AN INNOVATIVE APPROACH FOR DATA COLLECTION AND HANDLING TO ENABLE ADVANCEMENTS IN MICRO AIR VEHICLE PERSISTENT SURVEILLANCE

A Thesis

by

RYAN DAVID GOODNIGHT

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2009

Major Subject: Aerospace Engineering

AN INNOVATIVE APPROACH FOR DATA COLLECTION AND HANDLING TO ENABLE ADVANCEMENTS IN MICRO AIR VEHICLE PERSISTENT SURVEILLANCE

A Thesis

by

RYAN DAVID GOODNIGHT

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Helen Reed
Committee Members,	John Valasek
	William Schneider
Head of Department,	Dimitris Lagoudas

August 2009

Major Subject: Aerospace Engineering

ABSTRACT

An Innovative Approach for Data Collection and Handling to Enable Advancements in
Micro Air Vehicle Persistent Surveillance. (August 2009)

Ryan David Goodnight, B.S., Texas A&M University
Chair of Advisory Committee: Dr. Helen Reed

The success of unmanned aerial vehicles (UAV) in the Iraq and Afghanistan conflicts has led to increased interest in further digitalization of the United States armed forces. Although unmanned systems have been a tool of the military for several decades, only recently have advances in the field of Micro-Electro-Mechanical Systems (MEMS) technology made it possible to develop systems capable of being transported by an individual soldier. These miniature unmanned systems, more commonly referred to as micro air vehicles (MAV), are envisioned by the Department of Defense as being an integral part of maintaining America's military superiority.

As researchers continue to make advances in the miniaturization of flight hardware, a new problem with regard to MAV field operations is beginning to present itself. To date, little work has been done to determine an effective means of collecting, analyzing, and handling information that can satisfy the goal of using MAVs as tools for persistent surveillance. Current systems, which focus on the transmission of analog video streams, have been very successful on larger UAVs such as the RQ-11 Raven but have proven to be very demanding of the operator. By implementing a new and innovative data processing methodology, currently existing hardware can be adapted to effectively present critical information with minimal user input.

Research currently being performed at Texas A&M University in the areas of attitude determination and image processing has yielded a new application of photographic projection. By replacing analog video with spatially aware high-resolution images, the present MAV handheld ground control stations (GCS) can be enhanced to reduce the number of functional manpower positions required during operation. Photographs captured by an MAV can be displayed above pre-existing satellite imagery to give an operator a lasting reference to the location of objects in his vicinity. This newly generated model also increases the functionality of micro air vehicles by allowing for target tracking and energy efficient perch and stare capabilities, both essential elements of persistent surveillance.

DEDICATION

I would like to dedicate this work primarily to my grandfather who instilled the love of aerospace in my heart through his own display of hard work and passion for aviation. His constant encouragement has kept me focused on reaching my dreams and I am eternally grateful for the many lessons he has taught me throughout my youth. I would also like to dedicate this work to my parents who have done everything in their power to help me attain the degrees that will enable my career. I could not have made it this far without their sacrifices and their constant reminders that hard work results in the greatest rewards. I would finally like to dedicate this work to my fiancée, Laura, who has made her own personal sacrifices so that I may continue to work towards fulfilling my goals.

ACKNOWLEDGEMENTS

The road to graduate school has been significantly paved by many people and without their influence I would not be at this point in my life.

I would first like to thank Dr. Helen Reed for convincing me that I am capable of incredible things and enabling me to attain an advanced degree through her ceaseless support and guidance. Without Dr. Reed’s generosity and dedication, I would not have been able to realize my full potential and for this I am endlessly thankful.

I would also like to thank the professors who have kept me engaged throughout my time here at Texas A&M University. In both my undergraduate and graduate educations, Dr. David Hyland has been crucial in keeping me energized about my work. His classes and extracurricular projects were extremely influential in helping me realize that I had chosen the perfect educational path. Without Dr. John Valasek’s expertise in the field of digital control theory, I would not have been able to shape my research into its current form. I am extremely thankful for his advice and encouragement during the times where my progress was limited. A special thanks is also deserving to Dr. William Schneider who graciously joined my graduate committee. My passion for space related activity has made his presence irreplaceable, and I wish to express my thanks for his commitment.

Finally, I would like to thank all of my fellow graduate students whom I have worked with and grown with over the past few years. The struggles of graduate school could not have been weathered without their presence and friendship.

NOMENCLATURE

V_a	Aircraft Velocity
h	Altitude
φ	Bank Angle
u	x-component velocity
v	y-component velocity
w	z-component velocity
q	Pitch Rate
θ	Pitch Angle
p	Roll Rate
r	Yaw Rate
ψ	Heading Angle
β	Sideslip Angle
Ω	Fuel Mass
ACTD	Advanced Concept Technology Demonstration
AHRS	Attitude Heading Reference Sensor
ASCII	American Standard Code for Information Interchange
AVGCS	Aerovironment Inc. Ground Control Station
CCD	Charged-Coupled Device
CMOS	Complementary Metal-Oxide-Semiconductor
DARPA	Defense Advanced Research Projects Agency

DCM	Discrete Control Distribution Matrix
ECS	Electronic Speed Controller
FAA	Federal Aviation Administration
FPGA	Field-Programmable Gate Array
GCS	Ground Control Station
GNC	Guidance, Navigation, and Control
GPS	Global Positioning System
GUI	Graphical User Interface
HFDS	Human Factors Design Standard
IMAV	Integrated Micro Air Vehicle
IPNAR	Image Projection for Navigation and Reconnaissance
KB	Kilobyte
Kbps	Kilobits per Second
LCD	Liquid Crystal Display
LiPo	Lithium Polymer
LoL	Loss-of-Link
LQR	Linear Quadratic Regulator
MAV	Micro Air Vehicle
MEMS	Micro-Electro-Mechanical Systems
MHz	Megahertz
NMEA	National Marine Electronics Association

NZSP	Non-Zero Set Point
PCB	Printed Circuit Board
PGCS	Portable Ground Control Station
PID	Proportional-integral-control
PIF	Proportional Integral Filter
QPM	Quad Partition Matrix
R/C	Radio Control
RAM	Random Access Memory
SD	Secure Digital
SDR	Sampled Data Regulator
SSD	Solid-State Disk
STM	State Transition Matrix
TAMU	Texas A&M University
UAV	Unmanned Aerial Vehicle
UA	Unmanned Aircraft
UI	User Interface
WAAS	Wide-Area Augmentation System

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF EQUATIONS

EQUATION Page

LIST OF TABLES

1. INTRODUCTION: ADVANCING MICRO AIR VEHICLE RESEARCH

1.1 A Historical Perspective

Since their initial conception during the American Civil War, unmanned aircraft (UA) have endured a wide variety of successes and failures stemming from the broad and vast number of attempted applications. Originally thought of as an offensive weapon, the first UAs were uncontrolled balloons retro-fitted with explosives. Although their intent was to provide a means of penetrating into the opposition's territory and wreaking havoc behind enemy lines, the lack of control generated the common feeling that these UAs created more of a risk than a tactical advantage. It wasn't until after the Second World War that UAs would begin to show their true potential for success as tools of reconnaissance and surveillance.

On May 1st of 1960, the United States government officially redefined its level of acceptable risk for manned reconnaissance missions when a U-2 and its pilot, Francis Gary Powers, were shot down over the USSR. Though the dangers of manned reconnaissance flights were widely known from losses suffered during World War II, the 23 manned aircraft and 179 airmen lost during the Cold War were enough to spur the Air Force to invest in an unmanned system that could perform the previously human only task. The resulting UA, the AQM-34 Firebee, was far from perfect; however, it was a successful start to the unmanned systems movement and was generally met with wide acclaim. With an impending war in Vietnam, the U.S. Air Force decided to begin

This thesis follows the style of the *AIAA Journal.*

integrating UAs into their standing fleet. Over the course of the war, the AQM-34 proved to be an increasingly dependable source of surveillance. By the end of 1975, more than 34,000 missions had been flown with nearly 83% of all vehicles being recovered safely. The success of this first unmanned system did not go unnoticed and soon UAs were being equipped for a wide variety of missions ranging from simple daytime photography to surface-to-air missile radar detection. Unmanned aircraft had begun to solidify their reputation as a critical element of the United States armed forces.

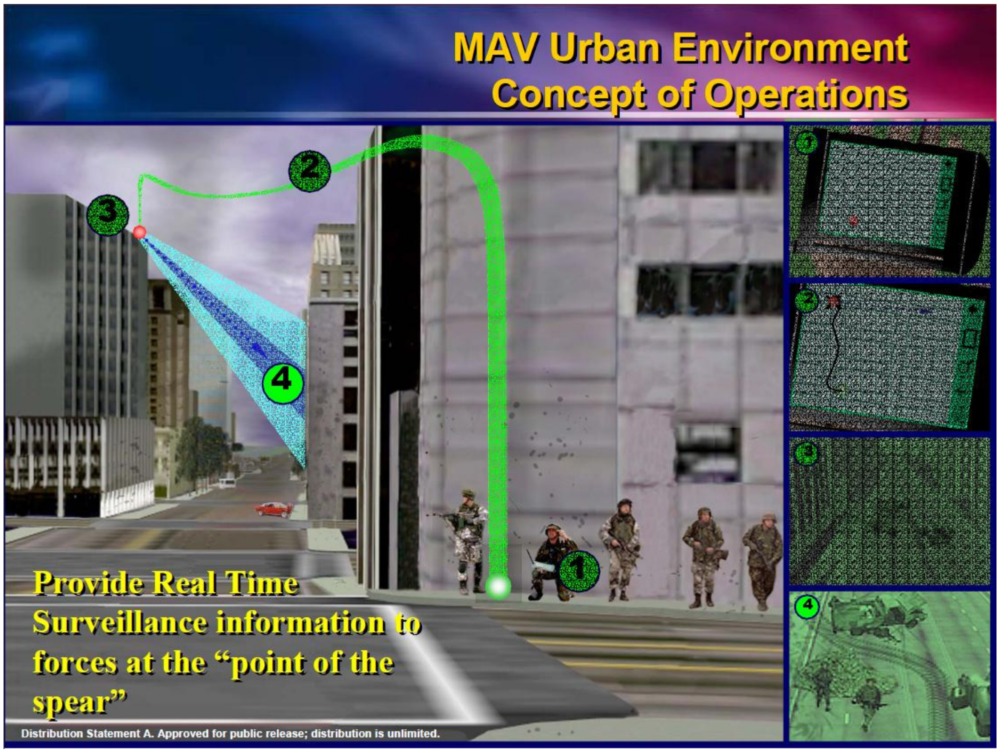
During the next 30 years, the United States would continue to invest in a plethora of unmanned aircraft as it looked to them to serve an expanding number of roles. The Gulf War saw UAs being used effectively as combat vehicles and less conventional aircraft such as the Helios began testing new concepts like a deployable communications system that could endlessly fly above ground forces. With the evolution of the computer age, unmanned systems technology began to develop at a highly accelerated rate and soon UAs were beginning to shrink in size. By 2003, squadrons of troops were already carrying back-packable UAs, like the RQ-11 Raven, that could be deployed on the ground for “over the hill” and predefined waypoint reconnaissance.[1] Systems that were previously operated from a command station were now able to be flown by an individual soldier with minimal training. Pushing the envelope even further, researchers have been designing micro air vehicles (MAV), a vehicle with a wing-span of less than 6 inches, for every soldier to carry.

Though the number of active UAs has increased, manned flight has continued to dominate airborne military activity. Losses, such as that of a Navy EP-3 carrying 24 men, resulted in the Department of Defense releasing a clearly defined vision for UAs. In its 2005 Unmanned Aircraft Systems Roadmap[2], the Office of the Secretary of Defense stated that it believed UAs should function primarily to reduce the "loss of human life in high threat environments." In addition, UAs should be used when it will create a "greater probability that the mission will be successful." Although these concepts may appear to be straight forward, they have played a significant role in defining the standards by which future UAs must be designed. As new systems are being developed, America's military command hopes to use this vision to continue successfully integrating UAs into its daily combat operations and further minimize the risk of human casualties.

With their past and present success as proof of their value, unmanned aircraft will play an important role in future conflicts. Researchers must continue to advance current systems and look for unique ways to solve the problems that are faced by the further miniaturization of UA hardware. By continuing to overcome these challenges, the United States looks to utilize UAs as a means of strengthening its position as the most powerful force in the world.

1.2 Background Motivation

The aforementioned evolution of unmanned aircraft has entered a new era during the past decade. Recognizing the continued miniaturization of hardware and the potential for an individually operated system, the Department of Defense issued an industry directive to begin developing micro air vehicle technology in 1996. The envisioned MAV operations are best illustrated by Figure 1.



MAV Urban Environment Concept of Operations

Provide Real Time Surveillance information to forces at the "point of the spear"

Distribution Statement A. Approved for public release; distribution is unlimited.

Figure 1: MAV Advanced Concept Technology Demonstration (ACTD) Concept of Operations[3]

Sam B. Wilson[3], the program manager of the Tactical Technology Office's (TTO) Micro Air Vehicle Project, identifies the "lack of situational awareness at the small unit level" as a critical problem in the new age of "hit and run" guerilla warfare. If we can develop a way to bring our commanders the "most accurate, real-time knowledge of the battlefield we will have an overwhelming advantage over [our] adversaries." Mr. Wilson and the Defense Advanced Research Projects Agency (DARPA) believe that MAVs are the technology that will give us this advantage.

Due to the infancy of the work, researchers are recognizing a number of problems associated with creating MAVs capable of performing their desired tasks.

Among these concerns is how to maximize controllability while concurrently minimizing the required user input. More specifically, what is the most effective and simple way of granting operators control of their vehicles? The general approach has so far been to mimic navigation and control software from larger UAs and apply them to smaller and more portable ground control stations. These systems can be difficult to use and have generally consisted of many different components. General Atomics Aeronautical's Portable Ground Control Station (PGCS) is an example of one such system. (Figure 2) This method of operation has the benefit of being able to provide detailed system information; however, field implementation has shown that these systems can require extensive training before they can be used effectively. The new paradigm of soldier supervised flight demands that "the MAV system be easy to use"[3] and many groups are looking for alternative solutions.

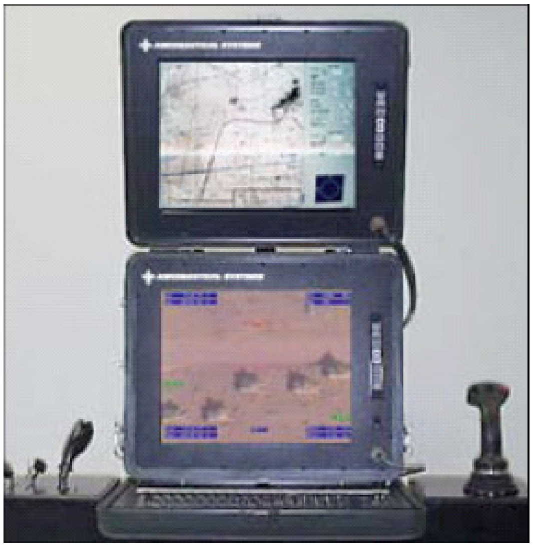


Figure 2: General Atomics PGCS[4]

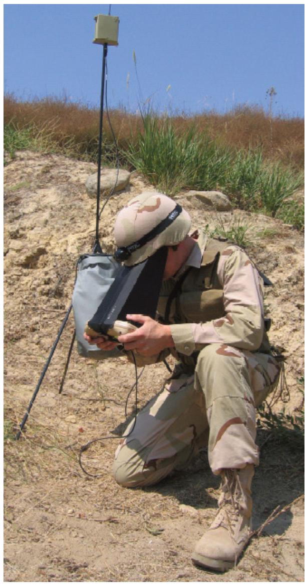


Figure 3: Aerovironment GCS[5]

Prospective MAV operations have also exposed a previously un-encountered data acquisition complexity. Larger unmanned reconnaissance vehicles such as the MQ-1 Predator are able to loiter and track targets with minimal interference from environmental disturbances. On the contrary, instability and disturbance mitigation has revealed itself as a defining challenge for the highly dynamic MAVs. Much work, which includes my research, is being done at Texas A&M University to better understand how MAVs can be designed to operate in an unsteady flight regime; however, this problem continues to pose a challenge for engineers as they attempt to collect data on current systems. While operators of larger platforms can rely on streaming video for target identification, Johansen et al.[6] of Brigham Young University has found that the high-frequency motion of MAV video streams makes it difficult for any operator to obtain useful information during flight. Some effort has been made to apply ground based frame stabilizers; however, these methods are computationally intensive and have generally demanded larger hardware than the envisioned MAV ground control stations. Attempts to mechanically stabilize the imaging hardware on-board the aircraft have also been investigated but due to physical payload limitations, MAVs simply cannot utilize this method.[7] With no clear solution available, DARPA has defined image stabilization as one of the enabling technologies required for future micro air vehicle integrations.

The concept of persistent surveillance, specifically “perch and stare” reconnaissance, is also unearthing new hurdles. Unlike conventional surveillance routines, “perch and stare” reconnaissance is a nascent tactic designed to collect information over extended periods of time. Researchers have historically struggled to design MAV systems capable of missions lasting longer than one hour, and facing the prospect of transmitting data over periods in excess of days, much work is currently being done to increase the energy density of battery power sources.[8] Multifunctional materials, rechargeable systems with solar cells, and hydride compounds are also being looked at as options for this problem; however, the only viable options currently available are lithium polymer (LiPo) batteries.[9] Further compounding the challenges of "perch and stare" reconnaissance is the need for high resolution imagery. As previously mentioned, currently used MAV systems depend on low resolution video streams for guidance and navigation and also, as is the case with Aerovironment's GCS, for data collection. (Figure 3) Though the quality of available sensors is constantly improving, physical constraints and bandwidth limitations have restricted MAVs to video cameras with resolutions in the range of a single megapixel. These video streams often suffer from granularity and their overall lack of clarity severely hinders their ability to make fine detections like the ones that will be required during a reconnaissance mission.

As these obstacles have been more clearly defined, DARPA has re-aligned its efforts to make a strong push for advancing MAV capabilities. The year 2008 marked the first time a contract was awarded with the specific goal of developing a technology directly related to "perch and stare" reconnaissance. Although the aforementioned challenges span a wide variety of sub-systems within the MAV class of UAs, they are collectively the focus of this research. With a simple shift in the applied data collection and handling methodology, it is believed that MAV research, particularly in the area of persistent surveillance, can be enabled with much greater efficiency.

1.3 Objective

The proceeding sections will show how the use of geo-spatially aware high resolution images, when presented in an appropriate way, offer a unique and effective substitute to the current paradigm of streamed video for MAV navigation and data collection. Three specific micro air vehicle areas will be targeted in this research:

- Image capturing and stabilization effects
- MAV systems integration and ground control software
- Control of semi-autonomous MAVs

By replacing an on-board complementary metal-oxide-semiconductor (CMOS) sensor with a higher resolution charged-coupled device (CCD), an MAV can be modified to take multi-megapixel snapshots with minimal increases of sensor mass and size. These images show a much higher level of detail than can be seen in individually captured frames of streamed video. Figure 4, which I created, shows how even a small portion of a 5-megapixel static image can provide detailed information where-as a screen capture from the analog video of a typical UAV is much poorer in quality. An example of a screen capture is shown in Figure 5.

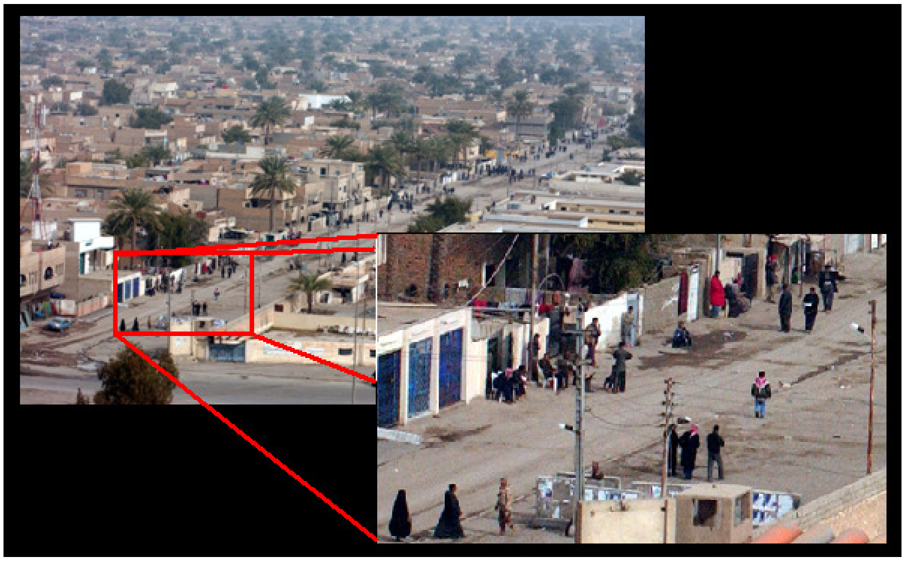


Figure 4: 5-megapixel Aerial Image and Regional Zoom

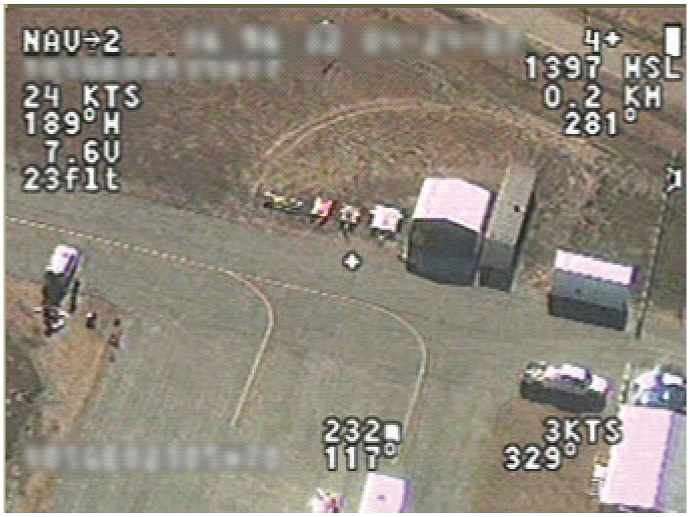
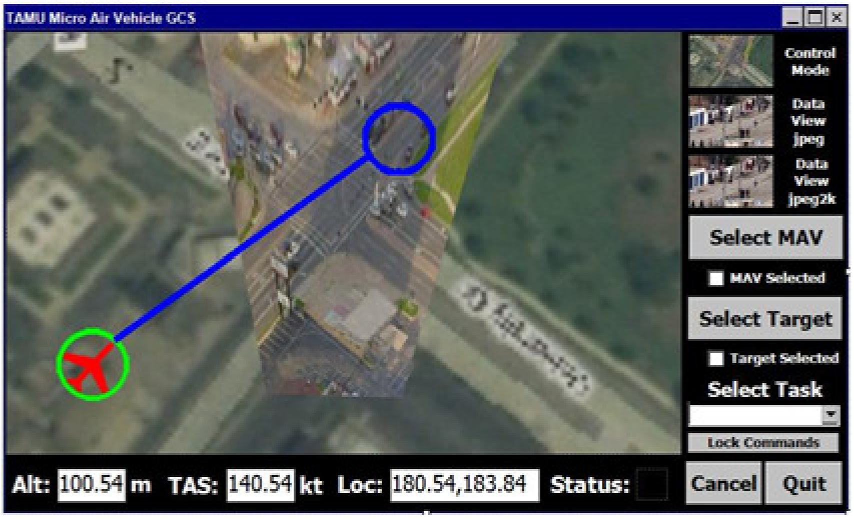


Figure 5: CMOS Screen Capture from Aerovironment UAV[5]

Compression rates and transmission limitations mean MAVs will not be able to send the images real-time; however, the well handled display of these images is enough to overcome that challenge. The process of integrating CCD imaging techniques is the first topic analyzed during this research.

Secondly, this research will look to improve upon current ground control stations and optimize for the handling of static images. Beginning with geo-referenced satellite imagery, a two-dimensional terrain map defines the visible control environment that an operator is presented with. Using vehicle state measurements that are collected using MEMS based sensors, the attitude and position from which the images are taken can be identified. Transformations can be applied to the images based off of the known capture attitude, and using the graphics processing capabilities of our custom portable ground control station, these images can be used to update pre-existing satellite imagery. An example of this technique is shown in Figure 6.



Alt: 100.54 m TAS: 140.54 kt Loc: 180.54,183.84 Status:

Figure 6: Attitude-Dependent Image Projection

By presenting the information in this manner, operators are no longer limited in their ability to detect targets because the high resolution imagery provides a lasting reference. An added benefit of static image display is the inherent stabilization effects. Using the estimated states of the MAV as a guide, software can be written to detect periods of minimum system motion and take pictures with the minimum probability of blurring. It will be shown how this data handling methodology aims to allow for the easier interpretation of mission critical information and therefore shortens the required training prior to use.

Finally, this research will look at the controls needed to achieve the necessary level of autonomy employed with this methodology. While there are many accepted definitions of autonomy, this research generally conforms to the Krashanitsa et. al. definition that identifies autonomy as the lack of "pilot or operator involved guidance and control".[10] Additionally, autonomous waypoint navigation will be explored; however, this falls within the aforementioned definition. It has been widely noted that in order to achieve persistent surveillance, a system must be capable of performing some steady state operation while its user directs his or her attention to more pertinent tasks. With a data handling approach like the one being presented, this requirement is even more stringent since it is completely dependent on the ability of the aircraft to maintain a sense of situational awareness. Specifically, this research aims to answer the question: In the face of limited computational power, what is the most well suited class of controller to provide waypoint navigation, target tracking, and disturbance mitigation? In addition, how can this controller be integrated into a ground control station to create a simple user interface with maximum functionality? The answers to these questions are an integral part of enabling future development of “perch and stare” technology. This research will not only aim to answer these questions but also show how there are many additional advantages to the adoption of this new approach.

1.4 Outline

Section 2 will introduce the elements which collectively define a micro air vehicle’s system configuration. For an MAV to successfully perform the tasks of retrieving information and relaying this data to a user, a certain set of flight hardware must be present. These necessary systems integrations are the focus of sub-section 2.1. A typical MAV will be described in detail with attention paid to the functionality of each component. This information will then translate into an overview of the specific hardware selections made in support of the proposed data handling methodology. Section 2 will continue with a look at the data processing elements of an MAV’s system configuration. Since the Department of Defense has clearly stated its desire for a small and portable ground control station, sub-section 2.2 will begin with an analysis of the hardware required to materialize this vision. State-of-the-art ground control stations will be compared to the GCS being utilized at Texas A&M and key differences between the systems will be highlighted. Sub-section 2.2 will then conclude with an in-depth description of the software functionality I developed for the Texas A&M handheld ground control station.

Section 3 will focus on the complications and considerations associated with the achievement of semi-autonomous MAV navigation. As flight hardware continues to be miniaturized, computational power limitations are playing a significant role in determining how MAVs will be controlled. Sub-section 3.1 provides a unique look at the source of these limitations while sub-section 3.2 details a number of viable controllers for use in commanding an MAV to its target location. Each of the proposed controllers will be written for a scenario which satisfies the requirements of the image projection methodology and their performances will be evaluated against their computational demands. The third section will conclude with a discussion of how the selected controller is integrated into the testing platform.

Section 4 is primarily concerned with outlining the procedure taken to verify the proposed methodology. The hardware and software described up to this point was taken into the field for testing and the performance of these systems is analyzed. In sub-section 4.2, the steps taken to prepare for testing, the testing procedure, and the future tests are all discussed in detail. The expected findings of these tests and an in-depth analysis of what additional steps will be taken in later stages of testing conclude this subsection.

Finally, in Section 5, a summary of all work done and the conclusions drawn from the results of that work are given in an effort to highlight the strengths of applying a new methodology to the practice of operating MAVs for the purpose of providing persistent surveillance with minimal user training.

2. MICRO AIR VEHICLE SYSTEM CONFIGURATIONS

Before applying image projection for navigation and reconnaissance (IPNAR) to MAV operations, the requirements of such a flight system must first be defined. In his lectures on the design of UAV systems, Dr. John Valasek of Texas A&M University, states that one should first look at the highest level of product definition before filtering down to the many subsystems and associated derived requirements. For the purpose of this research, the hardware and software utilized and developed was in support of the defined requirements below:

- An MAV for combat use is expected to be presented to an operator with all necessary hardware (flight vehicle, GCS, attitude determination components, communications systems, power supply, etc.).
- The MAV must be small and lightweight (6-in maximum dimension and 90g)[7] and all peripheral hardware should be able to be compactly carried by an operator with minimal impact on their ability to carry standard issue gear.
- MAV operations should have an endurance in excess of an hour, depending on the mission requirements, and be capable of travelling distances ranging from “over the hill” to several miles away.[7]
- The MAV should be easy to control and should be intuitive enough so that an untrained individual can quickly understand how to command the system.
- To allow for persistent surveillance capacities, the MAV should be capable of sustained flight configurations and data handling in such a way as to create a lasting reference of the point of reference.

The preceding 5 requirements are generally held as the primary drivers for an MAV flight system and while this research is more centralized around the data processing capabilities of a PGCS, any configuration utilized in the testing of IPNAR was done so as to conform to these standards. It is important to understand all hardware utilized in the entire MAV flight system so that available capabilities may be identified. This section will define these MAV system configurations and compare previously developed flight units with the ones developed in support of this research.

2.1 Vehicle Hardware Integration

The infrastructure required to take high resolution images from an airborne MAV, transmit them to a handheld ground control station, and present them in a manner so that a user gains a clear understanding of the current environmental state is significant. Figure 7 shows how this data acquisition and handling methodology would systematically take place.

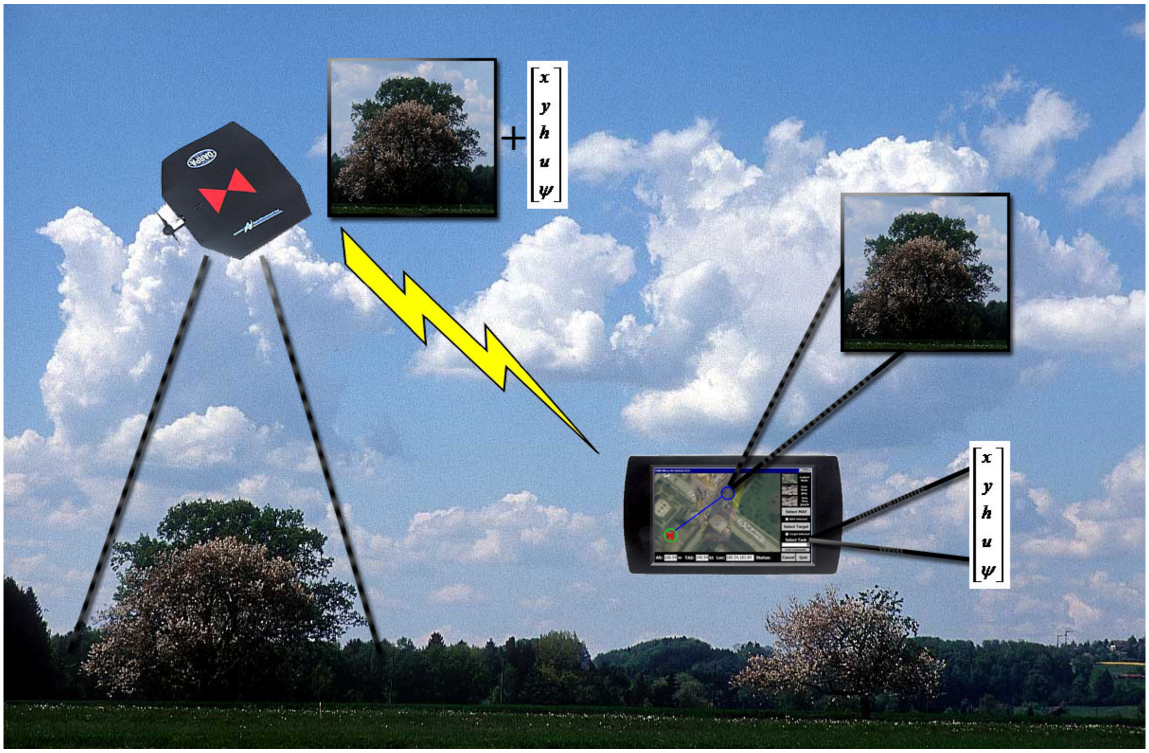


Figure 7: Systems Level View of MAV Data Acquisition and Handling

In the above figure it is shown that the flight unit will be responsible for navigating to a desired location, using an on-board imaging device for data collection, identifying the state (attitude, heading, and position) at which that information was collected, and passing this information back to a user along with the corresponding state information. To accomplish this task, it becomes essential that an MAV system be carrying a specific subset of hardware along with it during its mission.

2.1.1 Necessary Hardware and Examples

The selection of a propulsion system is one of the primary drivers in how an MAV flight unit will eventually take form. Because the propulsion of the MAV is directly related to its fuel source, several factors must be considered prior to making the final decision. For instance, a gas powered motor would not be a desirable selection for a vehicle that will be intended for multiple consecutive missions. A user would be required to carry reserve fuel which would directly impact the requirement for a compact system with little overhead. Other factors to consider include noise generation, reliability, and engine size. Again, referring to the requirements of an MAV system, it is essential to minimize these concerns while concurrently maximizing the unit's performance during flight operations. Additionally, it becomes important to select the means by which this propulsion system will be integrated into the MAV. Regardless of the motor selection, an MAV will need to be equipped with either an external propeller, an inset ducted fan, or in the case of flapping wing MAVs some propulsion producing mechanism like a wing. While each approach may have its own advantages, the efficiency and reliability of propellers cannot be denied.[11] The propulsion system and associated power source are also critical factors in the final system mass. These two subsystems together can make up as much as 50% - 62% of the total system mass.[12,13] These considerations have led most MAV design teams to the conclusion that small, lightweight electric motors combined with micro-propellers are currently the ideal solution.

Until recently, when rapid advancements in the energy density of lithium-ion polymer cells were made, MAVs were incapable of being powered electrically. Now that LiPo batteries have achieved energy densities in excess of 180 Wh/kg[14], they have become the popular choice for nearly all flight units. LiPo batteries are widely available and several cells can be configured in series to supply voltage multiples of 3.5 V, the typical load voltage of a single cell. The biggest challenge has now become the integration of these batteries into the flight unit. A number of unique configurations have been explored with varying degrees of success.

One example of these configurations can be seen in an early iteration of Aerovironment's WASP MAV. In the Block I version of the WASP, Figure 8, developed for DARPA, the LiPo batteries were placed into the wing of the MAV with the upper surface of the batteries serving to enhance lift generation.

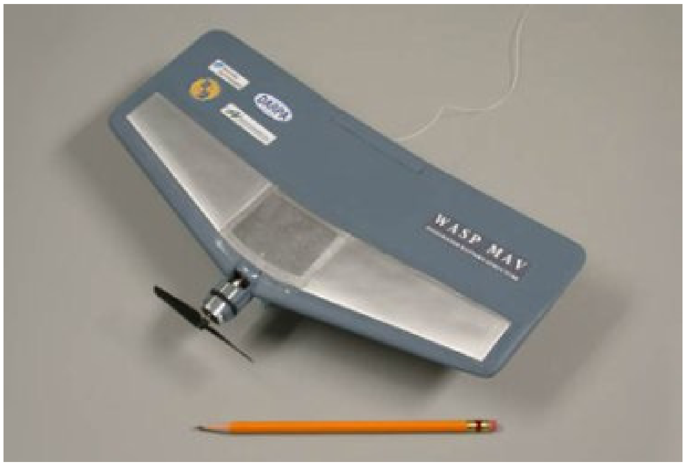


Figure 8: WASP Block I Configuration[7]

This battery placement was the first example of researchers exploring the possibility of further eliminating unnecessary infrastructure mass. Its implementation highlights a critical element needing attention during the design of these highly integrated MAVs, namely, the multi-functionality of components. Once the power supply and propulsion system have been selected, a bridge between this hardware and the operator must be chosen. This bridge consists of a mechanism to receive input signals from a controller and a second mechanism to convert these signals into a usable form by the propulsion system. For the case of the electric motor selection these components are an R/C receiver and electric speed controller (ECS). For extremely small micro air vehicles, brushless motors do not have any appreciable advantage in size and weight and brushed DC motors are powerful enough to satisfy most demands. This fact turns out to expose a second advantage in that brushed ECSs are smaller and less expensive than their brushless equivalent. These ECSs serve to vary the input voltage to a motor by way of a pulse width modulation signal that is received through radio control. These R/C systems have been very effective in commanding MAVs up to distances of several hundred meters and due their availability and ease of use, commercial R/C communication systems have become very popular among MAV developers. Commercial systems; however, have the disadvantage of forcing the MAV design to conform to the size and shape of the generically designed hardware. This drastically reduces the number of possible hardware configurations available to a developer during the design phase.

There are additional communication systems actively being employed in combat ready systems such as the one found on the WASP Block III which claims a line-of-sight operations limit of 5 km.[15] To achieve operating distances of this magnitude, US DoD Joint Spectrum Center JF-12/DD1494 Stage 4 approved frequencies must be used. Selecting these components and ensuring that they properly mate with the hardware in the ground control station is a critical requirement of successfully operating an MAV flight system. Integrating the payload onto an MAV is essential since it alone is the reason for designing a flight system in the first place. This becomes particularly important in the overall system composition because any mass that can be saved from other sub-systems can be redirected towards this sensor payload. Research has found that MAVs, whose flight performance is extremely susceptible to small variations in mass, are typically expected to be able to devote 12%-15% of the final system mass to a payload.[12,13] With a target overall mass of 90 grams, this does not leave much room for the data acquisition hardware. This challenge has led many researchers to look for innovative ways to further minimize any extraneous infrastructure. Typical payloads being used on MAVs range anywhere from a CCD imager, such as the one in this research, to atmospheric particle detection instruments.[16] Successfully integrating a sensor payload onto an MAV is not limited to just consideration of the sensor alone. There is additional hardware that must be taken into account that will enable the sensor to perform its desired tasks. In the case of a digital imager, some form of communications hardware will be needed for transmission and downlink of data. In some systems like the WASP III this communications infrastructure may be the same as the one used for controlling the aircraft. This is not always the case and often a separate dedicated transmitter is used. Over the past decade, more interest has been devoted to the miniaturization and advancement of autopilot systems and their associated control routines than any other subsystem found on an MAV.

The reason for this focus on autonomy work is a result of a number of challenges unique to low Reynolds number aircraft. Primarily, the highly turbulent flight regime under which MAVs operate has made stabilized flight nearly unobtainable. The smaller the aircraft becomes, the more these concerns play a role in determining how the system will interact with the environment during the course of completing its mission. To date, researchers have identified two potential approaches to solving this problem. The first of these solutions involves a continued effort to increase the accuracy and sampling rate of state measurement guided controllers.[17] With this method, designers are looking at the hardware to provide a means of physically counteracting the dynamic effects of the environment and allowing for stabilized flight. This approach has an eventual goal of being able to design computational systems capable of running full state estimators on-board the MAV. I, along with additional Texas A&M University aerospace students, am currently working to make the first fully integrated state-estimate driven MAV guidance and navigation computer. This will be discussed in more detail later in this section.

The second vehicle stabilization paradigm being utilized by researchers embraces the instability of MAV flight operations and looks to software for solutions to the problems associated with jittery data collection.[6] Because the processing capabilities of ground control stations have traditionally outperformed those of an MAV flight unit, this approach focuses on minimizing computations performed in the air and instead analyzing information from a post-processed perspective on the GCS. Blur reduction and target tracking through frame stabilization are a couple of the capabilities being explored on ground control stations and developers are finding some success. Recent advancements have been significant enough to demand that MAV developers now look at a combination of the two methodologies when designing their system.

Another cause for intense focus on advancing autopilot systems for MAVs was the realization that micro air vehicles would rely heavily on autonomous capabilities during their envisioned missions. In the dangerous combat scenarios in which these flight units might be deployed, an operator may not be able to devote full attention to the piloting of their air vehicle. A solution that can be launched with a target destination for hands-free persistent surveillance would necessitate a high level of computational ability that had previously not been available at such a small scale. Fortunately, the digital revolution in the cellular communications industry has helped make MEMS technology widely available. These MEMS components are ideal for developing lightweight and compact autopilot systems with processing capabilities orders of magnitude higher than when MAV development began. It is now feasible to purchase multi-axis accelerometers, gyroscopes, and magnetometers as well as GPS receivers, pressure sensors, thermal detectors, and fast micro-controllers at a fraction of the cost and size of similar hardware found on larger UAs. Integrating these sensors into a flight unit has become a critical factor in designs hoping to satisfy the core requirements laid out for MAVs. The final subsystems needing consideration when attempting to fully integrate a micro air vehicle into an operating flight configuration are the control surface actuators and the vehicle airframe.

For MAVs, it has become understood that, due to size and mass limitations, control surfaces must serve in the dual capacity of a lifting body as well as the element for creating controlled dynamic flight. Whether this includes using the trailing edge of a lifting body as an extended elevon or, in the case of some flapping wing MAVs, utilizing the stabilization effects of an "inverted V tail" with actuators for altitude and heading, researchers must look for unique ways of overcoming control difficulties when designing, new and advanced aerial vehicles.[18,19]

Historically, propeller driven MAVs have not been required to be designed to the same structural standards as larger unmanned systems. This is not to say that these micro air vehicle designers can ignore structural concerns when creating a new vehicle. In fact, many systems have suffered from increased susceptibility to disturbances as a result of the stiff materials being used in their construction. Structural considerations play a heavy role in the advancing of flapping wing MAVs; however. Dr. Gregg Abate, Team Leader for the Air Force Research Lab's Airframe Dynamics & Robust Control Weapon Dynamics & Controls Sciences Branch, explains that "wing flexibility is crucial in the generation of lift for a flapping wing" and researchers are trying to "exploit this interaction" and develop new light-weight materials.[20] As more progress continues to take place in advancing technology in each of the above subsystems, MAVs will begin to close the gap between their projected role in combat operations and their current state. Researchers must continue to look for new and innovative ways to integrate these systems as part of the ultimate pursuit of this goal.

An example of an MAV where similar hardware was integrated and successfully flown was the Dragonfly MAV developed at the University of Arizona.[18] This system, which was the winner of the first U.S.-European Micro-Aerial Vehicle Technology

Demonstration and Assessment held in Germany in 2005, was able to be programmed with GPS waypoints for navigation. In addition, the Dragonfly had built-in autonomous functionality such as climb and return-to-base as well as the ability to maintain flight stability in the presence of gusting winds. A schematic of the Dragonfly MAV, made available by the University of Arizona through R. Krashanitsa et al.[7], can be seen in Figure 9 with a breakdown of each component listed in Table 1.

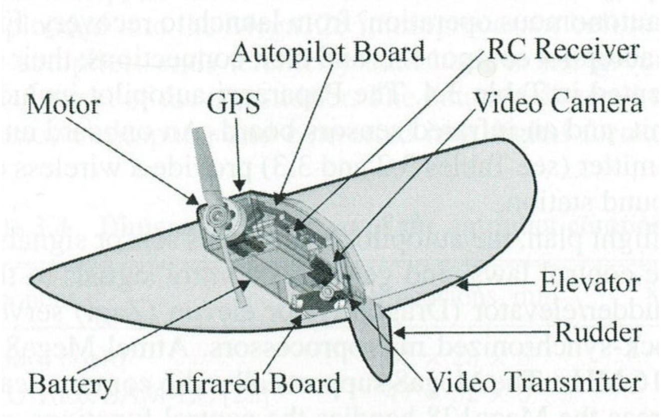


Figure 9: Subsystem Integration Schematic of Dragonfly MAV[7]

Table 1: Components of the Dragonfly[7]

Component	**Description**	**Mass, g**
Airframe	Kevlar/Rohacell foam	45
Autopilot	Paparazzi	30
R/C Receiver/Controller	PENTA 5/Phoenix-10	16
2 Micro Servos	Blue Arrow BA-TS	7
3-cell Li-Po Battery	Thunder Power, 730 mAh	46
Motor/Propeller	AC-DIYMOT-2207/APC 4.75 x 4.75 in.	35
Video Camera/Transmitter	MO-S508/BWAV240050	7
Total	----------	**186**

Many of the subsystems that were described earlier in this text can be seen here in this table. One important thing to note is that this system comes close to satisfying the envisioned specifications for an MAV and shows how as systems become smaller their subsystem mass distributions become more pronounced. There is an industry-wide push to change the mass distribution percentages for subsystems such as the airframe and batteries; however, at the current time, these ratios are pretty indicative of most MAVs. The autonomous functionality of the Dragonfly MAV was made possible by the Paparazzi autopilot system. This system has been widely used on a variety of UAVs and MAVs around the world and, due to its free and open-source nature, Paparazzi autopilots evolved extensively since their introduction in 2003.[21] The most recent version of the Paparazzi system is shown in Figure 10.

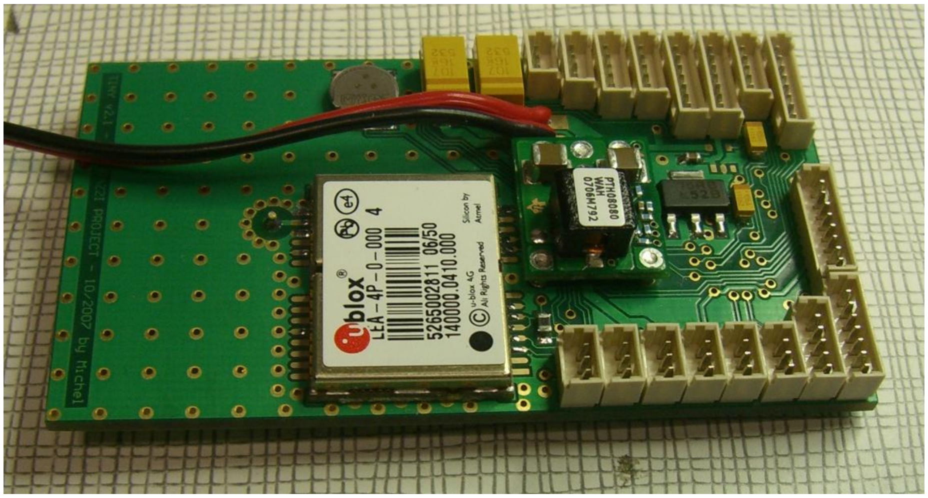


Figure 10: Paparazzi Tiny V2.1 Autopilot System[22]

This autopilot system was developed by a small group of engineers who recognized the fundamental lack of an easy to use and widely available autopilot system for groups developing and testing flight hardware. This newest Paparazzi system, including its GPS and 18 mm antenna, weighs in at a miniscule 24 grams and measures only 70.8 x 40 mm in size. The system is also extremely adaptable to a number of sensors that can be used to detect a number of parameters during flight. For the most part; however, most applications with the Paparazzi system rely on “6 orthogonal infrared temperature sensors to estimate the orientation of the aircraft relative to the warm earth and cold sky.”[22] This is regarded as an inaccurate way to determine a UAs attitude and orientation and this is the primary hindrance encountered when integrating this system onto an MAV. The reason for the success of the Paparazzi autopilot can be traced to the accessibility of its resource documentation and the lack of overhead needed when designing a custom autopilot system. Most of the teams participating in the MAV07 competition, which Texas A&M University attended, used this pre-made system for these reasons as well as the fact that it comes with a ground control station that has been designed specifically for use with the autopilot board. This reduced the amount of time that teams needed to spend developing their flight unit and allowed for additional focus on operations training. It should be noted, however, that one undesirable result of utilizing systems such as the Paparazzi is that it restricts a system to capabilities which may not satisfy the requirements of the mission to which it is being applied. This was especially evident at the MAV07 competition in Toulouse, France where no team using the Paparazzi system was able to complete all of the mission objectives. A second negative result of using a pre-built navigation system is that the MAV will need to conform to the dimensions of the hardware. This will clearly pose an ever growing problem as the size envelope for MAVs continues to shrink. Although multiple revisions were made to the airframe and structural composition of the Dragonfly, its continued reliance on outside hardware has limited its development and highlights a major hurdle in the development of MAVs.

The configuration of hardware integrated on the Dragonfly MAV is typical of many systems developed outside of industry. The availability of "off the shelf" components primarily designed for hobbyist use has made it easy to quickly gather the necessary pieces required to build a working flight unit. Despite the benefits from this component availability, the progress of MAV technology advancement has suffered as a result of this design methodology. For the vast majority of the past decade, the micro air vehicle community directed the bulk of their efforts towards improving the flight characteristics of MAVs and they were able to rely on the commercial RC community to continue developing smaller communications, actuator, and navigation hardware. As researchers have begun to develop better models of low Reynolds number aerodynamics, a technology bottleneck has been reached. Work is now shifting directions towards the continued miniaturization of faster and more powerful flight components. To develop these custom navigation systems is not only time-consuming but also very costly and this has left many research institutions and universities looking towards private companies such as Aerovironment to invest in developing this new hardware. Despite their progress in creating these new components, many of the early micro air vehicles developed in the private sector resembled those found in academia. One of the most famous MAVs, the Black Widow, was originally built using a methodology very similar to the University of Arizona's with their Dragonfly.

The Black Widow MAV was, like many other early micro air vehicle projects, a DARPA sponsored undertaking to provide a proof-of-concept demonstration of a system carrying a digital camera and providing a live video stream to a ground-based operator while maximizing mission endurance. The project's results were regarded as a success and confirmed that MAVs could potentially be a valuable tool to the armed forces. Although this system was developed many years before the Dragonfly, Figure 11 shows how its subsystem integration conformed to the aforementioned methodology.

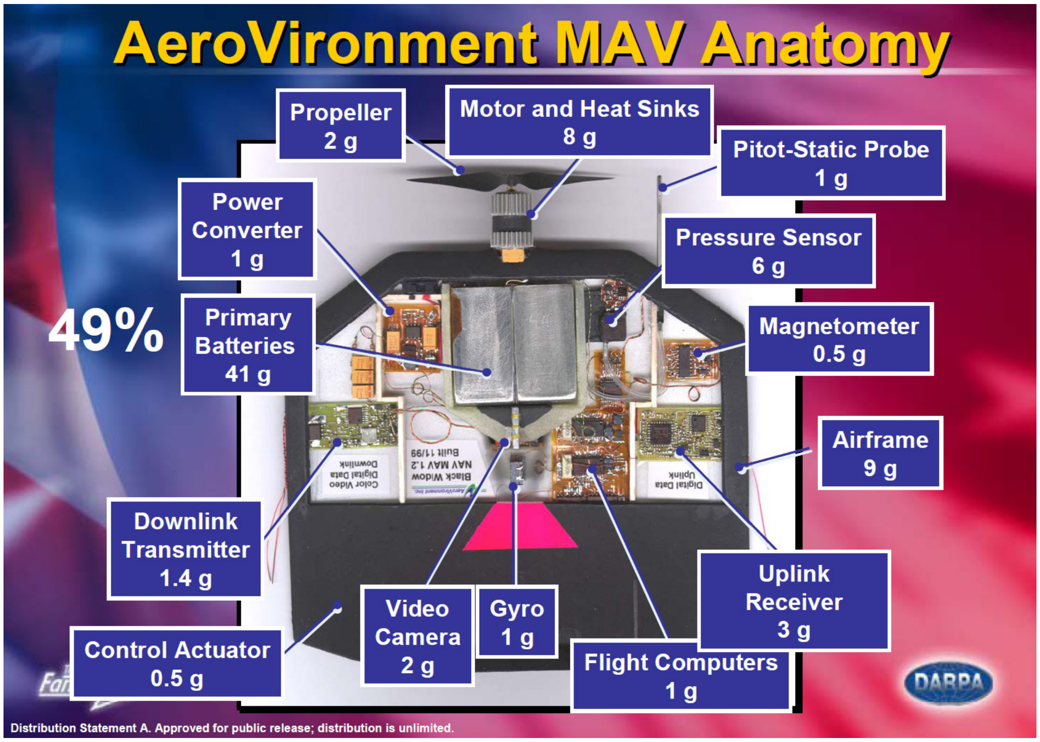


Figure 11: Aerovironment Black Widow MAV Subsystem Anatomy[3]

The Black Widow, like the Dragonfly, underwent several iterations before arriving at the award-winning configuration shown above, but it can be clearly seen how similar the two systems are in their integration. One striking difference between the two flight systems is the separation of the attitude sensors and navigation computer on the Black Widow compared to the all-inclusive flight computer utilized on the Dragonfly. This difference can be partially attributed to difference in design expertise and the contrasting levels of funding available for the two projects. Although the approach taken with the design of the Black Widow has the benefit of increasing the number of options for integrating sensors on-board a vehicle, it has a negative impact on the primary requirements of MAV design, namely, minimizing mass and size. All of the excess silicon used to make the separate circuits adds to the amount of infrastructure that researchers are so desperately trying to eliminate. Despite this inefficient approach, the Black Widow was able to supply its user with a low-resolution video feed complete with system information overlay such as magnetic heading, altitude, airspeed, and battery voltage. J. Grasmeyer et al. explains that it was also the first definitive proof that a micro air vehicle could be made with a span of less than 6-inches and weighing less than 85 grams without impacting its ability to carry enough power to fly for 30 minutes at ranges up to 1.8 km.[23] Another unique feature of the Black Widow was its custom-developed actuators located near the rudder and elevator for directional control. The mass difference between these actuators and those used on the Dragonfly, 0.5 grams versus 3.5 grams, is enough to highlight the advantages of being able to design proprietary hardware for your MAV. As Aerovironment secured new contracts for their micro air vehicle projects, they continued with this custom-design approach and now have many vehicles, such as the WASP III, in combat ready form.

The attention paid to these examples and what subsystems need to be included in the design of an MAV was intended to identify the areas where a new approach can be taken to the MAV design process.

2.1.2 Texas A&M Integrated MAV (IMAV)

Here at Texas A&M University, I have been involved in an active effort to eliminate the issues resulting from separate components being integrated together to form a complete system. The idea that every piece of electronics hardware could be integrated onto a single chipset is one that would have many advantages. Fundamentally, this would minimize the amount of extraneous material used in building up an infrastructure for the craft. Typical printed circuit boards (PCB) have a density of roughly 2 g/cm^3 and for a flight unit with a 6 inch span this would equal approximately 46 grams. Referring back to Table 1 regarding the Dragonfly, it can be seen that this is fairly equivalent to the amount of mass needed for the airframe. Taking into consideration the fact that the amount of excess PCB needed for the individual components in these other systems will be virtually eliminated, it is clear to see how this design approach could have immediate benefits. Additionally, current circuit manufacturing techniques could allow this defined component layout to be built on any unique board shape required by a given design. In this way, many of the problems associated with signal interference, magnetic field generation, and thermal fluctuations would already be accounted for. Depending on the specific requirements of the design, this board could also serve in the dual capacity of providing the structural rigidity needed during flight. For a vehicle that needs increased flexibility, a flexible printed circuit board could allow for any necessary bending of the structure. For fast moving MAVs that need to have increased structural integrity for damage mitigation, multi-layered PCBs would be a good choice. Designers will, however, need to consider some side effects when using these different types of PCB. For the case of using flexible printed circuit boards, developers will need to monitor the deflections of the board to ensure that no surface mounted components are in danger of popping off of the surface during flight.

In conforming to the core MAV requirements at the beginning of this section, a preliminary design for a portable, hand launched flight unit was developed and an Inventor© rendering of its layout can be seen in Figure 12.

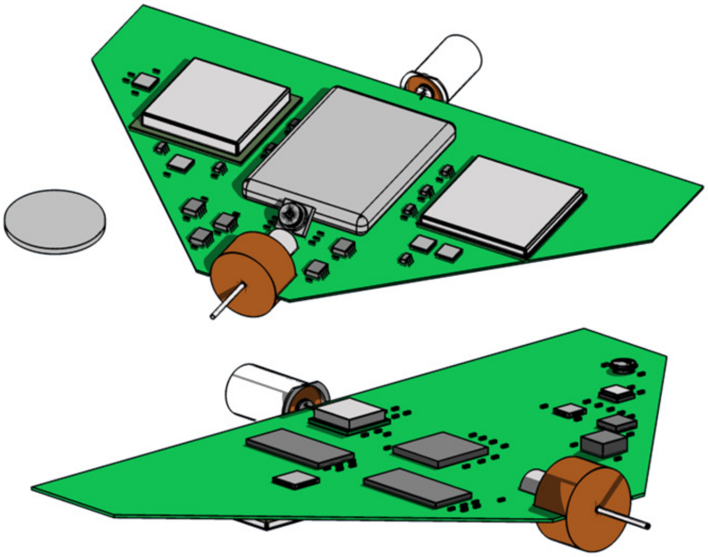


Figure 12: Texas A&M University Integrated MAV (IMAV) Inventor© Rendering

The diagram above gives a good visual representation of how the concept of a fully integrated MAV could take shape. Each of the blocks on the upper and lower surface of the PCB represents one of the required components needed to bring this flight unit into fruition. For propulsion, the TAMU IMAV, uses a brushed motor in combination with a high efficiency propeller. The possibility of a ducted fan was considered, but despite the advantage of limiting the number of moving external parts, the benefits of propellers as mentioned earlier could not be denied. The speed control for the motor is performed by an ECS that is printed onto the fully integrated circuit alongside the navigation hardware. Like most MAVs, the IMAV is powered by a lithium polymer battery and has a projected endurance in excess of 15 minutes at maximum operating conditions. Nominal performance drastically increases this projection. This battery is located along the underside of the vehicle's longitudinal axis. Because the battery is the largest of any on-board component, this placement has a stabilizing effect during flight.

For communication and data relay, the IMAV uses a 900 MHz radio model with a 3-mile line of sight range and 40 KB/s data transfer rate.[24] This range is comparable to vehicles such as the WASP III and, for its size and intended purpose of providing persistent surveillance of a near-by target, this satisfies the MAV requirements set forth by DARPA. The block directly behind the motor represents the sensor payload on the IMAV.

In this configuration a small, high-resolution CMOS camera is shown. This imager is similar to the ones often seen in cell-phones and is the smallest and lightest type of camera available. Once advancements in charge coupled device technology have miniaturized CCD sensors to a similar size, they will be the preferred choice; however, despite great progress, they are still significantly larger. The CMOS sensor is angled downward so that, in flight, a ground-based target can be selected and tracked while the IMAV loiters above. As previously mentioned, this payload is not limited to this location. The ability to re-configure the chip placements allows for the camera to be placed anywhere on the structure of the vehicle. The IMAV is also state-of-the art with its inclusion of all the necessary MEMS sensors for a full attitude heading reference sensor (AHRS). In addition to a 3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer, the IMAV has a 20 channel global positioning system complete with Wide-Area Augmentation System (WAAS). With these systems and an onboard extended Kalman filter it is possible to fully estimate the states of the vehicle in flight. States that can be estimated include position, altitude, airspeed, roll, pitch, yaw, roll rate, pitch rate, and yaw rate.

Through the measurement of these states, the Kalman filter, complimented with calibration data stored in flash memory, can provide a 3D orientation matrix with a 0.5° degree dynamic accuracy. This level of precision will be required if the image collection and projection methodology being proposed in this research is to be applied to an operating micro air vehicle. The antenna being used with the GPS receiver is also unique. Helical antennas are capable of receiving a signal lock from almost any orientation and this proves to be an important feature of the IMAV. During the many proposed MAV missions, such as perching and staring, there is no guarantee that any given surface will have exposure to the satellite array. With the potential for random vehicle orientations, a patch antenna, like the one used on the Paparazzi system, will be limited in the reliability of maintaining a signal lock. There has been an industry-wide interest in developing a fully integrated AHRS for MAV use and several small external units have recently hit the market.[25] The computer that handles the computational load of the extended Kalman filter is a 266 MHz ARM9 processor. These processors are most commonly used in personal digital assistants (PDA) and are exceptionally powerful for their size and energy requirements. Processors with faster clock speeds are available; however, at the time that this vehicle was designed, the ARM9 was the best performing option. Complimenting this ARM9 processor is 64 MB of synchronous dynamic random access memory (SDRAM) and 512 MB of Flash memory. The Flash memory houses any data that is collected during the course of the mission that has yet to be transmitted. Although it is not shown in the Inventor© drawing above, the IMAV takes a unique approach to control surface actuation. Rather than rely on micro and sub-micro servo motors, the IMAV developers looked to digital camera zoom mechanisms for the hardware to actuate the rear elevons. These extremely small piezoelectric linear actuators provide for rapid motion with minimal power required. As is the case with most flying wings, the IMAV gains its lateral stability from small winglets which are also not shown.

One final feature of this micro air vehicle is that it uses a never-before attempted means of shaping its lifting surface. It has already been mentioned that the circuit board will provide the structural support for the MAV and the components on this circuit board can be moved into many unique configurations, but there has been no mention of how the flow field around this vehicle will be controlled while in flight. The researchers at Texas A&M have converged upon a solution that will allow for nearly any imaginable surface profile. Using a lightweight and thin heat-shrinking plastic material, the entirety of the IMAV can be covered to create a smooth lifting body. To control the location of the root chord thickness and the amount of camber, chips of varying height can be arranged in such a way so that when the plastic shrinks around the components, the desired shape results. To date, no publically available information suggests that this approach has been taken before.

The unique features of the IMAV were developed in order to find a viable solution to the mass minimization problems associated with the sub-system integration on micro air vehicles. Concurrently, there was a secondary focus on developing a system that could be capable of expanding MAV persistent surveillance capabilities. This flight unit was able to provide a means of reducing the total system mass by eliminating unnecessary support infrastructure while at the same time pushing the envelope of MAV hardware capabilities by implementing a fully developed AHRS. By determining the aircraft state on-board in real-time, the IMAV is also unique in that it could stabilize the images it takes by recording the angular rates at which a given image was taken or by looking for periods of minimal acceleration and delaying its image capture until that time. These elements of state measurement and estimation will be critical in applying the data handling methodology studied in this research.

2.2 Common Handheld Ground Control Station

While a successfully designed flight unit is a major part of the full Micro Air Vehicle infrastructure, it alone is not enough to satisfy the operational requirements needed for persistent surveillance. Upon the completion of in-flight data collection, a medium for data processing and display will serve as the bridge between a user and his vehicle. In addition to raw data handling, this portable unit will need to function as the interface for commanding and navigating the unmanned aircraft. Like the micro air vehicle, ground control stations have evolved since their early developmental stages and the techniques being used to expand their capabilities will be critical if MAVs are to assume their desired role. The section will look at the necessary subsystems that need consideration when designing a GCS, briefly describe the capabilities and control methodology of current systems, and present an overview of the functionality of the Texas A&M University Portable Ground Control Station (PGCS).

2.2.1 Components and Current Systems

To understand the design driving requirements for a GCS, one can refer back to the core MAV requirements identified at the beginning of this section. Primarily, the hardware necessitated by the desire for a small and lightweight unit that an individual soldier could carry should be identified. The following are critical elements of MAV ground control stations:

- Visual Display
- Data Transmission and Communications System
- Computer and Data Storage

- Global Positioning System (GPS)
- Control Input Mechanism
- User Interface (UI)

PGCSs should, at a minimum, have some visual display capability. This is almost always done through the use of a liquid crystal display (LCD). LCD screens have become a very affordable means of digital information display. While they are available in a number of sizes, what gives LCD screens their utility in GCS design is their ability to output high resolution color images while maintaining a narrow profile. This aids developers in keeping space available for additional hardware that may be required as part of other control station subsystems. LCD screens have also been created with resistive and capacitive touch sensitivity; however, this feature has only become available on smaller LCD panels in recent years. Low power requirements and easy serial interfacing are also advantages of LCD screens, but they are not without problems. As with any digital display, outdoor visibility is limited by ambient sunlight. This has led many MAV ground control stations to be designed for use with a viewing shroud as can be seen in Figure 3. This too creates a number of operational problems, the least of which include the inability to quickly reference the GCS without having to divert all attention away from the current environment. In addition, a shroud increases the amount of support infrastructure that an operator must carry with him in order to properly view the screen. Both of these concerns make a system with this design ineffective at fulfilling the requirements of a PGCS for MAV persistent surveillance missions. There is some current research being done to build 9:1 contrast ratio LCD panels that are illuminated by sunlight instead of the traditional backlighting and as a result reduce power consumption by up to 75% to allow for excellent viewing in outdoor environments.[26] As these breakthrough panels become smaller and more advanced they could potentially pose a solution to the current viewing problems.

Clearly, an MAV PGCS should be able to maintain a data link between itself and the flight unit(s) from which it is receiving information. This communications system is responsible for not only receiving raw information from the vehicle(s) but also sending directional commands that instructs an unmanned craft to a desired location where it can perform a specific task. Several communications system configurations have become popular and vary greatly in design. The first of these two approaches involves separating the data transmission network from the control network.[27] This is generally the most popular form of communication because the separation of networks allows for the use of pre-built hardware in place of custom designed data links. Despite this benefit, the duplicity of receivers negatively impacts the flight unit mass. Additionally, dividing the communications tasks between the two networks brings with it an inherent redundancy. In the event that one signal is lost during flight, having the second will allow for some remaining functionality. This is a feature that was exploited by many of the teams competing in the outdoor flight portion of the MAV07 Competition. Systems that take the alternate approach of using a single communications bridge to both transmit data and control the aircraft are more adaptable to smaller vehicles. Beyond the obvious benefit of minimizing the hardware on-board the vehicle, this integration method also helps with signal interference concerns. The GCS is not impacted to the degree that a small MAV would be; however, there is still a risk of fluctuating signal strengths when two devices are used in close proximity.[28] This will become of even more concern as vehicles and GCSs become increasingly smaller. The Texas A&M IMAV is an example of a system that only uses one communication line between the vehicle and the ground control station. Regardless of the infrastructure, nearly all communications systems function under line of sight operations. This has left many ground control stations hindered by the bulky directional antenna that must be deployed before use. For missions with high data transfer rates, particularly those at a far distance where signal strengths begin to decay, this external antenna may be unavoidable. Research is underway though to determine if a work-around is available for close proximity missions, such as the perch and stare observation of an alleyway, where the amount of data needing transmission is minimal.

Like the MAV flight unit, the GCS has intense computational loads that it must bear. Whereas the vehicle computer only needs to perform sensor measurements, state estimation, and controller handling, the GCS computer will additionally be required to actively process data and display it to the user in an appropriate form. In general this means that the processor on the GCS will need the capability of handling graphics and serial devices. Again, the cellular phone industry has inconsequentially been an enabling driver in making miniature computers capable of handling these GCS demands. Devices such as the Apple iPhone© and Research in Motion Blackberry© have used new state-of-the-art 667 MHz ARM11 processors to handle a number of simultaneously running programs while actively engaged in graphical display.[29] In much the same way that Wilson[3] envisioned a palm pilot GCS, ground control stations can be designed to be very portable and power efficient if these computers are used. For data storage, a MAV ground control station will also need some form of memory. The low cost and high availability of secure digital (SD) Flash makes it the obvious choice for a GCS; however, there are alternative options that deserve mention. Solid-state disks (SSD) are one viable candidate. SSDs are a more rugged configuration of flash memory and have been developed for applications in consumer electronics. They have fast start-up time and are lightweight in comparison to standard hard drives of similar storage capacity. These SSDs are expensive though and although they are not excessively big, they are large enough to make standard SD flash memory a more preferable selection.

The presence of a GPS unit on an MAV ground control station serves a number of key roles. Predominantly, the GPS is used for its stated purpose of determining the geographical location of the ground control station at all times. This information is necessary for constructing the visual reference frame of topographical imagery. Before any updated information can be added to create a high-fidelity map of the operating environment, a low-resolution image-based map must be created. Knowing the location of the GCS and the resolutions of the screen and images, the field of view can be filled with the pre-existing information. As updated information is transmitted to the ground control station, this can be referenced to this framework and the model can be enhanced. A GPS receiver also serves to provide a point of reference for the vehicle during flight. It has already been discussed that a full AHRS would possess the capability to determine the flight unit's geographical position, but the combination of this information with the location of the GCS would allow for new autonomous functionality such as the ability to "return home".[18] Although there are additional sensors that a portable MAV GCS may use to collect more detailed environmental data, a GPS is necessary to enable a hands-off approach to persistent surveillance.

Granting a user the ability to command a vehicle and assign it target locations and tasks will demand that a GCS has some form of control input infrastructure. Historically, ground control stations have been designed using traditional methods such as joysticks, keypads, push buttons, and switches. An example of a MAV GCS that uses a control input model of this type is the Neural Robotics Incorporated Ground Control Station (NRI GCS) shown in Figure 13 below.

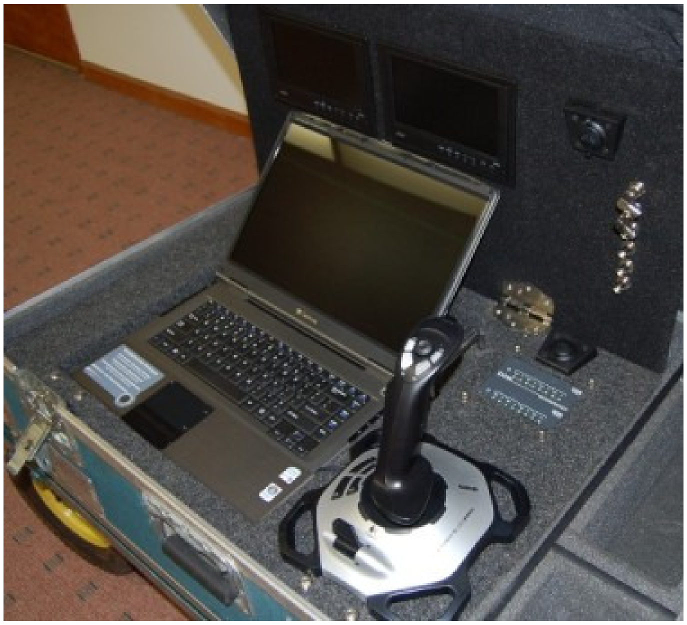


Figure 13: Neural Robotics Incorporated Ground Control Station[30]

Although ground control stations designed in this manner offer easy flight control and are fairly intuitive for today's video game generation, they pose several risks when used in the field. The biggest of these risks involves the complications introduced by moving parts. From our personal experience, it was easy to accidentally press a button or flip a switch while flying remotely operated vehicles. It is expected that this concern would be even greater in a highly tense combat environment and some of the newer ground control stations being developed have accordingly taken a more compact approach. The Aerovironment GCS, which can be seen in the figure on page 46 and will be discussed in greater detail later, is one ground control station that has limited the number of external control mechanisms. There is also the transportation problem associated with the cumbersome nature of this type of equipment. Even in systems like the NRI GCS, where organized packaging combines all equipment together, the possibility of an individual soldier carrying the GCS becomes impractical. This was noted in early versions of the RQ-11 Raven, where some soldiers said they would prefer additional supplies over the flight system because of its overbearing design. Again, this highlights the essential need for the development of a highly compact ground control station for use with MAVs. Referring back to the new developments in LCD panel technology, touch-screen GCSs are also a possibility. Accidental control inputs can be significantly reduced by eliminating external control infrastructure altogether. This is specifically the case of capacitive touch screens, where only a statically charged object like a finger can trigger an input.[31] Furthermore, touch screens provide for an easier user interface model than systems with buttons and switches. Whereas a button is restricted to a specific functionality, an LCD panel can be quickly updated with new options, windows, and configurations. Without having to become extremely comfortable with the GCS, a user can more easily make decisions when they are limited to touching adaptable icons as opposed to searching for the appropriate input device. From observations of the evolving state of MAV ground control stations, it is inferred that touch screens will be the control input mechanisms of the future.

The final major element of a ground control station is its graphical user interface (GUI). A graphical user interface is defined as a type of user interface (UI) that allows people to “communicate with a computer through the use of symbols, visual metaphors, and pointing devices.”[32] An example of a well developed user interface can be found on the Apple© iPhone. For the purpose of advancing persistent surveillance capabilities, a ground control station designer should make a concerted effort to develop an intuitive way of handling the interaction between the user and the vehicle. This is especially important if the user interacts with the GCS through a touch screen device. When designing this GUI, a developer must use human factors engineering to ensure the effectiveness of his product.

In his AERO 440 class on the development of heads-up-displays for aircraft, Dr. John Valasek of Texas A&M University, states that a human factor can be any part of “a body of knowledge about human abilities, limitations, and other human characteristics that are relevant to a design.” Human factors may include pattern recognition, chromostereopsis, and color memory and, when handled appropriately, they allow for reduced system training time, increases in productivity, and a reduction in operational errors. Maximizing the effectiveness of human factors engineering is achieved when the strengths of the machine and human are both utilized. For example, a processor on a GCS is able to handle many simultaneous complex operations but is less capable of exercising judgment as to the time when these operations should be executed. A human user, on the other hand, is capable of quickly observing detailed information, isolating the data of importance, and reacting accordingly. A well designed GUI would take advantage of both strengths and allow the processor to quickly collect, handle, and display information while presenting the user with well defined options for choosing a course of action. Upon receiving a user's response, the GCS could then rapidly put that decision into play. This process leaves the user in command of the highest level operational decisions yet allows the tedious low level work to be completed in the background.

Additional human factors come into play when developing the manner in which these options are presented. Pattern recognition is one major area where the expected behavior of a user should factor heavily into the design process. For example, the standard convention of the color green representing an affirmative and the color red representing a negative should not be ignored. The Federal Aviation Administration's (FAA) Human Factors Design Standard (HFDS) is a well versed template that aims to stress the importance of design considerations such as this. In section 8.6.2.1, the HFDS says that colors should be "consistent across a set of applications" and used to "augment a user's understanding of the information being presented."[33] Therefore, a poorly designed GUI would be one that either ignores color convention or relies on a user to react to information without a supportive color. Research has shown both of these approaches to result in a higher percentage of incorrect user reactions. Color blindness, chromatic aberrations of the eye, and verbal/visual conflicts are also factors which only further complicate GUI design. An abundance of information pertaining to these topics can be found online. A GCS developer is also tasked with the responsibility of delivering as much critical information as is necessary while preventing a user from experiencing an information overload. A purely text based approach to information display is generally overwhelming while a strictly graphical presentation may not provide enough data. Determining the appropriate balance between these two methodologies is essential for a system to be received well.

One of the defining elements of an MAV ground control station's GUI is the manner in which imagery data is displayed to the user. For manual flight control, it is expected that there will be a live video stream from which the operator can gather his decision making information. In the case of autonomous flight operations, the need for streaming imagery may not exist; however, it is still a requirement that the data be readily accessible. This holds true for all forms of data collected during flight. Although there are many additional GUI design criteria, this brief sampling is enough to convey the importance of developing a simple and informative user interface when directing MAV operations through a handheld ground control station. The above GCS elements, when combined together, form the backbone of an operable MAV handheld ground control station and though their discussion was superficial, it was intended to create a reference for the observation of currently operating GCSs.

The most advanced handheld MAV ground control station being used in active combat operations is the Aerovironment GCS (AVGCS). This system is modular and operates a variety of different vehicles, most notably, the WASP Block III. The AVGCS is a small unit that satisfies the requirement of being an individually portable system and contains a number of highly evolved capabilities. Figure 14 provides a close-up view of the AVGCS while Table 2 identifies the hardware that enables much of the GCS's functionality.

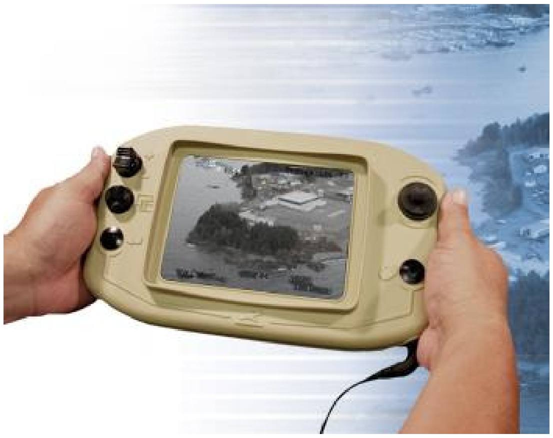


Figure 14: Aerovironment Ground Control Station Close-Up[5]

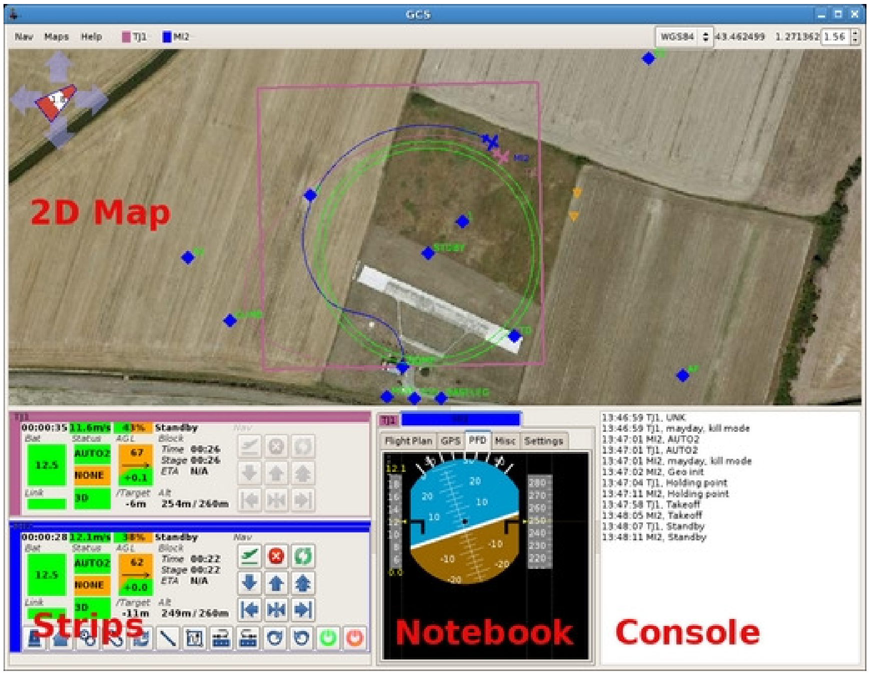
Table 2: Aerovironment Ground Control Station Features[5]

Feature:	Description:
Software	Proprietary, military-grade core operating system
Modes of Operation	Manual, Altitude-Hold, Loiter, Home, LoL, Follow
Batteries	BA-5590/U compatable
Interfaces	Ethernet and RCA (NTSC or PAL)
Datalinks	US DoD Joint Spectrum Center Stage 4
Weight	7.42 lb / 3.37 kg (fully packaged)
Product Status	In Full Rate Production

Like the larger systems that preceded it, the AVGCS is capable of handling the low level operations of a remotely controlled vehicle and giving a user full command of his craft. What makes this specific GCS unique among others, however, is the compact integration of advanced embedded functionality. The AVGCS primarily serves as a portal for live video stream transmissions from a UA. Figure 14 shows how information will be presented to a user during operation. The large screen is almost entirely devoted to imagery with the exception of minimal vehicle status data overlay. When an operator identifies an important piece of information, a simple screen capture feature allows the data to be saved for later analysis. This makes for an extremely user-friendly interface. In addition, the AVGCS can function as a Remote Video Terminal (RVT) and enable "command centers or monitoring stations to have the same viewing and analysis capability" as the UA operator.[5]

Using a custom operating system, the Aerovironment Ground Control Station can operate in several modes. Referencing a streaming video feed, a pilot can command the UA with the unobtrusive controls on the sides of the display panel. This mode requires the operator to constantly observe the feed and for situations where this is too demanding, pre-programmed missions can be activated. Some of these modes include the ability to hold an altitude, loiter around a target, or follow a user as he moves around on the ground. In the event of a signal interruption a vehicle can fly pre-defined patterns until the signal link is re-established. With industry standard interfaces and datalinks that operate at line-of-sight distances of several kilometers, the AVGCS is able to wrap an impressive amount of functionality into a small, lightweight, and ruggedized package.

Another GCS that deserves some consideration is Paparazzi's autopilot companion software, the Paparazzi Ground Control Station. This software package, which is expected to be run on a laptop or portable computer, offers far more advanced functionality than the AVGCS. Using popular waypoint navigation, the Paparazzi GCS, allows a user to monitor multiple UAs as they record and transmit data back to users on the ground. Flight plans can be created and real-time system state information can be visualized with a digital instrument panel complete with an artificial horizon, air speed indicator, and global positioning system. An example of the Paparazzi GCS in a flight configuration can be seen in Figure 15 below.



```
13:46:59 TJ1, UNK
13:46:59 TJ1, mayday, kill mode
13:47:01 MI2, AUTO2
13:47:01 TJ1, AUTO2
13:47:01 MI2, mayday, kill mode
13:47:02 MI2, Geo init
13:47:04 TJ1, Holding point
13:47:11 MI2, Holding point
13:47:58 TJ1, Takeoff
13:48:05 MI2, Takeoff
13:48:07 TJ1, Standby
13:48:11 MI2, Standby
```

Figure 15: Paparazzi Ground Control Station[34]

This autopilot software varies greatly from Aerovironment's software in that it is not a portal for live video streams or even static imagery. Rather, this system places a greater focus on building an environment where a user can control an MAV with top-level commands such as clicking on the map to drive the vehicle to a specific location. This system relies heavily on the controls capability found on the flight unit and simply updates the desired vehicle state vector rather than giving live manual control as seen in the AVGCS. In the upper section of the GCS in Figure 15 titled "2D Map", satellite imagery is used as a base-line for navigating the aircraft. This imagery is already tied to its respective global position and gives the user a decent situational awareness of the local region. The lower left section of the ground control software is used for vehicle status updates and point and click navigation routines. The digital cockpit is suppressed under a series of tabs in the notebook section and the console region provides a time history of each event that occurred during the mission. Beyond simply providing exceptional functionality, the Paparazzi team was able to build their GCS in accordance with the guidelines found in the FAA's HFDS. The system is intuitive and can be easily understood by a first-time user because of its effective integration of color and symbols. For these reasons, the Paparazzi GCS has become a popular choice for many research groups who are focusing on advancing flight hardware as opposed to developing new ground control station software.

2.2.2 Texas A&M Portable Ground Control Station

As research was underway at Texas A&M University to develop a new MAV system integration technique, I was heavily involved in a concurrent effort to design a

GCS that could explore the possibility of further enabling persistent surveillance capabilities for micro air vehicles. The results of these efforts produced a system that was easily adaptable to a variety of vehicles and allowed for quick operations with minimal training. While the system has not been fully completed, significant progress has been made in its development and this section will highlight the hardware and software utilized thus far.

The primary design driver when selecting the hardware for our portable ground control station was to maximize the hardware and software capabilities of the final product while keeping the overall system mass to a minimum. Simplicity of design was also a major contributing factor in how these selected components would be integrated together. This resulted in an early interest in using a touch-screen LCD panel for the primary display and control input infrastructure. By using a touch-screen, it was determined that the number of false input incidents could be kept to a minimum. Eliminating buttons and switches also makes room for a larger screen which expands the feasible output resolutions and viewable detail. To satisfy the computational demands of the image handling, control calculations, and associated graphical processes, a powerful micro-computer was needed. As previously mentioned, the ARM series offers the most power efficient processors capable of supporting these demands. While searching vendors for the display and processor, a development kit for a Windows CE based handheld controller was discovered. This unit, which is developed by Comfile Technology, combines a touch interface with an embedded graphic display to form a platform that developers can use to test the remote operation of machinery in an industrial setting. Although the controller was not intended for the use that we were planning, its integrated Flash, ARM9 processor, and Visual Basic support made the unit very adaptable to our needs. Figure 16 shows the Comfile CUWIN3100.

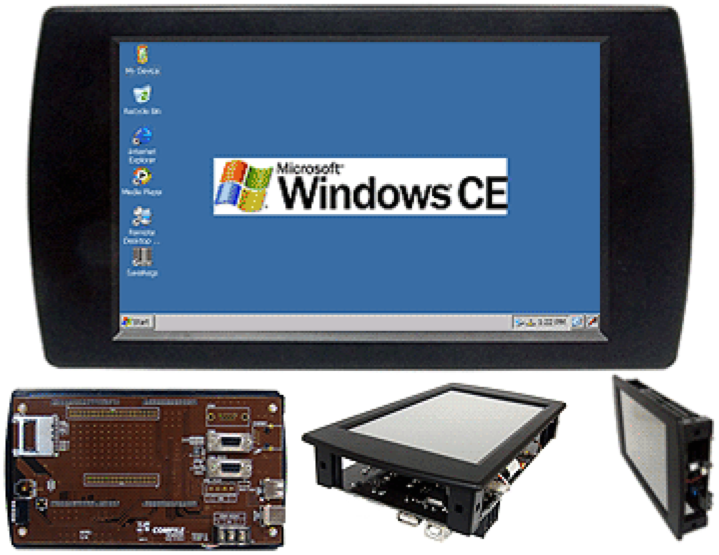


Figure 16: Comfile CUWIN3100 Windows CE Touch Controller[35]

Additionally, the CUWIN3100 offers SD card and USB support while also coming standard with two RS232 ports for the inclusion of serial devices. This would prove to be essential as we would need these connections for the GPS and communications systems. The Windows CE kernel was also attractive and the availability of the .NET Compact framework was enough to deter us from similar systems running Linux. Because a larger and more versatile UA ground control station was developed in conjunction with this project, it was decided that we would port the required functionality from the more powerful laptop to our handheld unit. The TAMU

UAV GCS was written in Visual Basic.NET and because the .NET Compact framework is a limited subset of the .Net framework, the porting of software was easy with the CUWIN3100. The CUWIN3100 is also easily adaptable to battery power and with dimensions of 216 x 120 x 30 mm, the goal of creating a small and portable system was attainable.

The RS232 serial connections on the CUWIN3100 offer a unique opportunity to easily integrate peripheral devices onto the ground control station with minimal hardware redesign. For communications and data transfer between the GCS and any micro air vehicle, a radio will be needed. Fortunately, the Aerocomm AC4490 was available in a development kit capable of serial connectivity. The AC4490 can operate at data rates up to 115.2 Kbps and, with ranges upwards of 6 kilometers, it offers a great solution to our data transmission needs. The TAMU PGS software package is being developed with the intention of using a split-process control methodology where the vehicle is responsible for self-stabilization and the ground control station, with its more powerful processor, will calculate the optimal trajectory between waypoints and upload this path to the aircraft. As such, the TAMU PGCS is being programmed to communicate these flight plans to the vehicle through the Aerocomm AC4490. Image transmission will also be handled by this radio. The RS232 serial connections also made it easy to integrate a global positioning system onto the CUWIN3100. This sensor was needed for the collection of position information to enable pre-programmed functionality on the vehicle such as hands-off return-to-base. Finally, the SD and USB interfaces allow for mission information to be saved and reviewed at a later time. Once the images are received by the PGCS, they can be stored on some peripheral media and transferred to another device for additional processing. Figure 17 shows the TAMU PGCS hardware with the communications system and a peripheral GPS attached in one of the developmental configurations.

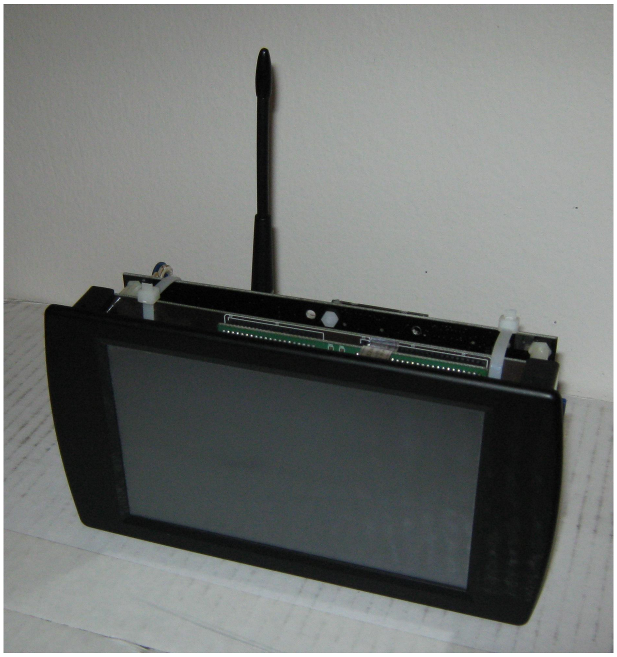


Figure 17: TAMU PGCS Developmental Hardware Configuration

Although the Texas A&M PGCS has not been completed and work remains to be done to raise its status to “field ready”, much of the software needed to test the applicability of IPNAR has been implemented. The infrastructure for building the pre-defined terrain map based off the global position has been implemented and the pixel-position correlations have been established. The map is presented from a birds-eye view and an imaginary camera which can be zoomed away and towards the imagery is given a reference altitude. This altitude is derived from the resolution of the images as presented on the screen. If a user zooms away from the map to increase his field of view, images taken by the MAV and projected onto this map are sized accordingly. Unfortunately, the .NET Compact framework does not support GDI+. This limited the number of graphical manipulations that could be programmed into the TAMU PGCS. There are several groups, such as XrossOne Mobile, who have built their own .NET Compact GDI+ support libraries; however, a complete library with the functionality we required was unavailable. While a custom GDI+ support library is still being developed, the current ground control station software limits the image projection to translation, skewing, rotation, and sizing. On the laptop-based UAV GCS, a shader was designed to allow the projected images to be "stamped" onto the terrain and persist over time. Once stamped these images were no longer individually accessible and considering the quick manipulation of information needed in MAV combat operations, a different approach was taken on the TAMU PGCS. Upon being received by the ground control station, an image will be confined to a window, manipulated based off of its recorded state, and overlaid appropriately on the map. This image box will be inaccessible unless requested by the user. Accidental image selection will therefore be prevented during normal data collection; however, the possibility of reviewing information is enabled.

The software on the TAMU PGCS is also being designed with the FAA's HFDS in mind. In compliance with all guidelines related to human factors and the effective sharing of responsibilities between man and machine, the TAMU PGCS has an intuitive and user-friendly GUI. In this GUI, users can select their vehicle through touching a representative icon on the map. This selection brings up several options that with a single screen tap can help an operator command the vehicle to a desired location and enter a specific flight configuration. Changing the target location and adding a secondary stop to a pre-existing flight plan are also done through simple icon selections. This approach was chosen to minimize the amount of attention that a user needed to devote to his vehicle operations. To maximize the data display region on the PGCS, the amount of system information presented to a user was limited to that which is critical to the mission. Parameters such as position, altitude, heading, and battery power remaining are discretely displayed where they can be quickly referenced but not distract from more important information like the most recent image taken. There are additional GUI features of the TAMU GCS that aid a user in controlling micro air vehicles with minimal confusion.

Due to the fact that the IPNAR has not reached a point where it can be fully field verified, it has been difficult to test the effectiveness of my graphical user interface in overcoming many of the human factor challenges. While there has been much preparation to handle these considerations, it is felt that the only effective way to truly verify this system will be to evaluate the performance of a group of users as they attempt to complete a simulated mission. Accordingly, a set of performance guidelines have been established to mimic what would be an acceptable level of operational affluence in the eyes of the Department of Defense. Primarily, it will be important to determined how long it takes before a user has a grasp on how to effectively and efficiently operate the vehicle. As mentioned a number of times previously, minimizing this learning time is one of the most critical desires of DARPA. Secondly, there will be a core set of operations that will be essential for all users to understand. The ability to utilize and repeat these operations, such as waypoint navigation, will be critical in determining how well a user understands the system. Finally, there will be some specific numerical criteria, such as the distance between and intended target and the actual vehicle location, that will help validate each user's ability. While the details of these human factors studies cannot at this time be finalized, it is understood that this will be a critical phase in bringing the PGCS to its final operating configuration.

Collectively, the aforementioned features of the CUWIN3100 provide a solid framework for the TAMU PGCS which is designed to handle the stresses of IPNAR. This paper will now shift into a discussion of the complications of integrating a multiple-input and multiple-output controller onto an MAV and what must be considered when trying to maximize the performance of this controller.

3. SEMI-AUTONOMOUS MAV NAVIGATION

One of the universally recognized challenges facing micro air vehicle development is how to effectively further the autonomous capabilities of the systems so that a user can be increasingly removed from the finer operational details. Howard Hamilton of the US Air Force Research Laboratory at Eglin Air Force Base explains that the mission challenges facing MAVs will need aircraft that can “satisfy a set of highly demanding requirements” among which include the ability to “[adapt] to changing and uncertain environments.”[36] In his publication about the navigation and control problems of MAVs, Hamilton reveals the ever-increasing demands being placed on micro air vehicle control techniques. The industry shift into persistent surveillance capabilities is only furthering these challenges. Recent hardware developments have begun to expand the possibilities for more advanced and refined control systems; however, this progress has exposed a new set of questions. Which type of controller will produce the necessary level of autonomy for persistent surveillance routines and what computational limitations accompany these controllers are two of the major questions developers must consider. This section will begin by looking at some of the historical and current obstacles facing the integration of autonomy enabling controllers on micro air vehicles and then identify what capabilities are required to successfully handle the demands of IPNAR.

3.1 Computational Concerns

Historically, autonomous capabilities for micro air vehicles have been achieved with the use of proportional-integral-derivative (PID) control. PID control offers a simple way of taking state measurements and translating them into a control response. The proportional control is used to determine how the control will react due to the most recent error in state-offset, the integral control is applied to tweak the control reaction to act as a function of the sum of all recent state-offset errors, and the derivative control adds a tertiary reaction derived from the rate of change of recent errors. Their ease of development and sole reliance on state measurements has made the PID control a widely applied tool. Hsiao et al.[37] used a collection of PID controllers in their autopilot system which was developed to demonstrate autonomous flight and image capturing capabilities. On their aircraft a proportional controller was used to establish longitudinal control while a PID maintained pitch attitude. A separate proportional controller worked with a PID to maintain lateral directional control and bank angle control respectively. This autopilot system is similar in composition to many of the controllers found in MAVs around the world. At the MAV07 Flight Competition, the PID controller was by-and-large the most common choice for teams attempting to demonstrate autonomous flight capabilities. While systems like this have been marginally successful in attaining some level of autonomy, there has been a vast realization that as micro air vehicles begin to expand their role in surveillance missions, more advanced control laws will be required. Confirming this conclusion, R. Krashanitsa et al.[10] found that "improvements in control effectiveness and the agility of MAVs will enable a further decrease in the size of MAVs and allow them to play a major role in the future."

3.1.1 Proportional Integral Derivative Control (PID)

Though this realization had been made, the question as to why researchers continue to use PID control as opposed to more robust and enhanced control laws remained unanswered. After looking into a number of vehicles where PIDs had been selected as the tool of choice, it became clear that several specific features were the cause for the popularity. Primarily, the PID controller is a very widely publicized control scheme. There is documentation of its application to nearly every type of control situation conceivable. For example, as an active controller, the PID can be used effectively to maintain a steady-state water temperature in a water heater or when using a horizon detector to keep an airplane in a wings-level flight configuration. Its applicability is unparalleled. Additionally, PIDs can be adapted for complex systems with many controllable states by integrating several controllers to work in unison. As seen in the Hsaio et al. system above, several layers of PIDs can be used to independently drive states to their desired values. This allows for slightly easier isolation of instability and ineffectiveness in the controllers written for a given system. The primary reason for the use of PID control, however, is the incredibly low computational demand as compared to some more robust controllers. Since micro air vehicle hardware has been very limited in its capability, specifically on the processing side, PID control has offered a solution where the vehicle will not be overwhelmed computationally.

3.1.2 Limitations and Considerations

Proportional-integral-derivative control is not without limitations though. One hurdle faced when using PID control is due to the fact that proportional-integral-differential controllers are themselves linear and face unpredictable results when used for highly non-linear systems such as the stabilization of a micro air vehicle. To account for this variable performance in non-linear systems, a PID controller can be enhanced with gain scheduling; however, the use of such methods can increase the amount of tuning that a system must undergo before becoming flight ready. Additionally, a PID controller can often be hindered by the sampling rate and measurement precision of sensors on the aircraft itself. Some sensors, such as a global positioning system may only report their current output once per second and a PID controller will suffer from a reduced reaction time when performing such functionality as waypoint navigation. This problem can, of course, be compensated for by using state estimation techniques to predict the system's current state and promote more responsive active control. One undesirable result of this fix is the loss of computational simplicity that PID advocates so greatly desire. The combination of several active and passive PID controllers on the same system is advantageous, but only furthers these computational problems. One complexity of PID control is that often the interdependency of states is difficult to account for. Finding the proper combination of weighting for each of the states can often lead to extensive tuning and many times an optimal solution cannot be reached. With the pre-existing challenges in accurately modeling the dynamics of MAVs, additional sub-optimality is highly undesirable and is one reason why more advanced controllers are being looked at for micro air vehicles. Although not the final limitation of PID controllers for MAV applications, set-point manipulation induced error is a significant problem that makes PID controllers very ineffective at persistent surveillance routines such as target tracking. Algorithms can be modified to include set-point weighting and derivative of output schemes to compensate for these errors; however, they too increase the computational demands and complexity of the proportional-integral-derivative controller.

While PID controllers have shown some limited success for MAV flight applications, a more advanced controller is needed to increase the level of autonomy that is needed for the IPNAR methodology to be successfully implemented. Among many other requirements, vehicles using IPNAR will need to be capable of waypoint navigation, self-stabilization, and pre-defined routines like target tracking. With the progress made in the development of micro-processors such as the ARM9 used on the IMAV, computational demands are still a concern but to a much lesser degree. It has already been explained that Texas A&M University research has produced an extended Kalman filter for estimation of more than 12 states which can operate at 80 Hz on an ARM9 processor. This same chip can be used to run a controller greatly superior to the lone PID while minimally impacting the processors ability to run the state estimator. Despite this fact, there is still a strong interest in finding the most capable controller that also has the minimal computational baggage so as to guarantee only a single chip is needed. With the critical micro air vehicle requirement of minimizing system mass, even a single gram for a second processor should be avoided. Also of importance is the goal of maximizing the flight endurance of newly developed MAVs. While the connection may be obvious, the optimality of a flight trajectory is not the only factor in increasing the duration of a vehicle's operation. Actuating the control surfaces in the most efficient manner is also a big contributor that, when done minimally, can translate into a greatly extended performance time. Controllers that exhibit a propensity for overshooting their control response should also be avoided and this is a common goal of developers. In this research, a number of different controllers that have the potential for expanding MAV autonomous functionality are reviewed.

3.2 Optimal Stabilization Controllers

One of the biggest shortcomings of using proportional integral derivative control is that it does not inherently converge on an optimal control response given a certain set of flight conditions. Tuning can help reach a solution that is close to optimal; however, the lack of available state interdependency makes it difficult to fully reach a completely ideal response. The linear quadratic regulator, or LQR, is a well documented controller that allows for optimization of control responses with full state interdependency. The LQR takes a linearized system model and aims to minimize a quadratic cost function in order to find the least costly manner of driving all states to a trimmed value of zero. Within the cost function, the states and controls are weighted so as to place significance on certain responses that should be avoided. The LQR would be useful in situations where an MAV is perturbed off of a trim state and requires a return to that condition. Aircraft stabilization with minimal control surface motion can certainly be advantageous when navigating between target locations and for collecting visual imagery with minimal motion induced blur. This is the first type of controller looked at for micro air vehicle applications. With IPNAR there will be a strong need for a controller that can drive a system to a set of desired steady state values. For example, if there is a static target on the ground that an operator decides is worthy of persistently monitoring, an MAV may need to enter into a circling configuration while keeping the camera pointed at said target. To do this, the vehicle will need to be commanded to a certain bank angle, airspeed, and yaw rate while maintaining a zero sideslip angle for coordination. Unless the system model is trimmed around this configuration, an LQR is incapable of driving the system to this orientation. With only a slight modification, the LQR can be transformed into a non-zero set point (NZSP) controller where this is possible. In essence the NZSP is one of the simplest tracking controllers available. In addition, waypoint navigation is easily achieved through implementation of the NZSP controller because it allows for convergence on any state that is controllable. With the NZSP, the optimality of the LQR is retained. The NZSP is the second controller class that this section will analyze in relation to micro air vehicle applications. For each of these controllers several specific flight configurations will be observed with the intention of showing how their implementation could further the autonomous capabilities needed to make persistent surveillance and hands-off MAV control a possibility with IPNAR.

The vehicle selected for use in the following controllers was the Aerosonde UAV (Figure 18). The Aerosonde UAV is a small unmanned vehicle platform that is available on the commercial market for a variety of applications including aerial imaging. This vehicle was chosen for a number of reasons but primarily because its flight dynamics reasonably compare to those of an MAV.

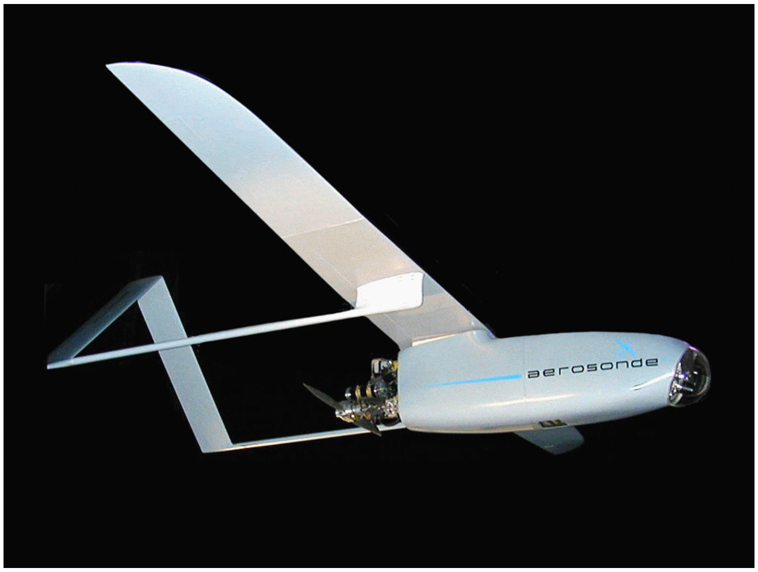


Figure 18: Aerosonde UAV (Image property of Aerosonde Ltd.)[38]

While the Aerosonde UAV will clearly be less agile than the IMAV or other similar vehicles, the platform is useful in providing a good testing bed to validate the utility of these controllers. Both longitudinal and lateral/directional models were studied and were made available by Aerosim Technologies Inc. The models were linearized around a trim flight condition of $\{V_a = 23 \text{ m/s}, h = 200 \text{ m}, \varphi = 0°\}$. These state space models are shown in Appendix A as Equations A-1 and A-2. The longitudinal states being handled are x-component velocity, z-component velocity, pitch rate, pitch angle, altitude, and angular velocity $\{u, w, q, \theta, h, \omega\}$. Rather than having the outputs the same as the states, this model converts the states into a more understandable form and finds the aircraft velocity, angle of attack, pitch rate, pitch angle, and altitude { V_a, α, q, θ, h}. These outputs make it easier to determine whether the system is performing as commanded longitudinally. Laterally, the monitored states in this model are y-component velocity, roll rate, yaw rate, bank angle, and heading angle {v, p, r, φ, ψ}. The output vector is the same as the state vector in the lateral model with the exception of the y-component velocity being replaced with sideslip angle. Many of the flight configurations that needed to be tested in the evaluation of these controllers involved coordinated turns and being able to visualize the sideslip angle was essential.

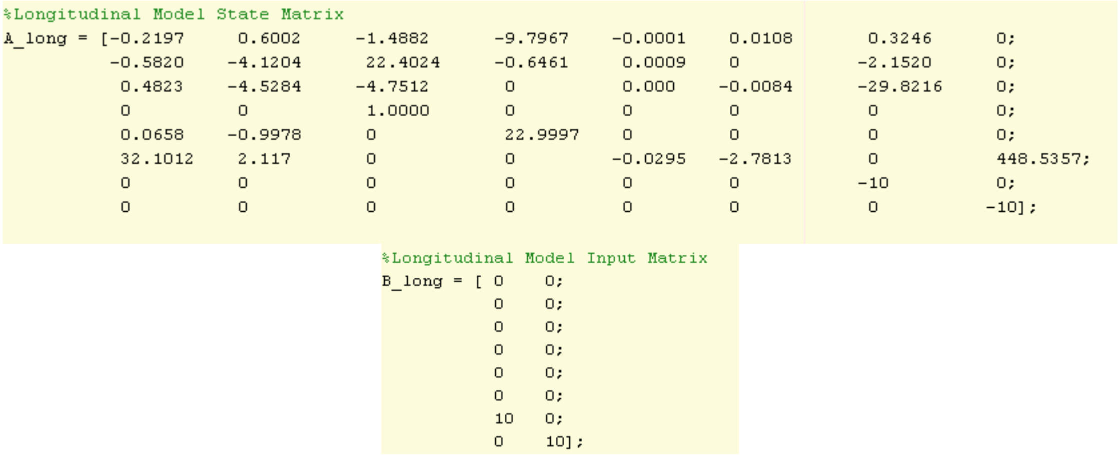
3.2.1 Linear Quadratic Regulator

The following will give an overview of the development of an LQR for micro air vehicle flight stabilization. The complete code for this discussion can be found in Appendix B and a full derivation of the linear quadratic regulator and all of the necessary conditions associated with satisfying the optimization problem can be found in Dr. John Valasek's Digital Control Lectures. Initially, it is important to note that the code to simulate the flight performance of the Aerosonde UAV was based on a sampled data regulator (SDR). This is due to the fact that the solution to the SDR is equivalent to the digital linear quadratic regulator. The performance of the LQR was evaluated for the condition where an aircraft is being hand-launched at the start of its mission. Because MAVs are so drastically light-weight, this is regarded as a critical time in the mission profile where the perturbations experienced by the aircraft are at an extreme. In this case, the initial conditions were φ = 25° and θ = 15°. These perturbations were chosen to ensure that both the longitudinal and lateral model responses were tested. The models were developed in MATLAB and were observed over a time span of 5 seconds.

After observing the longitudinal and lateral/directional dynamic models, it was decided that the dynamics of the actuators that will be used on an MAV should be taken into consideration. To make this modification to the state matrix, “A”, the control deflections were included as states and were replaced in the control vector with the new commanded states. Second order dynamics were used in this modeling and the equation representing these dynamics can be seen below in Equation 1.

$$G_{A(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{1}$$

The longitudinal state and control matrices in Figure 19 are modified to include the actuator dynamics. The lateral/direction model underwent the same alteration.



```
%Longitudinal Model State Matrix
A_long = [-0.2197     0.6002     -1.4882      -9.7967    -0.0001    0.0108      0.3246      0;
          -0.5820    -4.1204      22.4024     -0.6461     0.0009    0          -2.1520      0;
           0.4823    -4.5284     -4.7512       0          0.000    -0.0084     -29.8216     0;
           0          0           1.0000       0          0         0           0           0;
           0.0658    -0.9978      0            22.9997    0         0           0           0;
           32.1012    2.117       0            0         -0.0295   -2.7813      0           448.5357;
           0          0           0            0          0         0          -10          0;
           0          0           0            0          0         0           0          -10];
```

```
%Longitudinal Model Input Matrix
B_long = [ 0    0;
           0    0;
           0    0;
           0    0;
           0    0;
           0    0;
           10   0;
           0    10];
```

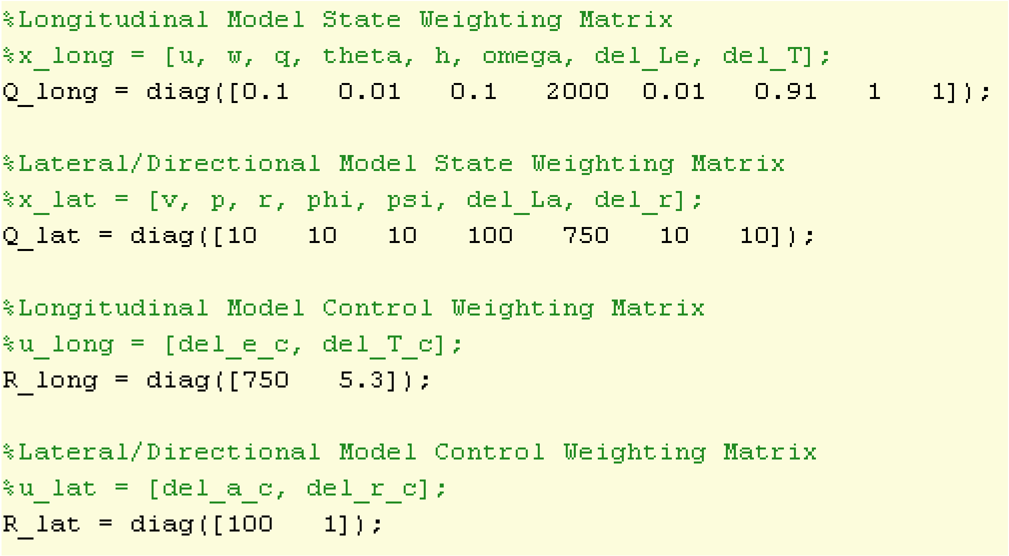
Figure 19: State and Control Matrices with Actuator Dynamics

By handling the actuator responses in this way, it was easy to visualize not only the actual motion of the control surfaces but also what was commanded by the controller in order to achieve stability. This visualization is also important because there are physical limitations on the maximum deflections reachable by the linear actuators and servos on the flight systems as well as limits on the possible rates of actuation. Since the commanded controls are now visible, it was possible to verify that these limitations were not exceeded.

From the development of the conditions of optimality, the discrete quadratic cost function to minimize for the linear quadratic regulator is as follows.

$$J = \tfrac{1}{2}\sum_{n=0}^{N}[\mathbf{x}_n{}^T Q\mathbf{x}_n + \mathbf{u}_n{}^T R\mathbf{u}_n] \tag{2}$$

The “Q” matrix is the weighting matrix that defines the relative importance of each state in determining the optimal convergence on a solution. In other words, if you wish to place precedence on driving one state to its final value, this state will be weighted heavily in the “Q” matrix so as to minimize the overall impact on the total cost value. The “R” matrix functions the same as the “Q” but weights the controls as opposed to the states. Although not required, the “Q” and “R” matrices were designed to be diagonal matrices. For the Aerosonde UAV models, the balanced weighting elements design technique was used to find appropriate starting values for the weighting matrices. This method, as described by Dr. John Valasek, uses the square of the maximum dimensionless values of each state and control to find a starting value for the diagonal weighting terms. The actuator limitations that were used to formulate these starting weights were a maximum aileron deflection of 35°, a rudder deflection of 45°, and an elevator deflection of 25°. These were chosen because they represent actual deflections that are attainable on RC aircraft. After observing the performance of the controller with these relative state and control weights, a process of tuning was used to converge on the desired performance. The final “Q” and “R” matrices for the flight stabilization LQR are given in Figure 20.



```
%Longitudinal Model State Weighting Matrix
%x_long = [u, w, q, theta, h, omega, del_Le, del_T];
Q_long = diag([0.1   0.01   0.1   2000  0.01   0.91   1   1]);

%Lateral/Directional Model State Weighting Matrix
%x_lat = [v, p, r, phi, psi, del_La, del_r];
Q_lat = diag([10   10   10   100   750   10   10]);

%Longitudinal Model Control Weighting Matrix
%u_long = [del_e_c, del_T_c];
R_long = diag([750   5.3]);

%Lateral/Directional Model Control Weighting Matrix
%u_lat = [del_a_c, del_r_c];
R_lat = diag([100   1]);
```

Figure 20: LQR State and Control Weighting Matrices

After specifying the starting state vector for each model and the desired output values, the optimal Kalman gains were found with LRQDJV.m, Dr. Valasek’s modified version of the embedded MATLAB linear quadratic regulator solver. LQRDJV.m is included with the rest of the code discussed in the section and can be found in Appendix B. Using the discrete control feedback law, LQRDJV.m minimizes the cost function, Equation 2, to find the gains associated with this optimal system response. This gain matrix, “K”, can then be used to rebuild a time history of the system and this was done next in LQR_AEROSONDE.m. The state transition matrix, STM, and discrete control distribution matrix, DCM, are the digital representations of the state and control matrix. MATLAB includes a function capable of making this conversion called "c2d". For re-building the state history of the system, these matrices are used. For identifying the control history, the Kalman gain matrix, "K" is needed. The process of how this was accomplished can be seen in the code. The frequency of the LQR controller was set at 20 Hz so as to mimic the rate at which data will be sampled when the controller is added to the microcontroller running the Texas A&M custom AHRS. After these state and commanded control time histories were generated, they were plotted for observation. The longitudinal and lateral/direction outputs for stabilizing the hand-launch simulation flight condition are given in Figures 21 and 22 respectively.

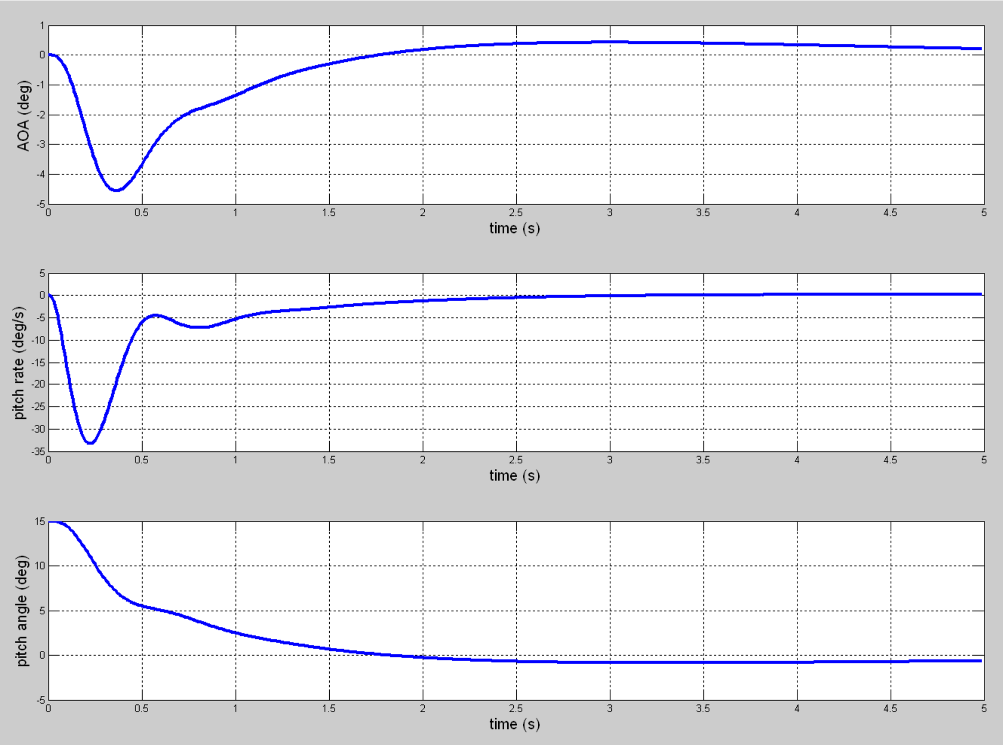


Figure 21: Longitudinal LQR State Response to Hand-Launch Disturbance

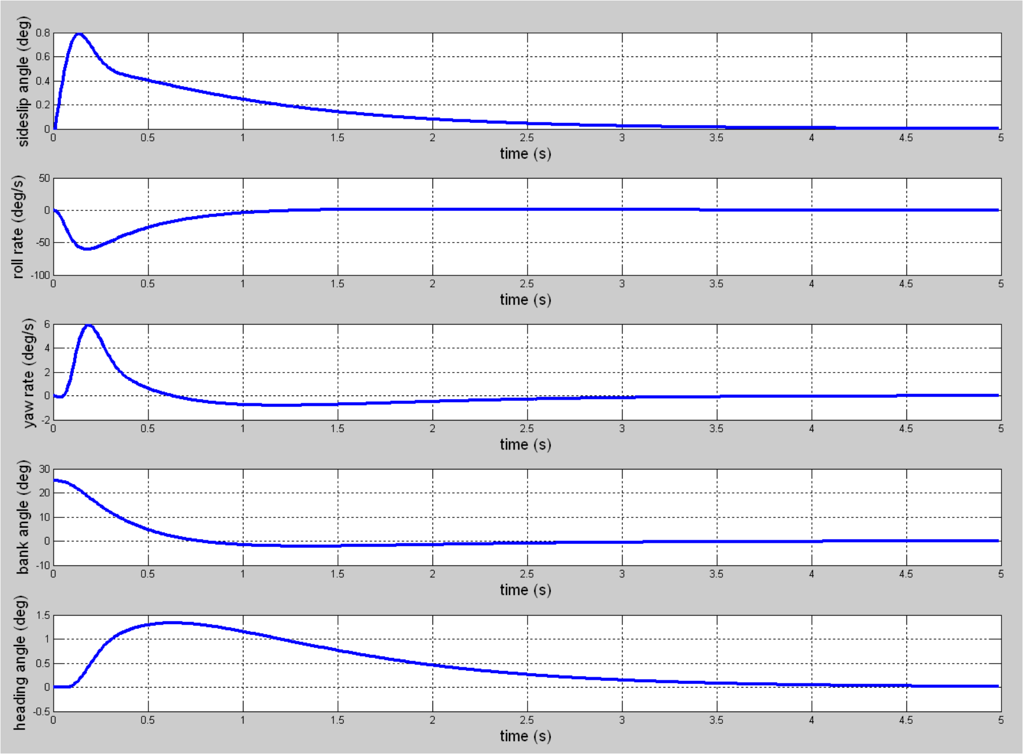


Figure 22: Lateral/Directional LQR State Response to Hand-Launch Disturbance

Figures 23 and 24 show the commanded controls needed to drive the system response as seen in the above figures. All control deflections conformed to the defined limitations and the units on the elevator, rudder, and aileron controls were in degrees while the throttle is given in percent increase over cruise.

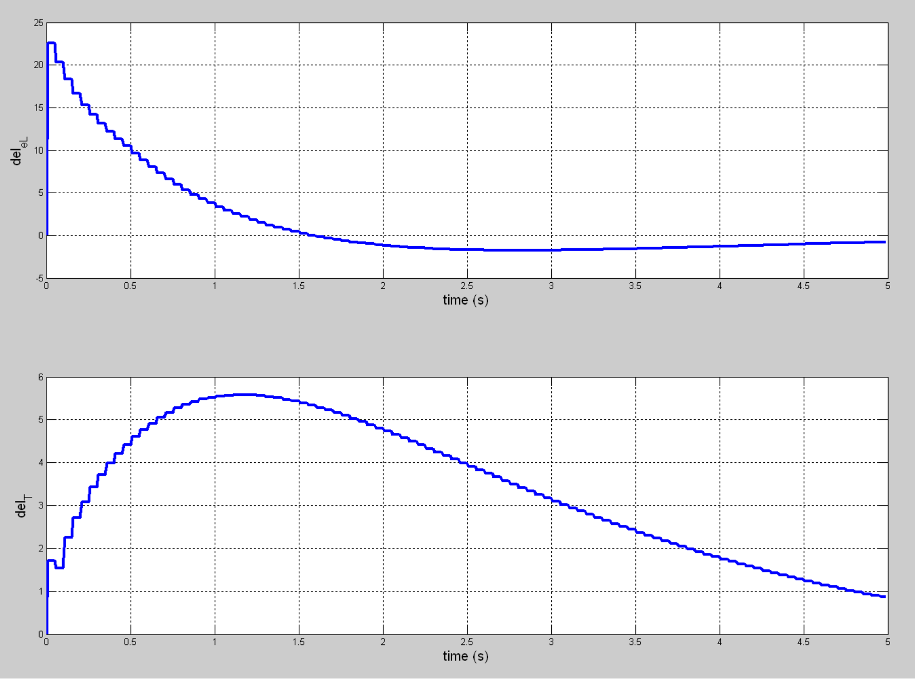


Figure 23: Longitudinal LQR Control Response to Hand-Launched Disturbance

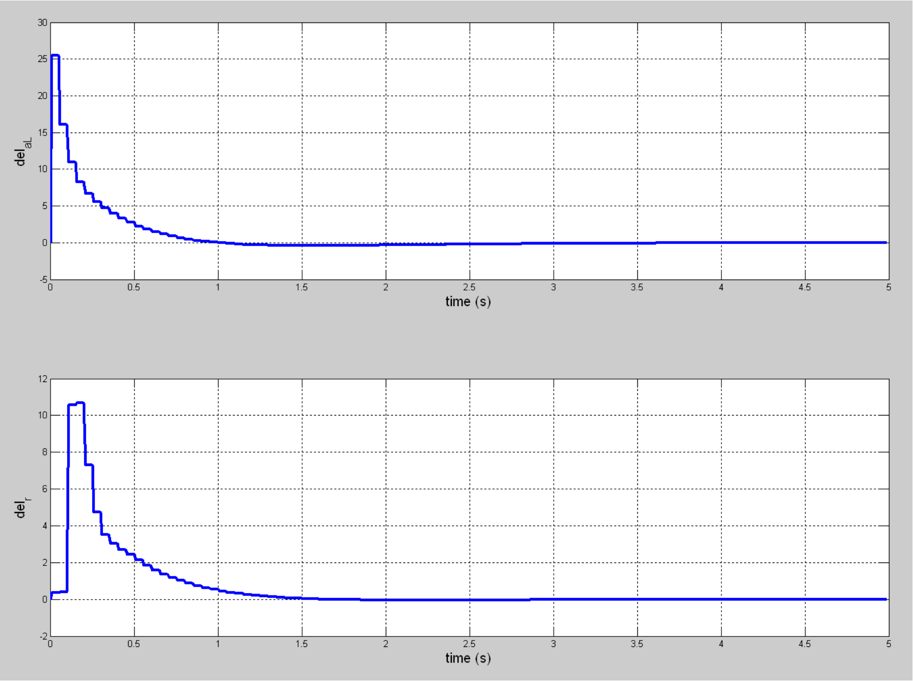


Figure 24: Lateral/Directional LQR Control Response to Hand-Launched Disturbance

The analysis of this data shows that the initial design specifications outlined for the hand-launched flight configuration are all capable of being met within the defined time parameter of 5 seconds. Clearly micro air vehicles such as the IMAV will require a controller that can drive the system to trim faster than this output shows; however, the tuning process revealed that the limitations on this convergence were due to the dynamic model of the Aerosonde UAV and not an insufficiency of the controller. When this model was tuned to simulate a more dynamic aircraft such as an MAV, it was observed that at 20 Hz, the control deflections could occur at a rate capable of compensating for typical disturbances generated by turbulent flow and gusts. In the longitudinal model it can be seen that the pitch rate and angle-of-attack do not follow the typical under-damped response. This is due to the heavy weighting placed on the pitch angle. There was a strong desire to create a smooth transition from the initial perturbation down to the trimmed value of zero and this was the result of this weighting. In dampening out the oscillatory pitching motion, the pitching rate began to slow and then accelerate a second time as it tried to compensate for the change in throttle resulting from the system attempting to maintain its airspeed and altitude. While these irregularities exist, they will not affect flight performance due to the haste of their occurrence. The lateral/directional state response performs as is to be expected and does so at a very acceptable rate. Convergence of the bank angle back to zero happens very smoothly and with minimal overshoot. This is helpful for a system that is attempting to maintain a heading. On a side note, the yawing motion in the lateral model appears to be lagged and the rudder deflection happens several samples into the time history. This lag is caused by the initial bank angle. While there is no original heading variation, the induced bank angle causes the system to drift as the system attempts to return to level flight. This heading change must be corrected for but is not recognized in the simulation until a few iterations have occurred. In application this would not occur because the sampling would happen on all states simultaneously and all responses would happen concurrently. One additional thing to note about the longitudinal response of the Aerosonde UAV was the presence of steady-state error in the pitch angle. This can be eliminated by applying some proportional integral (PI) control to that pitch angle state. This PI control functions in the same way as the PID controller discussed earlier; however, in this case it is limited to a single state since none of the other states display any steady state error tendencies. While the LQR provides a improved method of driving perturbations on a system back to their steady state value with efficiency, it is limited back its lack of ability to handle some of the configurations that may be required of a persistent surveillance MAV. One little known controller that can solve this problem is a derivative of the LQR and is known as the non-zero set point controller.

3.2.2 Non-Zero Set Point Control

The NZSP controller was developed and tested for two cases. The first case was the exact same as the case tested on the linear quadratic regulator. The purpose of this test was not only to confirm that the NZSP could be used for MAV flight stabilization but also to analyze its performance in such a case. For plots of the state and control responses from this test case, refer to Appendix A and Figures 37 – 40. The second case analyzed with the non-zero set point controller was intended to mimic a situation where an operator desired that the MAV change its heading to move toward a given target. Doing this requires being able to drive a system from a trim state to an off-trim state which is the primary benefit of the NZSP methodology. This is achieved by introducing a new definition for the state and control vectors. These new vectors, where the starred terms are the desired states and controls and the free vectors are the current states and controls, are shown below in Equation 3.

$$\begin{aligned}\tilde{\mathbf{x}} &= \mathbf{x} - \mathbf{x}^{*} \\ \tilde{\mathbf{u}} &= \mathbf{u} - \mathbf{u}^{*}\end{aligned} \tag{3}$$

These new vector definitions result in a new look at the state space model previously used in developing the LQR. (Equation 4)

$$\begin{aligned}\dot{\tilde{\mathbf{x}}} &= \dot{\mathbf{x}} - \dot{\mathbf{x}}^{*} = A\mathbf{x} + B\mathbf{u} - \left(A\mathbf{x}^{*} + B\mathbf{u}^{*}\right) \\ \dot{\tilde{\mathbf{x}}} &= A\tilde{\mathbf{x}} + B\tilde{\mathbf{u}}\end{aligned} \tag{4}$$

The quadratic cost function is also modified to handle these new state and control definitions and Equation 5 shows how the NZSP modifications have essentially just defined a new optimal control problem where the solution was already known.

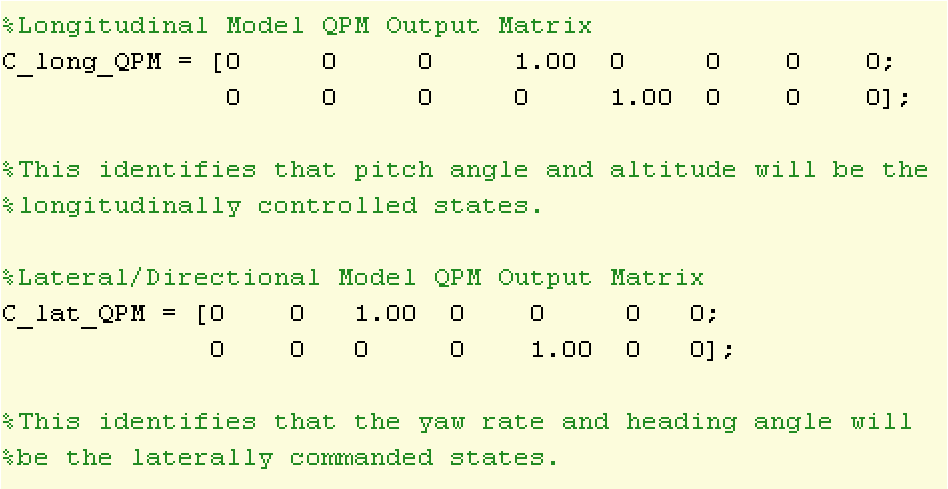
$$J = \frac{1}{2}\int_{0}^{\infty}\left(\tilde{\mathbf{x}}^{T}Q\tilde{\mathbf{x}} + \tilde{\mathbf{u}}^{T}R\tilde{\mathbf{u}}\right)dt \tag{5}$$

As opposed to the linear quadratic regulator, the NZSP formulation derives its control response from the error between the desired state and the current state. This allows the controller to be used in applications such as target circling or waypoint navigation. Once the existence of the new trim states and controls has been identified, an algebraic relationship (Equation 6) that correlates the states and controls to the state space model and desired output can be made with the use of the quad partition matrix (QPM). The quad partition matrix is the left-most matrix in Equation 6.

$$\begin{bmatrix} A & B \\ H & D \end{bmatrix} \begin{bmatrix} \mathbf{x}^* \\ \mathbf{u}^* \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix} \mathbf{y}_{\mathbf{m}} \tag{6}$$

A unique solution to this equation can only be found when the QPM is square and non-singular. This means that the number of states that can be driven to a new trim value using the NZSP technique is equal or less than the number of control inputs. In the case, where a tracking path was desired, the states that were controlled were pitch angle, altitude, yaw rate, and bank angle. With a final heading angle of 15°, yaw rate of 0°/s, altitude of 250 meters, and pitch angle of 0°, the aircraft flight configuration would mimic a typical MAV heading change.

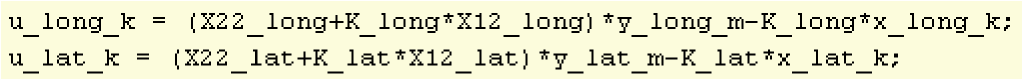
While the majority of the code written for NZSP was shared with the LQR methodology, several new modifications needed to be made. The first of these modifications was the redefining of the state-space model for use in building the QPM. Rather than simply acting to provide full state output, the "C" matrix is used to define which states would be driven by the controls. The new output matrices are shown in Figure 25.



```
%Longitudinal Model QPM Output Matrix
C_long_QPM = [0     0     0     1.00  0     0    0    0;
              0     0     0     0     1.00  0    0    0];

%This identifies that pitch angle and altitude will be the
%longitudinally controlled states.

%Lateral/Directional Model QPM Output Matrix
C_lat_QPM = [0    0   1.00  0    0     0   0;
             0    0   0     0    1.00  0   0];

%This identifies that the yaw rate and heading angle will
%be the laterally commanded states.
```

Figure 25: QPM Output Matrices for NZSP Controller

Additionally, the unique solution to the quad partition matrix relationship needed to be found in order to build a control history of the flight. This was done with the program QPMCALC.m written by Dr. John Valasek. Essentially, this code takes an inputted state-space model and returns the sub-matrices of the QPM. These sub-matrices are then used in the code seen in Figure 26 to build up a time history of the control. For a full look at the NZSP code and the associated programs, see Appendix B.



```
u_long_k = (X22_long+K_long*X12_long)*y_long_m-K_long*x_long_k;
u_lat_k = (X22_lat+K_lat*X12_lat)*y_lat_m-K_lat*x_lat_k;
```

Figure 26: Control History Using NZSP QPM

The state and control output from the heading change scenario shows that the NZSP offers a solution for MAV navigation and stabilization for imaging. Figures 27 and 28 show the longitudinal and lateral/directional state responses for the first test case.

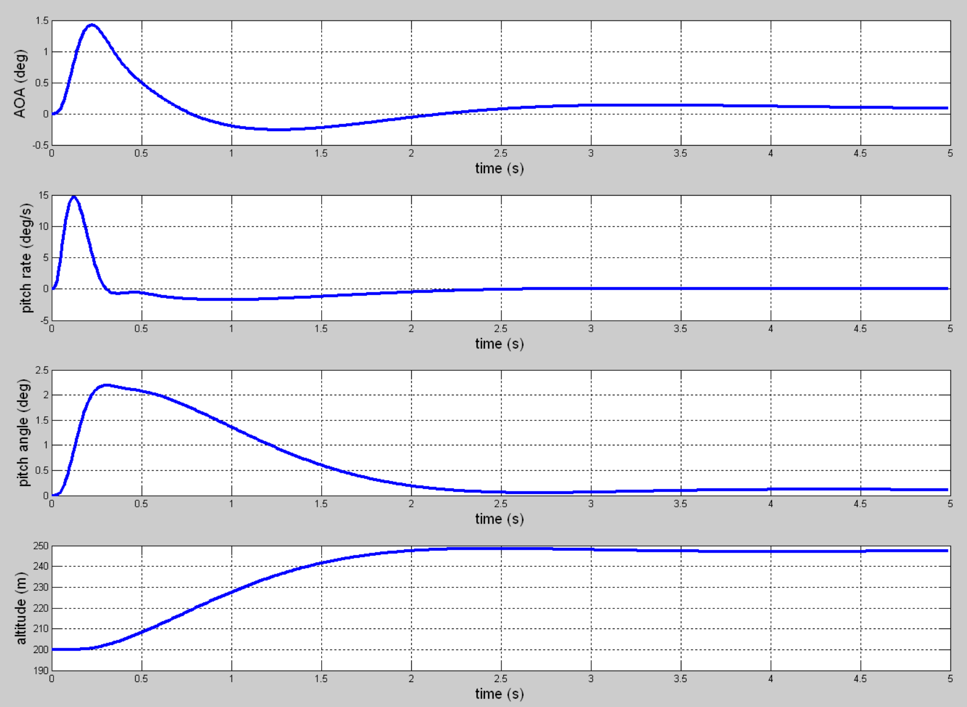


Figure 27: Longitudinal State Output for NZSP Heading Change

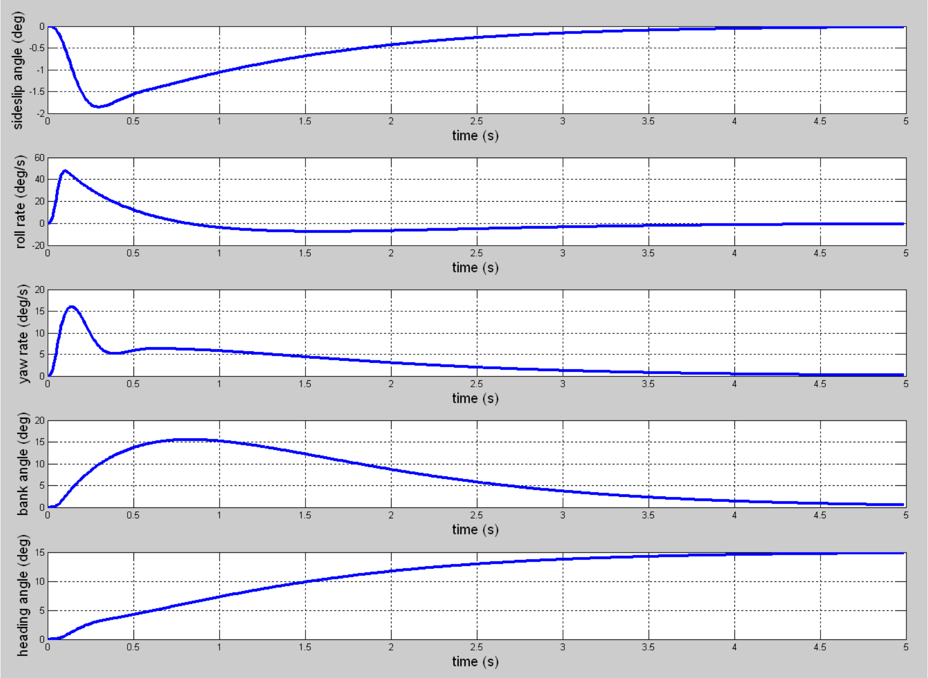


Figure 28: Lateral/Directional State Output for NZSP Heading Change

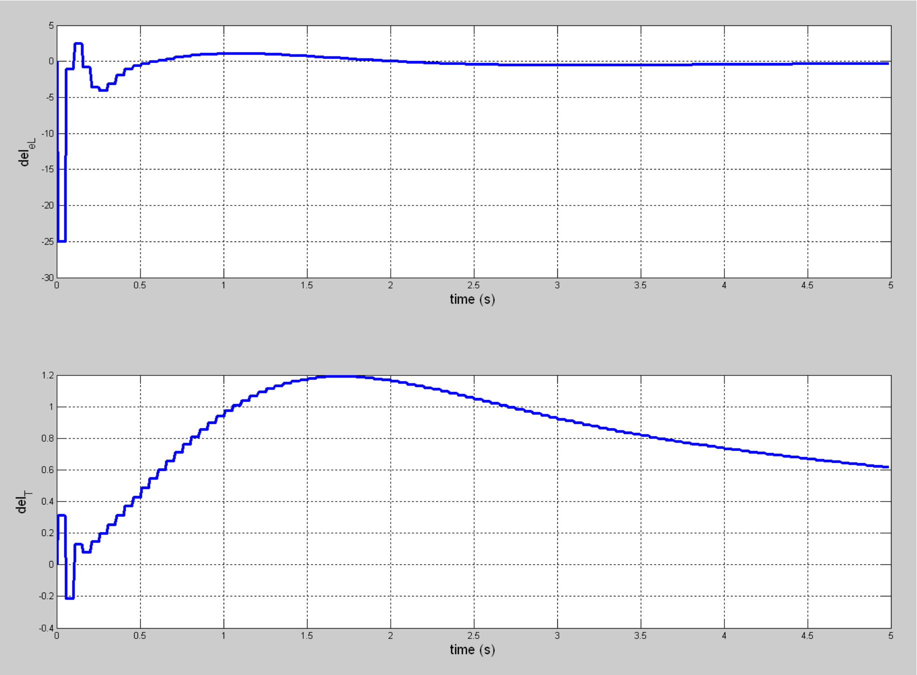


Figure 29: Longitudinal Control Response for NZSP Heading Change

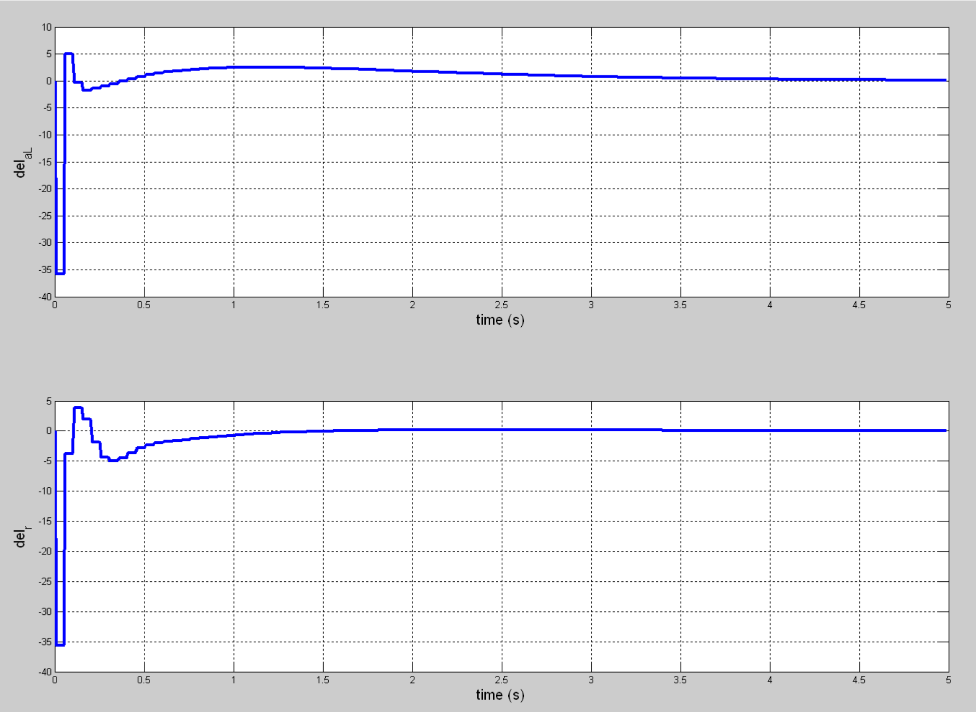


Figure 30: Lateral/Directional Control Response for NZSP Heading Change

The state and control responses, seen above in Figures 27 - 30, show that the NZSP is effective at driving the states to some non-zero trim value. Longitudinally it was found that stability could be reached quickly and in the case of the micro air vehicles for which this controller is being developed, this would be satisfactory. Like in the LQR, there is some steady-state error in one of the commanded states. Altitude has a tendency to converge on a value slightly lower than the intended 250 meters. A second variant of the NZSP was developed which included proportional integral control on the altitude state and this version was able to drive the error in the final altitude to zero. This version of the code is not discussed in this overview in order to compare the direct similarities of the NZSP with an unmodified LQR. Laterally, the state response in all states is very desirable. Each state makes a smooth transition to its desired value and there is minimal negative performance such as overshooting and extended settling times. Additional tuning can be done to ensure quicker convergence on the final heading; however, it was felt that for the case of navigation, haste was less of a concern than making sure that minimal control deflections were used. The control responses are self explanatory, yet it is worthy to note that this maneuver was able to be made with minimal change to the input throttle. This is important because on an MAV, the largest amount of energy consumption comes from propulsion. If changes to heading can occur without much throttling, precious energy can be saved which will in-turn increases the endurance of the vehicle. In addition to the heading change scenario, the NZSP controller was tested for alternate persistent surveillance flight routines such as circling a target while maintaining a camera pointing towards the ground as well as calculating an optimal trajectory through waypoints. Each of these cases performed with similar results and lead to the conclusion that the non-zero set point controller is a viable candidate for commanding an MAV.

3.3 Conclusions and Applications

The information presented in the previous section barely touches on the depth of work being done to identify new and more advanced controllers for micro air vehicle stabilization and navigation. The primary purpose of developing these controllers was because they offer a clear advantage over the currently frequented proportional integral derivative controller. One point not previously discussed was the inclusion of a runtime calculator in each of the controllers developed. This timer was placed in the code to determine which of the solutions offered the least computationally intensive solution while at the same time making sure not to sacrifice performance. There is a lot of concern that the NZSP technique would drastically increase the number of required clock cycles resulting from the matrix inversions in finding a unique solution from the quad partition matrix. This was found to be an over-estimated concern. A large sample of configurations and code runs revealed that the average run time of the NZSP controller was approximately 0.026 seconds. When compared to the LQR, which had a run time of about 0.024 seconds, there was only a net increase of 8% in run time. Taking into consideration the clock speed of the processor performing these operations, it was determined that the NZSP controller would require an average of 520 clock cycles to perform the necessary matrix calculations to find the control response for each data sampling. At 20 Hz this translates into 10,400 clock cycles per second which accounts for only 5.2% of the available horsepower of the ARM 9 processor used on the IMAV. The imbedded extended Kalman filter developed at Texas A&M for state measurement on MAVs runs at 80 Hz when there are no other process running concurrently, however, this becomes irrelevant because the control responses cannot perform at that frequency. The actuators can only operate at a fraction of this speed, and while the estimator should be run at least twice as fast as the controller, the NZSP does not appear to be a limiting factor. Therefore, it can be concluded that non-zero set point is a better controller selection than the standard linear quadratic regulator.

It is also important to note that during this analysis there was a constant understanding of the problems associated with controlling vehicles susceptible to gusts and other perturbations. While neither an LQR nor a NZSP controller will alone be able to handle the control of an MAV during flight, modifications to these controllers will enable the vehicle to be stabilized around a desired configuration. Specifically, adding a proportional integral filter (PIF) to the controllers will allow a designer to filter out any inputs into the control that fall within a set of bounds. Much like a low-pass or high-pass filter in circuitry, the PIF, will knock out undesired disturbances and keep the system controllable. Though this added control feature will slightly increase the computational demand of the controllers, it was determined that it was not significant enough to change the conclusions reached. Therefore, while important to note their consideration, the application of the PIF was irrelevant to the verification of the concept being tested in this research.

One further benefit of using non-zero set point control that has not previously been discussed is the impact on digital imaging techniques and how it is an enabling technology for IPNAR. With current limitations found in CMOS sensors, motion blur is a major hurdle that demands either some software correction or physical camera stabilization. To keep the computational load on the hardware to a minimum, the best solution for IPNAR to be successful is to identify periods of minimal system motion and collect images at those times. The Kalman estimator alone is a significant resource in finding these stabilization points, however, understanding the performance of the NZSP control responses can also aid in identifying when images should be taken. As a system is approaching a given target where it needs to take an image, the estimated period of stability can be refined with the knowledge of how the system may converge on a static flight configuration. These two tools can then be used to select the exact time to actuate the camera. This is one practice that will be explored in future iterations of my vehicle and control designs. In general; however, it can be stated that the study of how the optimality of NZSP control can positively impact the ability to perform persistent surveillance techniques on micro air vehicles was a success. While much work remains to implement these controls onto the hardware that will be used in eventual flight, the improved performance noted from the study in MATLAB© yields a better understanding of how more advanced controllers will not computationally overwhelm the processors onboard small aircraft like the IMAV.

4. FIELD OPERATIONS VERIFICATION

Before this methodology of IPNAR can be integrated into a fully operational MAV flight system, the field applicability of collecting and displaying imagery data requires verification. Rather than immediately test the concept in its complete capacity, IPNAR will be tested in stages. This will alleviate several of the complications encountered from trying to validate too many different functions at once and allows for the identification of any bottlenecks in the approach. The following sub-sections will chronicle the development of the hardware and software configurations needed to test the effectiveness of IPNAR on micro air vehicles. After these testing platforms are detailed, the process by which these systems will be operated in the field is described. Finally, the process of determining the effectiveness of the maps generated from the flight data is discussed and any potential problems encountered during collecting and interpreting this information are documented.

4.1 Hardware and Testing Configuration

The idea of using attitude determination hardware and a vehicle state estimator to associate a micro air vehicle’s orientation and position with imagery that is simultaneously collected brings with it many new concepts. Without too much re-iteration of previous statements, the desired objectives of this testing can be summarized as follows:

1) Determine the feasibility of using in-flight state information to associate an image with its corresponding location on a pre-existing terrain.

2) Identify the limitations of using software on a handheld ground control station to transform and display images using this state information.
3) Realize the effectiveness of image projection for navigation and target identification while noting how this methodology expands MAV persistent surveillance capabilities.

To accomplish these objectives, three hardware and software systems needed to be developed. The first element of hardware required was a flight unit to carry an imager above several selected locations. Although the IMAV provides a platform capable of satisfying this need, a larger and more stable aircraft was desired to help limit the control difficulties encountered by the operator. The next system required to satisfy these objectives was a sensor package to gather and record the state information of each image taken from the flight unit. In addition, this system was responsible for formatting this information in a way that the PGCS could easily interpret during post-processing. The final system that was developed was a program for the portable ground control station that would combine state information with the appropriate imagery and update a map with the newly acquired high-resolution images. This program was a limited subset of the full TAMU PGCS software package and the analysis of these maps was critical to the verification of the IPNAR methodology. The development of these three systems and their final testing configuration is the focus of this sub-section.

While the eventual goal of this research is to apply full state awareness to the images collected, it was identified that in order to fulfill the first testing objective a limited subset of state information should be used. After an extensive study to determine which states should be included in our analysis, the decision was made to restrict the image projection to a bird's eye view. The choice to eliminate roll and pitch from the initial study was driven by two factors. First, the lack of GDI+ support on the .NET Compact framework makes graphical processes, such as skewing, much more difficult and, without fronting the extra resources to develop this library, it was felt that the concept could be appropriately tested regardless of this elimination. Secondly, this modification would allow the testing to take place without the use of an embedded estimator since the sensors with inherent internal drift, i.e. gyros, are no longer required. This further minimized the possibilities of inducing error into the testing process. What this limitation left me requiring, however, was a platform that could gather images of the directly underlying terrain from a straight and level flight configuration and project them onto a map with only scale, rotation, and position as concerns. The state vector reduction can be seen in Figure 31 below.

$$\begin{bmatrix} x \\ y \\ h \\ \dot{x} \\ \dot{y} \\ \dot{h} \\ u \\ \theta \\ \phi \\ \psi \\ q \\ r \\ p \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ h \\ u \\ \psi \end{bmatrix}$$

Figure 31: Visualization of State Vector Reduction

The localization sensors needed to collect this state information were reduced from those found in a full AHRS, as described in Section 2, to simply a global positioning system and a two axis magnetic compass. Fortunately, for development purposes there are a number of these sensors available on break-out boards. Parallax Inc.^{TM} is one company who sells these components and images of them, Figures 32 and 33, show that in their standalone configuration all necessary support circuitry is included.

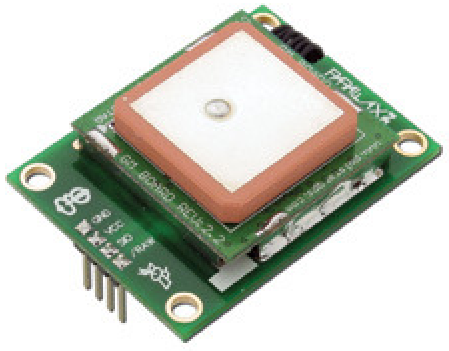


Figure 32: Parallax^{TM} GPS Receiver

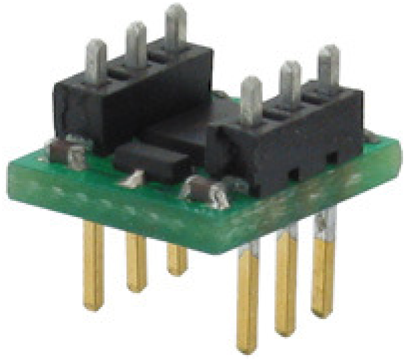


Figure 33: Parallax^{TM} Compass Module

The ease of integrating these sensors and the presence of pre-written component drivers in PBASIC, a custom version of BASIC written for use with microcontrollers, was the primary reason for the selection of the Parallax Inc.^{TM} platform. While the AHRS hardware for the IMAV was concurrently being developed during this verification phase, it was determined that a stand-alone unit should be created because of one other key feature of Parallax's Basic Stamp, serial data access. Rather than storing collected data in the microcontroller's random access memory (RAM), the Basic Stamp© microcontroller can directly output to a USB device using a serial data logger. This made collecting information for post-processing much easier. Additionally, it was desired that the operator be able to control when the vehicle would take an image. This would allow a user to orient the aircraft above a target location prior to recording information. The PBASIC language makes this possible by allowing for microcontroller interfacing through pulse width modulation. For these reasons, the data collection system was developed with the aforementioned hardware.

The imager used during the testing process was a Canon PowerShot A470. This camera captures 7.1 mega-pixel images using a CCD sensor. As previously mentioned, charged coupled device technology is superior in minimizing motion blur and the PowerShot A470 provides a light-weight and compact way of recording high resolution images while eliminating this concern. Additionally, it was decided that using the A470 was the best way to store the images for later analysis. Digital cameras, like the ones that can be purchased in electronics stores, are pre-loaded with drivers to handle image compression and disk writing. While this image compression could be offloaded to a microcontroller or FPGA, the integrated circuitry and software on the A470 was the easiest way to record images for later processing. Furthermore, the file creation system on the Canon PowerShot camera made image-state association easier due to the fact that images are saved with a sequential numbering system. This allowed for quick coupling of a specific data set to the image that was gathered at the same time.

In order to actuate the camera in flight, one of the I/O pins on the Basic Stamp© was used. To accomplish this, the A470 was taken apart and a lead was attached to the camera’s common ground plane. The opposite end of this lead was connected to the microcontroller reference ground. A second connection was made between the image capture button and an I/O pin. Whenever an image was requested by the flight operator, this I/O pin would have its voltage pulled low and the camera would be grounded for 100 ms which resulted in the camera taking an image.

Parallax Inc.^{TM} also offers a module for sensor testing with the Basic Stamp© microcontroller.[39] The module served as the interface between the GPS receiver, magnetic compass, serial data logger, Canon A470, and the BS2 microcontroller. My version of this board was designed to fly on the flight unit and record the heading, altitude, latitude, and longitude of the aircraft at the moment that each image was taken. This information was stored on a USB thumb drive in an ASCII text file. Communication with the module was achieved with an RC receiver through pulse width modulation. A channel was assigned to interpret a pulse being set high and then direct the BS2 microcontroller to initiate a set of pre-programmed commands. The state collection hardware in its full flight configuration can be seen in Figure 34.

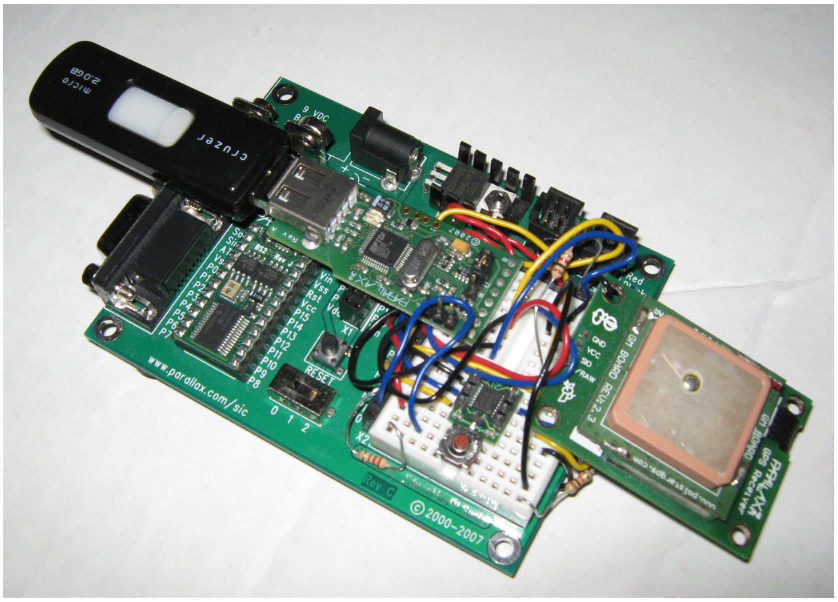


Figure 34: In-flight State Information Collection Module

To successfully collect the state information upon user request, a program needed to be developed in PBASIC and flashed to the EEPROM of the BS2 microcontroller. A full copy of this code can be seen in Appendix B and will now be briefly walked through for clarity. At the beginning, all variable declarations, constants, and pin assignments are made. The Basic Stamp 2 microcontroller only has 32 bytes of memory and this limitation nearly prevented its use for our desired application. Six of the thirty two bytes are reserved for internal variable declarations and each of the remaining 26 bytes was needed for the data collection and storing process. Following the definitions is the serial data logger initialization. Each time the program is run, a new file is created on the USB drive that will store all information from that given operating period. There was work done to determine if serial peripheral interface (SPI) could be used as opposed to standard serial, however, it was determined that this was not a possibility when using our specific GPS package. After the initialization period the main subroutine is entered. This is where the data collection procedure takes place and a flow chart of the activities can be seen below in Figure 35. The first action is to open a new data file on the USB flash drive. This file remains open for the duration of the data collection process. The file opening is followed by the first of two GPS information requests. Immediately following the primary collection and writing of the GPS data to the USB, the camera is actuated and a second sampling of the GPS data is taken. The reason for collecting the GPS information twice is to find a more precise set of states that describe the location that the image was taken at. A set of calibrations were done to determine the time between when the camera was instructed to capture an image and the time when the shutter actuated. This time, which was slightly less than one second, was then used to interpolate the exact state of the craft from the two state measurements.

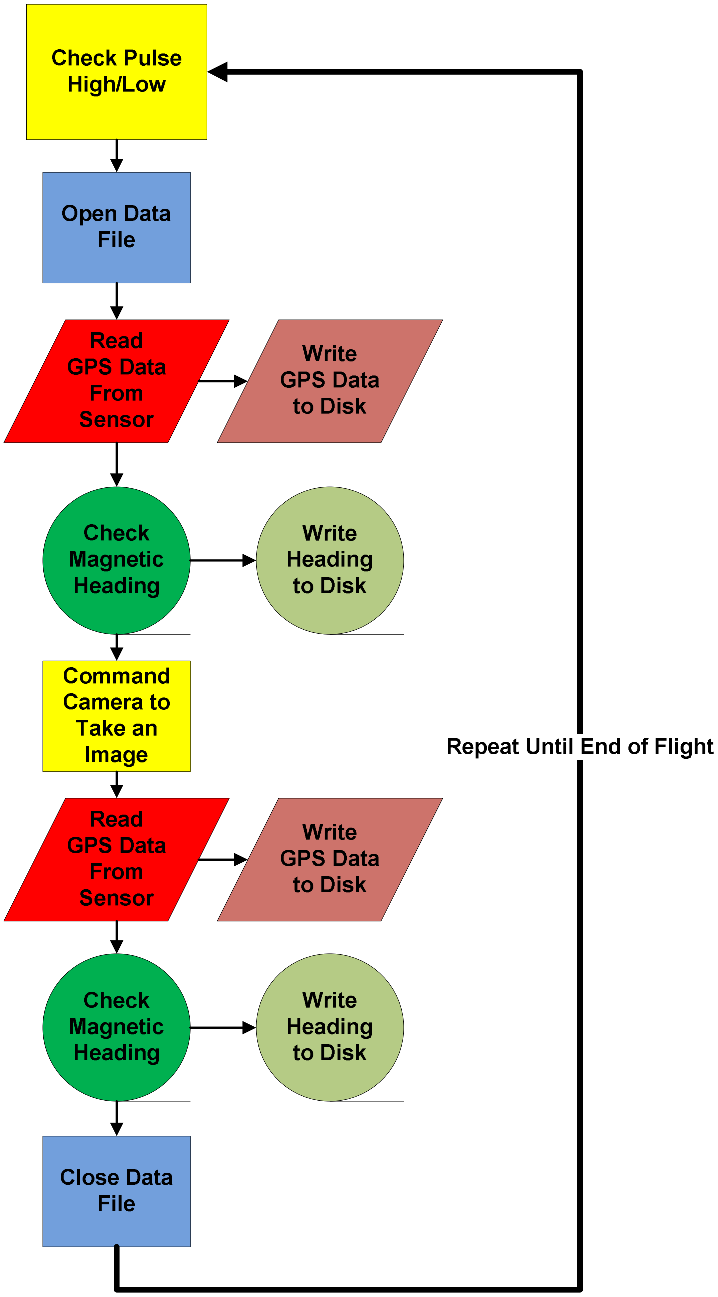


Figure 35: State Collection Procedure Visual Representation

A third item regarding the main subroutine is the placement of the magnetometer samplings. Since the interpolation was already implemented for the GPS information, it was an easy and obvious choice to make the same calculation on the heading data. Additionally, the data is calibrated to account for the magnetic declination at the location where the flight takes place. This is to account for the difference in the location of magnetic north and true north. Data processing will demand that this variation be considered. As a final note, the code was written in a way such that this process could be endlessly repeated over the course of the flight. Therefore, plenty of data could be collected in one run and limit the number of testing cycles. After the main subroutine, the sensor specific subroutines are defined and there-in lies the backbone of the code that enables the sensor operations.

Several complications were encountered during the development process; however, the most challenging to overcome was the serial communication rates through the USB. While the serial ports are more than capable of transmitting at the speed that was required for the small amount of data being collected in this scenario, the constant device switching and data array assigning made the system incapable of collecting two consecutive GPS transmissions during a single loop. This hurdle was overcome by defining each byte individually in the GPS string. Because the National Marine Electronics Association (NMEA) GGA format was used, it was possible to grab the necessary information through data parsing and discard the unnecessary characters. This enabled me to rapidly collect information for as many photos as I desired during a test flight.

The second hardware system that was developed for the testing procedure was a flight platform capable of carrying the aforementioned data acquisition hardware. Because the circuitry and imager were so much larger and heavier than the final systems would be, a larger and more powerful aircraft was needed. For the testing process, the Sig Kadet Mark II was selected. This aircraft is described as a trainer because of its exceptional flight stability. The wing dihedral causes the system to rest in level flight making it ideal for orthogonal ground imaging as was desired. The Kadet Mark II can be seen in Figure 36.

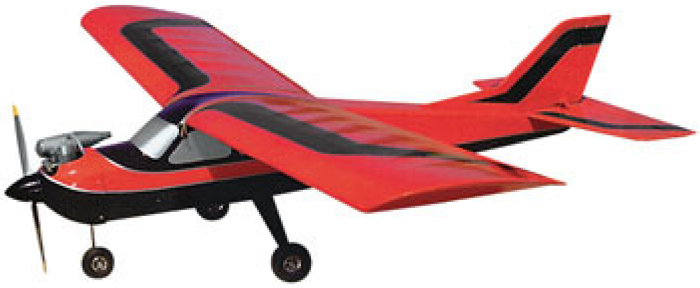


Figure 36: Sig Kadet Mark II Trainer Aircraft

Another appealing quality of this flight platform was the large and open fuselage.

Unfortunately, the state collection circuitry was oddly shaped and cumbersome. Several of the other aircraft considered did not have the internal capacity to house the board. The Kadet Mark II on the other hand was spacious enough to not only carry the hardware but also allow for easy repair access. Additionally, the Kadet Mark II sits high enough off the ground that there is ample room for the external mounting of the Cannon PowerShot A470. The camera was positioned to point directly down at the ground and was mounted in such a way as to provide access to the SD memory after each flight. The camera was longitudinally placed directly below the center of gravity in order to minimize the effect on the aircraft's stability. For flight control, a standard Futaba 7-channel RC transmitter was used with individual channels being assigned to the throttle, ailerons, elevator, rudder, and data acquisition board.

To validate the usefulness of the images collected during flight, a third system was needed during the testing phase. This software, which is a subset of the ground control station software package, serves to interpret the states associated with each image and use them to correlate that image to a pre-existing terrain. The pre-existing terrain was comprised of images provided by Google$^{\mathrm{TM}}$. This hybrid satellite and aircraft flyover photography serves as a guideline for estimating the position of the aircraft. When an image is handed to the GCS software, it interprets how it would appear if projected onto a two-dimensional surface from directly above. In this testing case, with the states available as previously mentioned, there will be limited image skewing required when attempting to orient the data. The software will determine how much image transforming is required and then using the interpolated state information to place the image over the pre-existing terrain imagery to update the fidelity of the model. This process will be repeated for each of the images captured during the test flights. After the map has been fully updated, a series of tests ranging from the calculation of pixel off-set to user-understandability will be performed to determine if the method is a valid approach to the presentation of accurate information.

4.2 Controlled Simulations and Future Testing

Before the data can be analyzed, a process for collecting the information is required. It was decided that the first round of testing would require close observation to make sure that the state collection hardware was operating properly. To perform these tests, the hardware needed to be connected to a ground station that would directly output the current readings. This was accomplished by attaching the data acquisition hardware to the hood of a moving vehicle as it drove around town. At regular intervals the hardware was sampled and the state information of the vehicle was collected and displayed on a connected laptop's debug terminal. This process helped ensure that the GPS and magnetic compass sensors were measuring in real-time and capable of accurately tracking the course of the vehicle. Once it was verified that the sensors were operating properly, the data needed to be correlated to the imagery taken at the same time. This correlation, as previously mentioned, necessitated that the state information be interpolated to match the moment of shutter actuation. To validate this correlation prior to flight, the unit was again operated as previously described and images were collected along the pre-defined route. Landmarks, in this case, houses, could be referenced with the position that was sampled concurrently. While these early tests were basic, they were necessary to perform prior to investing time in flight operations. Following this initial round of hardware verification, the next step involves taking to the skies.

The first thing to be considered during this second phase was specifically what type of visual imagery would be best to collect and what factors would play a role in properly collecting this information. In response to this concern, an effort was made to ensure that the location selected for image capturing encompassed a number of recognizable landmarks that could be easily distinguished from a birds-eye perspective. Buildings, parking lots, and intersections all make for good identifiers and easily discernable landmarks. Additionally, these objects simulate the urban environment that micro air vehicles are expected to fly in during missions of persistent surveillance. The altitude of operation was also a critical factor in this process. While increasing the altitude will help minimize the error in scaling, it decreases the fidelity of the model. In practice this is not necessarily a factor that will be left up to the operator's discretion and a variety of altitudes should be used during the testing runs. Finally, it was important to consider that a pilot would be operating the aircraft and collecting the imagery. Special precautions must be taken to make sure that the location for testing was not posing a danger to observers. Furthermore, human operation means that it will be difficult to hit a specific target location when collecting data. Therefore, the location selected must not be terribly dependent on a specific object making it into the field of view. After considering all of these factors, a location near the Texas A&M University campus was selected as the future site of IPNAR flight testing.

With regards to the testing process, an operator would simply repeat the steps taken during the first phase of testing but from an airborne perspective instead. During the course of the flight, a number of images should be captures at varying altitudes, positions, and headings. Additionally, it is important to maintain a level flight configuration during all the image captures. This is due to the limited states being considered. After these images are collected they are then interpreted by the PGCS software and analyzed for accuracy and understandability. It is felt that the testing of phase two would provide a solid compliment to many of the software simulations that have been performed during the development of Texas A&M University's more capable UAV GCS. While all hardware and software has been developed for testing during the second phase, the research from this point forward will be carried out at a later time.

The aforementioned testing stages are not the end of the testing required to fully verify that the idea behind IPNAR is sound. Once the preliminary test flights have taken place, the next step involves increasing the number of states that are included in the data sampling. Work on the full 6-DOF AHRS is currently underway at Texas A&M and this integrated circuit will be critical in evaluating how the additional states effect the ability to accurately project images onto the terrain. After the effectiveness of IPNAR has been analyzed, it becomes imperative that this methodology is tested on the IMAV or another similar micro air vehicle. With the highly turbulent motion of a small aircraft, the gathering of accurate state information may be hindered. Determining the level of hindrance, if any, is a key item needing evaluation. Once this step has been taken, making the process operate in real-time is the final hurdle. This will involved a full integration of semi-autonomous controllers of the variety previously mentioned as well as the establishment of a robust communications infrastructure for the transmission of data. Work is already underway to make these final steps toward bringing IPNAR into full operation.

5. SUMMARY AND CONCLUSIONS

At the onset of this research it was identified that in order to enable persist surveillance routines on micro air vehicles, a new approach would need to be taken to the data handling methodology used to bridge the gap between the user and the vehicle. While there is certainly much work yet to be done in areas such as advanced materials development, higher energy density batteries, and communications hardware, it was determined that currently available hardware could be used to satisfy many of the desired objectives for MAV operations. Breaking the paradigm of navigation though streaming analog video and replacing it with persisting high resolution imagery projected onto a pre-existing terrain was the solution that was converged upon. To adopt the Image Projection for Navigation and Reconnaissance (IPNAR) methodology, a wide variety of MAV subsystems would need to be analyzed and approached from new perspectives. Although many of these subsystems have and are currently under investigation, three specific areas, hardware integrations, semi-autonomous controls, and image projection effectiveness, were the focus of this research.

Early on, it was realized that the size reduction of micro air vehicles was a significant bottleneck in the quest for increasing operating performance to the level required for persistent surveillance. Not only do large systems increase the power requirements of a vehicle and therefore decrease possible operation time, but they also contradict one of the primary requirements of MAVs set forth by DARPA. With the eventual goal of being able to fly to a target and either loiter above or perch-and-stare, large vehicles lack the most basic element of stealth that researchers are striving for.

The first focus of this research was to identify what hardware components are essential to developing a capable micro air vehicle and then re-examining the method of integrating that hardware into a working configuration. The results of this study were visualized in the description of the IMAV design. Unlike current state of the art micro air vehicles, the IMAV design was centrally focused around the resourceful integration of every component. While most other MAVs make an effort to minimize the unnecessary support infrastructure, the IMAV takes this idea a step further and even looks at the potential of using irreplaceable components for dual purposes such as unique sensor placement to form the contours of the airfoil. In summary, the IMAV design highlights the importance of component integration for micro air vehicles that wish to operate for extended periods of time while maintaining remaining as unnoticeable as possible.

In addition to work done to advance MAV persistent surveillance through flight hardware, this research tackled the same problem from the ground as well. Throughout the past decade, a plethora of development groups have generated a wide variety of ground control station platforms for unmanned systems operations. This research showed that while there have been drastic advancements in the design and capabilities of these GCSs, there has been a fundamental lack of work done in making these systems ideal for the individual soldier's operation. Since the Department of Defense has envisioned each troop on the battlefield with their own personal flight system, there is an immense need for a portable ground control station that demands minimal operations training and user monitoring during missions. Current solutions simply do not offer this capability. Some ground control stations, such as the Aerovironment GCS, are moving in the right direction, however, they continue to rely on the old paradigm of analog video streams and do not enable enough direct removal of the user from basic flight operations. The solution in the TAMU PGCS was to keep with the majority of the same physical hardware as other systems but take an alternate approach to the data display and user interface. Through these changes, all of which conform to industry standards, it is believed that a more intuitive vehicle control infrastructure has been created that will enhance a user's ability to identify points of interest and instruct his vehicle to a target location. With the design centered on promoting self sustaining flight configurations, the TAMU PGS will also aid in helping bring additional persistent surveillance capabilities to smaller vehicles such as the IMAV.

By removing the user from the low level operations that are currently needed with MAV flight systems; there is an increased demand for the integration of semi-autonomous flight controllers. Research has shown that PID control offers a computationally simple way of inducing moderate stability on unmanned vehicles and is a popular choice with developers. Although some autonomous navigation can be achieved with PID control, it is inefficient and decreases a crafts operational time with non-optimal control dynamics. Prior to recent developments in advanced micro-processors, the low computational demands of the PID controller were a justifiable means for their implementation. With this new hardware now available to developers, the second major focus of this research was centered on determining how effective optimal controllers such as the LQR and NZSP are for navigation and stabilizing of micro air vehicles. Two cases were looked at for each of the two controllers and their appealing performance proved that optimal control theory increases the autonomous functionality of micro air vehicles without imposing any significant computational burden on the system. Some of the benefits of using the NZSP controller technique that directly relate to persistent surveillance capabilities are the minimization of control surface motion, decreased power requirements, and the ability to navigate through multiple waypoints in minimal time. NZSP control also allows the system to be commanded to an off-trim state. This is critical for applications such as target tracking and loitering because it opens up the possibility of maintaining a specific camera orientation for the duration of a flight. While there is more work currently underway, the results shown in this report convey that optimal control does in fact offer a computationally mild option for enabling persistent surveillance capable control on future micro air vehicles.

The third and final focus of this research involved the process of collecting state information and correlating that data with imagery in support of the IPNAR data handling methodology. The advancements made in the first two focus areas were both done with the intent of supporting the presentation of imagery as projections onto a pre-existing terrain. In order to fully enable those advancements, the process of collecting the data needed validation. A simplified version of the Texas A&M AHRS was developed and the states of interest were reduced down to latitude, longitude, altitude, and heading. With these four states as references, an infrastructure was built to fly hardware above landmarks and collect imagery that was then capable of being used to analyze the effectiveness of IPNAR for navigation and target identification. Preliminary stages of testing showed promising results and have laid the foundation for future increases in the scope of implementation.

The presentation of this material, although broad in scope, was intended to show how research currently being performed at Texas A&M University in the area of micro air vehicles has yielded a new and innovative approach to the visual information collection and presentation methodology which is a potential enabler of many previously unobtainable persistent surveillance capabilities. The specific focus on systems integration, optimal control, and data acquisition is a limited subset of the information revealed during this research but is undeniably critical to the full understanding and development of the IPNAR methodology. As research continues in these areas, it remains the eventual goal of Texas A&M University's Aerospace Engineering Department to develop a completely integrated personal MAV flight system capable of fulfilling the Department of Defense's vision for individually operated micro air vehicles.

REFERENCES

[1] Puels, R., "TCOM 598 Independent Study of Telecommunications Unmanned Aerial Vehicles (UAVs) Enabled by Technology," <http://telecom.gmu.edu/sites/default/files/publications/TCOM598-2006-Puels-UAVs.doc>, 2006.

[2] Cambone, S.A., Krieg, K.J., Pace, P., Wells II, L., "Unmanned Aircraft Systems (UAS) Roadmap, 2005-2030," Office of the Secretary of Defense, < http://www.uavforum.com/library/uav_roadmap_2005.pdf >, August 2005.

[3] Wilson III, S.B., "Micro Aerial Vehicle (MAV) ACTD," DARPATech 2002 Symposium, <http://www.darpa.mil/darpatech2002/presentations/tto_pdf/slides/WilsonSBIR.pdf>, 2002. pp. 9.

[4] General Atomics Aeronautical, Ground Control Stations, <http://www.ga-asi.com/products/gcs.php>, Accessed on April 13, 2009.

[5] Aerovironment Inc., UAS:GCS Data Sheet, <http://www.avinc.com/uas_product_details.asp?Prodid=106>, Accessed on February 15, 2009.

[6] Johansen, D. L., Hall, J. K., Beard, R. W., Taylor, C. N., "Stabilization of Video from Miniature Air Vehicles," AIAA Guidance, Navigation and Control Conference and Exhibit, August 20-23, 2007, AIAA Paper No. 2007-6862.

[7] Mueller, T. J., Kellogg, J. C., Ifju, P. G., Shkarayev, S. V., *Introduction to the Design of Fixed-Wing Micro Air Vehicles*, Reston, Virginia: American Institute of Aeronautics and Astronautics Inc., 2007, ISBN-10: 1-56347-849-8.

[8] Naimer, N., Koretz, B., Putt, R., "Zinc-Air Batteries for UAVs and MAVs," Electric Fuel Corporation, Auburn, Alabama and Bet Shemesh, Isreal, < http://66.102.1.104/scholar?q=cache:yxD0qr5BUbUJ:scholar.google.com/&hl=en >, December 2002.

[9] Michelson, R., Helmick, D., Reece, S., Amarena, C., "A Reciprocating Chemical Muscle (RCM) for Micro Air Vehicle "Entomopter" Flight," Georgia Tech Research Institute, < http://avdil.gtri.gatech.edu/RCM/RCM/Entomopter/AUVSI-97_EntomopterPaper.html>, 1997.

[10] Krashanisa, R., Platanitis, G., Silin, B., Shkarayev, S., “Aerodynamics and Controls Design for Autonomous Micro Air Vehicles,” AIAA Atmospheric Flight Mechanics Conference and Exhibit, Keystones, Colorado, August 21 -24 2006, AIAA Paper No. 2006-6639.

[11] Peterson, B., Erath, B., Henry, K., Lyon, M., Walker, B., Powell, N., Fowkes, K., Bowman, W. J., “Development of a Micro Air Vehicle for Maximum Endurance and Minimum Size,” 41st Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 6-9, 2003, AIAA Paper No. 2003-416.

[12] Keennon, M. T., Grasmeyer, J. M., “Development of the Black Widow and Microbat MAVs and a Vision of the Future of MAV Design,” AIAA/ICAS International Air and Space Symposium and Exposition: The Next 100 Years, Dayton, Ohio, July 14-17, 2003, AIAA Paper No. 2003-2901.

[13] Pralio, B., Guglieri, G., Quagliotti, F., “Design and Performance Analysis of a Micro Aerial Vehicle Concept,” 2nd AIAA “Unmanned Unlimited” Systems, Technologies, and Operations – Aerospace, San Diego, California, September 15-18, 2003, AIAA Paper No. 2003-6546.

[14] Armand, M., Tarascon, J. M., “Building Better Batteries,” Nature: International Weekly Journal of Science. <http://www.nature.com/nature/journal/v451/n7179/full/451652a.html?message=remove>, February 6, 2008.

[15] Aerovironment Inc., UAS:WASP III Data Sheet, <http://www.avinc.com/downloads/Wasp_III.pdf>, Accessed February 15, 2009.

[16] Gupta, S., Raj, C. V., “1st US-Asian Demonstration and Assessment of Micro Aerial Vehicle (MAV) and Unmanned Ground Vehicle (UGV) Technology Press Release,” <http://www.nal.res.in/MAV08/Indian_Press_Release-MAV08_Final%20US%20Edit%2020%20Feb%2008.pdf>, 2008.

[17] Quix, H., and Alles, W., “Design and Automation of Micro Air Vehicles,” 8th International Micro Air Vehicle Competition, University of Arizona, Tuscon, Arizona, July 2004.

[18] Shkarayev, S., “Development and Evaluation of MAV System for Surveillance and Reconnaissance,” Department of Aerospace and Mechanical Engineering, University of Arizona, Tucson, AZ, March 1, 2008.

[19] Mols, B., “Flapping micro plane watches where it goes,” Delft Outlook, < http://www.tudelft.nl/live/pagina.jsp?id=5ba8080d-6331-49cb-9d68-658e450299f9&lang=en&binary=/doc/DO05-4-1microplane.pdf>, April 2004. pp. 6.

[20] Abate, G., Personal Conversation Regarding MAV Technology Demands.

[21] Brisset, P., Hattenberger, G., “Multi-UAV Control with the Paparazzi System,” ENAC, Toulouse, France, 2008.

[22] The Paparazzi Project, <http://paparazzi.enac.fr/wiki/Main_Page>, Accessed on October 20, 2008.

[23] Grasmeyer, J. M., Keennon, M. T., “Development of the Black Widow Micro Air Vehicle,” 2001, AIAA Paper No. 2001-0127.

[24] AEROCOMM, AC4490 Transceiver Data Sheet, <http://www.aerocomm.com/rf_transceiver_modules/ac4490_900mhz_rf_transceiver.htm>, Accessed on June 1, 2008.

[25] SBG Systems, IG-500A Data Sheet, <http://www.sbg-systems.com/docs/IG-500A-Leaflet.pdf>, Accessed on September 30, 2007.

[26] Lugmayr, L., “CES 2009: LG Unveils Sunlight Illuminated LCD Display,” 14UNews, <http://www.i4u.com/article22370.html>, December 22, 2008.

[27] Anderson, P. S., “Development of a UAV Ground Control Station,” M.Sc. Thesis, <http://api.ning.com/files/ga5AVWy8xTu8cx9rHdzJ73Epc*dzb0uy*UPe0O8wFxogMF6WNRW8StK3x-xzkG2iBk16kNFpYTZe80NgM8kbXOiyxrEwUXIt/GroundControlStation.pdf>, 2002.

[28] Arning, R. K., Sassen, S., “Flight Control of Micro Aerial Vehicles,” AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, Rhode Island, August 16-19, 2004, AIAA Paper No. 2004-4911.

[29] Block, R., “iPhone Processor Found: 620 MHz ARM CPU,” Engadget, <http://www.engadget.com/2007/07/01/iphone-processor-found-620mhz-arm/>, July 1, 2007.

[30] Neural Robotics Inc., GCS Information Page, <http://www.neural-robotics.com/Products/GCS.html>, Accessed on April 10, 2009.

[31] Lee, S., Zhai, S., “The Performance of Touch Screen Soft Buttons,” 27th Conference on Human Factors in Computing Systems, Boston, Massachusetts, April 6th, 2009.

[32] “Graphical User Interface,” Encyclopedia Britannica Online, <http://www.britannica.com/EBchecked/topic/242033/graphical-user-interface>, Accessed on April 21, 2009.

[33] “Human Factors Design Standard,” U.S. Department of Transportation: Federal Aviation Administration, <http://hf.tc.faa.gov/hfds/download_received.htm>, Accessed on March 24, 2009.

[34] Paparazzi: The Free Autopilot, GCS Information Page, <http://paparazzi.enac.fr/wiki/GCS>, Accessed on October 20, 2009.

[35] Comfile Technology, CUWIN3100 Information Page, <http://cubloc.com/product/05_04cw3100.php>, Accessed on March 10, 2008.

[36] Hamilton, H. H., “Navigation and Control Problems for Classes of Micro Air Vehicles,” AIAA Atmospheric Flight Mechanics Conference and Exhibit, Keystones, Colorado, August 21-24, 2006, AIAA Paper No. 2006-6642.

[37] Hsiao, F., Chien, Y., Liu, T., Lee, M., Chang, W., Han, S., and Wang, Y., “A Novel Unmanned Aerial Vehicle System with Autonomous Flight and Auto-Lockup Capability,” AIAA Paper 2005-1050, Jan. 2005.

[38] Aerosonde Ltd., Aerosonde UAV Brochure, 2009, <http://www.aerosonde.com/html/Products/images/Aerosonde02-20-09.pdf>, Accessed on March 15, 2009.

[39] Parallax Inc.^{TM}, Board of Education Revision C (28150) Data Sheet, <http://www.parallax.com/Portals/0/Downloads/docs/prod/sic/boeman.pdf>, Accessed on May 9, 2009.

APPENDIX A

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{h} \\ \dot{\Omega} \end{bmatrix} = \begin{bmatrix} -0.2197 & 0.6002 & -1.4882 & -9.7967 & -0.0001 & 0.0108 \\ -0.5820 & -4.1204 & 22.4024 & -0.6461 & 0.0009 & 0 \\ 0.4823 & -4.5284 & -4.7512 & 0 & 0 & -0.0084 \\ 0 & 0 & 1.0000 & 0 & 0 & 0 \\ 0.0658 & -0.9978 & 0 & 22.9997 & 0 & 0 \\ 32.1012 & 2.1170 & 0 & 0 & -0.0295 & -2.7813 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \\ h \\ \Omega \end{bmatrix} + \begin{bmatrix} 0.3246 & 0 \\ -2.1520 & 0 \\ -29.8216 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 448.5357 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_T \end{bmatrix}$$

$$\begin{bmatrix} V_a \\ \alpha \\ q \\ \theta \\ h \\ \Omega \end{bmatrix} = \begin{bmatrix} 0.9978 & 0.0658 & 0 & 0 & 0 & 0 \\ -0.0029 & 0.0434 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \\ h \\ \Omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_T \end{bmatrix}$$

Equation A-1: Longitudinal State-Space Model for the Aerosonde UAV

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} -0.6373 & 1.5135 & -22.9498 & 9.7967 & 0 \\ -4.1919 & -20.6283 & 9.9282 & 0 & 0 \\ 0.6798 & -2.6757 & -1.0377 & 0 & 0 \\ 0 & 1.0000 & 0.0659 & 0 & 0 \\ 0 & 0 & 1.0022 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \\ \psi \end{bmatrix} + \begin{bmatrix} -1.2510 & 3.1931 \\ -109.8373 & 1.9763 \\ -4.3307 & -20.1754 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix}$$

$$\begin{bmatrix} \beta \\ p \\ r \\ \phi \\ \psi \end{bmatrix} = \begin{bmatrix} 0.0435 & 0 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \\ \psi \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix}$$

Equation A-2: Lateral/Directional Model for the Aerosonde UAV

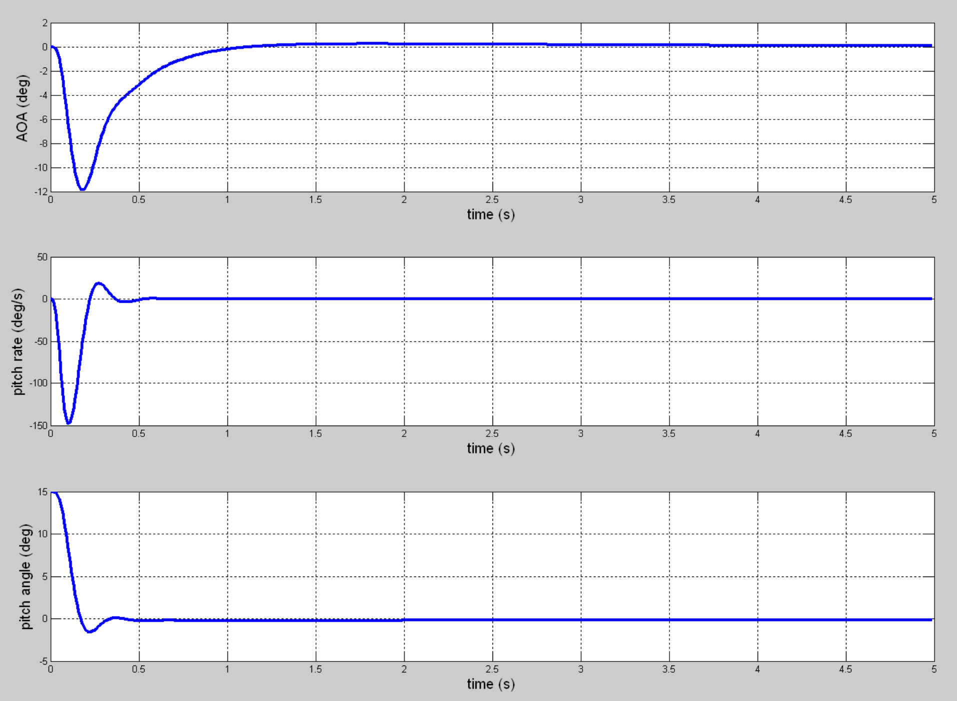


Figure 37: NZSP Longitudinal State Stabilization for Aerosonde UAV

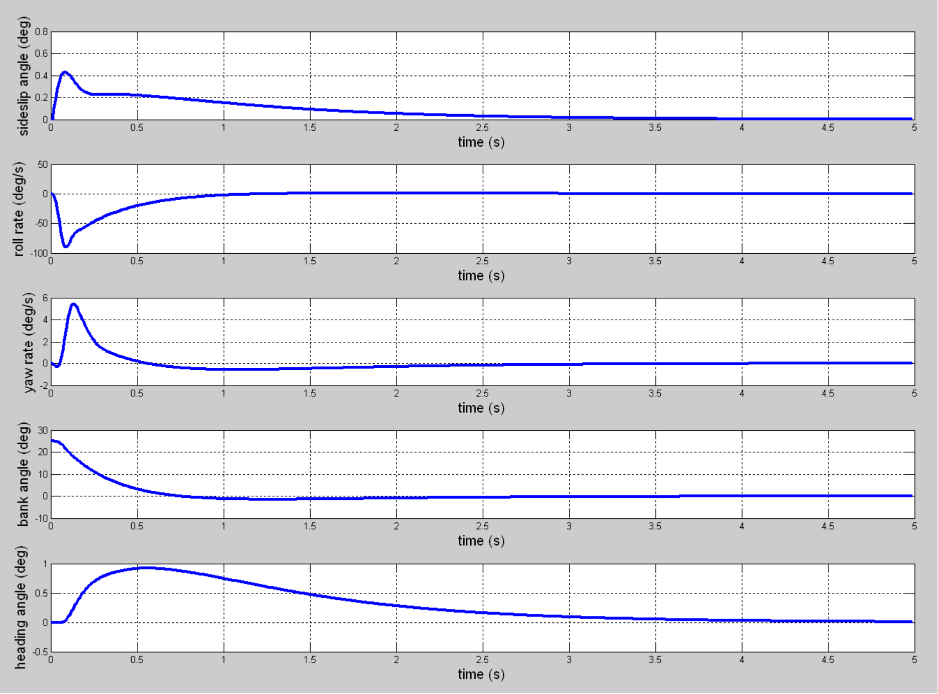


Figure 38: NZSP Lateral/Directional State Stabilization for Aerosonde UAV

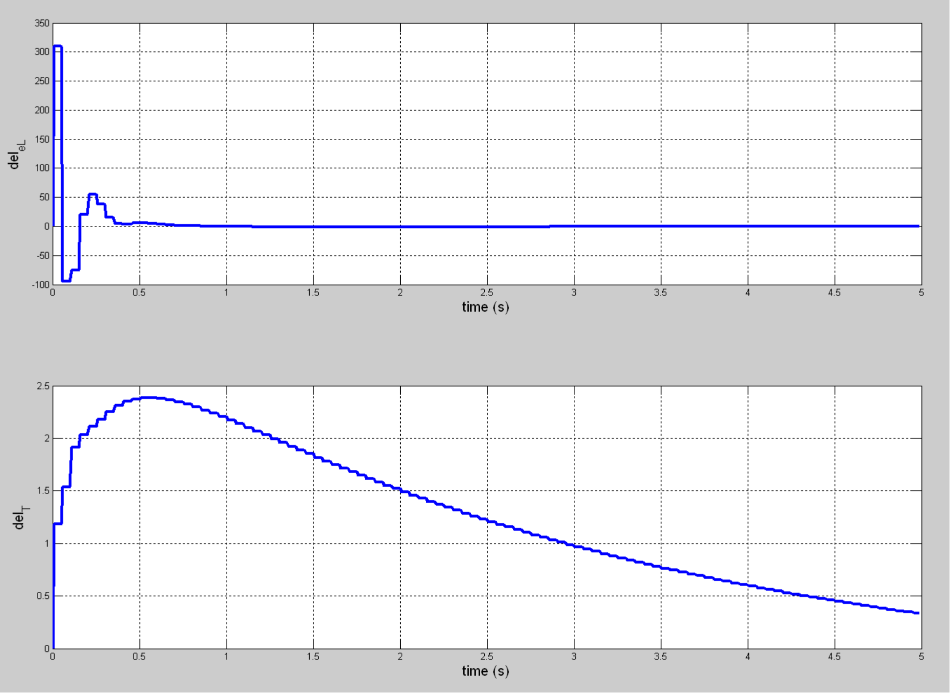


Figure 39: NZSP Longitudinal Control Stabilization for Aerosonde UAV

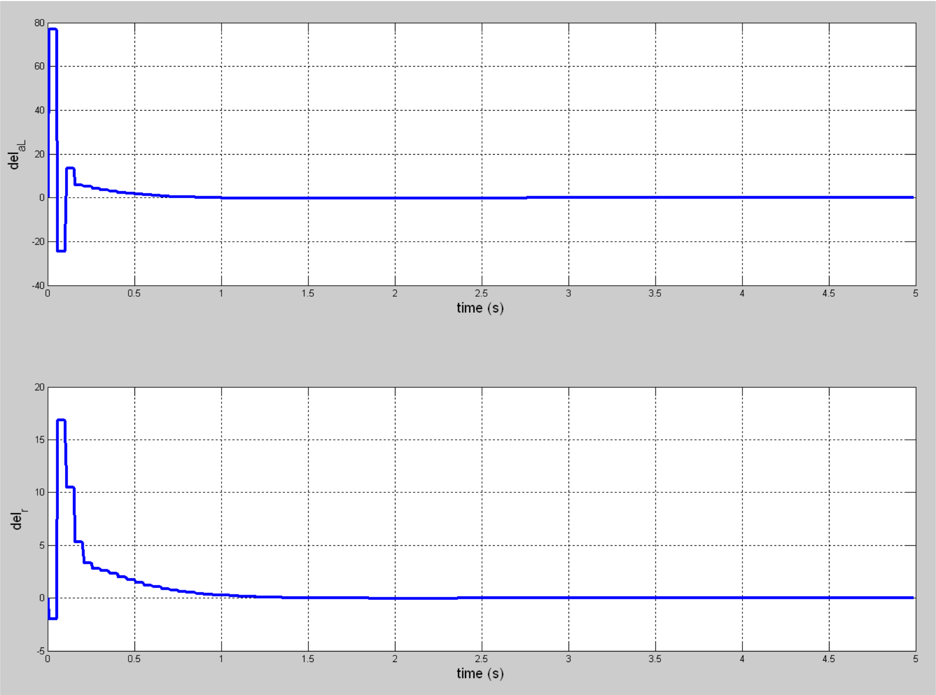


Figure 40: NZSP Lateral/Directional Control Stabilization for Aerosonde UAV

APPENDIX B

LQR_AEROSONDE.M

```
%Ryan Goodnight
%Micro Air Vehicle Controls Research
%Texas A&M University
%Advisor: Dr. Helen Reed
%12/06/08

function SDR = LQR(Ts)
%The following is a code that applies a linear quadratic regulator to
an
%Aerosonde UAV model in a disturbed state with the intent of bringing
the
%system back to a straight and level configuration.

%% Define States
%Longitudinal Model
% u = x-vel in body frame (m/s)
% w = z-vel in body frame (m/s)
% q = pitch rate (deg/s)
% theta = pitch angle (deg)
% h = altitude (m)
% omega = fuel mass (kg)
% del_Le = Left Elevator (deg)
% del_T = Throttle (deg)

%Lateral/Directional Model
% v = y-vel (m/s)
% p = roll rate (deg/s)
% r = yaw rate (deg/s)q
% phi = bank angle (deg)
% psi = yaw angle (deg)
% del_La = Left Aileron (deg)
% del_r = Rudder (deg)

%% Define Outputs
%Longitudinal Model
% Va = vel (m/s)
% aoa = angle-of-attack (deg)
% q = pitch rate (deg/s)
% theta = pitch angle (deg)
% h = altitude (m)
% del_Le = Left Elevator (deg)
% del_T = Throttle (deg)

%Lateral/Directional Model
% beta = sideslip angle (deg)
% p = roll rate (deg/s)
% r = yaw rate (deg/s)
% phi = bank angle (deg)
% psi = yaw angle (deg)
% del_La = Left Aileron (deg)
% del_r = Rudder (deg)

%% Define Controls
%Longitudinal Model
% del_Le_c = Left Elevator (deg)
% del_T_c = Throttle (deg)

%Lateral/Directional Model
% del_La_c = Left Aileron (deg)
% del_r_c = Rudder (deg)

%% Start Computational Stopwatch
tic

%% Initialize Constants
h = 0.01;
t_f = 5;

%% Define you state, input, output, and feedforward matrices:

%The following matrices were found from flight data at a flight
condition
%of 200 meters and a velocity of 23 m/sec.  The trim conditions are an
%elevator deflection of 0 degrees and an bank angle of 0 degrees.

%Longitudinal Model State Matrix
A_long = [-0.2197     0.6002     -1.4882      -9.7967    -0.0001
0.0108       0.3246      0;
          -0.5820    -4.1204      22.4024     -0.6461      0.0009    0
-2.1520      0;
           0.4823    -4.5284     -4.7512       0           0.000    -
0.0084      -29.8216     0;
           0          0           1.0000       0           0         0
0           0;
           0.0658    -0.9978      0            22.9997    0         0
0           0;
           32.1012    2.117       0            0          -0.0295   -
2.7813       0           448.5357;
           0          0           0            0           0         0
-10          0;
           0          0           0            0           0         0
0          -10];

%Lateral/Directional Model State Matrix
A_lat = [-0.6373     1.5135     -22.9498     9.7967     0      -1.2510
3.1931;
         -4.1919    -20.6283     9.9282      0          0      -
109.8373   1.9763;
          0.6798    -2.6757     -1.0377      0          0      -4.3307
-20.1754;
```

```
          0          1.00        0.0659      0          0       0
0;
          0          0           1.0022      0          0       0
0;
          0          0           0           0          0      -10
0;
          0          0           0           0          0       0
-10];

%The state matrices are separated into the longitudal and
%lateral/directional models.  In previous work it was found that while
%these two models are not coupled, the large matrices dimensions is a
%potential source of complications.  Actuator dynamics have been added
for
%the control surfaces so as to model the difference in commanded
control
%and actual control.  The actual control response is treated as a state
%now.

%Longitudinal Model Input Matrix
B_long = [ 0    0;
           0    0;
           0    0;
           0    0;
           0    0;
           0    0;
           10   0;
           0    10];

%Lateral/Directional Model Input Matrix
B_lat = [ 0    0;
          0    0;
          0    0;
          0    0;
          0    0;
          10   0;
          0    10];

%Like the state matrices, the input matrices are separated into the
%longitudinal and lateral/directional models so as to minimize the
%complications of dealing with large matrices.

%Longitudinal Model Output Matrix
C_long = [0.9978  0.0658  0      0      0      0      0      0;
         -0.0029  0.0434  0      0      0      0      0      0;
          0       0       1.00   0      0      0      0      0;
          0       0       0      1.00   0      0      0      0;
          0       0       0      0      1.00   0      0      0;
          0       0       0      0      0      1.00   0      0;
          0       0       0      0      0      0      1.00   0;
          0       0       0      0      0      0      0      1.00;
          0       0       0      0      0      0      0      0;
          0       0       0      0      0      0      0      0];
```

```
%Lateral/Directional Model Output Matrix
C_lat = [0.0435 0       0      0      0      0      0;
         0      1.00    0      0      0      0      0;
         0      0       1.00   0      0      0      0;
         0      0       0      1.00   0      0      0;
         0      0       0      0      1.00   0      0;
         0      0       0      0      0      1.00   0;
         0      0       0      0      0      0      1.00;
         0      0       0      0      0      0      0;
         0      0       0      0      0      0      0];

%Like the state and input matrices, the output matrices are separated
%into the longitudinal and lateral/directional models so as to minimize
%the complications of dealing with large matrices.  The output matrices
%have had the control terms concatinated in for analysis.

%Longitudinal Model Feedforward Matrix
D_long = [zeros(8,2); eye(2)];

%Lateral/Directional Model Feedforward Matrix
D_lat = [zeros(7,2); eye(2)];

%Both feedforward matrices have had an identity matrix matrix of a
%dimension matching the number of controls concatinated to the bottom
of
%the bottom of a zeroed matrix of a dimension matching the input
matrices
%so that the control reactions may be observed during analysis.

%% Define state, control, and final state weighting matrices for a
quadratic cost function:

%Longitudinal Model State Weighting Matrix
%x_long = [u, w, q, theta, h, omega, del_Le, del_T];
Q_long = diag([0.1   0.01   0.1   2000  0.01   0.91   1   1]);

%Lateral/Directional Model State Weighting Matrix
%x_lat = [v, p, r, phi, psi, del_La, del_r];
Q_lat = diag([10   10   10   100   750   10   10]);

%The first iteration towards solving for the state weighting matrices
was
%to use the balanced weighting elements technique.  After this first
%iteration, weighting tweaking was done through simulation and
observation.

%Longitudinal Model Control Weighting Matrix
%u_long = [del_e_c, del_T_c];
R_long = diag([750   5.3]);
```

```
%Lateral/Directional Model Control Weighting Matrix
%u_lat = [del_a_c, del_r_c];
R_lat = diag([100   1]);

%The first iteration towards solving for the control weighting matrices
was
%to use the balanced weighting elements technique.  After this first
%iteration, weighting tweaking was done through simulation and
observation.

%Longitudinal Final State Weighting Matrix
N_long = zeros(8,2);

%Lateral/Directional Final State Weighting Matrix
N_lat = zeros(7,2);

%In both models these matrices are set to zero for simplicity.

%% Define the initial perturbed flight configurations and the desired
final flight configuration to stabilize around:

%In this scenario we are considering the case where an MAV is being
%hand-launched and needs to return to a straight and level flight
%configuration.  The initial conditions are a pitch angle of 15 degrees
in
%the longitudinal model and a bank angle of 25 degrees in the
%lateral/directional model.

%Initial state vectors are in the following form:
%x_long' = [u, w, q, theta, h, omega, del_Le, del_T];
x_long_k = [0;0;0;15;0;0;0;0];

%x_lat' = [v, p, r, phi, psi, del_La, del_r];
x_lat_k = [0;0;0;25;0;0;0];

%Desire state vectors are of the following form:
%y_long_m' = [Va, alpha, q, theta, h, del_Le, del_T];
y_long_m = [0;0;0;0;0;0;0;0];

%y_lat_m' = [beta, p, r, phi, psi, del_La, del_r];
y_lat_m = [0;0;0;0;0;0;0];

%These intial and final state vectors are varied based off of the case
%being analyzed.

%% Control vectors are of the following form:
%u_long' = [left elevator commanded, throttle commanded];

%u_lat' = [left aileron commanded, rudder commanded];

%Output vectors are of the following form:
```

```
%y_long' = [Va, alpha, q, theta, h, del_Le, del_T];

%y_lat' = [beta, p, r, phi, psi, del_La, del_r];

%% Calculate the optimal control gains, K, for each model using
LQRDJV.m:

%Longitudinal Model Optimal Kalman Gains
[K_long,Q_long_h,R_long_h,M_long,P_long,E_long] =
LQRDJV(A_long,B_long,Q_long,R_long,N_long,Ts);

%Lateral/Directional Model Optimal Kalman Gains
[K_lat,Q_lat_h,R_lat_h,M_lat,P_lat,E_lat] =
LQRDJV(A_lat,B_lat,Q_lat,R_lat,N_lat,Ts);

%% Define the state transition matrix and the discrete control
distribution matrix for both models using the c2d function:

%Longitudinal Model State Transition and Discrete Control Matrices
[Phi_long,Gamma_long] = c2d(A_long,B_long,h);

%Lateral/Directional Model State Transition and Discrete Control
Matrices
[Phi_lat,Gamma_lat] = c2d(A_lat,B_lat,h);

%% Initialize a starting control vector for each model:

%Longitudinal Model Starting Control Vector
u_long_k = K_long*(y_long_m - x_long_k);

y_long(:,1) = C_long*x_long_k + D_long*u_long_k;

%Lateral/Directional Model Starting Control Vector
u_lat_k = K_lat*(y_lat_m - x_lat_k);

y_lat(:,1) = C_lat*x_lat_k + D_lat*u_lat_k;

%% Using the Kalman gains and the STM and DCM build a time history of
the control and command history of both models.

%Define the number of intervals
n_frames = t_f/h;

%Create a time array and fill the first entry with t = 0
time(1) = 0;

%Initialize a counter for rebuilding the command and control history.
counter = 0;

%Build control and command history
```

```
for i = 2:n_frames
    if abs(counter-Ts) < 1e-8
        u_long_k = K_long*(y_long_m - x_long_k);
        u_lat_k = K_lat*(y_lat_m - x_lat_k);
        counter = 0;
    else
        u_long_k = u_long_k;
        u_lat_k = u_lat_k;
    end
  counter = counter + h;
  x_long_plus1 = Phi_long*x_long_k + Gamma_long*u_long_k;
  x_lat_plus1 = Phi_lat*x_lat_k + Gamma_lat*u_lat_k;
  y_long(:,i) = C_long*x_long_k + D_long*u_long_k;
  y_lat(:,i) = C_lat*x_lat_k + D_lat*u_lat_k;
  u_long(:,i) = u_long_k;
  u_lat(:,i) = u_lat_k;
  time(i) = time(i-1) + h;
  x_long_k = x_long_plus1;
  x_lat_k = x_lat_plus1;
end

%% Stop Computational Stopwatch
toc

%% Output the results in graphical form for both models.

%Longitudinal Model State History
figure(1);
subplot(3,1,1);
AoA = plot(time,y_long(2,:));
xlabel('time (s)', 'fontsize', 15);
ylabel('AOA (deg)', 'fontsize', 15);
set(AoA, 'Linewidth', 3);
grid on;
subplot(3,1,2);
q = plot(time,y_long(3,:));
xlabel('time (s)', 'fontsize', 15);
ylabel('pitch rate (deg/s)', 'fontsize', 15);
set(q, 'Linewidth', 3);
grid on;
subplot(3,1,3);
theta = plot(time,y_long(4,:));
xlabel('time (s)', 'fontsize', 15);
ylabel('pitch angle (deg)', 'fontsize', 15);
set(theta, 'linewidth', 3);
grid on;

%Lateral/Directional Model State History
figure(2);
subplot(5,1,1);
beta = plot(time,y_lat(1,:));
xlabel('time (s)','fontsize', 15);
ylabel('sideslip angle (deg)','fontsize', 15);
```

```
set(beta,'linewidth',3);
grid on;
subplot(5,1,2);
p = plot(time,y_lat(2,:));
xlabel('time (s)','fontsize', 15);
ylabel('roll rate (deg/s)','fontsize', 15);
set(p,'linewidth',3);
grid on;
subplot(5,1,3);
r = plot(time,y_lat(3,:));
xlabel('time (s)','fontsize', 15);
ylabel('yaw rate (deg/s)','fontsize', 15);
set(r,'linewidth',3);
grid on;
subplot(5,1,4);
phi = plot(time,y_lat(4,:));
xlabel('time (s)','fontsize', 15);
ylabel('bank angle (deg)','fontsize', 15);
set(phi,'linewidth',3);
grid on;
subplot(5,1,5);
psi = plot(time,y_lat(5,:));
xlabel('time (s)','fontsize', 15);
ylabel('heading angle (deg)','fontsize', 15);
set(psi,'linewidth',3);
grid on;
%Longitudinal Model Control History
figure(3);
subplot(2,1,1);
del_e = plot(time,u_long(1,:));
xlabel('time (s)','fontsize', 15);
ylabel('del_e_L','fontsize', 15);
set(del_e,'linewidth',3);
grid on;
subplot(2,1,2);
del_T = plot(time,u_long(2,:));
xlabel('time (s)','fontsize', 15);
ylabel('del_T','fontsize', 15);
set(del_T,'linewidth',3);
grid on;
%Lateral/Directional Model Control History
figure(4);
subplot(2,1,1);
del_a = plot(time,u_lat(1,:));
xlabel('time (s)','fontsize', 15);
ylabel('del_a_L','fontsize', 15);
set(del_a,'linewidth',3);
grid on;
subplot(2,1,2);
del_r = plot(time,u_lat(2,:));
xlabel('time (s)','fontsize', 15);
ylabel('del_r','fontsize', 15);
set(del_r,'linewidth',3);
grid on;
```

NZSP_AEROSONDE.M

```
%Ryan Goodnight
%Micro Air Vehicle Controls Research
%Texas A&M University
%Advisor: Dr. Helen Reed
%12/06/08

function SDR = NZSP(Ts)
%The following is a code that applies a non-zero set point controller
to an
%Aerosonde UAV model in a disturbed state with the intent of bringing
the
%system back to a straight and level configuration.  A second condition
is
%tested where a heading change is requested.

%% Define States
%Longitudinal Model
% u = x-vel in body frame (m/s)
% w = z-vel in body frame (m/s)
% q = pitch rate (deg/s)
% theta = pitch angle (deg)
% h = altitude (m)
% omega = fuel mass (kg)
% del_Le = Left Elevator (deg)
% del_T = Throttle (deg)

%Lateral/Directional Model
% v = y-vel (m/s)
% p = roll rate (deg/s)
% r = yaw rate (deg/s)q
% phi = bank angle (deg)
% psi = yaw angle (deg)
% del_La = Left Aileron (deg)
% del_r = Rudder (deg)

%% Define Outputs
%Longitudinal Model
% Va = vel (m/s)
% aoa = angle-of-attack (deg)
% q = pitch rate (deg/s)
% theta = pitch angle (deg)
% h = altitude (m)
% del_Le = Left Elevator (deg)
% del_T = Throttle (deg)

%Lateral/Directional Model
% beta = sideslip angle (deg)
% p = roll rate (deg/s)
% r = yaw rate (deg/s)
% phi = bank angle (deg)
% psi = yaw angle (deg)
% del_La = Left Aileron (deg)
% del_r = Rudder (deg)

%% Define Controls
%Longitudinal Model
% del_Le_c = Left Elevator (deg)
% del_T_c = Throttle (deg)

%Lateral/Directional Model
% del_La_c = Left Aileron (deg)
% del_r_c = Rudder (deg)

%% Start Computational Stopwatch
tic

%% Initialize Constants
h = 0.01;
t_f = 5;
%% Define you state, input, output, and feedforward matrices:

%The following matrices were found from flight data at a flight
condition
%of 200 meters and a velocity of 23 m/sec.  The trim conditions are an
%elevator deflection of 0 degrees and an bank angle of 0 degrees.

%Longitudinal Model State Matrix
A_long = [-0.2197     0.6002     -1.4882      -9.7967    -0.0001
0.0108       0.3246      0;
          -0.5820    -4.1204      22.4024     -0.6461     0.0009    0
-2.1520      0;
           0.4823    -4.5284     -4.7512       0          0.000    -
0.0084      -29.8216     0;
           0          0           1.0000       0          0         0
0           0;
           0.0658    -0.9978      0            22.9997    0         0
0           0;
           32.1012    2.117       0            0         -0.0295   -
2.7813       0           448.5357;
           0          0           0            0          0         0
-10          0;
           0          0           0            0          0         0
0          -10];

%Lateral/Directional Model State Matrix
A_lat = [-0.6373     1.5135     -22.9498     9.7967     0      -1.2510
3.1931;
         -4.1919    -20.6283     9.9282      0          0      -
109.8373   1.9763;
          0.6798    -2.6757     -1.0377      0          0      -4.3307
-20.1754;
```

```
          0          1.00        0.0659      0          0       0
0;
          0          0           1.0022      0          0       0
0;
          0          0           0           0          0      -10
0;
          0          0           0           0          0       0
-10];

%The state matrices are separated into the longitudal and
%lateral/directional models.  In previous work it was found that while
%these two models are not coupled, the large matrices dimensions is a
%potential source of complications.  Actuator dynamics have been added
for
%the control surfaces so as to model the difference in commanded
control
%and actual control.  The actual control response is treated as a state
%now.

%Longitudinal Model Input Matrix
B_long = [ 0    0;
           0    0;
           0    0;
           0    0;
           0    0;
           0    0;
           10   0;
           0    10];

%Lateral/Directional Model Input Matrix
B_lat = [ 0    0;
          0    0;
          0    0;
          0    0;
          0    0;
          10   0;
          0    10];

%Like the state matrices, the input matrices are separated into the
%longitudinal and lateral/directional models so as to minimize the
%complications of dealing with large matrices.

%Longitudinal Model Output Matrix
C_long = [0.9978  0.0658  0      0      0      0      0      0;
         -0.0029  0.0434  0      0      0      0      0      0;
          0       0       1.00   0      0      0      0      0;
          0       0       0      1.00   0      0      0      0;
          0       0       0      0      1.00   0      0      0;
          0       0       0      0      0      1.00   0      0;
          0       0       0      0      0      0      1.00   0;
          0       0       0      0      0      0      0      1.00;
          0       0       0      0      0      0      0      0;
          0       0       0      0      0      0      0      0];
```

```
%Lateral/Directional Model Output Matrix
C_lat = [0.0435 0       0      0      0      0      0;
         0      1.00    0      0      0      0      0;
         0      0       1.00   0      0      0      0;
         0      0       0      1.00   0      0      0;
         0      0       0      0      1.00   0      0;
         0      0       0      0      0      1.00   0;
         0      0       0      0      0      0      1.00;
         0      0       0      0      0      0      0;
         0      0       0      0      0      0      0];

%Like the state and input matrices, the output matrices are separated
%into the longitudinal and lateral/directional models so as to minimize
%the complications of dealing with large matrices.  The output matrices
%have had the control terms concatinated in for analysis.

%Longitudinal Model QPM Output Matrix
C_long_QPM = [0     0     0     1.00  0     0    0    0;
              0     0     0     0     1.00  0    0    0];

%This identifies that pitch angle and altitude will be the
%longitudinally controlled states.

%Lateral/Directional Model QPM Output Matrix
C_lat_QPM = [0    0   1.00  0    0     0   0;
             0    0   0     0    1.00  0   0];

%This identifies that the yaw rate and heading angle will
%be the laterally commanded states.

%Longitudinal Model Feedforward Matrix
D_long = [zeros(8,2); eye(2)];

%Lateral/Directional Model Feedforward Matrix
D_lat = [zeros(7,2); eye(2)];

%Both feedforward matrices have had an identity matrix matrix of a
%dimension matching the number of controls concatinated to the bottom
of
%the bottom of a zeroed matrix of a dimension matching the input
matrices
%so that the control reactions may be observed during analysis.

%Longitudinal Model QPM Feedforward Matrix
D_long_QPM = eye(2);

%Lateral/Directional Model QPM Feedforward Matrix
D_lat_QPM = eye(2);
```

```
%% Define the initial perturbed flight configurations and the desired
final flight configuration to stabilize around:

%In this scenario we are considering the case where an MAV is being
%hand-launched and needs to return to a straight and level flight
%configuration.  The initial conditions are a pitch angle of 15 degrees
in
%the longitudinal model and a bank angle of 25 degrees in the
%lateral/directional model.

%Initial state vectors are in the following form:
%x_long' = [u, w, q, theta, h, omega, del_Le, del_T];
x_long_k = [0;0;0;0;0;0;0;0];

%x_lat' = [v, p, r, phi, psi, del_La, del_r];
x_lat_k = [0;0;0;0;0;0;0];

%Desire state vectors are of the following form:
%y_long_m' = [theta, h];
y_long_m = [0;50];

%y_lat_m' = [r, psi];
y_lat_m = [0;15];

%These intial and final state vectors are varied based off of the case
%being analyzed.

%% Control vectors are of the following form:
%u_long' = [left elevator commanded, throttle commanded];

%u_lat' = [left aileron commanded, rudder commanded];

%Output vectors are of the following form:
%y_long' = [Va, alpha, q, theta, h, del_Le, del_T];

%y_lat' = [beta, p, r, phi, psi, del_La, del_r];

%% Define State Transition Matrix and Discrete Control Distribution
Matrix

%Longitudinal Model STM and DCDM
[Phi_long,Gamma_long] = c2d(A_long, B_long, h);
[Phi_long_d,Gamma_long_d] = c2d(A_long,B_long,Ts);

%Lateral/Directional Model STM and DCDM
[Phi_lat,Gamma_lat] = c2d(A_lat, B_lat, h);
[Phi_lat_d,Gamma_lat_d] = c2d(A_lat,B_lat,Ts);

%% Find the X1 and X2 Values from the QPM using QPMCALC.m

%Longitudinal Model
```

```
[X12_long,X22_long] = QPMCALC(Phi_long_d-
eye(8),Gamma_long_d,C_long_QPM,D_long_QPM);

%Lateral Directional Model
[X12_lat,X22_lat] = QPMCALC(Phi_lat_d-
eye(7),Gamma_lat_d,C_lat_QPM,D_lat_QPM);

%% Define state, control, and final state weighting matrices for a
quadratic cost function:

%Longitudinal Model State Weighting Matrix
Q_long = diag([1   1   1   1   0.8   1    1    5]);
%x_long = [u, w, q, theta, h, omega, del_Le, del_T];

%Lateral/Directional Model State Weighting Matrix
Q_lat = diag([1   3    1   10   50   1   1]);
%x_lat = [v, p, r, phi, psi, del_La, del_r];

%The first iteration towards solving for the state weighting matrices
was
%to place a "1" in each of the diagonal terms of a square matrix with
the
%dimension n of the state vector.  An alternate approach is to use the
%balanced weighting elements technique.

%Longitudinal Model Control Weighting Matrix
%u_long = [del_e_c, del_T_c];
R_long = diag([1  1]);

%Lateral/Directional Model Control Weighting Matrix
%u_lat = [del_a_c, del_r_c];
R_lat = diag([1  1]);

%The first iteration towards solving for the control weighting matrices
%was to place a "1" in each of the diagonal terms of a square matrix
with
%the dimension n of the state vector.  An alternate approach is to use
the
%balanced weighting elements technique.

%Longitudinal Final State Weighting Matrix
N_long = zeros(8,2);

%Lateral/Directional Final State Weighting Matrix
N_lat = zeros(7,2);

%In both models these matrices are set to zero for simplicity.

%% Calculate the optimal control gains, K, for each model using
LQRDJV.m:
```

```
%Longitudinal Model Optimal Kalman Gains
[K_long,Q_long_h,R_long_h,M_long,P_long,E_long] =
LQRDJV(A_long,B_long,Q_long,R_long,N_long,Ts);

%Lateral/Directional Model Optimal Kalman Gains
[K_lat,Q_lat_h,R_lat_h,M_lat,P_lat,E_lat] =
LQRDJV(A_lat,B_lat,Q_lat,R_lat,N_lat,Ts);

%% Initialize a starting control vector for each model:

%Longitudinal Model Starting Control Vector
u_long_k = (X22_long+K_long*X12_long)*y_long_m-K_long*x_long_k;

y_long(:,1) = C_long*x_long_k + D_long*u_long_k;

%Lateral/Directional Model Starting Control Vector
u_lat_k = (X22_lat+K_lat*X12_lat)*y_lat_m-K_lat*x_lat_k;

y_lat(:,1) = C_lat*x_lat_k + D_lat*u_lat_k;

%% Using the Kalman gains and the STM and DCM build a time history of
the control and command history of both models.

%Define the number of intervals
n_frames = t_f/h;

%Create a time array and fill the first entry with t = 0
time(1) = 0;

%Initialize a counter for rebuilding the command and control history.
counter = 0;

%Build control and command history
for i = 2:n_frames
    if abs(counter-Ts) < 1e-8
        u_long_k = (X22_long+K_long*X12_long)*y_long_m-K_long*x_long_k;
        u_lat_k = (X22_lat+K_lat*X12_lat)*y_lat_m-K_lat*x_lat_k;
        counter = 0;
    else
        u_long_k = u_long_k;
        u_lat_k = u_lat_k;
    end
  counter = counter + h;
  x_long_plus1 = Phi_long*x_long_k + Gamma_long*u_long_k;
  x_lat_plus1 = Phi_lat*x_lat_k + Gamma_lat*u_lat_k;
  y_long(:,i) = C_long*x_long_k + D_long*u_long_k;
  y_lat(:,i) = C_lat*x_lat_k + D_lat*u_lat_k;
  u_long(:,i) = u_long_k;
  u_lat(:,i) = u_lat_k;
  time(i) = time(i-1) + h;
  x_long_k = x_long_plus1;
  x_lat_k = x_lat_plus1;
end

toc

%% Output the results in graphical form for both models.

%Longitudinal Model State History
figure(1);
subplot(4,1,1);
AoA = plot(time,y_long(2,:));
xlabel('time (s)', 'fontsize', 15);
ylabel('AOA (deg)', 'fontsize', 15);
set(AoA, 'Linewidth', 3);
grid on;
subplot(4,1,2);
q = plot(time,y_long(3,:));
xlabel('time (s)', 'fontsize', 15);
ylabel('pitch rate (deg/s)', 'fontsize', 15);
set(q, 'Linewidth', 3);
grid on;
subplot(4,1,3);
theta = plot(time,y_long(4,:));
xlabel('time (s)', 'fontsize', 15);
ylabel('pitch angle (deg)', 'fontsize', 15);
set(theta, 'linewidth', 3);
grid on;
subplot(4,1,4);
h = plot(time,y_long(5,:)+200);
xlabel('time (s)', 'fontsize', 15);
ylabel('altitude (m)', 'fontsize', 15);
set(h,'linewidth',3);
grid on;

%Lateral/Directional Model State History
figure(2);
subplot(5,1,1);
beta = plot(time,y_lat(1,:));
xlabel('time (s)','fontsize', 15);
ylabel('sideslip angle (deg)','fontsize', 15);
set(beta,'linewidth',3);
grid on;
subplot(5,1,2);
p = plot(time,y_lat(2,:));
xlabel('time (s)','fontsize', 15);
ylabel('roll rate (deg/s)','fontsize', 15);
set(p,'linewidth',3);
grid on;
subplot(5,1,3);
r = plot(time,y_lat(3,:));
xlabel('time (s)','fontsize', 15);
ylabel('yaw rate (deg/s)','fontsize', 15);
set(r,'linewidth',3);
```

```
grid on;
subplot(5,1,4);
phi = plot(time,y_lat(4,:));
xlabel('time (s)','fontsize', 15);
ylabel('bank angle (deg)','fontsize', 15);
set(phi,'linewidth',3);
grid on;
subplot(5,1,5);
psi = plot(time,y_lat(5,:));
xlabel('time (s)','fontsize', 15);
ylabel('heading angle (deg)','fontsize', 15);
set(psi,'linewidth',3);
grid on;

%Longitudinal Model Control History
figure(3);
subplot(2,1,1);
del_e = plot(time,u_long(1,:));
xlabel('time (s)','fontsize', 15);
ylabel('del_e_L','fontsize', 15);
set(del_e,'linewidth',3);
grid on;
subplot(2,1,2);
del_T = plot(time,u_long(2,:));
xlabel('time (s)','fontsize', 15);
ylabel('del_T','fontsize', 15);
set(del_T,'linewidth',3);
grid on;

%Lateral/Directional Model Control History
figure(4);
subplot(2,1,1);
del_a = plot(time,u_lat(1,:));
xlabel('time (s)','fontsize', 15);
ylabel('del_a_L','fontsize', 15);
set(del_a,'linewidth',3);
grid on;
subplot(2,1,2);
del_r = plot(time,u_lat(2,:));
xlabel('time (s)','fontsize', 15);
ylabel('del_r','fontsize', 15);
set(del_r,'linewidth',3);
grid on;
```

LQRDJV.M

```
function [k,Qd,Rd,Nd,s,e] = lqrdjv(a,b,q,r,nn,Ts)
%%%%%%function [k,s,e,Qd,Rd,Nd] = lqrdjv(a,b,q,r,nn,Ts)
%LQRDJV  Discrete linear quadratic regulator design from continuous
%       cost function.
%
%
*********************************************************************
%
%           =====>  modified by J. Valasek, 5 Dec 94   <=====
%
%           This routine now sends back the Q^, R^, and M^ matrices.
%
%           NOTE: the order of the passed-back arguments is NOT the
%                 same as the original MATLAB version.
%
%
*********************************************************************
%
%   [K,S,E] = LQRD(A,B,Q,R,Ts) calculates the optimal feedback gain
%   matrix K such that the discrete feedback law  u[n] = -K x[n]
%   minimizes a discrete cost function equivalent to the continuous
%   cost function
%       J = Integral {x'Qx + u'Ru} dt
%                                                        .
%   subject to the continuous constraint equation:  x = Ax + Bu
%
%   Also returned is S, the discrete Riccati equation solution, and
%   the closed loop eigenvalues E = EIG(Ad-Bd*K).
%
%   The gain matrix is determined by discretizing the continuous plant
%   (A,B,C,D) and continuous weighting matrices (Q,R) using the sample
%   time Ts and the zero order hold approximation. The gain matrix is
%   then calculated using DLQR.
%
%   [K,S,E] = LQRD(A,B,Q,R,N,Ts) includes the cross-term N that
%   relates u to x in the cost function.
%       J = Integral {x'Qx + u'Ru + 2*x'Nu}
%
%   See also: C2D, LQED, DLQR, and LQR.

%   Clay M. Thompson 7-16-90
%   Copyright (c) 1986-93 by the MathWorks, Inc.

% Reference: This routine is based on the routine JDEQUIV.M by
Franklin,
% Powell and Workman and is described on pp. 439-441 of "Digital
Control
% of Dynamic Systems".

error(nargchk(5,6,nargin));
error(abcdchk(a,b));
[nx,na] = size(a);
[nb,nu] = size(b);

[nq,mq] = size(q);
if (nx ~= nq) | (nx ~= mq), error('A and Q must be the same size.');
end
[nr,mr] = size(r);
if (mr ~= nr) | (nu ~= mr), error('B and R must be consistent.'); end

if nargin==5,
  Ts = nn;
  nn = zeros(nb,nu);
else
  [nnn,mn] = size(nn);
  if (nnn ~= nx) | (mn ~= nu), error('N must be consistent with Q and
R.'); end
end

% Check if q is positive semi-definite and symmetric
if any(eig(q) < -eps) | (norm(q'-q,1)/norm(q,1) > eps)
  disp('Warning: Q is not symmetric and positive semi-definite');
end
% Check if r is positive definite and symmetric
if any(eig(r) <= -eps) | (norm(r'-r,1)/norm(r,1) > eps)
  disp('Warning: R is not symmetric and positive definite');
end

% Discretize the state-space system.
[ad,bd] = c2d(a,b,Ts);

% --- Determine discrete equivalent of continuous cost function ---
n = nx+nu;
Za = zeros(nx); Zb = zeros(nx,nu); Zu = zeros(nu);
M = [ -a' Zb   q  nn
      -b' Zu  nn'  r
      Za  Zb   a   b
      Zb' Zu  Zb' Zu];
phi = expm(M*Ts);
phi12 = phi(1:n,n+1:2*n);
phi22 = phi(n+1:2*n,n+1:2*n);
QQ = phi22'*phi12;
QQ = (QQ+QQ')/2;        % Make sure QQ is symmetric
Qd = QQ(1:nx,1:nx) ;
Rd = QQ(nx+1:n,nx+1:n) ;
Nd = QQ(1:nx,nx+1:n) ;

% Design the gain matrix using the discrete plant and discrete cost
function
[k,s,e] = dlqr(ad,bd,Qd,Rd,Nd);
```

QPMCALC.M

```
%  qpmcalc.m
%  This M-file assembles the Quad Partition Matrix (QPM) and returns
%  the sub-matrices X12 and X22 from passed A, B, C, and D.
%
%  =================================================================
%
%  written by:   J. Valasek
%                WMU Aircraft Design and Control Laboratory
%                19 January 1996
%
%  =================================================================
%       DIGITAL FLIGHT CONTROL SYSTEMS: Analysis and Design
%                              by
%                         David R. Downing
%                         John Valasek

%   The University of Kansas - Division of Continuing Education

%                         1 September 2003
%  =================================================================
function [X12, X22] = QPMCALC(A, B, C, D) ;

%
% .. determine dimenstions of passed matrices and vectors
%

[rowa, cola] = size(A) ;
[rowb, colb] = size(B) ;
[rowc, colc] = size(C) ;
[rowd, cold] = size(D) ;

%
% .. quad partition matrix and its inverse
%

qpm  = [A, B ; C, D]   ;
qpmi = inv(qpm)       ;

%
% .. break out the X12 and X22 quadrants
%

X12 = qpmi(1:rowa, cola+1:cola+colb)           ;
X22 = qpmi(rowa+1:rowa+rowc, cola+1:cola+colb) ;
```

MAVIMAGECOLLECTION.BS2

```
' MAV Image Collection Software - Version 0.0
' Ryan Goodnight - 2009

' {$STAMP BS2}
' {$PBASIC 2.5}

' -----[ Pins/Constants/Variables ]------------------------------------------

Sio           PIN    15             ' connects to GPS Module SIO pin
ServoIn       PIN    14             ' Receives image collection request
Camera        PIN    1              ' connects camera to SIO pin
DinDout       PIN    6              ' P6 transceives to/from Din/Dout
Clk           PIN    5              ' P5 sends pulses to HM55B's Clk
En            PIN    4              ' P4 controls HM55B's /EN(ABLE)
TX            PIN    8              ' Transmit Data   --> 27937.4 (RXD)
RTS           PIN    9              ' Request To Send --> 27937.6 (CTS)
RX            PIN    10             ' Receive Data    <-- 27937.5 (TXD)
CTS           PIN    11             ' Clear To Send   <-- 27937.2 (RTS)

Reset         CON    %0000             ' Reset command for HM55B
Measure       CON    %1000             ' Start measurement command
Report        CON    %1100             ' Get status/axis values command
Ready         CON    %1100             ' 11 -> Done, 00 -> no errors
NegMask       CON    %1111100000000000  ' For 11-bit negative to 16-bits
BaudReceive   CON    188               ' Baud rate for receiving GPS
T4800         CON    188
Open          CON    $8000

Baud          CON    Open | T4800      ' Open mode to allow daisy chaining
BaudUSB       CON    84                ' Serial Baud Rate 9600 bps (BS2)

x             VAR    W0             ' x-axis data
y             VAR    W1             ' y-axis data
status        VAR    Nib            ' Status flags
angle         VAR    W2             ' Store angle measurement
ioByte        VAR    Byte
buffer        VAR    Byte(15)       ' Input Buffer
index         VAR    Byte           ' Index Variable
flag          VAR    Bit            ' Event Status Flag
Length        VAR    Byte

'Initiailize camera TO NOT take a picture
```

```
INPUT Camera
'LOW 13                          'Don't take picture until signalled to do so.

'Initialize the USB Flash Disc
'DEBUG CLS, "Memory Stick Datalogger Demo V1.0", CR, CR, "Initializing..."
PAUSE 200                       ' Allow Time To Settle

HIGH TX                         ' Initialize Transmit Line
LOW RTS                         ' Take Vinculum Out Of Reset

PAUSE 600                       ' Allow Time To Settle
DEBUG "Done!", CR, "Synchronizing..."

DO
  SEROUT TX\CTS, BaudUSB, ["E", CR]       ' Sync Command Character
  GOSUB Get_Data                    ' Get Response
  PAUSE 250
LOOP UNTIL ioByte = $0D                         ' Wait For Carriage Return

DO
  SEROUT TX\CTS, BaudUSB, ["e", CR]        ' Sync Command Character
  GOSUB Get_Data                    ' Get Response
  PAUSE 250
LOOP UNTIL ioByte = $0D                      ' Wait For Carriage Return

Main:
  DEBUG "Done", CR, "Switching to Short Command Mode..."
  SEROUT TX\CTS, BaudUSB, ["SCS", CR]   ' Switch To Short Command Mode
  GOSUB Get_Data                    ' Purge Receive Buffer
  DEBUG "Done!", CR, "Waiting for Memory Stick..."

Check_Drive:
  DO
    SEROUT TX\CTS, BaudUSB, [CR]        ' Prompt Device For Status
    GOSUB Get_Data                          ' Purge Receive Buffer
    IF buffer(0) = ">" THEN                 ' Check For Ready Prompt
      EXIT                         ' If Ready Then Exit Loop
    ELSEIF buffer(0) = "N" AND buffer(1) = "D" THEN
      DEBUG "."                     ' Device Ready But No Memory Stick
    ELSEIF buffer(0) = "D" AND buffer(1) = "D" AND flag = 0 THEN
      DEBUG "Connected!", CR, "Accessing..."
      flag = 1                                  ' Memory Stick Ready
    ELSE
      DEBUG "."
    ENDIF
    PAUSE 250                               ' Command Retry Delay
  LOOP
  DEBUG "Ready!", CR

  DEBUG "Deleting Data File...",CR      ' First Delete File
  SEROUT TX\CTS, BaudUSB, [$07, $20, "datafile.txt", CR]
  GOSUB Get_Easy_Data                ' Purge Receive Buffer
                                    ' Then Create File
  DEBUG "Creating Data file", CR
  SEROUT TX\CTS, BaudUSB, [$09, $20, "datafile.txt", CR]
  GOSUB Get_Easy_Data                ' Purge Receive Buffer

  DEBUG "Open!", CR, CR', "Writing Data...", CR

' -----[ Main Routine ]------------------------------------------------------
DO

  PULSIN ServoIn, 1, Length

'  DEBUG DEC Length, CR
'  PAUSE 1000

  IF Length < 100 THEN
  'Open the data file
    SEROUT TX\CTS, BaudUSB, [$09, $20, "datafile.txt", CR]
    GOSUB Get_Easy_Data
    DEBUG "Opened File", CR

  'Read GPS Info from Sensor
    GOSUB GPS_Get_Info                      ' Request GPS information
    GOSUB FindGpsString

  'Write GPS Info To Disk
    GOSUB GPS_Write_Info                    ' Output the GPS information
    GOSUB Get_Easy_Data

  'Command Camera to Take Picture
    DEBUG "Wrote GPS INFO", CR
    LOW Camera
    PAUSE 100
    INPUT Camera                             ' Flag Camera I/O Low

  'Check Magnetic Heading
    GOSUB Compass_Get_Heading              ' Get x, and y values

    angle = x ATN -y                       ' Convert x and y to brads
    angle = angle */ 360                   ' Convert brads to degrees

  'Write Heading to Disk
    GOSUB Compass_Write_Heading            ' Output the heading angle
    GOSUB Get_Easy_Data
    DEBUG "Wrote compass info", CR

  'Read GPS Info from Sensor
    GOSUB GPS_Get_Info                     ' Request GPS information

  'Write GPS Info to Disk
    GOSUB GPS_Write_Info                   ' Output the GPS information
    GOSUB Get_Easy_Data
    DEBUG "wrote second GPS" , CR
    GOSUB Compass_Get_Heading              ' Get x, and y values

    angle = x ATN -y                  ' Convert x and y to brads
    angle = angle */ 360              ' Convert brads to degrees

  'Write Heading to Disk
    GOSUB Compass_Write_Heading
    GOSUB Get_Easy_Data
    DEBUG "wrote second compass heading"
    DEBUG "Closing Data File..."      ' Close File (MUST CLOSE!)
    SEROUT TX\CTS, BaudUSB, [$0A, $20, "datafile.txt", CR]
    GOSUB Get_Easy_Data               ' Purge Receive Buffer

  ELSE
    PAUSE 40

  ENDIF

LOOP


' -----[ Subroutines ]-----------------------------------------------------

Compass_Get_Heading:                ' Compass module subroutine

    HIGH En: LOW En                   ' Send reset command to HM55B
    SHIFTOUT DinDout,clk,MSBFIRST,[Reset\4]
    HIGH En: LOW En                    ' HM55B start measurement command
    SHIFTOUT DinDout,clk,MSBFIRST,[Measure\4]
    status = 0                         ' Clear previous status flags

    DO                                 ' Status flag checking loop
      HIGH En: LOW En                  ' Measurement status command
      SHIFTOUT DinDout,clk,MSBFIRST,[Report\4]
      SHIFTIN  DinDout,clk,MSBPOST,[Status\4]  ' Get Status
    LOOP UNTIL status = Ready          ' Exit loop when status is ready

    SHIFTIN  DinDout,clk,MSBPOST,[x\11,y\11]   ' Get x & y axis values
    HIGH En                            ' Disable module

    IF (y.BIT10 = 1) THEN y = y | NegMask      ' Store 11-bits as signed word
    IF (x.BIT10 = 1) THEN x = x | NegMask      ' Repeat for other axis

    RETURN

Compass_Write_Heading:

 'DEBUG DEC angle, " degrees", CR
    SEROUT TX\CTS, BaudUSB, [$08, $20, $00, $00, $00, $07, CR,
    DEC5 angle, CR,LF,
    CR]

    RETURN

GPS_Get_Info:

    SERIN 15, BaudReceive, 2100, noGPSfound, [WAIT("GGA,"), B0, B1, B2, B3, B4,
B5, SKIP 1, B6, B7, B8, B9, SKIP 1, B10, B11, B12, B13, SKIP 3, B14, B15, B16, B17,
B18, SKIP 1 , B19, B20, B21, B22, SKIP 15 ,B23, B24, B25]

    RETURN

FindGpsString:

    RETURN

GPS_Write_Info:
  SEROUT TX\CTS, BaudUSB, [$08, $20, $00, $00, $00, $22, CR, B2, B3, B4, B5, ",",
B6, B7, B8, B9, ".", B10, B11, B12, B13, ",N,", B14, B15, B16, B17, B18, "." , B19,
B20, B21, B22, ",W,", B23, B24, B25, ",", CR]

  RETURN

noGPSfound:

  DEBUG "No GPS Found!", CR

  RETURN

' -----[DataLogger Subroutines ]---------------------------------------------------

Get_Data:
 index = 0                    ' Reset Index Pointer
 DO                           ' Receive Data
   SERIN RX\RTS, BaudUSB, 100, Timeout, [ioByte]
   buffer(index) = ioByte          ' Add Received Byte To Buffer
   index = index + 1               ' Increment Index Pointer
   IF index > 14 THEN Timeout         ' Check For Overflow
 LOOP

Timeout:
 RETURN

Get_Easy_Data:
 B1 = 0                       ' Reset Index Pointer
 DO                           ' Receive Data
   SERIN RX\RTS, BaudUSB, 100, Timeout, [B0]
 B1 = B1 + 1
   IF B1 > 14 THEN Timeout         ' Check For Overflow
 LOOP
```

VITA

Ryan David Goodnight received his Bachelor of Science degree in aerospace engineering from Texas A&M University in College Station in 2007. He began his pursuit of a M.S. degree in aerospace engineering the following fall. His research interests include advancing the state of micro air vehicle combat operations, deep space object identification and imaging, and space mission design for future exploration and touristic endeavors. Upon the completion of his M.S. degree, Ryan will begin working toward a Master's of Business Administration in the Mays School of Business at Texas A&M University.

Mr. Goodnight can be reached at H.R. Bright Building, Rm. 701, Ross Street – TAMU 3141, College Station, TX 77843-3141. His email address is ryan.goodnight@gmail.com.