REAL TIME HARDWARE AND SOFTWARE SYSTEMS FOR MICRO AIR VEHICLE FLIGHT CONTROL TESTING

by

CHRISTOPHER DALE MCMURROUGH

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2010

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor, Dr. Frank L. Lewis, for his endless dedication to his students. Working with Dr. Lewis has been one of the many great blessings of my life, and for that I am sincerely grateful. I would also like to thank my defense committee co-chair, Dr. Sajal K. Das, for his encouragement and wisdom. His guidance over the course of my graduate study is greatly appreciated.

I would also like to thank the brilliant men and women of the Air Force Research Laboratory; in particularly Dr. Dave Doman. Dr. Doman is the core source of inspiration for this thesis. He is an exceptional engineer and an equally exceptional mentor. Dr. Siva Banda is also thanked for facilitating my summer research activities at Wright Patterson AFB.

Additionally, I would like to thank Dr. Harry Stephanou of the Automation and Robotics Research Institute. Without his enthusiastic support, this work would not have been possible. Also recognized are my colleagues in the ACS lab for sharing with me their skills, common passion for robotics, and friendship. I have learned a great deal from these truly gifted people.

I would also like to acknowledge the Computer Science and Engineering department at The University of Texas at Arlington for facilitating my education throughout my undergraduate and graduate careers. The expertise and warmth of the faculty and staff is sincerely appreciated.

Finally, I would like to thank my parents and my fiancé for their constant love and support. Their encouragement is the root source of all my successes.

April 13, 2010

ABSTRACT

REAL TIME HARDWARE AND SOFTWARE SYSTEMS FOR
MICRO AIR VEHICLE FLIGHT CONTROL TESTING

Christopher McMurrough, M.S.

The University of Texas at Arlington, 2010

Supervising Professors: Frank L. Lewis and Sajal K. Das

The trend of increasing capabilities of micro scale electrical, mechanical, and computing systems is opening up the possibilities of new unmanned vehicle platforms. Both unmanned aerial and ground vehicles have traditionally been created from large conventional platforms such as automobiles, rotorcraft, and fixed wing aircraft. These conventional chassis designs are durable and proven, but the benefits of automating these types of platforms are limited by their inherent human-centric design. Many real-world applications would benefit greatly from vehicle classes designed with unconventional means such as biologically inspired locomotion.

Micro air vehicles are a class of vehicles that have recently began moving from the realm of the probable to that of the practical. Current micro aerial vehicle (MAV) designs are generally nothing more than scaled down fixed wing airplanes or rotorcraft, but biologically inspired flapping wing platforms at the insect scale are now a distinct possibility. Flapping wing MAVs have the potential to perform tasks in environments that conventional aerial vehicles fail

due to constraints imposed by size and weight, robustness to environmental perturbations, and endurance.

Flapping wing MAV development requires new experimentation techniques for control design and validation. These types of vehicles require rapid actuator responses to a degree where human remote control is impossible without electronic assistance. This and other caveats make it difficult to develop MAV platforms and controllers without complete avionics and power systems, which are in turn difficult to develop when the vehicle specifications are unknown.

In this thesis, a *Real-Time Testing Environment for Development of MAV Flight Controls* is presented. The system makes it possible to test MAV flight controls without integrated avionics by using real-time computing hardware within a vision based motion capture environment. A *Split-Cycle Wingbeat Modulated MAV Platform for Hardware-in-the-loop Control Analysis* is also presented. This test platform was designed and fabricated to verify the motion capabilities of a proposed flapping MAV controller on a practical vehicle. The vehicle platform is actuated by brushless DC motors (BLDC) and controlled by real-time computing hardware. The instantaneous forces and torques generated by the vehicle are measured with a 6 DOF force/torque transducer for processing by the real-time computer. Finally, *A Reinforcement Learning Approach to BLDC Motor Commutation* is also proposed in the context of actuating a flapping wing with finite timing constraints. This approach involves the use of a Q-learning algorithm to learn the optimal amount of power to apply at a given motor commutation state to achieve a desired transition time.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background

The natural flight abilities of insects and birds have long been studies by humans. Observation of flying creatures has inspired the development of human flight in the form of airplanes, helicopters, and gliding aircraft. While engineers have developed artificial flying machines of multiple scales and capabilities, biologically inspired flapping wing flight is still relatively immature.

The development and successes of unmanned aerial vehicles has created a significant interest in flapping wing flight research. Micro aerial vehicles (MAVs) would particularly benefit from flapping wing flight for reasons of efficiency, robustness to environmental perturbation, and biomimicry. For these reasons, biologists and engineers alike have tried to gain a greater understanding of nature's design.

Many works have been produced related to flapping wing animal flight. The wing motion characteristics of birds and bats have been studied in [9, 11]. The kinematics and dynamics of bat flight are studied in [10]. The aerodynamics of insect flight are explored in [6, 8]. The effects of wing morphology in insects are explored in [3].

Several research efforts have attempted to expand upon observation by modeling insect flight. Works like [4, 5, 7] propose control models for flapping wing flight. Studies such as these provide a bridge between biology and engineering, and lead the way for efforts to reproduce insect flight mechanically. Mechanisms to reproduce the motion of insect wings are

proposed in [7, 8, 9, 10, 11, 12]. These generally involve the use of a 4 bar linkage mechanism to convert the rotary motion of small motors to flapping motion.

The first successful demonstration of net lift forces created by an insect sized mechanism was the Harvard Robofly [2]. This device featured a vibrating piezoelectric actuator linked to 2 wings. The wings were mounted to the chassis with a flexure joint, which allowed the wings to rotate passively during flapping. The lift capabilities of this design were demonstrated with a vertical flight up a guide wire.

Expanding on the Robofly design is a vehicle proposed in [1]. This vehicle actuates the 2 flexure joint wings independently with parallel piezoelectric actuators. In addition to providing a vehicle design concept, the work includes a set of control laws for hover capable 5 DOF motion. The split cycle controller presented in [1] works by flapping the left and right wings independently at a given rate. The speeds of forward and backward wing strokes are changed relative to each other to create the aerodynamic forces necessary for hovering. 5 DOF control authority is demonstrated in software simulation. A physical vehicle implementation was not available prior to the work of this thesis.

1.2 Purpose

The purpose of this thesis is to provide 3 key contributions to the field of MAVs. The 3 contributions are part of a larger vision of creating a real-world hovering MAV based on the design presented in [1]. The first contribution is the creation of a real-time motion capture based system for MAV flight control. Motion capture technology has been successfully used for flight control in [13, 14], but these test beds lack the real-time determinism presented in this work. The second contribution is a system for testing the performance of a split cycle controlled MAV in the form of a hardware-in-the-loop-simulation. The fabrication of a practical split cycle controlled MAV using brushless DC (BLDC) motors is detailed. The final contribution is a reinforcement learning based approach for optimal BLDC motor control. Reinforcement learning

has been used effectively in related applications [20, 21, 22], and this thesis attempts to extend the concept to the context of MAVs.

1.3 Thesis Organization

Chapter 2 of this work describes the implementation of the motion capture based MAV test environment. Chapter 3 describes the fabrication of the MAV test platform, along with the design of the hardware-in-the-loop simulation for split cycle control analysis. Chapter 4 presents the reinforcement learning based approach to BLDC motor control for use in flapping wing MAVs, and chapter 5 concludes the work with final thoughts and future work.

CHAPTER 2

A REAL-TIME TESTING ENVIRONMENT FOR DEVELOPMENT OF MAV FLIGHT CONTROLS

2.1 System Overview

The Real-Time Testing Environment is intended to assist research efforts in MAV control by eliminating the need for onboard sensors and computers in flight testing. The system is designed around an optical motion capture system by Vicon Motion Systems. This motion capture subsystem provides position and orientation information of multiple marked bodies with a resolution of approximately 0.1 mm through a network with a refresh rate of 120 Hz.

The vehicle controller is deployed on a dedicated real-time target from a development computer. When running, the development computer acts as a host, displaying diagnostic information from both the real-time target and Vicon IQ server. The real-time target performs all data processing; taking inputs from the Vicon server and producing outputs for the vehicle radio transmitter. The real-time target, programmed using LabVIEW, is capable of running vehicle controllers written in LabVIEW or Matlab/Simulink.

Actuator commands are sent to the vehicle through a standard radio control (RC) transmitter. The transmitter has a trainer port, or “buddy box”, which allows input commands to be received from another device using a Pulse Position Modulation (PPM) protocol. A microcontroller based transmitter interface device was created to translate control outputs from the real-time target to a continuously generated PPM signal.

The complete system, when provided with an appropriately designed vehicle and flight controller, is able to autonomously fly the vehicle through the test environment using feedback

from the motion capture system. Vehicles and controllers can be quickly replaced for rapid prototyping.

2.2 Hardware Organization

The system is distributed across several different hardware entities and subsystems. Fig. 2.1 shows the distribution of hardware and their applicable interfaces. The Vicon motion capture subsystem is composed of 36 cameras, 4 MX Ultranet units, and a Windows XP based server PC. The cameras each connect to an MX Ultranet unit through a proprietary network. The Windows XP based server PC, development PC, and real-time target are all connected through a TCP/IP connection provided by one of the MX Ultranet units. This particular MX Ultranet unit acts as a network hub, allowing all of the MX Ultranets and system computers to communicate with each other.

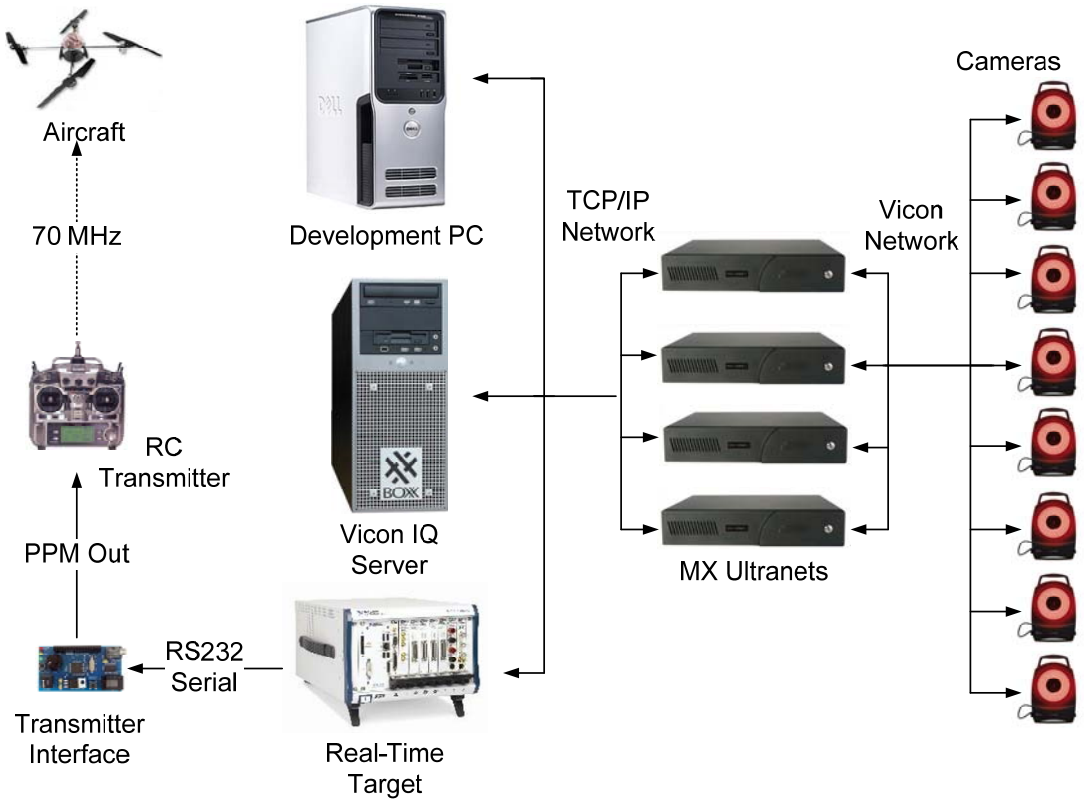

Figure 2.1 Testing Environment Hardware Organization

The real-time target is connected to a transmitter interface device through an RS232 serial port. This PIC microcontroller based device connects to the RC transmitter through the transmitter trainer port. The RC transmitter then sends actuator commands through a radio signal to the vehicle receiver. This simple RC receiver then drives the vehicle control surfaces and propulsion devices.

2.3 Software Organization

Several software entities are required to coordinate the activities of the test environment. The distribution of these entities across the various subsystems is shown in Fig. 2.2.

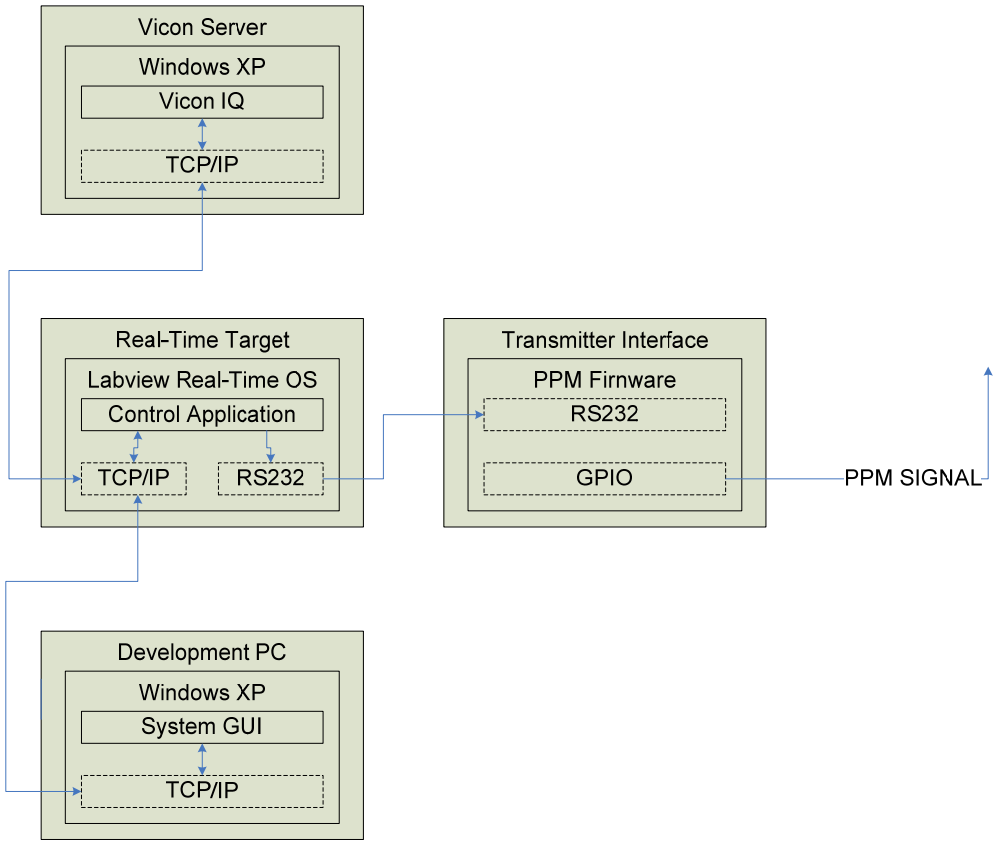

Figure 2.2 Testing Environment Software Distribution

The real-time control application requests position/orientation data from the Vicon Server using a TCP/IP network connection. Requests from the control application and responses from the Vicon IQ executable are transferred through TCP/IP using the Tarsus Communication Protocol specified by the Vicon documentation.

The control application processes the position/orientation data, computes the appropriate RC transmitter channel outputs, and packages this information using a simple checksum protocol. Data packaged using this protocol is sent through an RS232 COM port to the PPM generation device, which then generates a PPM signal with channel pulse widths corresponding to the packaged data received from the real-time control application.

The development PC is used to program the real-time PC and to provide a GUI control panel while the real-time control application is in run mode. In both development and run mode, communication is established using a TCP/IP network. The real-time control application is invoked from the development PC. Upon invocation, the LabVIEW program is compiled and transferred to the real-time PC. Once completely loaded by the real-time machine, program execution begins. While the GUI control panel is hosted on the development PC, all computation is handled by the real-time PC (including the graphical data displayed by the GUI).

2.3.1 Real-Time Control Application

The real-time target software was designed using the LabVIEW Development Suite. This platform allows a system user to easily test vehicle controllers written in LabVIEW or MATLAB/Simulink. The system software has been designed in a modular fashion; such that modifications can be created easily without a detailed understanding of low-level functions and processes.

One key design aspect of the system software is that wrappers are provided for the vehicle controller and PPM transmitter. This allows a new control model or transmitter profile to

be added to the system without having to reroute any of the connections in the tightly integrated high-level Virtual Instrument (VI).

The real-time system uses four independent loops for critical system processes. Camera input processing, control computation, PPM transmitter output, and GUI control are all handled in these individual loops. The loops are decoupled, such that different refresh rates and priorities can be assigned. This decoupling is achieved using LabVIEW local variable structures, which allow data to be shared between the independent loops. The loops are prioritized in a way that guarantees determinism for the critical control computation loop. This loop is given access to a dedicated processing core on the multicore real-time target.

Figure 2.3 and Table 2.1 illustrate the hierarchy of the control application VIs and their functions.

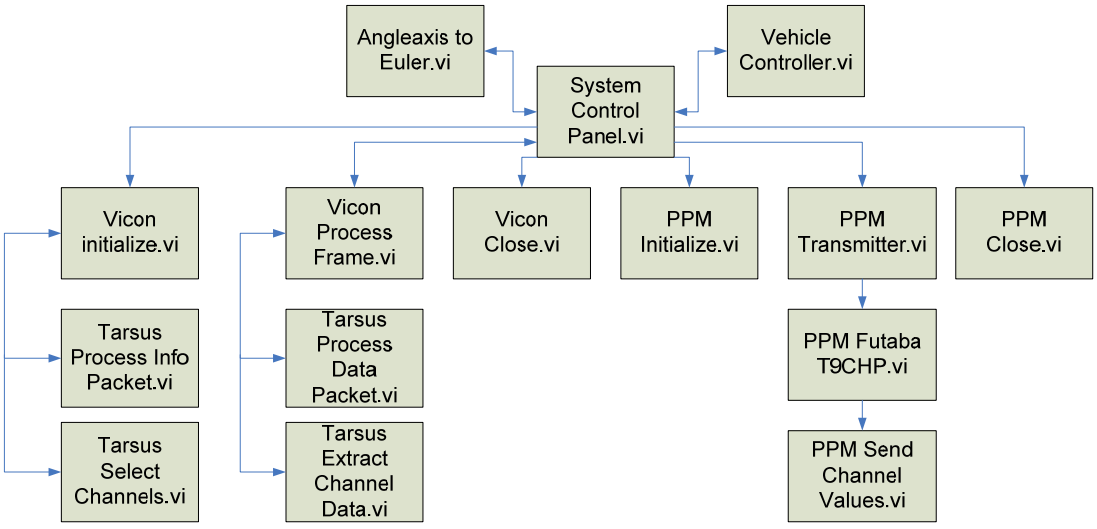

Figure 2.3 Control Application VI Hierarchy

Table 2.1 Control Application VI descriptions

System Control Panel	Provides a user interface for high level vehicle commands
Vicon Initialize	Establishes a connection to the Vicon IQ server
Vicon Process Frame	Parses position/orientation data from a received Vicon data frame

Table 2.1 – *Continued*

Vicon Close	Terminates a connection to the Vicon IQ server
Tarsus Process Info Packet	Requests and processes a Tarsus info type packet from the Vicon IQ server
Tarsus Process Data packet	Requests and processes a Tarsus data type packet from the Vicon IQ server
Tarsus Select Channels	Returns the packet indices of desired vehicle data from a Tarsus packet
Tarsus Extract Channel Data	Parses the desired channels of a data packet for a tracked object
PPM Initialize	Initializes a connection to the transmitter interface device
PPM Transmitter	Provides a generic interface wrapper for a particular RC transmitter
PPM Close	Closes the connection to the transmitter interface device
PPM Futaba T9CHP	Implements communication parameters specific to the T9CHP RC transmitter
PPM Send Channel Values	Packages and forwards the PPM waveform parameters to the interface device
Vehicle Controller	Provides a generic interface wrapper for a particular vehicle controller

2.3.2 Vehicle Transmitter Interface Firmware

The transmitter interface firmware was written in the C programming language and compiled with the CCS C Compiler. The device is programmed to listen for valid RS232 packets and create PPM waveforms as shown in Fig. 2.4 based on these commands. The signal generation timing is accomplished using a 16 bit counter on the PIC 18F4550. As valid packets of serial data are received, individual parameters are extracted and loaded into an array of individual pulse times. The counter is set to generate an interrupt at the time specified by the array, with the index of the array corresponding to the position of the individual pulse to be created. At each interrupt the signal pin is toggled, the next pulse time is loaded into the counter, and the signal array index is incremented. Once the synch pulse is reached, the index is reset and a new train of pulses begins.

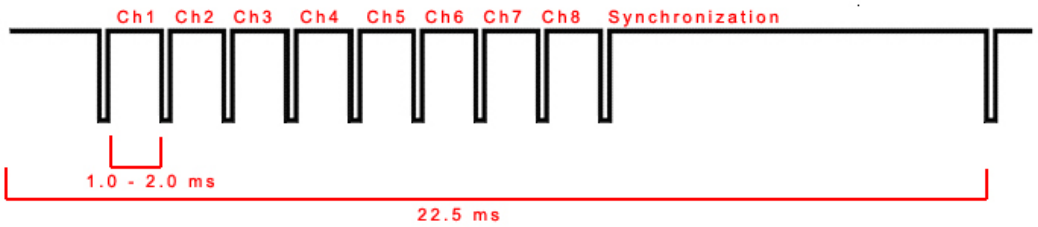

Figure 2.4 Visualization of a PPM Pulse Train

The transmitter interface device receives packets of 16-bit data from the real-time target through an RS232 serial port. Data arriving through this port is formatted using a simple protocol that prevents index confusion and erroneous data. The packet structure is shown in Fig.2.5.

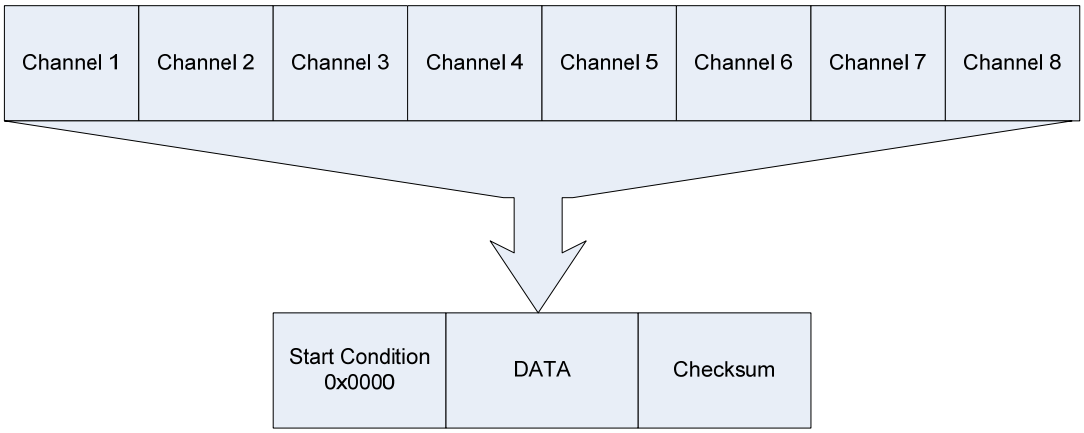

Figure 2.5 Transmitter Interface Communication Packet Structure

2.4 Test Results

The functionality of the system was verified in a series of flight tests with commercially available aerial vehicles. The first series of tests were conducted using a Draganflyer quadrotor with a classical control structure. Inner and outer control loops were implemented for regulation of angular rates, angular position, linear rates, and linear position. Position and orientation

holding capabilities were verified, followed by figure 8 and circular maneuvers. These tests were carried out successfully with and without physical interference by the system operators. Autonomous flight of the quadrotor vehicle is depicted in Fig. 2.6

Figure 2.6 Demonstration of Position and Orientation Hold with Quadrotor

The next test was to verify the system's ability to track multiple targets simultaneously. The system was set to track the position and orientation of both the quadrotor and calibration wand. The quadrotor was then commanded to follow the position of the calibration wand, which it did successfully.

The final test was to verify the modularity of the system with a new vehicle. For this test, the quadrotor was replaced with a collective pitch helicopter. The size and mass parameters of the vehicle controller were modified, and the previous tests were successfully

repeated. The helicopter was able to fly autonomously on the same day that it was brought into the test environment.

2.5 Future Enhancements

The functionality of the system was demonstrated with relatively stable conventional vehicles, but has yet to be validated with a flapping wing MAV. This is mostly due to the fact that such a platform was unavailable during testing validation. Additional features such as extended flight maneuvers and waypoint following can also be added. The system's ability to track multiple targets was demonstrated, but the ability to simultaneously control more than one vehicle was never attempted. These unimplemented items would extend the capability of the system beyond single vehicle control and allow rapid prototyping of swarm control techniques.

CHAPTER 3

DESIGN AND FABRICATION OF A SPLIT-CYCLE WINGBEAT MODULATED MAV PLATFORM FOR HARDWARE-IN-THE-LOOP CONTROL ANALYSIS

3.1 Simulation Overview

The purpose of the MAV platform and supporting hardware in this section is to create a practical implementation of a flapping wing MAV utilizing the split-cycle wingbeat modulation controller presented in [1]. The working MAV platform is intended to be mounted on a 6 degree of freedom force/torque measurement device for post processing of generated forces and moments. This hardware-in-the-loop simulation includes unsteady aerodynamic forces that are not represented in the model presented in [1]. These forces are essential to understand how the controller performs on a real world vehicle implementation with a practical wingspan of 5 inches. A vehicle of this size would have to perform approximately 20 wingbeats per second through an angle close to 120 degrees. These design parameters were chosen for the MAV platform presented in this chapter.

3.1.1. Split Cycle Control with BLDC Motors

The desired motion characters of the MAV platform require high speed motors with the ability to react quickly to speed changes. Because of the nature of the split cycle controller, forward and backward wing motion must be possible at different speeds. The 20 Hz average flapping rate of the platform requires instantaneous motor speed changes at a rate of 40 Hz (one speed for the forward stroke and one speed for the backward stroke).

The hypothetical vehicle presented in [1] uses a pair of piezoelectric actuators similar to those used in the Harvard Robofly [2]. These actuators perform best within a certain resonant

frequency, and have yet to be demonstrated with split cycle motion characteristics. Conventional DC motors are capable of generating the necessary speeds, but the built up rotor inertia limits their response to abrupt speed changes.

Stepper motors were initially considered for split cycle motion due to their driving properties. The position of the shaft in a stepper motor at a given instant can be controlled accurately by issuing step commands at set intervals. This property allows the motor to overcome the response problem, but the maximum attainable RPM is not sufficient for 20 Hz motion. Stepper motors also tend to be much heavier than comparable DC motors, which make them unrealistic for MAV use.

BLDC motors were eventually chosen for their high RPM and stepped control nature. Similar to stepper motors, BLDC motors can be actuated in a discrete nature. Their use of high power magnets and high current motor windings allow them to achieve speeds and torques much higher than a stepper motor. They are also much more efficient in terms of power usage than a stepper motor or conventional DC motor. The main caveat of BLDC motors is cost. While currently much more expensive than conventional DC motors, BLDC motor prices have shown a downward trend due to increased production for the hobby market.

Split cycle wingbeat motion can be obtained with BLDC motors by commuting the motor poles at specific intervals. The BLDC motors are used to drive crank-rocker 4 bar linkages for conversion of rotary motion to angular displacement. Each discrete motor commutation results in a constant angular shaft displacement. This angular shaft displacement creates a given angular wing tip displacement through the linkage, resulting in precise flapping motion. Commuting the motor at a set of specific rates for a single wing stroke can be used to create a constant wing velocity in a given direction. Switching the set of rates between forward and backward wing strokes then produces split cycle motion.

3.1.2. Experiment Architecture

The architecture of the hardware-in-the-loop simulation experiment is shown in Fig. 3.1. A real-time control application runs the vehicle controller and sends the actuator commands to the BLDC motor controllers. The motor controllers produce wing motion through a 4 bar linkage. The wing beats produce lift forces and moments on the chassis, which is mounted to the force/torque sensing device. These force/torque measurements can then be used as feedback for the split cycle controller, or saved for post processing analysis.

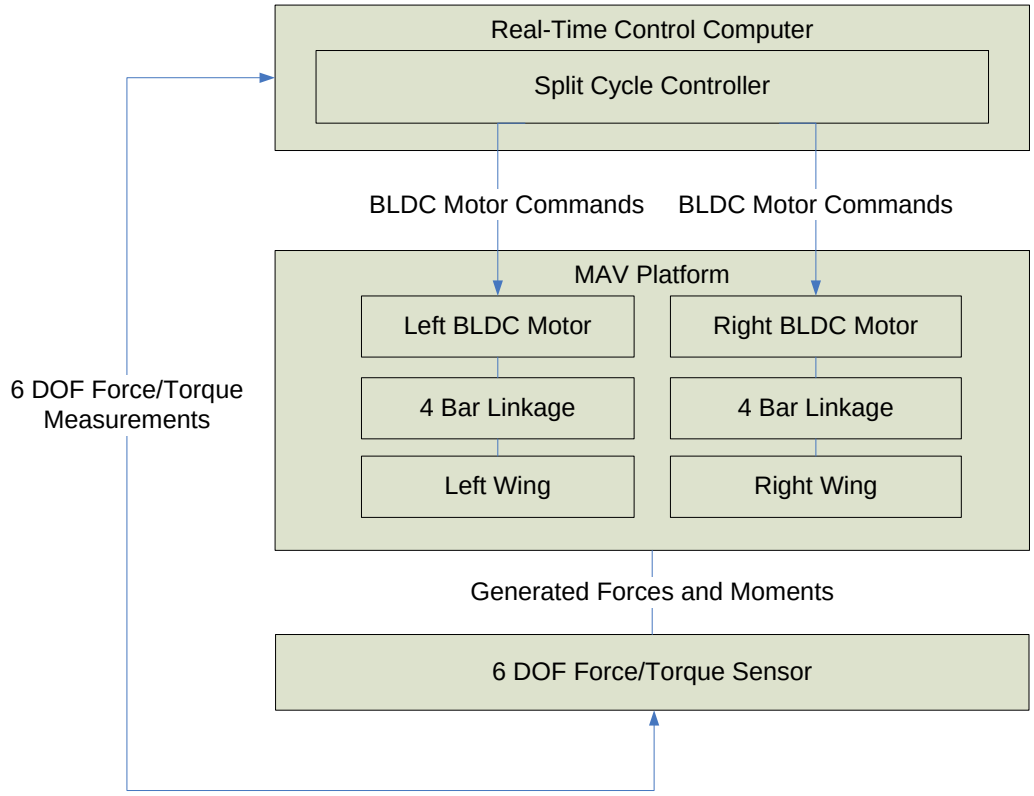

Figure 3.1 Hardware-in-the-loop Simulation Architecture

The architecture shown Fig. 3.1 is common to both fabricated vehicle prototypes. Details on their exact hardware implementations are given in the proceeding subsequent sections.

3.2 First Generation MAV Prototype

An initial prototype was created at a larger scale to test the split cycle motion capabilities of BLDC motors. This prototype was used as the basis for the realistically sized vehicle presented in section 3.3.

3.2.1. Transmission Design

To convert the rotation of the BLDC motors to a flapping motion, a four bar linkage was chosen. The linkage was designed such that a complete rotation of link B would result in a 112 degree rotation of link D. It was determined that a 112 degree rotation was sufficiently close to the 120 degrees proposed in section 3.1. Increasing the rotation angle increases the footprint of the linkage, which introduces size and symmetry problems to the platform. Link A is the fixed distance between the moving linkages, and link C is the transmission linkage between B and D. The linkage proportions used to generate 112 degree flapping motion were A = 7, B = 2, C = 7.5, D = 3. This linkage design is depicted in Fig. 3.2.

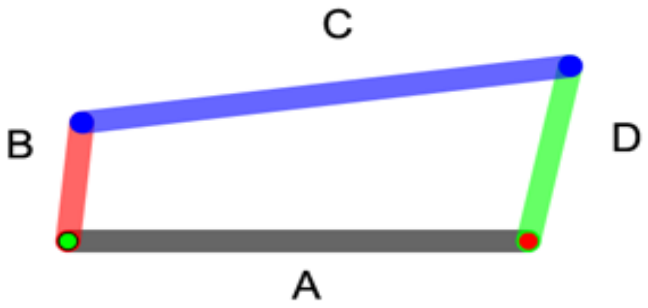

Figure 3.2 Linkage design used in the MAV prototype

A simple 2 gear train is used to reduce the output speed and increase the torque of the rotating motor shaft. Using the four bar linkage also requires that the drive link (link B) be sufficiently large, so a gear with radius greater than or equal to the length of B is required to

serve as link B (in other words, a radii of the gear becomes link B when one end of link C is attached to the edge of the gear). The gear train design is illustrated in Fig. 3.3.

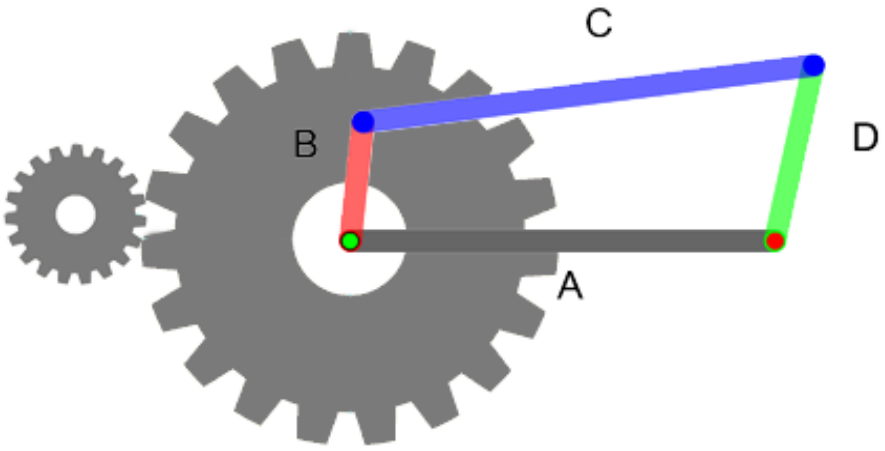

Figure 3.3 Illustration of complete flapping transmission

The gear driving link B is rotated by a smaller pinion gear which is mounted directly to the motor shaft. Using a 5:1 gear reduction ratio, 5 complete rotations of the pinion gear result in one complete 112 degree upward and downward wing beat.

3.2.2. Chassis Design

Once the flapping transmission was designed, a chassis was created for mounting of all moving parts. A single chassis plate was designed for simplicity and strength using the Solidworks design suite. Fabrication of the chassis was achieved using a rapid prototyping 3D printer. The chassis plate was designed with mounting points for the BLDC motors, drive gears, and wing rotation points. The position of each mounting point was according to the geometry of the transmission mechanism. The use of the 3D printer facilitated rapid prototyping with a high degree of precision. A 3D rendering of the chassis plate is shown in Fig. 3.4.

Figure 3.4. 3D Rendering of Chassis Plate

3.2.3. BLDC Motors and Controllers

For the first MAV prototype, commercially available BLDC motors were selected as a proof of concept. The selected motors were model Three-80 from Novak Electronics. These motors feature integrated Hall Effect sensors for detection of the current commutation state. This increases motor performance by preventing premature commutation attempts.

Motor drivers for these types of BLDC motors are readily available, but focus on overall RPM control rather than instantaneous shaft position control. In order for the BLDC motors to be run in split cycle, custom driving circuits had to be created. The motors drivers, based on the BLDC control circuit in [25], expose the gate inputs of the 6 MOSFET transistors required to control the flow of current through the 3 motor phases. This feature allows the real-time computer to exert full control authority of the individual motor commutations. The completed motor drivers are pictured in Fig. 3.5.

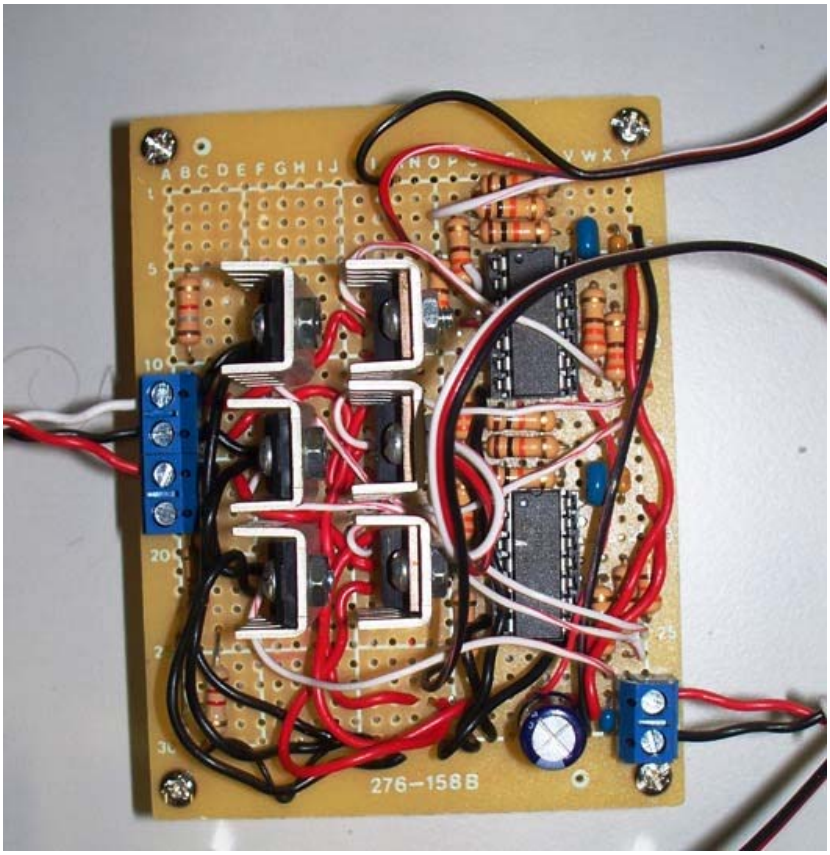

Figure 3.5. BLDC Motor Driver Circuit

3.2.4. Wing Design

The wings used by the MAV must be able to passively rotate in order to generate lift forces. The MAV designs presented in [1, 2] propose the use of flexure joints. This solution was successfully demonstrated by the Harvard Robofly, but the fabrication techniques required are difficult to master. A rotation stop limit solution is used in the MAV prototype, which uses a carbon fiber wing structure mounted through a linkage tube. The air pressure generated by the moving wing forces the wing to rotate away from the direction of motion. Shaft collars used to hold the wing in the tube are stopped after +/- 45 degrees of rotation by a stop cap mounted on the linkage tube.

The actual wing structure was created by mounting small carbon fiber struts into precisely drilled 45 degree holes on the main wing rod. The struts are held in place with a composite epoxy solution, and the resulting skeletal structure is wrapped with a lightweight adhesive plastic. The working wings are displayed with their shaft collars in Fig. 3.6.

Figure 3.6. First Generation MAV Wings

3.2.5. Platform Architecture

After the Initial MAV Prototype was successfully assembled, the various hardware components were integrated into a working subsystem. The National Instruments PXI-8106 Embedded Real-Time Controller changes the motor commutation states through digital signals generated by a DAQ breakout board. This real-time computer runs a multi-threaded control program coded in LabVIEW. The integrated Hall Effect sensors of the BLDC motors are read through the DAQ breakout for use in the control program. The complete system architecture is shown in Fig. 3.7.

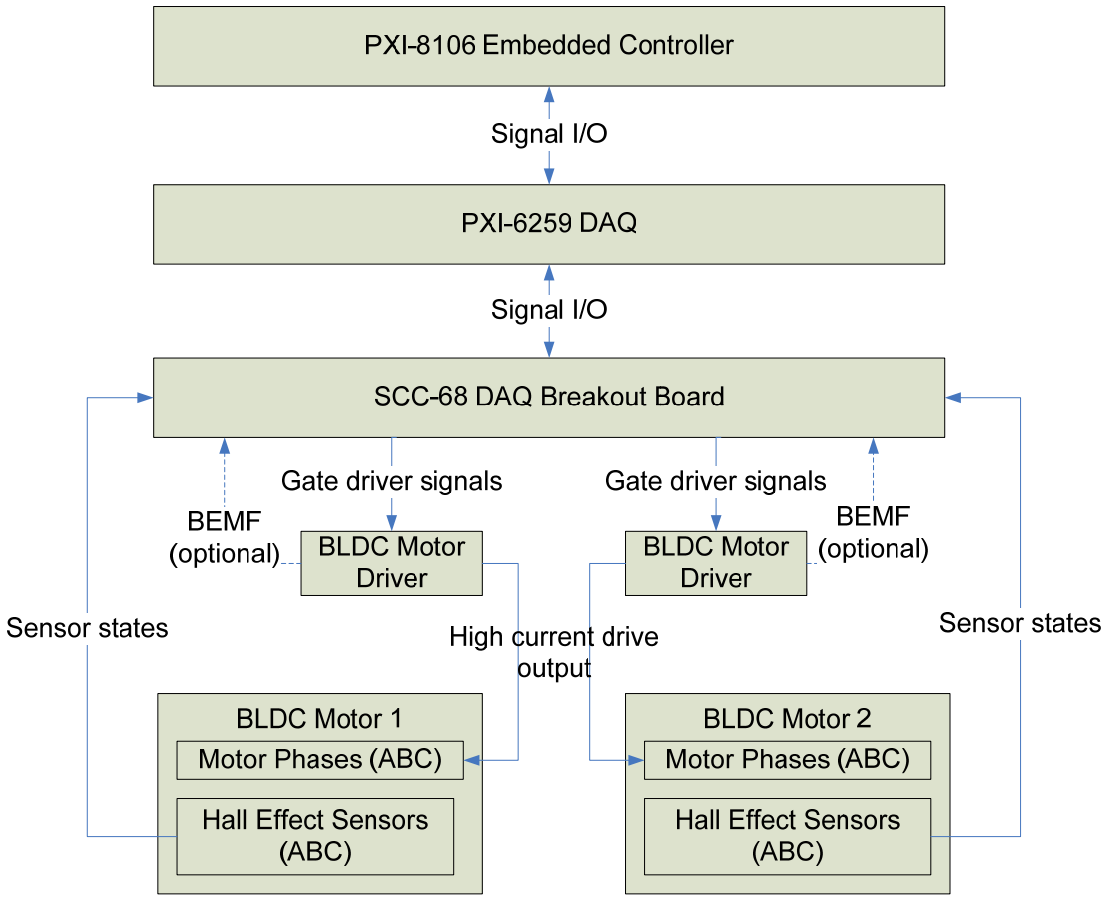

Figure 3.7. First Generation MAV Platform Hardware Architecture

3.2.6. Test Results

The initial MAV prototype successfully demonstrated the mechanical capability of the linkage design, BLDC motors, and stop limit method for passive wing rotation. The concept of precise commutation driving was proven to be feasible with flapping speeds up to 20 Hz for short durations, indicating that high speed split cycle motion is attainable with BLDC motors.

The main limitation of the mechanism was the mounting of the wing support bearings with a press fit method. The vibration caused by high flapping frequencies would force the bearings from their mounting holes causing the test run to halt. Sustained split cycle motion at a

safe speed of 3 Hz was eventually demonstrated on the platform, proving the feasibility of BLDC motors in future split cycle MAV designs. The completed test mechanism is displayed in Fig. 3.8. The test mechanism is shown with support electronics in Fig. 3.9.

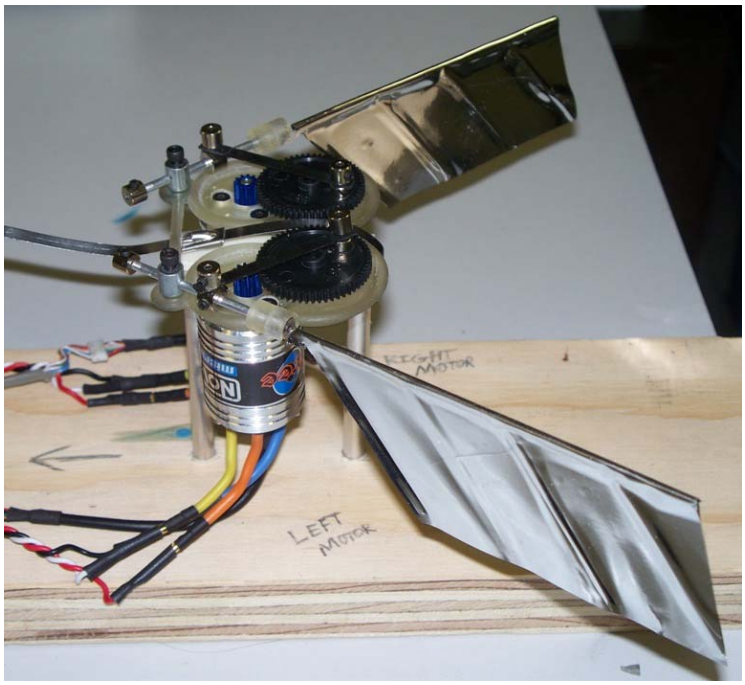

Figure 3.8. Completed First Generation MAV Mechanism

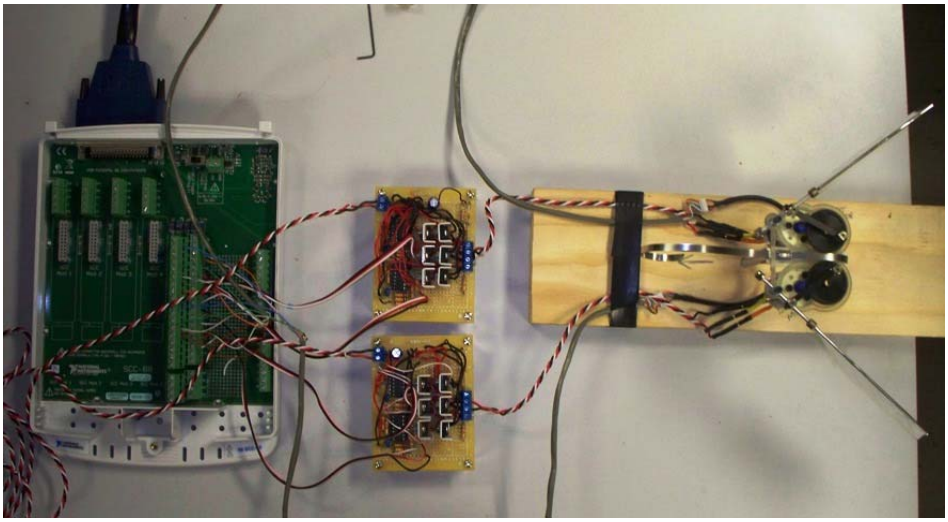

Figure 3.9. First Generation MAV Platform with Support Electronics

3.3 Second Generation MAV Prototype

The first generation MAV prototype demonstrated core capabilities necessary to achieve split cycle wingbeat control with BLDC motors. The second generation platform built on the proven strengths of its predecessor while addressing some of its shortfalls. The size and weight were drastically reduced; meeting the original goal of a 5 inch wingspan. Flapping performance was also greatly increased with the use of high performance micro BLDC motors. The section describes the key enhancements and results of the second generation platform.

3.3.1. Transmission Design

The 4 bar linkage used by the first generation MAV was scaled down by a factor of 2 to reach the vehicle goal size. This decreased the size of the drive gear necessary to function as the linkage crank. In addition, the pinion gear was eliminated in favor of an integrated motor planetary gearbox. This allowed the drive gear to be directly mounted to the motor shaft, further reducing the vehicle footprint. The planetary gearbox increased the gear ratio from 1:5 to 1:15, which resulted in higher output torque.

All transmission components were custom fabricated by the author. This was done to address the lack of commercially available precision parts made from proper materials. Fig. 3.10 illustrated the fabrication of drive gears on a precision micro lathe from aluminum stock.

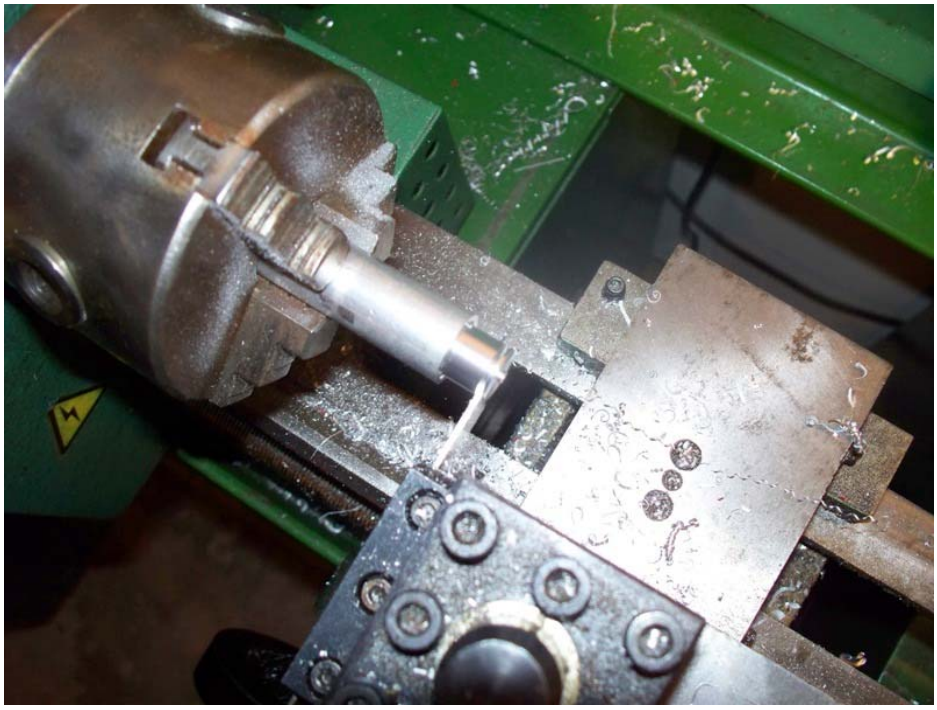

Figure 3.10. Precision Drive Gear Fabrication

3.3.2. Chassis Design

The chassis was completely redesigned for the second generation vehicle. The single plate approach was eliminated in favor of a lightweight carbon fiber frame structure. Four main chassis components were created to hold the motors and wing post bearings. A second bearing was added to each wing post to eliminate the problems caused from high frequency vibration. Carbon fiber rods were precisely cut for the frame structure, and mounting holes were included in the CAD model for each chassis part to guarantee vehicle symmetry. Fig. 3.11 shows the 3D rendering of the second generation chassis.

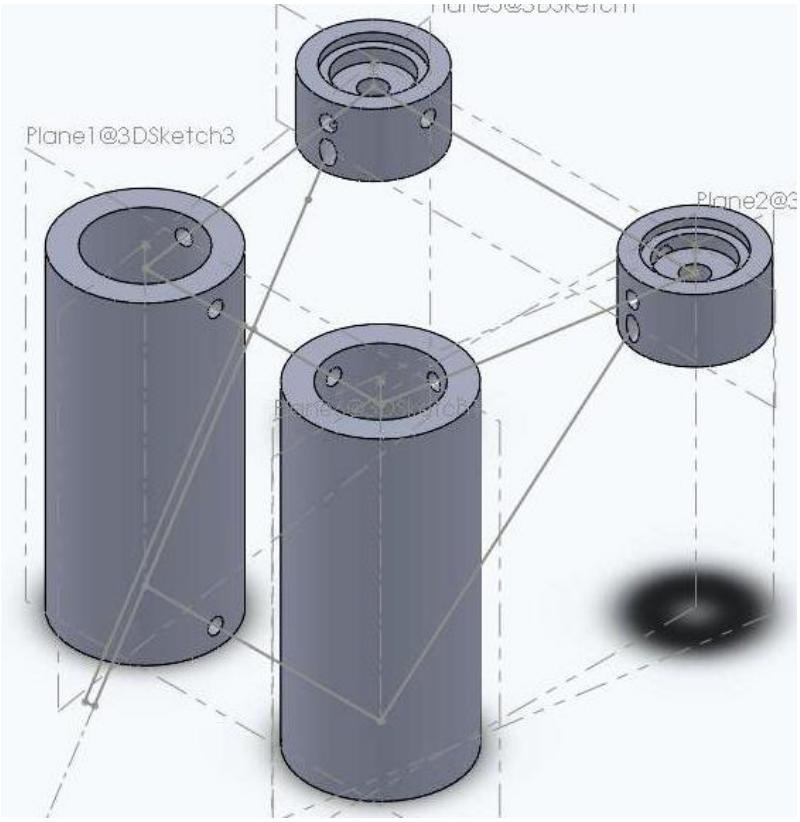

Figure 3.11. Second Generation MAV Chassis

3.3.3. BLDC Motors and Controllers

The single most important improvement added to the second generation MAV was the use of high performance micro BLDC motors. The Maxon EC-6 motors offer drastic improvements in size, weight, and speed over the Three-80 motors from the first generation platform. The inclusion of an integrated 15:1 planetary gearbox and 64 counts per turn encoder further enhance the utility of the EC-6. The total motor weight of the second generation MAV was reduced from over 100 grams to 5.6 grams. The maximum shaft speed of 40000 RPM is capable of producing unloaded drive gear rotation frequencies of 44.4 Hz after gear reduction. Fig. 3.12 illustrates the completed MAV chassis equipped with both EC-6 motors.

Figure 3.12. Second Generation MAV Chassis Equipped with EC-6 Motors

The second generation MAV replaces the custom BLDC driver circuits with modules created by the manufacturer specifically for the EC-6 motor. The EPOS 24/1 position controller allows a wide range of control modes that are useful for evaluation of the MAV platform. The position control mode and stepping control mode are both capable of producing split cycle motion similar to the method used in the first generation controllers.

3.3.4. Wing Design

The wing carrying tube method from the first generation MAV was reused in the second generation platform. The design was optimized by using removing excess length from the tube, resulting in increased space for the moving wing surface. Hollow carbon fiber tubing was used in place of the aluminum tubing to reduce weight and prevent any permanent bending from sustained high frequency flapping. The stop cap was also replaced with set of thin carbon fiber limiting bars that travel in plane with the wing. These bars proved to be more durable than the stop cap design due to the reduced shock experienced as the wing changes direction.

The wing fabrication method was improved by replacing the plastic membrane with a transparent Mylar film. The scaled down carbon fiber wing structure is coated with a cyanoacrylate adhesive and wrapped with the Mylar film. This process produces veins of adhesive that cure to produce structures resembling those found in an insect wing. These structures greatly increase the strength of the membrane, while still allowing a degree of flexibility. A pair of wings created with this process is displayed in Fig. 3.13.

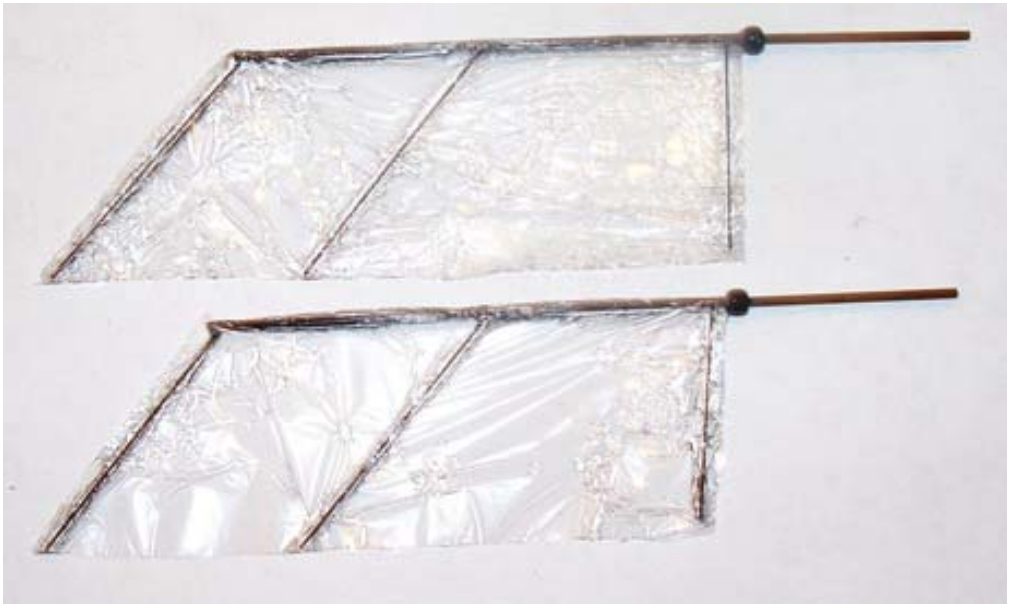

Figure 3.13. Second Generation Wings with Mylar Membrane

3.3.5. Platform Architecture

The hardware architecture for the second generation MAV closely resembles that of the first generation platform with a few exceptions. First, the need for a DAQ breakout as an interface between the real-time computer and BLDC controllers was eliminated due to the RS232 interface provided by the Maxon EPOS controllers. Second, the Hall Effect sensor feedback is replaced with high resolution data from the integrated MILE encoder. Third, the

force/torque measuring transducer and ethernet interface are included. Fig. 3.14 shows the hardware architecture of the second generation MAV platform.

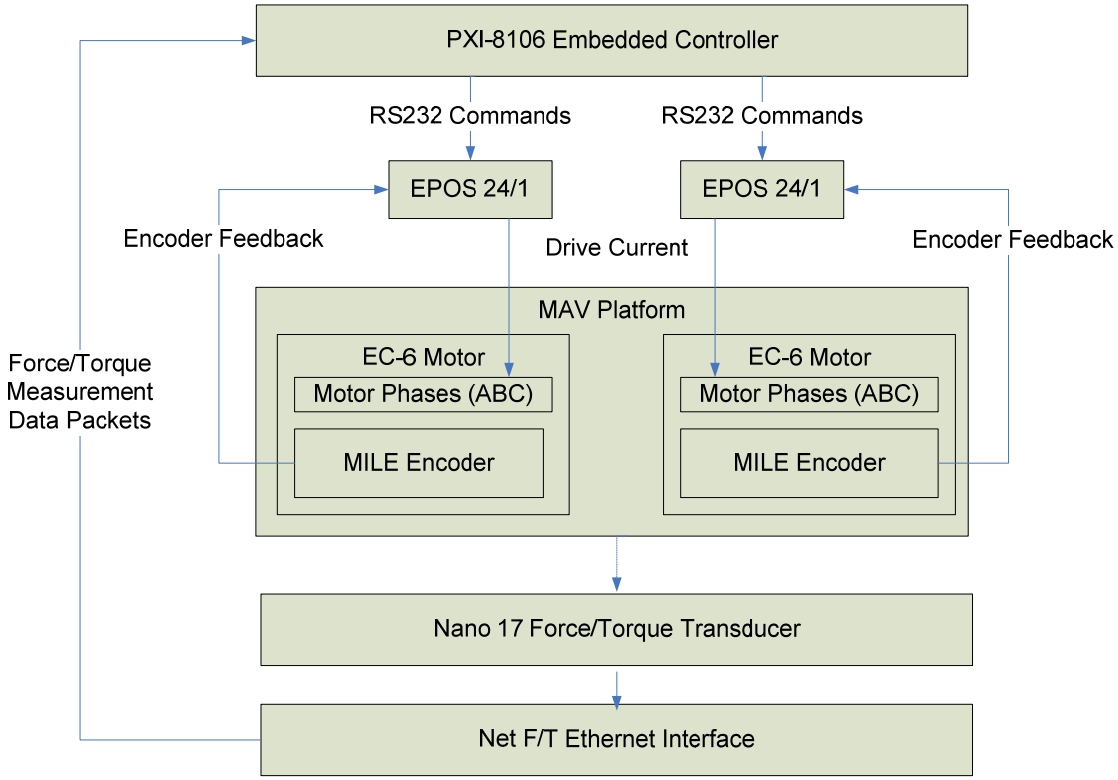

Figure 3.14. Second Generation MAV Platform Hardware Architecture

3.3.6. Test Results

Once construction of the MAV was completed, a test stand was created to hold the vehicle and support electronics. All electrical connections between hardware modules were then established. Fig. 3.19 showcases the completed MAV platform mounted on the test stand.

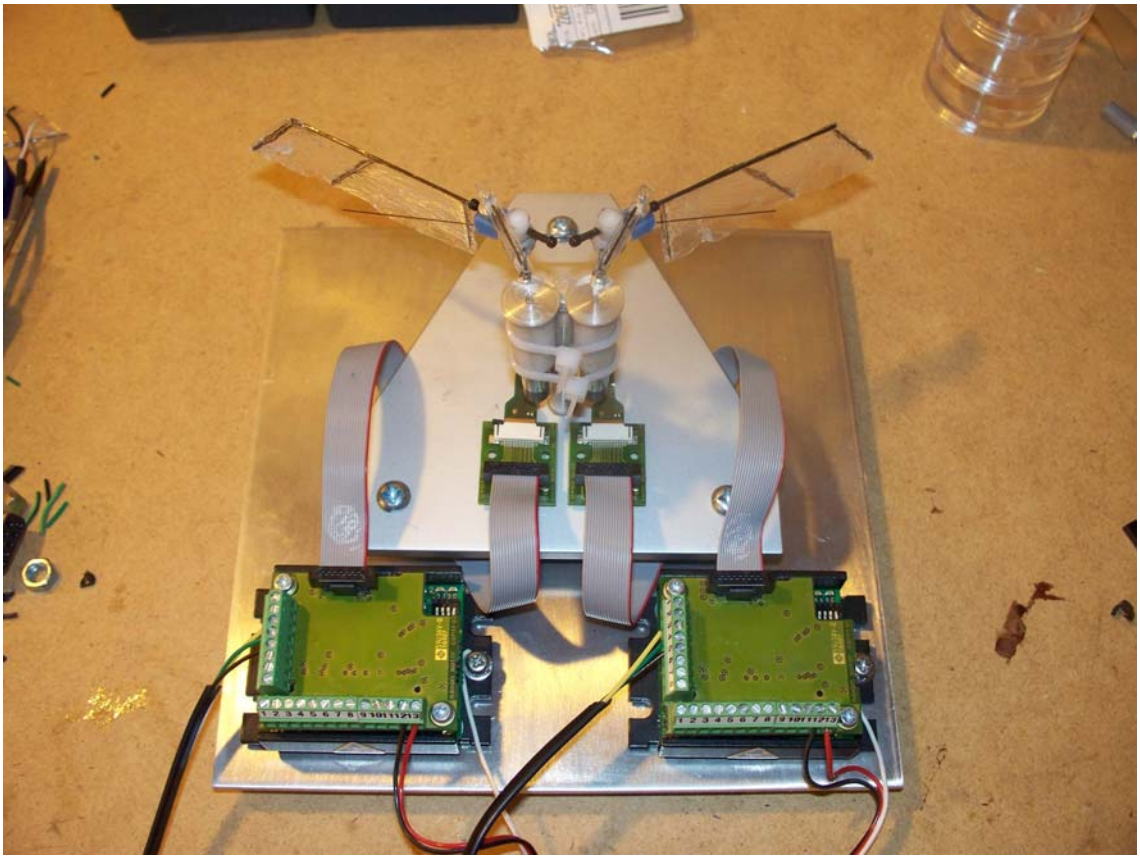

Figure 3.19. Completed Second Generation MAV on Test Stand

The first test of the second generation MAV platform was a wingless linkage test. The mechanism was placed on a test stand and given constant speed commands of 20,000 RPM. The resulting speed was measured with the MILE encoder and verified to produce a 22.2 Hz flapping rate.

The next test was to verify passive wing rotation and durability at low flapping speeds. The wings were mounted and flapped at a constant rate of 4 Hz. The passive wing rotation and limiting bar functionality were verified visually.

The next test was to verify full speed flapping at 20+ Hz with the wings mounted. The motors were slowly brought up to a speed of 25,000 RPM, producing a flapping rate of 27.7 Hz.

This speed was sustained for roughly 5 seconds before the experiment was stopped for a damage check. No damage was observed, so the experiment was repeated successfully for several 30 second durations.

The final test performed was a shaft position command test. This test was critical to implementing successful split cycle operation. The motors were given instantaneous shaft position commands, and then checked for accuracy with the encoder data. Several position commands were given to the motors resulting in random angular displacements. Each time a position command was given, the motor shafts turned to the position at maximum speed with no overshoot. No value was found that would result in an erroneous activity, regardless of the displacement or direction. A summary of experimental results is shown in Table 3.1.

Table 3.1 Second Generation Flapping MAV Platform Test Results

Experiment Description	**Motor Speed (RPM)**	**Flapping Rate (Hz)**	**Duration (s)**	**Pass/Fail**
Low speed wingless linkage test	3600	4	30+	Pass
High speed wingless linkage test	25,000	27.7	30+	Pass
Low speed passive wing rotation test	3600	4	30+	Pass
High speed passive wing rotation test	25,000	27.7	5	Pass
Gradual wing speed increase test	0-25,000	0-27.7	30	Pass
Sustained high speed flapping test	25,000	27.7	30	Pass
Direct position control test	NA	NA	NA	Pass

3.4 Future Enhancements

The second generation MAV platform has demonstrated the capabilities necessary to implement a split cycle controller for 5 DOF motion. The only remaining action item necessary to fulfill this goal is to mount the vehicle on the Nano17 device for analysis of generated forces and torques. Due to equipment availability, this experiment has yet to be performed.

Once a working controller is verified to generate the predicted 5 DOF motion capabilities, an onboard electronic control system with an integrated radio receiver will be built for untethered flight testing. This control unit will be based on the BLDC controller from the first

generation platform, and will have to meet strict size and weight requirements. The flight testing for the MAV will take place in the testing environment presented in chapter 2.

CHAPTER 4

A REINFORCEMENT LEARNING APPROACH TO BLDC MOTOR COMMUTATION

4.1 Motivation

The rotary motion generated by a BLDC motor is created by applying a driving current in a particular direction through 2 of the motor poles at a given instant. In order for this synchronous electronic motor commutation to work, each pole must be able to be driven high, pulled to ground, or left unconnected in a “floating” state. The amount of current allowed to pass between 2 motor poles in a given commutation step has a direct effect on the speed at which the rotor spins, and thus the amount of time elapsed before the poles can be switched to the next commutation state.

A simple method for maintaining a desired rotor speed in a BLDC motor involves switching the poles at specific intervals while providing enough current to guarantee that the rotor arrives at the next commutation state before the transition time elapses. This method, implemented in the BLDC control circuits from the version 1 flapping mechanism, maintains RPM at the expense of power efficiency and instantaneous speed regularity.

An ideal approach to BLDC speed regulation involves modulating the provided current at each commutation step such that the error in arrival time is minimized. This would reduce the changes in instantaneous rotor speed while minimizing energy consumption and maintaining overall RPM. The difficulty in implementing such a control scheme exists because the position of the rotor is only measurable to a resolution equal to the number of commutation states per shaft rotation. The use of an external shaft encoder could provide the necessary rotor position readings to implement an optimal control scheme, but the added size and weight could make this unrealistic. A method for determining the optimal amount of power to apply at a commutation state without the addition of extra sensing hardware is desirable.

4.2 Learning Optimal Power with Q-Learning

In this section, a reinforcement learning approach for determining the optimal amount of power provided by a given the current commutation state and rotor travel time is proposed. The Q-learning function given in [23] can be applied, in a modified form, to select the optimal power PWM values from a finite discrete range. First, some system parameters are defined.

- Let p be the resolution of the PWM signal, where $1/p$ is a whole number and $0 \leq p \leq 1$
- Let g be the number of complete drive shaft rotations per wingbeat (the gear ratio)
- Let c be the number of motor commutations per shaft turn
- Let $T_{\max}$ and $T_{\min}$ be the maximum and minimum considered transition times, respectively
- Let t be the transition time resolution, where $\frac{T_{\max} - T_{\min}}{t}$ must be a whole number
- Let $t_{desired}$ and $t'_{desired}$ be the current and previous desired transition times, respectively
- Let s and s' be the previous and current commutation states, respectively
- Let r and r' be the previous and current rewards, respectively
- Let a be a PWM action, where a is a multiple of p

The Q-learning function in [23] is then modified to use a 3 dimensional Q table, with the added dimension $t_{desired}$. The necessary static global variables are then defined below.

- Let Q be a matrix of action values indexed by a, s, and $t_{desired}$
- Let N(s, a, $t_{desired}$) be a frequency table for commutation state, action, desired commutation time triplets.

The modified version of the Q-learning function in [23] is then given below in Fig. 4.1. Prior to the first run, s , a , and r are set to null.

function BLDC-Q-LEARNING(s ',r', $t_{desired}$) **returns** a

if s is not null **then do**

increment N(s , a , $t_{desired}$)

$Q(a,s,t_{desired}) \leftarrow Q(a,s,t_{desired}) +$

$\alpha(N(s,a,t_{desired})(r + \gamma \max_{a'} Q(a',s',t'_{desired}) - Q(a,s,t_{desired}))$

if TERMINAL?[s'] **then** s,a,r, $t_{desired} \leftarrow null$

else s,a,r, $t_{desired} \leftarrow s', \arg\max_{a'} f(Q(a',s',t'_{desired}), N(s',a',t'_{desired}), r', t'_{desired}$

return a

Figure 4.1 Q-learning Function for BLDC Motor Commutation

The values in Q are updated as the function explores actions. After a sufficient number of training iterations have occurred, the optimal power PWM value for the desired commutation time from the current commutation state is located by selecting the value of a that maximizes Q(a , s , $t_{desired}$).

The practical implementation of this method would initially take place with the vehicle mounted on a test stand, since the initial Q values would result in unpredictable motor behavior. After a sufficient number of iterations have occurred such that the function selects power PWM values that facilitate flight, though they may not be optimal, the vehicle can be removed from the test stand for online learning. As more iterations are performed, the function will start selecting the optimal discrete PWM value. This "lifelong learning" property ensures that the optimal PWM values are adjusted as the vehicle experiences mechanical wear or undergoes certain gradual environmental changes (such as barometric pressure or humidity).

4.3 Limitations

Caveats of this approach exist due to space complexity and training time. The space complexity of the Q matrix is:

$$O(cg \cdot \frac{T_{\max} - T_{\min}}{t} \cdot \frac{1}{p})$$

The time complexity of algorithm iteration is negligible, but the number of iterations required to learn useful values increases with the resolution of transition time, power PWM outputs, and gear transmission ratios. The time between function iterations is dominated by one of its own input parameters: the desired commutation time. The iterations per second (and thus the real world learning rate) would be limited by the flapping rate of the mechanical platform.

This method does not take into account the inertia of the spinning rotor or changes in torque loads. The assumption is made that the commutation commands change gradually over several wing beats such that rotor inertia does not significantly affect the arrival time of the rotor. The changes in rotor torque loads are also assumed to change gradually over several wing beats. This limits the effectiveness of the reinforcement learning approach in applications where the real world optimal PWM values change rapidly.

4.4 Future Enhancements

The reinforcement learning approach could be improved by reducing the size of the Q table. The formulation as presented in this thesis makes no attempt to remove state/action pairs that produce no utility. A large amount of table entries could be eliminated in this application due to the fact that low PWM output values will cause the motor to stall. This is a condition which has no chance of producing utility. Reducing the size of the Q table would decrease the time required to obtain useful values.

This approach has yet to be implemented on a hardware platform. A test mechanism would give valuable performance metrics pertaining to the training time and optimality of chosen actions. Special care must be taken on any hardware implementation to limit the amount of time that current is supplied during a commutation state. If the motor is stalled for any reason, continuous supply of current will have damaging effects on the internal windings. BLDC motors capable of actuating the type of MAVs proposed in chapter 3 tend to be cost prohibitive, so extra care would have to be taken to guard against damage.

One interesting possibility for improvement would be to attempt to learn a continuous function for optimal state/action pairs using a best fit technique. This approach could reduce the number of discrete actions in the Q tables, and hence improve learning speed. A continuous function would be able to provide precise PWM values as opposed to limited discrete choices, which would eliminate error due to imprecise resolution.

CHAPTER 5

CONCLUSION

5.1 Final Thoughts

This thesis presented 3 main contributions to the research field of micro aerial vehicles. The purpose was not to present the contributions as individual projects or topics, but rather to show the progressive development of a new class of flying machines. Chapter 2 discussed the first major development step of creating a test environment to facilitate future work in MAV platform design. Chapter 3 discussed the fabrication of a practical vehicle designed with the purpose of implementing a previously developed set of control laws. Chapter 4 expanded on one of the key successes from the previous chapter by proposing an optimization approach for BLDC motor control.

MAVs are inherently small entities that present big challenges. These challenges are multidisciplinary in nature, requiring a wide range of engineering contributions. While many complex multidisciplinary systems exist in our modern world, MAVs are particularly interesting because of the tightly coupled nature of the vehicle subsystems. Traditional engineering development models simply will not facilitate the kind of innovation required to produce these and other types of future machines. While some may see this as problematic, the author sees this as an exciting opportunity for specialists of all types to redefine the meaning of collaboration in the context of engineering.

5.2 Future Work

Future enhancements for the contributions discussed in this thesis were presented in each chapter. The “bigger picture” of future work lies in the combination of these solutions. The steps required to create MAVs that are truly useful beyond simple novelty are described below.

First, a vehicle controller must be verified on a stationary platform as described in chapter 3. This will prove the flight capability of a given aircraft platform. The testing method presented in the chapter is not limited to the proposed split cycle MAV, but could be used virtually and platform of similar scale.

Second, a vehicle with a verified controller must be tested in real flight. The system proposed in chapter 2 will facilitate this type of testing. It is only after passing this phase that a platform can truly be deemed “flight worthy”.

Third, an avionics package must be designed for the platform. This presents particularly tough challenges in electronics design due to the size constraints for any onboard sensors and instruments. Many sensor types used in modern small aircraft will either have to be scaled down further or eliminated from consideration. New sensors and estimation techniques that are robust to the dynamic nature of MAV motion will have to be designed.

Fourth, power systems must be developed to provide the necessary endurance for the vehicle to perform useful tasks. The old approach of “build a better battery” will not suffice. Energy harvesting systems that are small enough to be integrated into the vehicle will have to be created.

Finally, collaborative behaviors will have to be advanced further. The vast majority of useful MAV applications will require cooperation between many vehicles. It is likely that these vehicles will have to possess capabilities that are unique from other team members, which will require advancements in heterogeneous systems.

REFERENCES

[1] Doman, D. B., Oppenheimer, M. W., Sigthorsson, D. O., “Wingbeat Shape Modulation for Flapping Wing Micro Air Vehicle Control”, American Institute of Aeronautics and Astronautics, 2009.

[2] Wood, R., “The First Takeoff of a Biologically Inspired At-Scale Robotic Insect," IEEE Transactions on Robotics, Vol. 24, No. 2, 2007, pp. 341-347.

[3] Wootton, R. J., “The functional morphology of the wings of Odonata,” Adv. Odonatol. 5, 1991, pp. 153-169.

[4] Sane, S. P. Dickinson, M. H., “The aerodynamic effects of wing rotation and a revised quasi-steady model of flapping flight,” J. Exp. Biol., vol. 205, 2002, pp. 1087–1096

[5] Taylor, G.K., Bomphrey, R.J., Hoen, J., “Insect Flight Dynamics and Control," AIAA Paper 2006-32, Jan. 2006.

[6] Wakeling, J. M. Ellington, C. P., “Dragonfly flight I. Gliding flight and steady-state aerodynamic forces,” J. Exp. Biol. 200, 1997, pp. 543 -556.

[7] Dickson, W. B., Straw, A. D., Poelma, C. Dickinson, M. H., “An integrative model of insect flight control. In 44th AIAA Aerospace Sciences Meeting and Exhibit. Reno, NV; USA, 2006.

[8] Ellington, C. P., “The Aerodynamics of Hovering Insect Flight, Parts I-VI," Philosophical Transactions of the Royal Society of London: Series B: Biological Series, Vol. B305, No. 4, 1984.

[9] Tobalske, B. W., “Biomechanics of Bird Flight,” The Journal of Experimental Biology 210, 3135-3146, 2007.

[10] Tian, X., Iriarte-Diaz, J., Middleton, K., Galvao, R., Israeli, E., Roemer, A., Sullivan, A., Song, A., Swartz, S., Breuer, K., “Direct Measurements of the Kinematics and Dynamics of Bat Flight,” Bioinspiration & Biomimetics, Vol. 1, Dec. 2006, pp. S10–S18.

[11] Hedenstrom, A., Johansson, L., Wolf, M., von Busse, R., Winter, Y., Spedding, G., “Bat Flight Generates Complex Aerodynamic Tracks,” Science, Vol. 316, No. 5826, May 2007, pp. 894–897.

[12] Madangopal, R., Khan, Z.A., and Agrawal, S.K., "Biologically Inspired Design of Small Flapping Wing Air Vehicles Using Four-Bar Mechanisms and Quasi-Steady Aerodynamics," J. Mech. Design, 127, 2005, pp. 809-816.

[13] Galinski, C., Zbikowski, R., "Materials Challenges in the Design of an Insect-Like Flapping Wing Mechanism Based on a Four-Bar Linkage," Materials & Design, 28, No.3, 2007, pp. 783-796.

[14] Banala, S.K., Agrawal, S.K.,, "Design and Optimization of a Mechanism for Out-of-Plane Insect Winglike Motion with Twist," J. Mech. Design, 127, No.4, 2005, pp. 841-844.

[15] Conn, A.T., Burgess, S.C., Ling, C.S., "Design of a Parallel Crank-Rocker Flapping Mechanism for Insect-Inspired Micro Air Vehicles," Proceedings of the Institution of Mechanical Engineers Part C - J. Mechanical Engineering Science, 221, No.10, 2007, pp. 1211-1222.

[16] Jones, K.D., Bradshaw, C.J., Papadopoulos, J., Platzer, M.F., “Bio-inspired Design of Flapping-Wing Micro Air Vehicles," The Aeronautical Journal , Vol. 109, No. 1098, Aug. 2005.

[17] Lian, Y., Shyy, W., Viieru, D., and Zhang, B. N., “Membrane Wing Aerodynamics for Micro Air Vehicles,” Progress in Aerospace Sciences, Vol. 39, Nos. 6–7, 2003, pp. 425–465.

[18] Valenti, M., Bethke, B., Fiore, G., How J.P., Feron, E., “Indoor Multi-Vehicle Flight Testbed for Fault Detection, Isolation, and Recovery,” Proceedings of the AIAA Guidance, Navigation and Control Conference, Keystone, CO, August 2006.

[19] Troy, J.J., Erignac, C.A., Murray, P., “Closed-Loop Motion Capture Feedback Control of Small-Scale Aerial Vehicles," AIAA Paper 2007-2905, May 2007.

[20] Bagnell, J., Schneider, J., “Autonomous helicopter control using reinforcement Learning policy search methods,” Int’l Conf. Robotics and Automation. IEEE, 2001.

[21] Ng, A.H., Kim, J., Jordan, M., Sastry, S., “Autonomous helicopter flight via reinforcement learning,” Advances in Neural Information Processing Systems, Vancouver, BC, 2003.

[22] Smart, W.D., Kaelbling, L.P., “Effective reinforcement learning for mobile robots,” 2002 IEEE Intl. Conf. on Robotics and Automation (ICRA), vol. 4, Washington, D.C., 2002, pp. 3404–3410.

[23] Russel, S.J., Norvig, P.S., Artificial Intelligence: A Modern Approach. 2nd ed., Upper Saddle River: Prentice Hall/Pearson Education, 2003, pp. 775-776.

[24] Shoham, Y., Leyton-Brown, K., Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations., New York: Cambridge University Press, 2009, pp. 204-209.

[25] Brown, W., “Brushless DC Motor Control Made Easy: AN857”, Microchip Technology Inc., 2002.

BIOGRAPHICAL INFORMATION

Christopher Dale McMurrough received his Bachelor of Science in Computer Science Engineering in 2008 and his Master of Science in Computer Engineering in 2010 from the University of Texas at Arlington. He has worked in the Advanced Control Systems Laboratory at the Automation and Robotics Research Institute under the guidance of Dr. Frank L. Lewis since 2006. He was selected as a summer researcher for the Air Force Research Laboratory in Dayton, OH in 2008 and 2009. His research interests include mobile robots, micro air vehicles, real-time systems, robotic swarms, and embedded systems.