The Pixhawk was a simple way to provide active flight test data to the controller as AFIT has worked with the Pixhawk extensively in the past. However, the Pixhawk was never designed to supply active data for a controller. The data rate over the telemetry port is only at 4Hz. At that data rate during fast maneuvers the wing control surface will move in jumps instead of smoothly transitioning. Future work can be done to either try to increase the Pixhawk data rate or to replace the Pixhawk with some other system.

Another potential improvement would be to find a better method of sending the pilot aileron commands to the control system. While the method used in this study worked, there was an increase in noise and delay in the signal due to the filter. One method could be to send the PWM signal from the RC receiver directly to the Raspberry Pi and use the system clock to identify the rising and falling edges of the signal. Another could be to use another controller with a clock dedicated to identifying the PWM signal and then sending the data to the Pi. These two methods could prove to be better than using the PWM-to-analog method used in this research.

Appendix

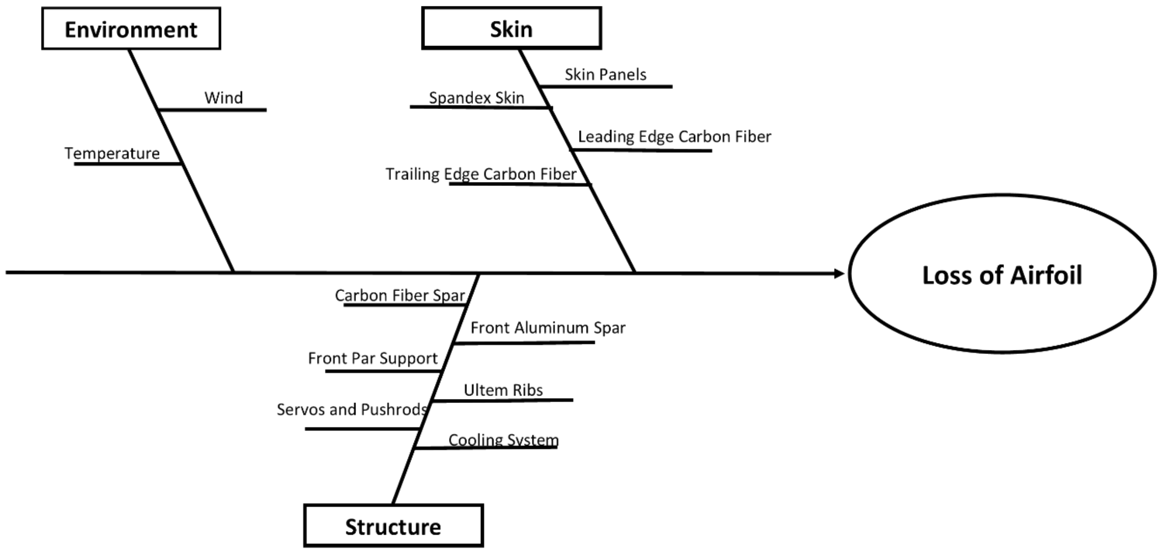

Figure 38: Failure Mode Tree for Morphing wing created by AFRL.

Bibliography

[1] S. Barbarino, O. Bilgen, R. M. Ajaj, M. I. Friswell, and D. J. Inman, "A review of morphing aircraft," *J. Intell. Mater. Syst. Struct.*, vol. 22, no. 9, pp. 823–877, 2011.

[2] D. F. Hunsaker and W. F. Phillips, "Aerodynamic Shape Optimization of Morphing Wings at Multiple Flight Conditions," no. January, pp. 1–14, 2017.

[3] D. F. Hunsaker, Z. S. Montgomery, and J. J. Joo, "Lifting-Line Analysis of Wing Twist to Minimize Induced Drag During Pure Rolling Motion," *AIAA Scitech 2019 Forum*, no. January, pp. 1–24, 2019.

[4] P. D. R. Santos, D. B. Sousa, P. V. Gamboa, and Y. Zhao, "Effect of design parameters on the mass of a variable-span morphing wing based on finite element structural analysis and optimization," *Aerosp. Sci. Technol.*, vol. 80, pp. 587–603, 2018.

[5] W. W. Gilbert, "Development of a mission adaptive wing system for a tactical aircraft," *AIAA Aircr. Syst. Meet.*, 1980.

[6] J. Kudva *et al.*, "Design, fabrication, and testing of the DARPA/Wright Lab 'smart wing' wind tunnel model," *38th Struct. Struct. Dyn. Mater. Conf.*, no. 1, 1997.

[7] J. A. Hetrick, R. F. Osborn, S. Kota, P. M. Flick, and D. B. Paul, "Flight Testing of Mission Adaptive Compliant Wing," *48th AIAAASMEASCEAHSASC Struct. Struct. Dyn. Mater. SDM Conf.*, no. April, pp. 1–17, 2007.

[8] Y. Jo, S. Choi, L. Zientarski, and J. J. Joo, "Aerodynamic Characteristics and Shape Optimization of a Variable Camber Compliant Wing," *34th AIAA Appl. Aerodyn.*

Conf., no. June, pp. 1–19, 2016.

[9] J. J. Joo, C. R. Marks, L. Zientarski, and A. J. Culler, “Variable Camber Compliant Wing - Design,” *23rd AIAA/AHS Adapt. Struct. Conf.*, no. January, pp. 1–14, 2015.

[10] C. R. Marks, L. Zientarski, and J. J. Joo, “Investigation into the Effect of Shape Deviation on Variable Camber Compliant Wing Performance,” *AIAA SciTech Forum* , no. January, pp. 1–15, 2016.

[11] C. R. Marks, L. Zientarski, A. J. Culler, B. Hagen, B. M. Smyers, and J. J. Joo, “Variable Camber Compliant Wing - Wind Tunnel Testing,” *23rd AIAA/AHS Adapt. Struct. Conf.*, no. January, pp. 1–16, 2015.

[12] L. Prandtl, “REPORT No . 116 APPLICATIONS OF MODERN HYDRODYNAMICS,” 1923.

[13] W. F. Phillips, D. F. Hunsaker, and J. J. Joo, “Minimizing Induced Drag with Lift Distribution and Wingspan,” vol. 2, no. 1, pp. 1–11, 2018.

[14] W. F. Phillips, “Aircraft Equations of Motion,” in *Mechanics of Flight*, John Wiley & Sons, Inc, 2004, pp. 599–690.

[15] G. Franklin, D. Powell, and M. Wrokman, *Digital Control of Dynamic Systems*, Third. Ellis-Kagle Press, 2006.

[16] Mathworks Matlab, “Identify Linear Models Using System Identification App,” 2018. [Online]. Available: https://www.mathworks.com/help/ident/gs/identify-linear-models-using-the-gui.html. [Accessed: 25-May-2018].

[17] W. F. Phillips, “Linearized Lateral Dynamics,” in *Mechanics of Flight*, John Wiley & Sons, Inc, 2004, pp. 765–776.

[18] B. Blanchard and W. Fabrycky, "Design for Reliability," in *Systems Engineering and Analysis*, Fourth., W. Fabrycky and J. H. Mize, Eds. Pearson Prentice Hall, 2006, pp. 394–400.

[19] W. and O. Wright, "Flying Machine Automatic Stabilizer," U.S. Patent 821;393, 1913.

[20] F. Afonso, J. Vale, F. Lau, and A. Suleman, "Performance based multidisciplinary design optimization of morphing aircraft," *Aerosp. Sci. Technol.*, vol. 67, pp. 1–12, 2017.

[21] W. Su, S. S.-M. Swei, and G. G. Zhu, "Mission adaptive wing shape determination for highly flexible aeroelastic aircraft," *15th Dyn. Spec. Conf.*, no. January, pp. 1–22, 2016.

[22] A. Moosavian, F. Xi, and S. M. Hashemi, "Design and Motion Control of Fully Variable Morphing Wings," *J. Aircr.*, vol. 50, no. 4, pp. 1189–1201, 2013.

[23] J. B. Samuel and D. J. Pines, "Design and Testing of a Pneumatic Telescopic Wing for Unmanned Aerial Vehicles," *J. Aircr.*, vol. 44, no. 4, pp. 1088–1099, 2007.

[24] D. Inman, K. Kochersberger, O. Bilgen, E. Diggs, and A. Kurdila, "Morphing Wing Aerodynamic Control via Macro-Fiber-Composite Actuators in an Unmanned Aircraft," no. May, pp. 1–17, 2012.

[25] O. Bilgen *et al.*, "A novel unmanned aircraft with solid-state control surfaces: Analysis and flight demonstration," *J. Intell. Mater. Syst. Struct.*, vol. 24, no. 2, pp. 147–167, 2013.

[26] J. Kimaru and A. Bouferrouk, "Design, manufacture and test of a camber morphing

wing using MFC actuated mart rib," *2017 8th Int. Conf. Mech. Aerosp. Eng. ICMAE 2017*, pp. 791–796, 2017.

[27] O. Bilgen, M. Friswell, K. Kochersberger, and D. Inman, "Surface Actuated Variable-Camber and Variable-Twist Morphing Wings Using Piezocomposites," no. April, pp. 1–13, 2014.

[28] L. L. Gamble, A. Moosavian, and D. J. Inman, "Effects of Speed on Coupled Sweep and Camber in Morphing Wings," no. January, pp. 1–10, 2017.

[29] Z. S. Montgomery and D. F. Hunsaker, "A Methodology for Roll Control of Morphing Aircraft," no. January, pp. 1–18, 2019.

[30] M. E. Lloyd, "The Effect of System Lag on Unmanned Air System Internal Pilot Manual Landing Performance," 2012. Dissertations and Theses. 93

[31] R. Keim, "Low-Pass Filter a PWM Signal into a Analog Voltage," *All About Circuits*, 2016. [Online]. Available: https://www.allaboutcircuits.com/technical-articles/low-pass-filter-a-pwm-signal-into-an-analog-voltage/. [Accessed: 22-Apr-2018].

[32] T. DiCola, "Raspberry Pi Analog to Digital Converters," *Adafruit*. [Online]. Available: https://learn.adafruit.com/raspberry-pi-analog-to-digital-converters/overview. [Accessed: 15-May-2018].

[33] B. Earl, "Adafruit PCA9685 16-Channel Servo Driver," *Adafruit*. [Online]. Available: https://learn.adafruit.com/16-channel-pwm-servo-driver?view=all. [Accessed: 15-May-2018].

[34] "DroneKit-Python's Documentation." [Online]. Available:

http://python.dronekit.io/. [Accessed: 15-May-2018].

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
21-03-2019	Master's Thesis	August 2017 - March 2019

4. TITLE AND SUBTITLE

Active Control of a Morphing Wing Aircraft and Failure Analysis for System Reliability

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Montgomery, Madison J., Capt

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
Wright-Patterson AFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-MS-19-M-045

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL Advanced Structural Concepts Team
James J. Joo, Ph.D.
2790 D St.
WPAFB, OH 45433
james.joo.1@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RQVS

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution Statement A.
Approved for public release:Distribution Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A morphing wing aircraft has the ability to increase the efficiency of an aircraft by better optimizing lift and drag characteristics during a flight. A morphing wing UAV was designed and constructed by AFRL/RQVS and required a means of control and method of characterizing the performance of the aircraft through flight testing. This research presents the design and construction of a control system capable of adjusting the morphing wing shape based on pilot commands and current flight status. The control system was tested and improved following a flight test crash utilizing failure mode analysis.

15. SUBJECT TERMS

morphing wing control, aircraft control, UAV flight test, active aircraft control

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	UU	108	Dr. Robert Leishman AFIT/ENG

19b. TELEPHONE NUMBER *(Include area code)*

(937) 255-3636 x4755 robert.leishman@afit.edu

Reset

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g. 30-06-1998; xx-06-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATES COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33615-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report, e.g. AFOSR-82-1234.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. 61101A.

5d. PROJECT NUMBER. Enter all project numbers as they appear in the report, e.g. 1F665702D1257; ILIR.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, J, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/ monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/ restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.