NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

CONTROL AND IN-FLIGHT DIAGNOSTIC OF B-52–TYPE AIRCRAFT

by

Benjamin W. Hogin

June 2020

Thesis Advisor:	Vladimir N. Dobrokhodov
Co-Advisors:	Kevin D. Jones
	Mariano Lizarraga, MathWorks
Second Reader:	Christopher S. Clay

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2020	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
CONTROL AND IN-FLIGHT DIAGNOSTIC OF B-52–TYPE AIRCRAFT

5. FUNDING NUMBERS

6. AUTHOR(S) Benjamin W. Hogin

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

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

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

The democratization of rapidly advancing drone technology has enabled access to a warfare space previously denied to malicious actors with limited resources. This thesis explores potential weaponization of commercial-off-the-shelf (COTS) autopilot units for use in a ballistic aerial delivery mission by adopting the well-established approach of a computed aerial release point (CARP). Simulation of a model B-52 aircraft equipped with a COTS autopilot to deliver a payload to the CARP was used to establish the prerequisite technological and expertise requirements inherent in achieving desired performance of the CARP mission, despite adverse flight conditions. The current state of the art in unmanned aerial vehicle (UAV) hardware, sensors, and navigation algorithms was evaluated and used to inform the level of augmentation to a nominal COTS autopilot to enable higher levels of precision. To accomplish the objective, a framework was established to rapidly evaluate the aerodynamics, guidance, navigation, and control (GNC) components of a UAV. In the developed simulation environment, a properly tuned and implemented COTS autopilot presents a menacing threat. As a result of this study, GPS denial was identified as a promising avenue in mitigating the UAV weapons delivery threat and is recommended as the subject of future research.

14. SUBJECT TERMS
autopilot, control, simulation, aerodynamics, B-52, stability, PID, Simulink, COTS, commercial-off-the-shelf, computing aerial release point, CARP, unmanned aerial vehicle, UAV, guidance, navigation, and control, GNC

15. NUMBER OF PAGES
83

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

CONTROL AND IN-FLIGHT DIAGNOSTIC OF B-52–TYPE AIRCRAFT

Benjamin W. Hogin
Ensign, United States Navy
BS, United States Naval Academy, 2019

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN ENGINEERING SCIENCE
(AEROSPACE ENGINEERING)**

from the

**NAVAL POSTGRADUATE SCHOOL
June 2020**

Approved by: Vladimir N. Dobrokhodov
Advisor

Kevin D. Jones
Co-Advisor

Mariano Lizarraga
Co-Advisor

Christopher S. Clay
Second Reader

Garth V. Hobson
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The democratization of rapidly advancing drone technology has enabled access to a warfare space previously denied to malicious actors with limited resources. This thesis explores potential weaponization of commercial-off-the-shelf (COTS) autopilot units, for use in a ballistic aerial delivery mission by adopting the well-established approach of a computed aerial release point (CARP). Simulation of a model B-52 aircraft equipped with a COTS autopilot to deliver a payload to the CARP was used to establish the prerequisite technological and expertise requirements inherent in achieving desired performance of the CARP mission, despite adverse flight conditions. The current state of the art in unmanned aerial vehicle (UAV) hardware, sensors, and navigation algorithms was evaluated and used to inform the level of augmentation to a nominal COTS autopilot to enable higher levels of precision. To accomplish the objective, a framework was established to rapidly evaluate the aerodynamics, guidance, navigation, and control (GNC) components of a UAV. In the developed simulation environment, a properly tuned and implemented COTS autopilot presents a menacing threat. As a result of this study, GPS denial was identified as a promising avenue in mitigating the UAV weapons delivery threat and is recommended as the subject of future research.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1 A Review of Autonomy **1**
1.1 Motivation 1
1.2 Introduction to Feedback Control 2
1.3 The Trade Space 4
1.4 Study Methodology 12

2 Task Formulation **17**
2.1 Task Formulation Methodology 17
2.2 Hardware Augmentation 18
2.3 Navigation Algorithm 26
2.4 Value Consideration of COTS Units 28

3 Modeling of the B-52 **29**
3.1 Aerodynamic Analysis 29
3.2 Open Loop System Response 31
3.3 Guidance, Navigation, and Control 37

4 Results and Discussion **45**
4.1 Test Procedure 45
4.2 Gyro and Accelerometer Noise 47
4.3 GPS Navigation Update Rate. 49
4.4 GPS Noise 52
4.5 Wind Modeling 53
4.6 Ground Station Interface 56

5 Conclusions and Recommendations **59**

List of References **61**

Initial Distribution List **65**

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

Figure 1.1 Closed loop feedback diagram. 2

Figure 1.2 PID control architecture. 3

Figure 1.3 The Piccolo family of autopilots. 6

Figure 1.4 The Piccolo desktop ground station. 7

Figure 1.5 Piccolo hardware in the loop simulation diagram. 7

Figure 1.6 Piccolo hardware in the loop simulation. 8

Figure 1.7 Autopilot selection study. 10

Figure 1.8 Pixhawk port diagram. 11

Figure 1.9 QGroundControl waypoint planning interface. 12

Figure 1.10 Software in the loop simulation workflow. 13

Figure 1.11 COTS system evaluation workflow. 14

Figure 1.12 COTS system evaluation tools. 15

Figure 2.1 Concept of operations for B-52 weapons delivery mission. 18

Figure 2.2 Diagram of CPU interaction with input/output devices. 19

Figure 2.3 Signal to noise ratio of minimum receiver sensitivity to thermal noise. 22

Figure 2.4 Force inducing a voltage in a piezoelectric material. 24

Figure 2.5 Position estimation from rate measurements. 24

Figure 2.6 Kalman filter conceptual diagram. 25

Figure 2.7 Unmanned aerial vehicle navigation model for curved path following. 27

Figure 3.1 B-52 in MachUp 4 environment. 30

Figure 3.2 B-52 wireframe in MachUpX environment. 31

Figure 3.3 Pole-zero map of B-52 longitudinal dynamics. 33
Figure 3.4 Phugoid and short period dynamic simulation. 34
Figure 3.5 Pole-zero map of B-52 lateral dynamics. 36
Figure 3.6 System response to rudder doublet command. 37
Figure 3.7 System response to step aileron command. 38
Figure 3.8 Longitudinal inner loop. 39
Figure 3.9 Implementation of logical switches including the guidance mode. 40
Figure 3.10 Computation of model lookahead vector. 41
Figure 3.11 Addition of Gaussian noise to sensor signal. 42

Figure 4.1 Proposed trajectory for test. 46
Figure 4.2 Probability density function of miss distance data. 47
Figure 4.3 Cumulative probability function of miss distance data. 48
Figure 4.4 Miss distance for increasing gyroscope and accelerometer noise. . 49
Figure 4.5 Histogram of miss distance for increasing gyroscope and accelerometer noise. 50
Figure 4.6 Miss distance as a function of navigation update rate. 51
Figure 4.7 UAV states as a function of navigation update rate. 52
Figure 4.8 Miss distance as a function of GPS measurement noise. 53
Figure 4.9 Miss distance as a function of wind direction. 54
Figure 4.10 Occurrences of miss distances for wind test. 55
Figure 4.11 Miss distances as a function of wind direction, increased distance to computed air release point (CARP). 56
Figure 4.12 Simulated waypoint mission using QGroundControl (QGC) user interface (UI). 57

List of Tables

Table 3.1 Eigenvalues, natural frequency, and damping ratio of longitudinal modes. 33

Table 3.2 Eigenvalues, natural frequency, and damping ratio of lateral modes. 36

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

UAV	unmanned aerial vehicle
CFD	computational fluid dynamics
HIL	hardware in the loop
LTI	linear time invariant
UI	user interface
PID	proportional integral derivative
USN	U.S. Navy
PI	proportional integral
COTS	commercial-off-the-shelf
UHF	ultra-high frequency
GPS	global positioning system
DGS	desktop ground station
PCC	Piccolo Command Center
DOF	degrees-of-freedom
IED	Improvised Explosive Device
CONOPS	concept of operations
CPU	central processing unit
FPU	floating point unit
SRAM	static random-access memory

IMU inertial measurement unit

CARP computed air release point

VLOS visual line of sight

QGC QGroundControl

SIL software in the loop

SUT system under test

GCS ground control station

PIO pilot induced oscillation

GNC guidance, navigation, and control

MAC mean aerodynamic chord

CG center of gravity

CEP circular error probable

SEP spherical error probable

SLUF steady, wings level, unaccelerated flight

Acknowledgments

I would like to thank Dr. Dobrokhodov for his leadership and guidance. His commitment to personal excellence and dedication to his students provided a great example. Additionally, the help and insight provided by Dr. Lizarraga was a great gift and significantly advanced my project. To Dr. Jones and Chris Clay, the knowledge and passion you brought to the project was a constant motivator. I would also like to thank my family for their constant support and encouragement throughout my life. I am deeply grateful for the gift of attending the Naval Academy, and for the teachers and mentors who encouraged me to be a lifelong learner and a person of integrity. Specifically, I would like to thank Dr. Niewoehner, CAPT, USN (ret.), Dr. Miklosovic, Maj. Graves, USMC, and Maj. Martino, USMC; I regard the lessons you taught me as among the most valuable I possess. Lastly, to ADM Bill Moran, USN (ret.), and CAPT Chuck Zingler, USN (ret.), thank you for your mentorship and example, as I have grown as a man and young officer.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
A Review of Autonomy

This chapter introduces the motivation of the research, as well as establishes the methodology to be used in selecting a commercial-off-the-shelf (COTS) autopilot as candidate for augmentation to military specification. Success in the study is defined as mission success by the unmanned aerial vehicle (UAV) in a ballistic payload delivery mission flown to the computed air release point (CARP) , requiring only primitive understanding of the system by the end user.

1.1 Motivation

Since the first flight of a powered aircraft at the turn of the 20th century, the aviation industry has evolved at an incredible rate. With the blistering pace of technological development, the contemporary limitations of many aerospace platforms are the imaginations of those who strive to devise novel and exciting ways to implement that which is now readily available. Whereas the skies once were reserved for corporations, governments, and those with the means to overcome the prohibitive cost of personal air travel, airspace access is undergoing a rapid democratization. While the availability of personal aerial vehicles is an exciting prospect, unfortunately, there now exists the possibility that those with malicious intentions have another weapon at their disposal. As early as 2016, the Islamic State used drones as weapons in its defense of Mosul, loading them with makeshift payloads of grenades and small explosives [1]. Drones have similarly been used to facilitate prison escapes, and to spread hateful speech to large audiences. Therefore, it is imperative to understand the limitations of commercially available unmanned vehicles from the perspective of the engineering efforts required to weaponize a COTS UAV. To this end, a study was carried out in order to determine the level of engineering knowledge required to bring a commercially available autopilot unit to complete a military mission. As a case study in guidance and control of a military aircraft, a model of the B-52 Stratofortress was constructed. The B-52 as system under test (SUT), aimed to complete a CARP weapons delivery mission, controlled by a COTS autopilot unit.

1.2 Introduction to Feedback Control

Though an automatic control problem can vary in complexity or method of solution, certain elements of the problem are fundamentally the same. Every system to be controlled has a set of desired outputs, for the case of a UAV altitude, heading, and airspeed are all examples of output parameters an operator would wish to control. Additionally, a control system must meet the criteria of controllability, or rather, the system must have the means to provide inputs in order to affect the desired outputs. Examples of input control in a UAV are adequate servo control in the elevator to stabilize the attitude of the craft in pitch. A control system will encounter disturbances, or conditions external to the system that interfere with the ability of inputs to produce the desired output. Common disturbances in aerospace applications tend to involve wind or sudden changes in ambient conditions. Lastly, a fundamental trait of all control systems concerns imperfect knowledge of the object under control—the plant. While a process or plant can be understood via the physical laws governing its behavior, the essence of linear control theory involves the imperfect assumption that around an operating point the system remain linear [2].

In order to manage inputs, disturbances, and outputs and meet output requirements, a typical control system utilizes feedback. So called "closed loop" feedback utilizes inputs and measured outputs to calculate an error signal. The purpose of a closed-loop feedback system is to regulate the error between the input and the output to zero. Figure 1.1 illustrates the concept of closed-loop feedback.

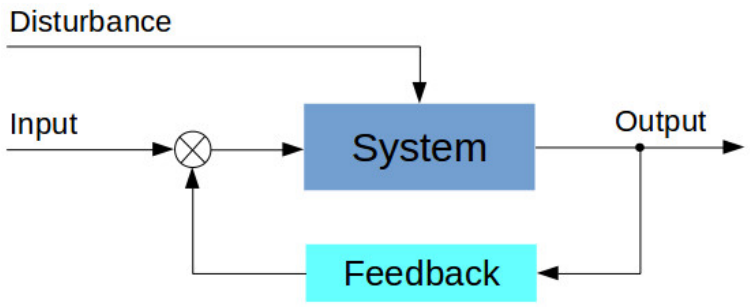


Figure 1.1. Closed loop feedback diagram. Source: [3].

While feedback is a mechanism by which most systems achieve the desired output, there are numerous ways that the feedback is processed in order to minimize error. Proportional control describes a feedback process in which the change in input is directly proportional

to the difference between output and desired state. Proportional control is the simplest to implement, however, it results in a finite steady state error for a system with no integrators. Since the change in input due to error is proportional to the error, the input decreases proportional to the decrease in error. In order to regulate the steady state error to zero, a proportional-integral controller proportional integral (PI) is utilized. A PI controller relies on cumulative (integral) error and instantaneous (reactive proportional) error, vice a proportional controller which only considers instantaneous error. A PI controller is especially useful when tracking a "reference point"—a constant or a variable command to follow. If an integrator is attached to a system, and the integrator relies on the error between output and the reference value, the error will tend to zero over time; classical feedback control theory calls it "at steady state." A proportional integral derivative (PID) controller adds a derivative element to the feedback process. In doing so, the slope of the error that captures the short-term forecast is considered. PID controllers are useful to regulate system overshoot and regulate control effort. Figure 1.2 illustrates the control scheme of a PID controller.

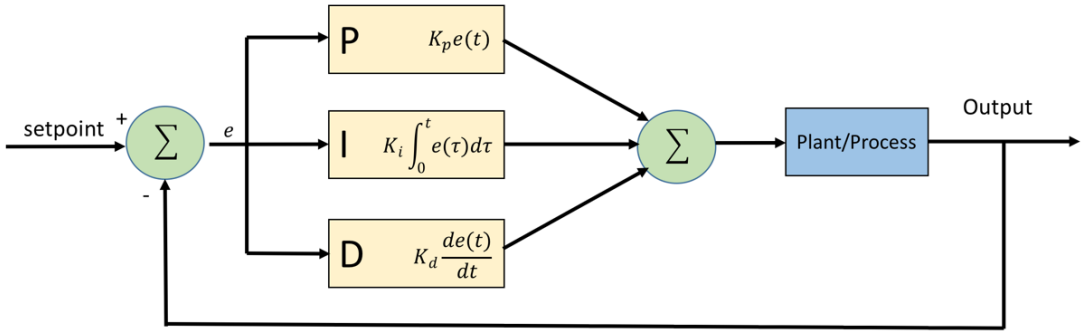


Figure 1.2. PID control architecture. Source: [4].

The robustness of a PID controller applied to a linear time invariant (LTI) system has led to their widespread use across industries and disciplines. PID controllers tailor their performance via their control gains, seen in Figure 1.2 as K_p, K_i, and K_d. Controller gains dictate the performance metrics of the system (e.g., rise time, percent overshoot, settling time) and therefore allow application across systems, requiring only the gains to be tuned for a specific performance requirement; although the performance criteria can be made more complex by a convolution of multiple comparable metrics. Despite the simplicity of a PID controller, tuning methods vary in methodology as well as complexity. Several

heuristic tuning methods were once standard, that involved modifying the gains manually and relying on empirical data to reach a desired performance. Many modern systems rely upon self-tuning algorithms in order to achieve a prescribed performance: they include the simplest self-tuning regulators and more complex adaptive control systems. Many of the self-tuning approaches "excite" a mode of a system and measure the system response. From the system response, the real-time learning algorithms rapidly identify what gain adjustment is necessary to achieve the desired performance. Some of the commercial autopilot units use a form of self-tuning algorithms in order to realize required performance. Limitations of the PID controller include that they have no knowledge of the changing system dynamics, and therefore cannot perform optimally across the entire life span of the system. In aerospace applications, PID algorithms are often paired with a form of feed-forward control, which relies upon a mathematical model of the system to estimate dynamics. PID controllers can be paired with a myriad of other protocols, such as gain scheduling, or adaptive control.

1.3 The Trade Space

Despite the numerous methods of designing automatic controllers for the class of LTI systems the ultimate goal is always to find the best architecture (the combination of the available sensors along with the elementary PID loops and their placement within the system) and the specific values of the PID gains that achieve the desired performance specifications. While the design of control system architecture is primarily based on an understanding of the fundamental physics of the system and the availability of signals for measurements, the tuning of control gains is a "well-automated" task; a variety of "semi-manual" and fully automated methods have been developed over the last few decades. Regardless of the approach, tuning for the best achievable performance always represents a challenge where the desired objectives conflict with the fundamental limitations within the system. The inherent compromise in design is quantified in what is typically referred to as the trade space.

Currently, the trade space is awash with commercially available guidance and control units, which are capable and robust out of the box. In order to understand the trade space, it is important to understand the constituent components that combine to enable autonomous flight. An often-overlooked delineation is that between a flight controller and an autopilot. A flight controller only implies system stabilization, and implements the aforementioned

control loop in order to achieve the input commanded by a pilot. While a flight controller is a component on an autopilot, a system with only a flight controller does not imply autonomous operation. An autopilot implies that the composite system contains a flight controller, the requisite sensors, and a guidance component that can traverse a predefined flight plan. Colloquially, flight controller and autopilot may be used interchangeably, but it is understood that an autopilot implies navigation while a flight controller does not.

When evaluating the parameters that drive autopilot selection, cost, weight, resolution of sensors, ease of augmentation, and user interface are all to be considered. Cost is a factor when considering the perspective of a potential adversary who would endeavor to use a UAV maliciously. Currently, COTS autopilots sell in the range of thirty to three hundred dollars. In the cost arena, COTS units have a considerable advantage over traditional military grade units. The only conceivable instance in which cost of an autopilot would be prohibitive is in a large swarm scenario; the avionics cost of a swarm is a sum of the cost of multiple UAVs.

Weight is a parameter to be considered when selecting an autopilot, as current autonomous systems are limited in range owing to poor energy density. For an adversary considering an autopilot for use in a weaponised drone, the weight of a heavier autopilot may not be justifiable, owing to the penalty in range, endurance, and payload that will be incurred. Rotorcraft are especially sensitive to increases in weight, thus the weight fraction of the autopilot should be minimized.

Sensor resolution is very important when considering an autonomous system. Using a simple PID controller, which implies that the controller has no knowledge of the plant, resolution of sensors is very important. Every signal has a measure of noise, and a PID controller has to be robust enough in order to operate. By reducing noise in measurements, PID controllers are able to operate with a higher degree of accuracy and precision. Additional measures can be taken to mitigate noise in measurements, such as employing a Kalman Filter, but regardless of the controller, it is advantageous to reduce the amount of noise in measurements.

Many commercially available autopilot units are “open source,” meaning that the source code of the autopilot, its schematics and drawings, as well as the ground control software, can be edited by the user. Employing an “open source” autopilot has distinct advantages and disadvantages, which are the double-edged sword of complexity. To a user who is

familiar with programming and control theory, the flexibility that an "open source" package provides is unparalleled. For example, additional control channels could be added into the autopilot software which, if utilized, could greatly improve system performance. However, augmenting an existing autopilots software is no trivial task, with many prerequisite skills. If a user is unacquainted with control theory or programming, it would be a disadvantage to utilize such a bare bones approach to flying a UAV.

While some may find editing the source code of an autopilot overwhelming, or very useful, the user interface (UI) of the ground station ought to be user friendly, regardless. An autopilot relies on a unit known as a ground control station (GCS), for mission planning, to display telemetry data, and to initialize auxiliary equipment. Typically the GCS exists on a laptop, smartphone, or tablet, and is the link between the user and the vehicle. For the purposes of this study, the UI was judged relative to a user who has a low level of familiarity with UAV utilization.

1.3.1 Piccolo Autopilot

The Piccolo Autopilot, produced by Cloud Cap Technologies, is an autopilot currently in use by the United States Military. For the purposes of this study, the Piccolo Autopilot serves as a benchmark for how a military grade system ought to perform, as well as a familiarization with the various flight simulations. Among others, the Piccolo Autopilot is used in the Northrup Grumman Bat and Arcturus T-20 [5]. Figure 1.3 shows the Piccolo family of autopilots.

The Piccolo unit evaluated was the Piccolo Plus model. The Piccolo Plus includes a ultra-

Figure 1.3. The Piccolo family of autopilots. Source: [6].

high frequency (UHF) antenna for communication with a desktop ground station (DGS) or controller, a pitot static system, including a pitot and static port, and a global positioning system (GPS) antenna. The DGS is shown in Figure 1.4.

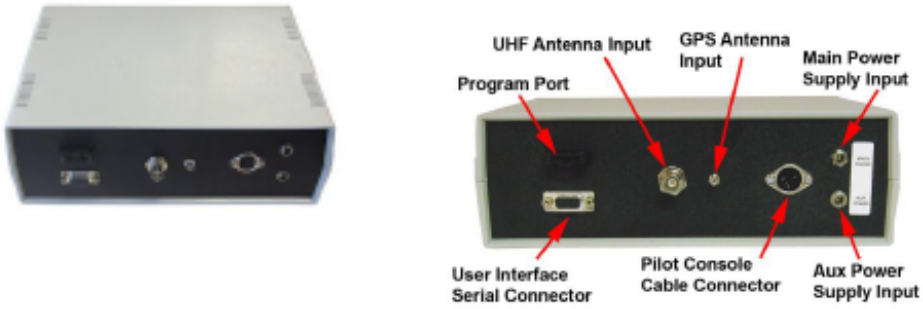


Figure 1.4. The Piccolo desktop ground station. Source: [6].

The DGS interfaces with the user via a serial/ethernet communication line to a computer running the Piccolo Command Center (PCC) software [6]. The PCC software was evaluated by running a hardware in the loop (HIL) simulation. Figure 1.5 details the setup of a HIL simulation, and Figure 1.6 shows the experimental setup.

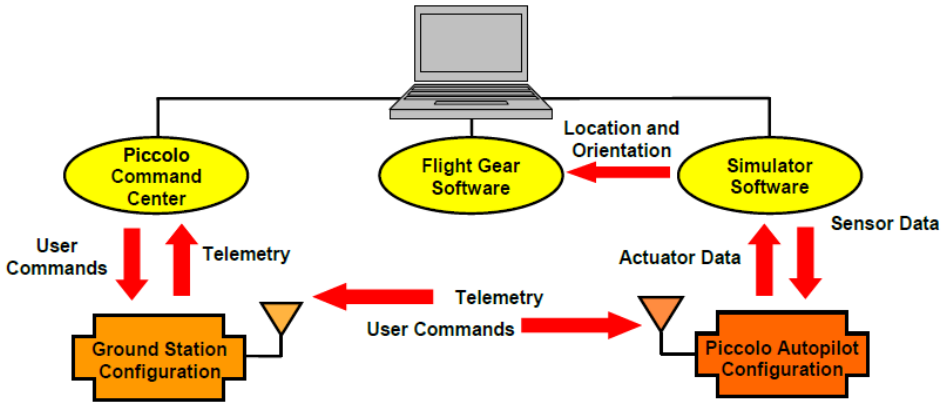


Figure 1.5. Piccolo hardware in the loop simulation diagram. Source: [6].

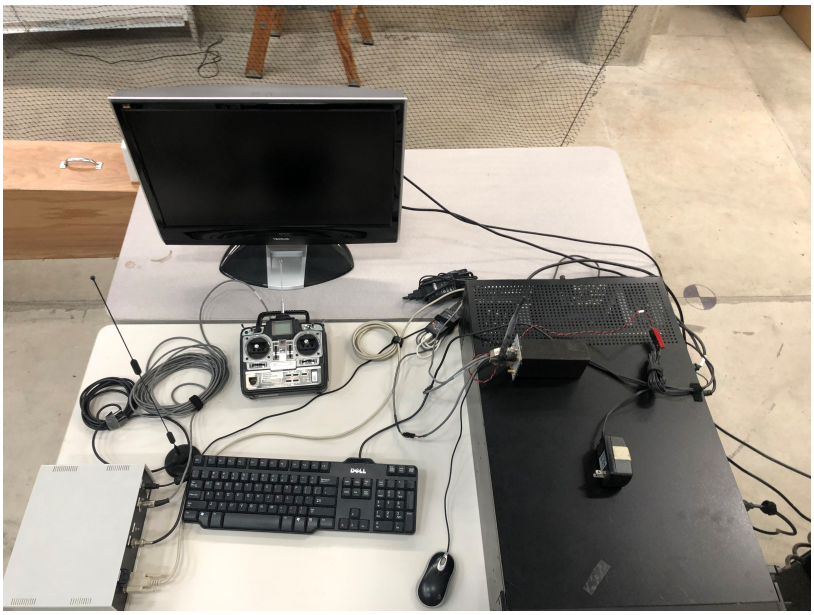


Figure 1.6. Piccolo hardware in the loop simulation.

The Piccolo autopilot undergoes HIL simulation via simulator software, noted in Figure 1.5. The simulator software is a six degrees-of-freedom (DOF) model of the aircraft dynamics. It is important to note that while the Piccolo autopilot unit does not necessarily require knowledge of the system to operate, the simulator software utilizes a mathematical model of the UAV to estimate its dynamics. The simulator software receives actuator data from the autopilot unit, and uses the actuator data as an input into the dynamics model. Computing the effect of an actuator movement, the simulator software returns data that sensors understand to be the flight condition. Effectively, the autopilot unit "believes" that it is flying, despite being on the ground. The Piccolo autopilot communicates with the DGS unit via its UHF antenna. The transmitted data from the autopilot includes altitude, airspeed, attitude, and position. The DGS receives this telemetry data, and presents it to the user via the PCC software. Similarly, the user is able to issue commands to the autopilot via the PCC software, which is sent to the autopilot via the DGS.

1.3.2 Autopilot Selection

When considering an autopilot for use in the B-52, several parameters were considered. As mentioned previously, cost, weight, resolution of sensors, ease of augmentation, and user

interface are all fundamental standards that must be considered when choosing an autopilot. For the purposes of selecting a suitable autopilot for the B-52, the criteria were specified further. The cost and weight parameters were defined as relative to the Piccolo II autopilot. Quality of sensors was considered as the quality of built in, or included sensors. While all autopilots considered can be improved with higher quality or additional sensors, out of the box performance was the priority. Augmentability speaks to the ability of the user to change system parameters to impact performance. UI and sophistication encompass both the quality of the user interface of the ground station software, and also the prerequisite knowledge necessary to initialize a system. For example, a lack of, or poorly compiled, documentation constituted a decrease in the sophistication score. Lastly, computing speed was considered. A more powerful processor is advantageous in the B-52 application, as it frees up computing power for auxiliary processes. The following autopilots were considered:

- Pixhawk
- Snap Dragon
- Navio2
- Beaglebone Blue

A trade study comparing the merits of each system was conducted, with the results shown in Figure 1.7. The decision-making matrix includes the previously mentioned parameters, which were assigned values relative to the most capable in each category. For example, the Beaglebone Blue scores the highest in cost, as it was the least expensive, and thus provides the benchmark against which to score the other options.

Since all of the autopilots considered, with the exception of Piccolo as the exemplar, are based on the ArduPilot software [7] , they all scored similarly in user interface, but varied significantly in sophistication. As seen in Figure 1.7, the Pixhawk was the most well rounded autopilot, with little compromise across the decision-making matrix.

Restricting autopilot consideration to the ArduPilot software package was a binary measure taken to simplify the decision-making process. ArduPilot is one of the most widely used open source autopilot software systems, and therefore has a vast library of widely available and tested documentation. Additionally, PX4, from the same developers as ArduPilot, was considered. ArduPilot is the more robust of the packages, as PX4 is mainly a developer tool. From a technological maturity standpoint, ArduPilot stands out due to the depth and

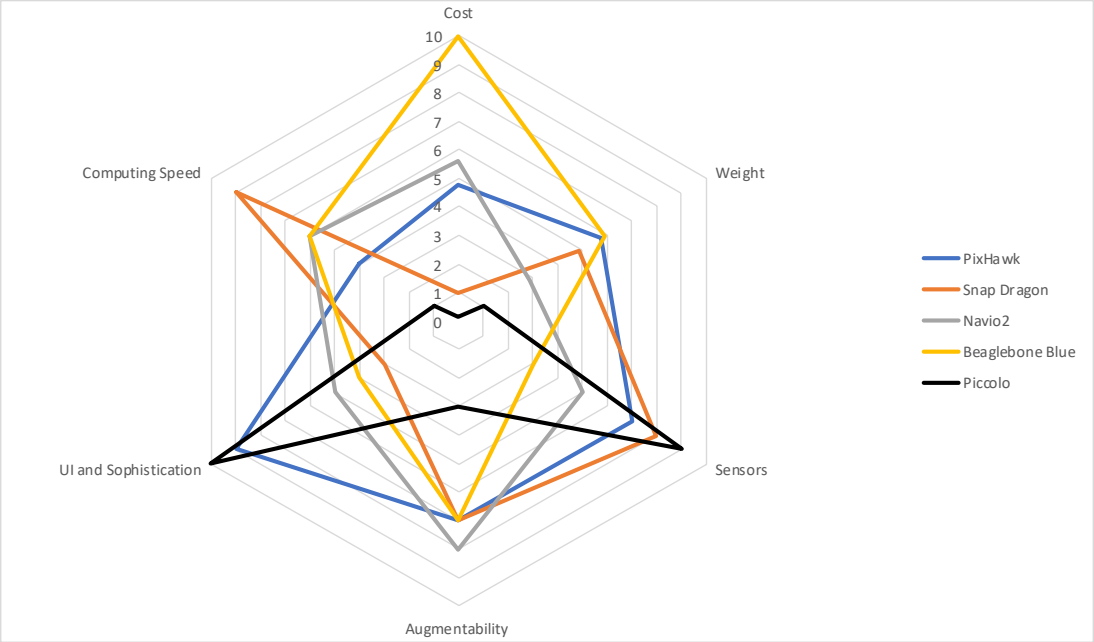


Figure 1.7. Autopilot selection study.

breadth of documentation and developer support, which is a weakness of the PX4 software. The ArduPilot autopilot framework is not specific to one autopilot unit. Rather, ArduPilot is a software suite that is run on a diverse number of units, and is not limited to fixed wing aerial vehicles. ArduPilot it is also capable of controlling:

- Multirotors
- VTOL aircraft
- Helicopters
- Ground rovers
- Submarines
- Boats

The Pixhawk combines a 32-bit processor, with a 32-bit failsafe co-processor, a MPU6000 accelerometer which samples at 8kHz, a 16-bit gyroscope, a 14-bit magnetometer, and a barometer. Figure 1.8 shows the setup and ports found on a Pixhawk unit.

There exists many open source ground control station for the Pixhawk unit, such as QGroundControl (QGC), Mission Planner, APM Planner, and MAVProxy. QGC was be used as a ground control station for the Pixhawk unit. QGC is an open source software

Figure 1.8. Pixhawk port diagram. Source: [8].

that is compatible with a variety of vehicle configuration. The flexibility of QGC extends to the platforms on which it can run, which include Windows, OS X, Linux, iOS, and Android. Additionally, it provides user friendly waypoint management, and the ability to log and replay telemetry after a mission. Figure 1.9 displays the QGC waypoint planning interface. QGC is an interface built around the MAVLink protocol, which supports PX4 Pro, ArduPilot, or any software which uses MAVLink protocol. In this study, which focuses on the capabilities of COTS units, the MAVLink focus of QGC has the benefit of easing the implementation of software in the loop (SIL) simulation.

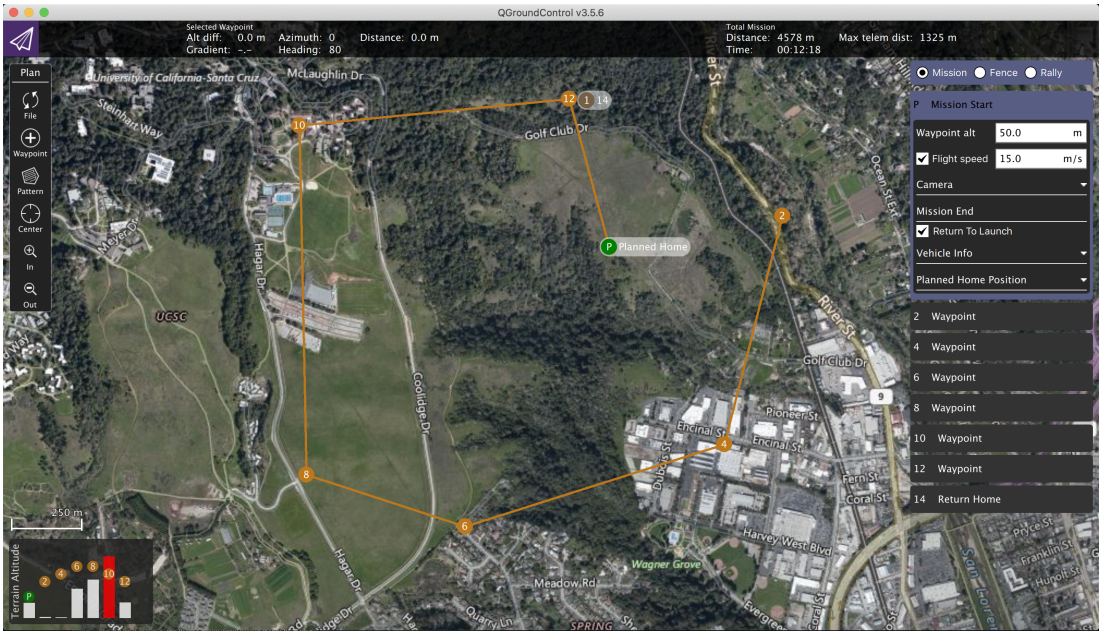


Figure 1.9. QGroundControl waypoint planning interface.

To evaluate the performance of the aircraft and autopilot, in a safe environment, SIL simulation was utilized. SIL simulation can be considered an intermediate step between conceptual design, and hardware testing. SIL simulation relies upon a computer which simulates the dynamics of a vehicle, a controller, which executes commands necessary to accomplish a given mission, and a GCS, which serves as a visualization tool for the mission. Figure 1.10 shows the workflow of SIL simulation.

In the context of the COTS autopilot study, PC1 was implemented by a Simulink mathematical model. PC2, the control element of the simulation, was represented as both a Simulink diagram and the commands delivered by the ground station. PC3 represents the GCS, as viewed by the user during a mission. During a simulation, the effective user interface was restricted to the GCS, from which the user can specify commands to be implemented at the level of PC2, and drive the dynamics of the plant on the PC1 level.

1.4 Study Methodology

In many respects evaluating the suitability of a COTS autopilot to military specifications follows the logic of prototyping an aerial vehicle, from conceptual design to simulation.

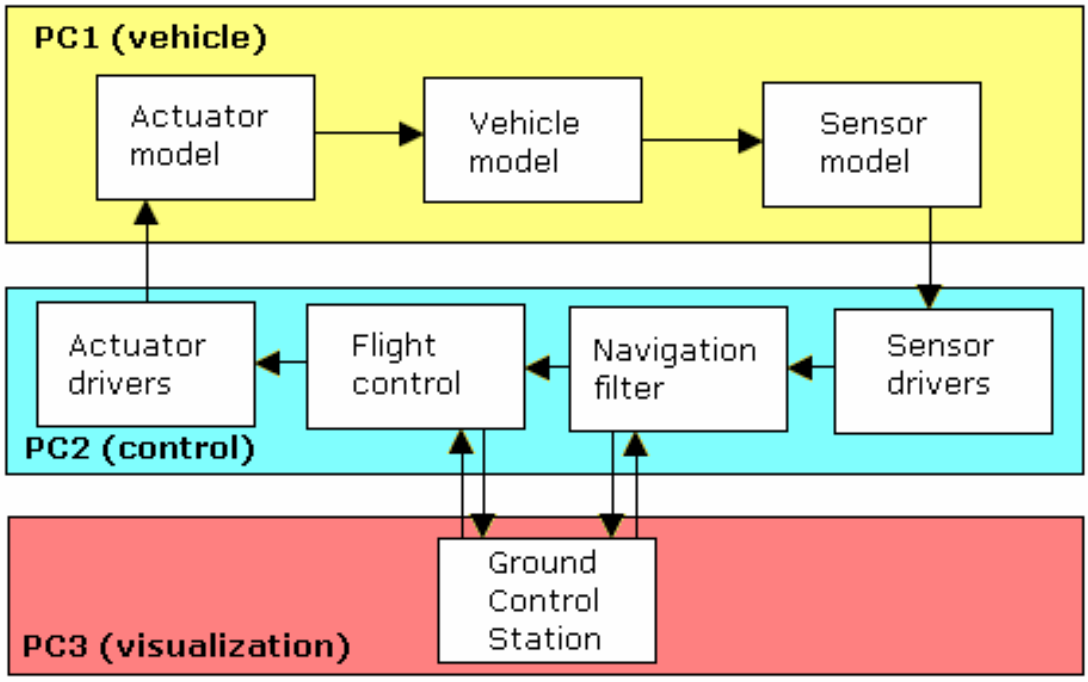


Figure 1.10. Software in the loop simulation workflow. Source: [9].

Figure 1.11 describes the steps taken to evaluate the COTS unit.

The first step in investigating the suitability of a COTS autopilot unit for an aerial weapons delivery mission, as the model B-52 aims to do, is to understand the state of the art. In the context of this study that focuses on guidance, navigation, and control (GNC) components of the system, the concern with state of the art encompasses those parameters that can be tuned in order to achieve a desired mil-grade flight and mission performance level. Hardware, sensors, and navigation algorithms were investigated in order to determine what is representative of a COTS unit, how it differs from a unit utilized in a military mission, and how each constituent contributes to the system.

Next, a model of the SUT was developed. In the case of an aerial vehicle, estimating the dynamics of the system involves an aerodynamic analysis. In the design of the B-52 model for evaluation, the result of an aerodynamic analysis were flight stability and performance derivatives, which are used to initialize a model of the SUT dynamics.

Following aerodynamic analysis, and verification that the aircraft under control exhibits expected behavior, next the architecture of the autopilot loops was established. A functional

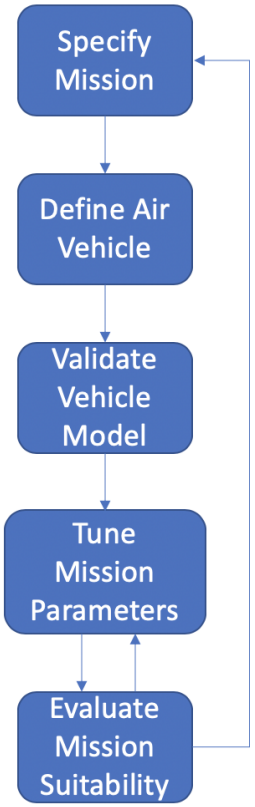


Figure 1.11. COTS system evaluation workflow.

model of the autopilot includes sensor models, navigation algorithms, and stabilization and reference tracking control loops. As mentioned previously, there exists a myriad of ways to tune a control system, yet the autopilot model ought to emulate the performance of the control, the architecture of the Pixhawk, and subsequently enable experimentation of different hardware, sensors, and navigation parameters.

The last step in the evaluation of the COTS autopilot unit involves evaluating the performance of the modeled plant and autopilot in a simulated mission, with a COTS GCS as the user interface. The tasks mentioned previously relied upon numerous computational tools, and Figure 1.12 describes the chronology of the research and the associated tool utilized for each task.

The study aims at establishing what augmentation to hardware, sensors, and algorithms, is

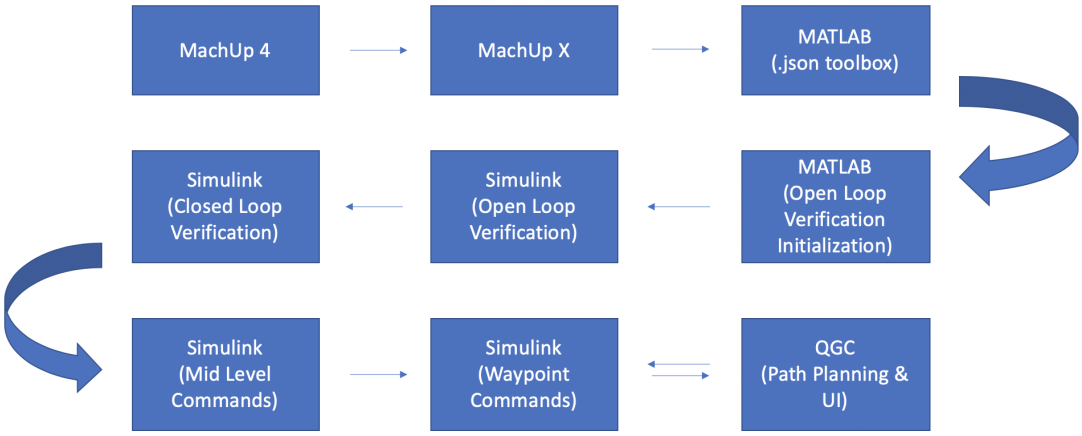


Figure 1.12. COTS system evaluation tools.

necessary to achieve a level of performance expected of a military specification, and what is the associated level of knowledge prerequisite to that implementation. Success in the study is defined as mission success by the UAV, requiring primitive understanding of the system necessary by a user, beyond the GCS UI.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Task Formulation

The purpose of this chapter is to qualitatively describe the constituent components of an autopilot unit. The components discussed include the central processing unit (CPU), sensors, and navigation algorithms. Comparisons between components found in a COTS unit and those found in a military UAV autopilot seek to clarify the prerequisite question of what differentiates a COTS unit from those used in a military application. This approach allowed for identification of potential parameters to be varied in future testing. Understanding the differences in components will scope the approach of answering the research question, which seeks to define the technical and expertise requirements to emulate military performance from an inexpensive and readily available COTS autopilot.

2.1 Task Formulation Methodology

In order to define performance to a military specification, it is important to understand the expected employment of a weaponized COTS system. Though many applications of drones being used by an insurgent or terrorist organization constitute single use weapons, which detonate in some proximity to a target, it is conceivable that the next evolution of malicious drone utilization will migrate from the expendable domain. Dr. Robert J. Bunker, Futurist in Residence, Behavioral Research and Instruction Unit, FBI Academy, places the insurgent drone threat into three distinct categories: tactical, operational, and strategic [10]. Dr. Bunker describes tactical use of a UAV as a single vehicle employing an Improvised Explosive Device (IED) against a crowd, or attempting to damage a commercial airplane. The threat presented by a tactical malicious UAV is equally present both from domestic terrorists targeting civilians or insurgents overseas. The operational UAV threat is considered to be human controlled and semiautonomous vehicles, which consists of, "various types of flying IED, weapons platform[s], and human insurgent fighter combined arms' hybrid threats" [10]. Because the prerequisites exist for the operational UAV threat, Dr. Bunker classifies the threat to be imminent in the near future. The last category of drone threat is the strategic threat, which consists of "normal and micro sized UAVs" [10]. Dr. Bunker compares the present to the interwar years, between World War I and II, which

led to the vast mechanized formations encountered in the latter. It is the aim of this study to anticipate and understand the sophistication (technology and the expertise) required to implement a UAV which could be utilized in an operational manner.

In order to understand the challenges associated with implementing a UAV for a malicious operational purpose, a concept of operations (CONOPS) was developed. The CONOPS considers a single UAV, launched from a range beyond the visual line of sight (VLOS) from a target. The UAV will travel to the target using GPS-based waypoint navigation, en route to deliver an explosive air launched device. The figure of merit for the study will be the UAV position and velocity error over the CARP that is calculated with respect to the target and the weather conditions. Figure 2.1 depicts a possible profile in which the UAV flies a waypoint mission en-route to the CARP. In Figure 2.1, the red sphere represents a theoretical tolerance (exaggerated to 80 meters in Figure 2.1 for scale) in the launch conditions, within which is considered a hit. Assuming unguided ballistic munitions as the most challenging case, the miss distance of the projectile is dependent upon the states of the launch platform with respect to the CARP. The velocity and position errors of the Pixhawk autopilot will be evaluated in simulation, and comparison with military specification will be enabled by changing the hardware and software model.

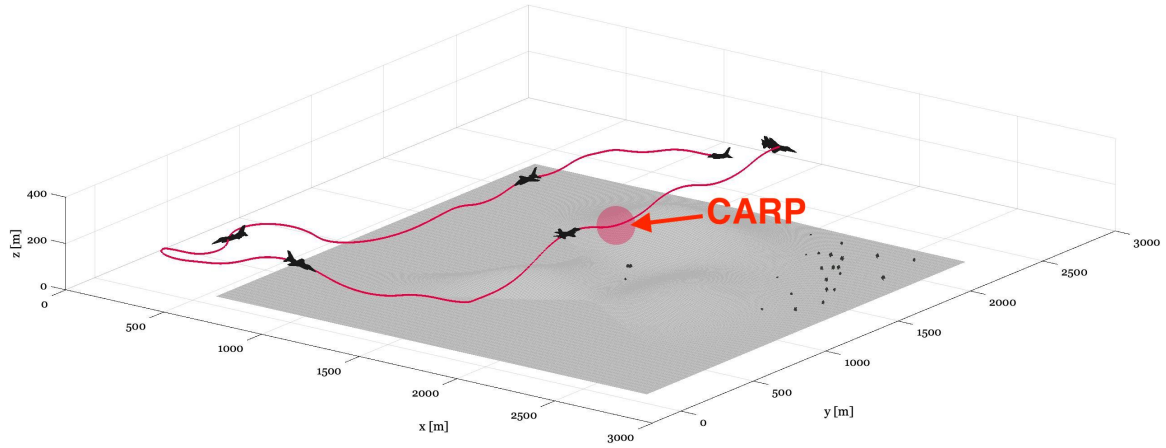


Figure 2.1. Concept of operations for B-52 weapons delivery mission.

2.2 Hardware Augmentation

In order to augment the performance of a UAV carrying out an ordinance delivery mission, several hardware models will be considered. The onboard instrumentation to be considered

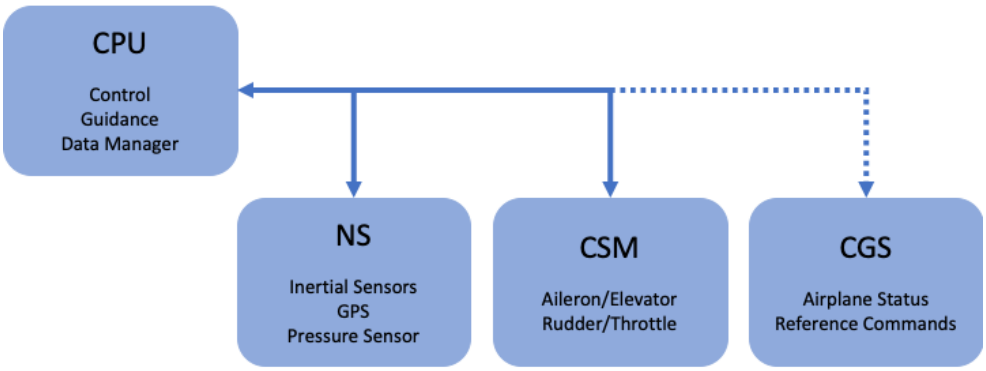


Figure 2.2. Diagram of CPU interaction with input/output devices. Source: [11].

are the CPU and sensors.

2.2.1 Central Processing Unit

The responsibility of the CPU is to receive and manage the data coming from the sensors (input), transforming sensor readings to the desired controls, which are sent to the actuators (output) as commands [11]. Figure 2.2 displays the routing of messages to and from input and output devices through the CPU.

The CPU of the Pixhawk 1 is a 32-bit STM32F427 Cortex-M4F microcontroller with a floating point unit (FPU). A FPU is a complementary coprocessor, also known as a math coprocessor, which is optimized for fast execution of mathematic operations with floating point numbers. In addition to the main CPU, the Pixhawk 1 employs a failsafe co-processor, should the main processor fail. The addition of a co-processor to an autopilot is increasingly common in COTS units, and increases reliability and robustness of the package. For reference, the Piccolo autopilot uses a variation of the Motorola MPC555 Microcontroller [12]. The MPC555 microcontroller represents the state of the art in small form factor computing technology circa the early 2000s. Similar to the STM32F427 Cortex-M4F, the MPC555 also is a 32-bit processor, with a FPU.

A figure of merit in evaluating a CPU is the clock speed of the processor. The CPU of the Pixhawk 1 operates at 168 MHz. The CPU clock speed of a specific platform must be tailored to the sort of processing it will be tasked with. Many contemporary UAV systems are employing on board image processing, for example, which requires a faster clock speed than one which does not. The MPC555 CPU operates at a clock speed of 40MHz, approximately a fourth the speed of the Pixhawk 1. The significant advantage the Pixhawk 1 has in clock speed speaks to the high quality of COTS autopilots, and additionally highlights the democratization of small, inexpensive processing power.

Another specification important to UAV operation is CPU cache memory; a limited memory space with an order of magnitude faster access speed that is used to enable in-place vectorized computations, as well as high speed IO operations with the external sensors and peripherals.

Though a CPU can carry out computations extremely quickly, it is limited by how fast it can access queried values from memory. In order to decrease the lookup time, a memory cache is utilized. The MPC555 incorporates 26 KB of static random-access memory (SRAM), whereas the Pixhawk 1 utilizes 256 KB SRAM (level 1), for its quick access memory. Level 1 memory resides closest to the CPU, for quick access. The downside of level 1 SRAM, is its volatility. When the power to the SRAM is lost, so is the data stored there. Therefore, information saved in this registry is not meant for longevity, and if it is to be stored must be written to a less volatile form of memory. Consequently, the amount of memory allocated for the SRAM ought to be enough to store pertinent data sufficiently long, but an excess of SRAM drives up cost unnecessarily.

The comparison with the Piccolo autopilot CPU, the MPC555, leads to the conclusion that relatively tame computing power, by modern standards, is sufficient to achieve the precision necessary for a military tasking. Therefore, given that the CPU and SRAM are sufficient for the input and output devices compatible with the Pixhawk 1, it is expected that should modifications be made to the resolution of sensors, the processing hardware need not be augmented.

2.2.2 GPS

GPS is instrumental in providing accurate position and groundspeed data to a UAV. GPS relies on a constellation of satellite, each with an atomic clock. By measuring the time it

takes for a signal to travel from the satellite to the receiver, the distance to the satellite is calculated. Using multiple (at least four) satellites to acquire a fix, it is possible to accurately calculate latitude, longitude, and height. GPS-based states are also fundamental in a UAV because they provide a reference off which other sensors can calibrate or the overall state estimation errors can be improved: the idea is typically called a complementary filtering of multiple heterogeneous sensors. While certain sensors may accumulate errors over time, GPS does not; therefore, the effects of accumulated sensor errors can be mitigated by refreshing their knowledge of vehicle states at a certain interval. The interval at which the GPS received these signals is known as the maximum navigation update rate. The Piccolo autopilot utilizes a GPS that updates at 4 Hz [13], whereas the Pixhawk GPS receivers update at 10 Hz. However, like any signal, a GPS signal is susceptible to corruption by noise. Measurement noise in a GPS system comes primarily from hardware internal sources and the propagation of signals through an unsteady atmosphere (see [14] for more details).

Internal thermal noise is due to the random movement of electrons in a conductor when the temperature is above absolute zero. Thermal noise is generated across broad frequencies, and takes the form of white noise, in practice. The movement of electrons causes a slight fluctuation in voltage about the mean value. When the power is calculated, however, the integral of the square of the voltage is non-negligible. Thermal, or Johnson noise, is given as

$$p = kTB, \tag{2.1}$$

where p is the thermal noise power, k is the Boltzmann constant, T is the temperature in Kelvin, and B is the bandwidth in hertz [14]. The Pixhawk-1 is compatible with the u-Blox 7 and 8 GPS units [15]. The navigation sensitivity, or minimum received power for a fix, for each is given as -162 and -167 dBm, respectively. Converting the navigation sensitivities to Watts, arrives at an estimated signal power at the receiver end of $6.3096x10^{-20}$ and $1.9953x10^{-20}$ Watts; in fact this high sensitivity has an obvious drawback that a nearby jammer with very limited power ($\ll 100mW$) can wipe out GPS receivers in a radius of few kilometers. Calculating the thermal power at 290 Kelvin, bandwidths ranging from $1x10^2$ to $1x10^6$ Hz, the signal to noise ratio was calculated. Figure 2.3 displays the difference in

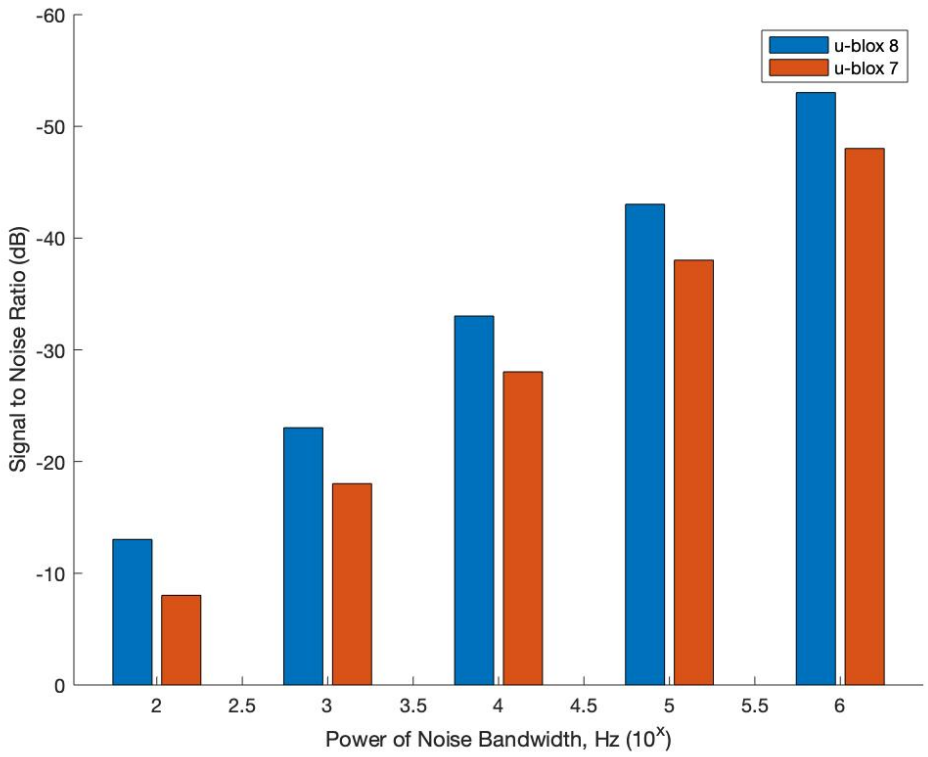


Figure 2.3. Signal to noise ratio of minimum receiver sensitivity to thermal noise.

signal to noise ratio between the u-Blox 7 and 8, at the minimum GPS signal strength.

Despite the u-Blox 8 receiver being three times more sensitive to the GPS signal than the u-Blox 7 receiver, both systems have a negative power of the signal to noise ratio in dB across a wide bandwidth spectrum. Therefore, receiver sensitivity cannot compensate entirely for thermal noise.

Another type of noise, antenna noise, is a result of ambient electromagnetic radiation being absorbed by the antenna. One such type of noise is "sky noise" [14]. "Sky noise" is the effect of radiation from the Sun, the galaxy, and other cosmic bodies, as well as radiation caused by the Earth's atmosphere. However, "sky noise" is considered negligible because of the bandwidth that the radiation is observed at. Because the bandwidth of sky noise is sufficiently different than the GPS carrier frequency, it can be neglected.

The last form of GPS noise considered is system noise. System noise is not a discrete form of noise, but rather the aggregate noise from external sources, and internal mechanisms such as cable loss or receiver temperature. A figure of merit for GPS systems if the noise factor, given by

$$F = N_{out}/GkT_0B, \tag{2.2}$$

where N_{out} is output noise power, G is receiver gain, B is noise bandwidth, and an effective noise temperature, T_0, which relates noise intensity [14]. Implementation of the noise factor allows for modeling of different qualities of GPS receivers. By simulating noise as white noise of differencing power levels, it is possible to understand the effect that a GPS receiver has on the position and velocity error of the UAV in a simulated ordinance delivery mission.

2.2.3 Accelerometers, Gyroscopes, and Inertial Measurement Units

The Pixhawk-1 is equipped with an ST Micro L3GD20H, 16-bit gyroscope, a LSM303D, 14-bit accelerometer/magnetometer, and an Invensense MPU 6000 3-axis accelerometer/gyroscope. Before describing the metrics of quality of accelerometers, gyroscopes, and inertial measurement unit (IMU)s, it is important to understand the difference between them. An accelerometer, as the name suggests, is used to measure accelerations. A modern COTS accelerometer is typically a piezoelectric or piezoresistive device. When a force is applied to a piezoelectric, a voltage is induced. By calibrating a piezoelectric materials voltage response to a force, it is possible to estimate an applied force due to the induced voltage. Figure 2.4 displays how a piezoelectric material creates a potential difference in a circuit.

When multiple piezoelectric circuits are oriented in an orthogonal manner, such that they measure forces in three orthogonal directions, the combined system is known as a 3-axis accelerometer. A gyroscope operates in much the same way, but instead of measuring forces, the gyroscope measures angular rates. Combining force measurements obtained from the accelerometers, and angular rate estimation from the gyroscopes, it is possible to integrate the body rates in order to dead-reckon position, given an initial absolute reference. Figure

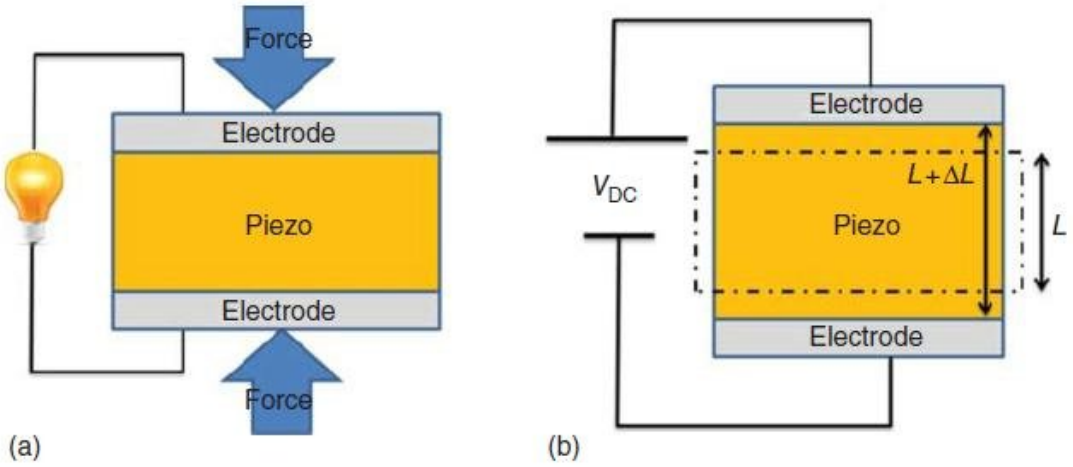


Figure 2.4. Force inducing a voltage in a piezoelectric material. Source: [16].

2.5 represents the modern concept of the position estimation achieved using a combination of gyroscopes and accelerometers.

As seen in Figure 2.5, the estimation of position relies upon an integration operation. Therefore, noise in sensors' measurements is accumulated over time. The resulting phenomenon is known as drift of navigation system. In order to overcome drift, measurements are often multiplexed via various forms of estimation algorithms with the Kalman Filter being the most widely used. A Kalman Filter relies upon an approximate model of the system, in order to estimate its states; in the classical case it is the linear model. Figure 2.6 describes

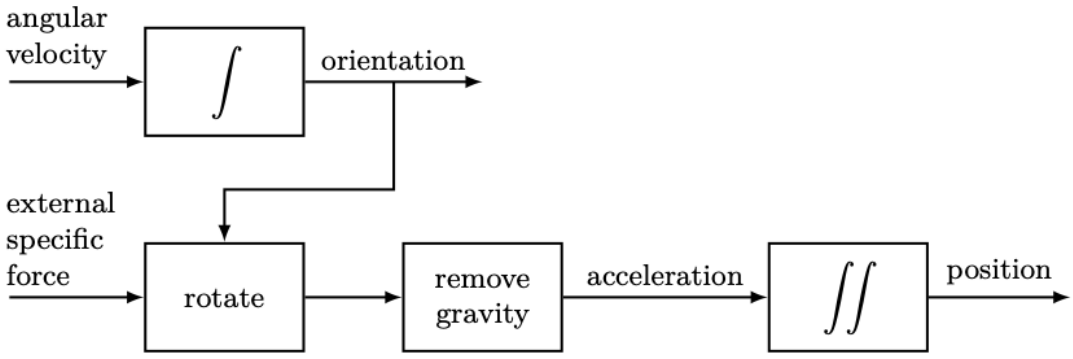


Figure 2.5. Position estimation from rate measurements. Source: [17].

the concept of a Kalman Filter mechanization.

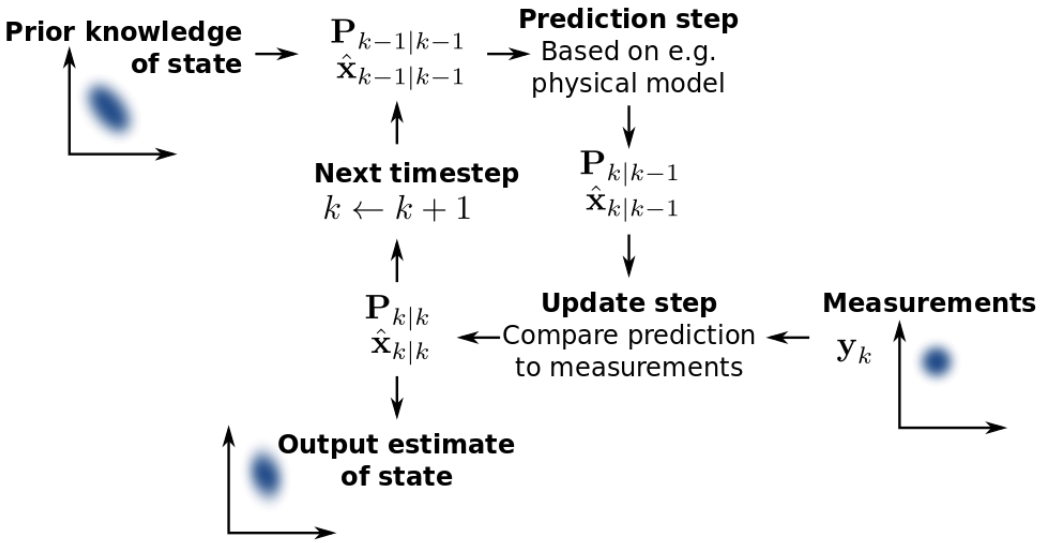


Figure 2.6. Kalman filter conceptual diagram. Source: [18].

Because a Kalman Filter relies upon a model of the system, the position estimation is not impervious to noise. However, the Kalman mechanization allows for "multiplexing" of multiple sensors with different levels of noise in measurements. As a result, one of the benefits of Kalman filtering is the achievement of the lowest error at the state estimation that is characterized by the corresponding noise in the measurements (i.e., the error in state estimation cannot be made smaller then the worst sensor in the system). Therefore, it is advantageous to minimize the noise in measurements, despite filtering, in order to reduce the magnitude of navigation drift.

An IMU combines an accelerometer, a gyroscope, and a magnetometer into a single unit, with a small form factor. A combination of an accelerometer and gyroscope provides sensor measurements in surge, sway, and heave translations, and yaw, pitch, and roll rotations. Six-DOF resolution is increased to 9-DOF, with the addition of a 3-axis magnetometer. The augmentation of an IMU with a magnetometer, in addition with utilization of a GPS, further compensates and isolates the system from individual sensor noise.

Therefore, the effective sensor fusion of a GPS and IMU system results in repeatability, subject to random noise in measurements, minimization of measurement drift and bias.

2.3 Navigation Algorithm

The navigation algorithm used by the ArduPilot software is based on the works of Park, Dryst, and How [19]. The navigation algorithm is known as an L1 path-following controller. The L1 controller projects a vector of length, L1, connecting the current position of a robot to a point along the desired trajectory, as shown in Figure 2.7. The angle between the velocity vector of the aircraft, and the L1 vector, along with the magnitude of the velocity, are used to compute the desired lateral acceleration that rotates the current path toward the desired.

$$a_{s_{cmd}} = 2\frac{V^2}{L_1}sin(\eta) \tag{2.3}$$

An advantage of L1 control is that when the cross-track error is large, the angular error between the velocity vector and L1 vector will be large, commanding a larger acceleration and rotating the velocity vector toward the desired track, at a steep angle of approach. Conversely, when the cross-track error is small, the angle will be shallow, and small corrections will be applied. In this respect, L1 navigation is analogous to proportional navigation [19]. One of the most important parameters in a L1 controller is the L1 vector length, or lookahead distance. The ArduPilot software aboard the Pixhawk dynamically computes the length of the L1 vector, such that the period of the guidance loop is constant [20]. Following the methodology presented in [19], it is proven that the L1 vector length impacts the system response significantly. Figure 2.7 shows a scenario of a UAV following a curved reference.

The corresponding cross-track error dynamics is described by the differential equation

$$\ddot{d} + \frac{2V}{L_1}\dot{d} + \frac{2V^2}{L_1^2}d = \frac{2V^2}{L_1^2}d^*_{ref.pt}. \tag{2.4}$$

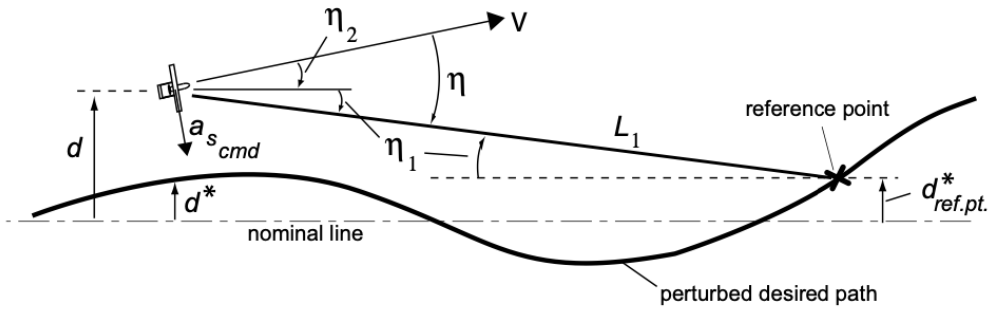


Figure 2.7. Unmanned aerial vehicle navigation model for curved path following. Source: [19].

Correspondingly, in the frequency domain:

$$\frac{d(s)}{d^*_{ref.pt}(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \tag{2.5}$$

where the ζ is the damping coefficient, and a degree of freedom for tuning, and the undamped natural frequency, ω_n, is $\frac{\sqrt{2}V}{L_1}$. Therefore, it is advantageous that the navigation algorithm dynamically computes the lookahead distance for consistent performance, further highlighting the quality of COTS units available today. The effect of the lookahead distance will be considered in the study of the B-52's effectiveness in achieving the CARP.

Another consideration in navigation is waypoint radius, or distance from a waypoint that a UAV initiates a turn to the next waypoint. The ArduPilot software defaults to the calculated L1 length to initiate a turn to the next waypoint, yet the radius is limited by a user specified value, which the UAV will not exceed. Given an appropriate distance to settle on a track, the L1 navigation algorithm is very effective at navigating to waypoints with little cross track error. However, if the CARP immediately follows or precedes a turn, the cross track error is only bounded by the waypoint radius. Therefore, when considering a CARP as a waypoint in a mission, path planning is essential.

2.4 Value Consideration of COTS Units

Comparing the technical specifications of the Piccolo to the Pixhawk autopilots yielded an interesting picture of the technology that exists in small military UAVs, and that which is freely available to the public. While the Piccolo is a proven unit which is trusted by the military to accomplish a myriad of tasks, a comparison of technical specifications implies that the Pixhawk represents a more modern and capable unit. There are a few conclusions to be drawn from this comparison; the first being the exemplary capability presented by a COTS unit, for very little cost. Noting that a quality COTS unit can be bought for a fraction of the cost of a Piccolo, or other military system, the implication exists that a COTS unit could meet or surpass the performance of a military system. However, when considering the apparent disparity in value between the classes of autopilots, the reason for the difference in price must be questioned. Robustness in operation is a quality of a system which is of vital import in military applications. The Piccolo has proven it can operate in adverse conditions, and that capability is what sets it apart from similar technologies. While the Pixhawk represents an advanced and dangerous level of capability for malicious actors given benign conditions, the robustness of the system given a electromagnetically denied space or adverse weather conditions is in question. Therefore, assessing the challenge presented in meeting or exceeding military performance given adverse conditions is truly necessary to understand the potential for adversary exploitation.

CHAPTER 3:
Modeling of the B-52

Navigation performance of the aerial system is contributed by a number of components. The flight performance of the aircraft is one of them. Therefore it is critical to enable high-fidelity modeling of the aircraft's flight dynamics. This chapter describes the methodology undertaken to initialize a model of the B-52 dynamics and GNC components, using MATLAB, Simulink, and QGC. The Simulink Drone Reference App is used as the framework for the subsequent modeling. Tuning of an inner loop autopilot, navigation algorithms, and sensors modeling are described, with the intent of creating a framework for definition of the technical and expertise requirements necessary to achieve military performance from a COTS autopilot unit.

3.1 Aerodynamic Analysis

The first step in developing a suitable model for evaluating the capabilities of a COTS autopilot, was to develop a sufficiently fidelity model of the plant. In the case of this research, the plant was the B-52 model, however a similar procedure can be followed for any type and class of aerial platform.

Typically, when considering a plant model of an aerial vehicle, the first consideration is an aerodynamic analysis. There are many options when considering how to approach an aerodynamic analysis. For some applications, manual estimation may be sufficient, whereas in the late stages of conceptual design for a safety critical or expensive application, a computational fluid dynamics (CFD) or wind tunnel approach may be appropriate. Regardless of the method, the objective of any such analysis is to understand the stability and dynamics (forces and moments) of the system, where a linearization technique is used most widely. Around an equilibrium, or trim point, and assuming perturbations from that point are small, the resulting forces and moments due to perturbations can be expressed as the first order term in the corresponding Taylor series expansion. The Taylor series expansion terms are known as force and moment coefficients, or flight dynamics derivatives. Furthermore, the derivative are further delineated into control, or response to an actuator command, stability, the tendency to converge or diverge under no control to an equilibrium

given incidence, and damping, the way in which excitations impact the system. All of the aforementioned traits provide a means to estimate with a high fidelity, operation of a system around the trim point. To estimate the key characteristics of the B-52 aerodynamics, a lifting line analysis was undertaken. For details on the lifting line approach used, see [21].

Predicating the lifting line analysis of the B-52, details of the lifting surfaces had to be identified. Because the B-52 model's main wing is a modified geometry, conventional airfoil databases were ineffective. Therefore, a CFD analysis of the main wing was necessary; the results of which can be found in [22]. The horizontal and vertical stabilizers were both assumed to be NACA 0010 sections, which were considered representative. With the parameters of the B-52 main wing computed, a software program called MachUp 4 [23] was utilized to model the geometry of the B-52. MachUp 4 operates in the HTML environment. Figure 3.1 shows the B-52 in the MachUp 4 environment.

Once the geometry of the B-52 was built in the MachUp 4 environment, the aerodynamic analysis was exported to a Python interface known as MachUpX [24]. MachUpX is a more robust package, offering advantages like asymmetric control deflections, nonlinear solvers, and simulation batching. Conversely, it requires prerequisite knowledge of Python programming language and basic Python scripting.

The main wing was divided into four distinct segments for analysis, in order to accurately model the location of the flaps and ailerons. Despite the B-52 utilizing spoilerons for roll control, in order to obtain an estimate of pitch and yaw stability, the configuration was simplified to asymmetric aileron deflection. Figure 3.2 displays the lifting surface placement

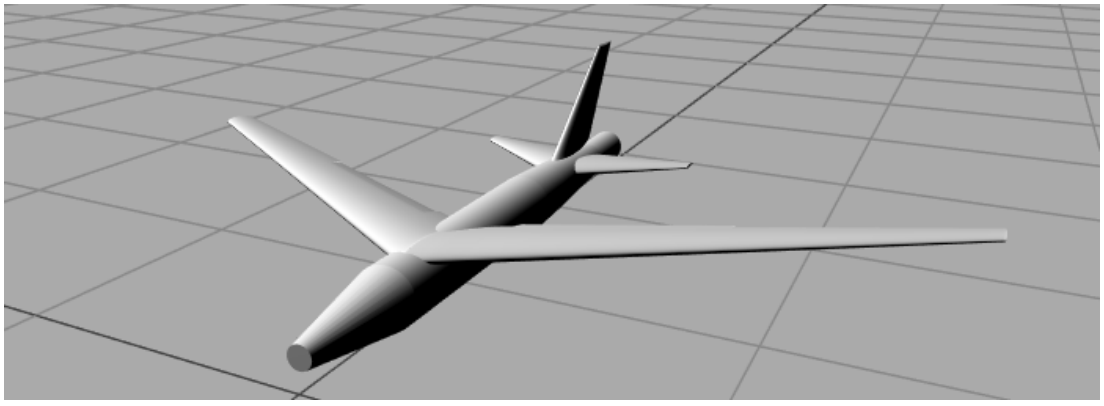


Figure 3.1. B-52 in MachUp 4 environment.

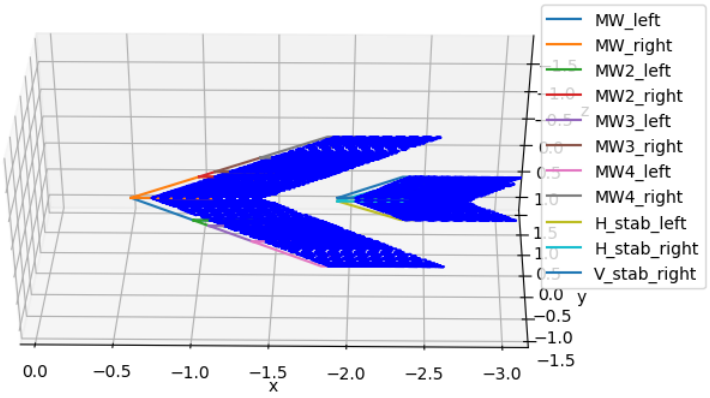


Figure 3.2. B-52 wireframe in MachUpX environment.

and the corresponding horseshoe vortex distribution. Each lifting surface utilized twenty spanwise vortices, for a total of 80 on the main wing.

The results of the aerodynamic analysis show that at 25 m/s (expected nominal airspeed of the model under development), the aircraft can trim with a center of gravity (CG) located at 25% of the mean aerodynamic chord (MAC) with zero elevator deflection. CG location is generally expressed as a percentage of the MAC, and zero elevator deflection with a CG situated at 25% MAC builds confidence in the model. The pitch stability and trim state are notable for the conceptual design phase, as takeoff and steady, wings level flight will be one of the initial milestones for the B-52.

3.2 Open Loop System Response

3.2.1 Longitudinal Channel

MachUpX outputs the results of the aerodynamic evaluation in the form of a JSON file, which allows for powerful implementation in MATLAB [25]. JSON is a modern file format used to conveniently integrate structured data into numerical algorithms. The force and moment coefficients, stability, damping, and control derivatives were read into a MATLAB script [26], with the intent of developing a state-space model for evaluation of the longitudinal dynamics. The state-space longitudinal model implemented is

$$\begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{x} \\ \dot{z} \\ \dot{\theta} \end{bmatrix} = A \begin{bmatrix} u \\ \alpha \\ q \\ x \\ z \\ \theta \end{bmatrix} + B\delta_e$$

where A and B are matrices containing the coefficients which characterize the dynamics of the vehicle [27]

$$A = \begin{bmatrix} R_{x,u} & R_{x,\alpha} & R_{x,q} & 0 & 0 & -R_{gx}cos(\theta_0) \\ R_{z,u} & R_{z,\alpha} & (R_{x,q}+1) & 0 & 0 & -R_{gx}sin(\theta_0) \\ R_{m,u} & R_{m,\alpha} & R_{m,q} & 0 & 0 & 0 \\ cos(\theta_0) & sin(\theta_0) & 0 & 0 & 0 & -sin(\theta_0) \\ -sin(\theta_0) & cos(\theta_0) & 0 & 0 & 0 & -cos(\theta_0) \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} R_{x,\delta e} \\ R_{z,\delta e} \\ R_{m,\delta e} \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The state-space representation of the system allows for an eigenvalue analysis, which will confirm expectations, and therefore build confidence in the plant model. In the longitudinal channel, the two dynamics modes for consideration are the phugoid and the short period mode. The phugoid mode is a dynamic response to a non-equilibrium pitch angle condition; it primarily involves large amplitude variations of pitch angle, airspeed, and altitude, and almost no angle of attack. Whereas the short period mode is the dynamic system response primarily of the angle of attack due to a disturbance. Figure 3.3 displays the phugoid and short period modes (roots of the characteristic equation) of the developed prototype model in the complex plane.

Immediately apparent from Figure 3.3 is the poles corresponding to the phugoid mode very close to the imaginary axis. Therefore, it is expected that the B-52 is considerably underdamped, and while stable, will have a large settling time. Table 3.1 displays the eigenvalues, natural frequency, damping ratio, and time constant of the phugoid and short period mode of the modeled version of the B-52.

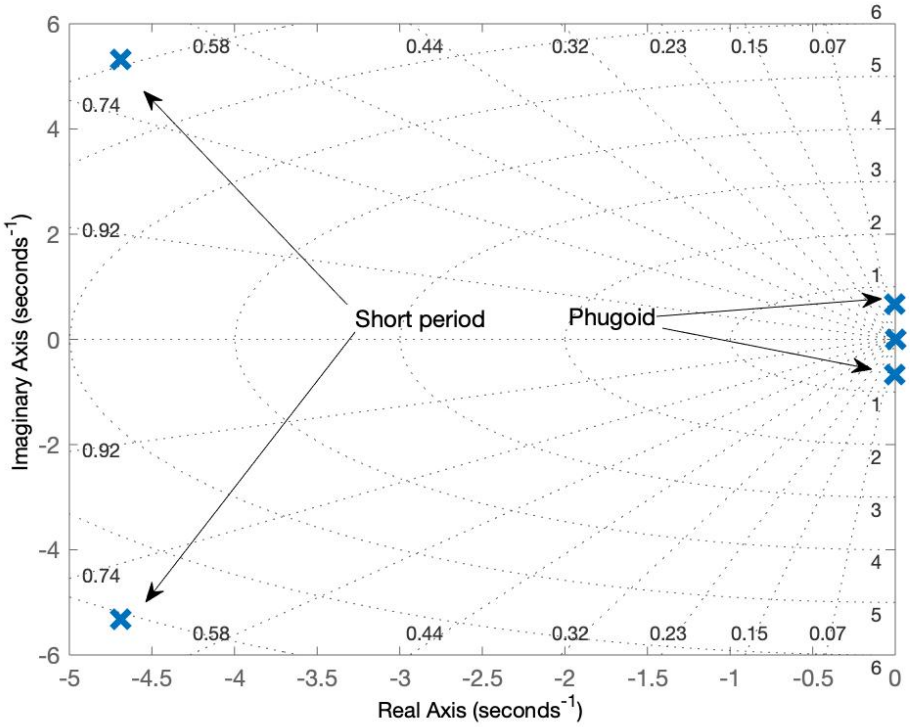


Figure 3.3. Pole-zero map of B-52 longitudinal dynamics.

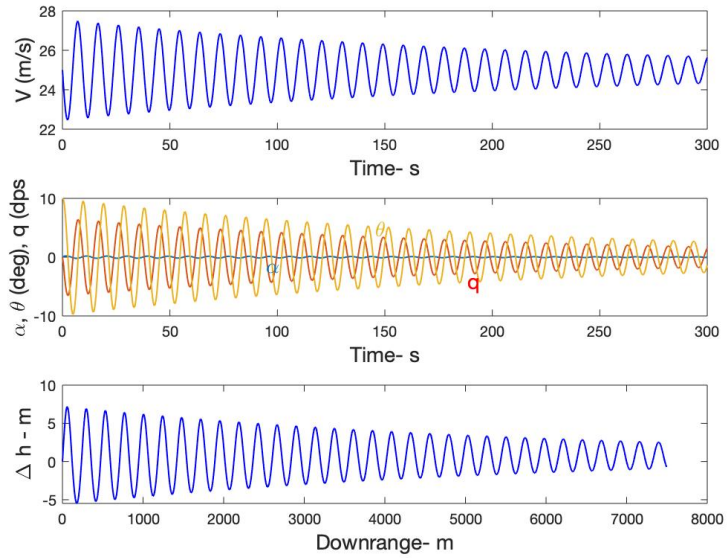
Table 3.1. Eigenvalues, natural frequency, and damping ratio of longitudinal modes.

	$\lambda\ (sec^{-1})$	$\omega_n\ (rad/sec)$	ζ	$T(sec)$
Short Period	$-4.69 \pm 5.33i$	7.1	0.661	0.213
Phugoid	$-0.00443 \pm 0.664i$	0.664	0.00667	226

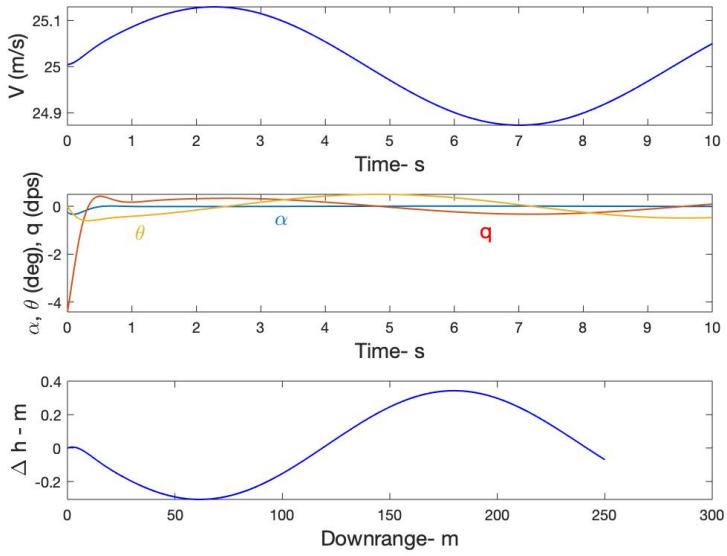
As seen in Table 3.1, the phugoid mode is in fact underdamped. In a piloted aircraft these conditions would be a warning of potential pilot induced oscillation (PIO); however, for an unmanned platform, the only consequence is the requirement that the inner loop autopilot regulate pitch attitude faster than the period of oscillations.

Figure 3.4 displays the result of a dynamic simulation from the state-space model. As expected, the phugoid mode is described by short period oscillations which slowly decrease in amplitude. The short period, despite overshooting, bounds the oscillations to a modest amplitude.

A potential source of uncertainty in modeling pitch damping was the lack of downwash propagation; the vortex emanating from the main wingtip aids pitch damping in dynamic



(a) Phugoid



(b) Short period

Figure 3.4. Phugoid and short period dynamic simulation.

maneuvers. Because an inner loop autopilot was to be implemented, the effect of this uncertainty was considered negligible. Therefore, the aerodynamic parameters pertaining to the longitudinal channel matched expectations, and were considered accurate for the following autopilot study.

3.2.2 Lateral Channel

The next step taken in validating the plant model of the B-52 was to evaluate the lateral dynamic stability. Similar to the longitudinal channel, a state-space representation of the system was utilized. However, because the angle of incidence, sense of rotations, and displacements are now oriented differently, the implementation becomes

$$\begin{bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{y} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = A \begin{bmatrix} \beta \\ p \\ r \\ y \\ \phi \\ \psi \end{bmatrix} + B \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix}.$$

In the lateral channel, the modes of interest are the "Dutch roll," or the oscillation incited by the induced drag of control surface deflection, the roll mode, the restorative moment created by a roll rate, and the spiral mode, the tendency of the aircraft to initiate a spiraling motion in the absence of inputs. The corresponding modes are observed in Figure 3.5.

The stable roll mode lies on the real axis, and therefore will not overshoot. The Dutch roll is stable, yet the damping ratio and frequency will determine the acceptability of the mode. The unstable spiral mode, as expected, is in the right half plane of the pole zero map, yet this poses no danger for an automated vehicle which will be controlled at a frequency much greater than that of the spiral mode. Table 3.2 displays the eigenvalues, natural frequency, damping ratio, and time constant of the Dutch roll and roll mode.

In order to visualize the effect of the Dutch roll on the aircraft states, a simple test was carried out simulating a 5 degree rudder doublet. Figure 3.6 describes how the input propagates through the system states.

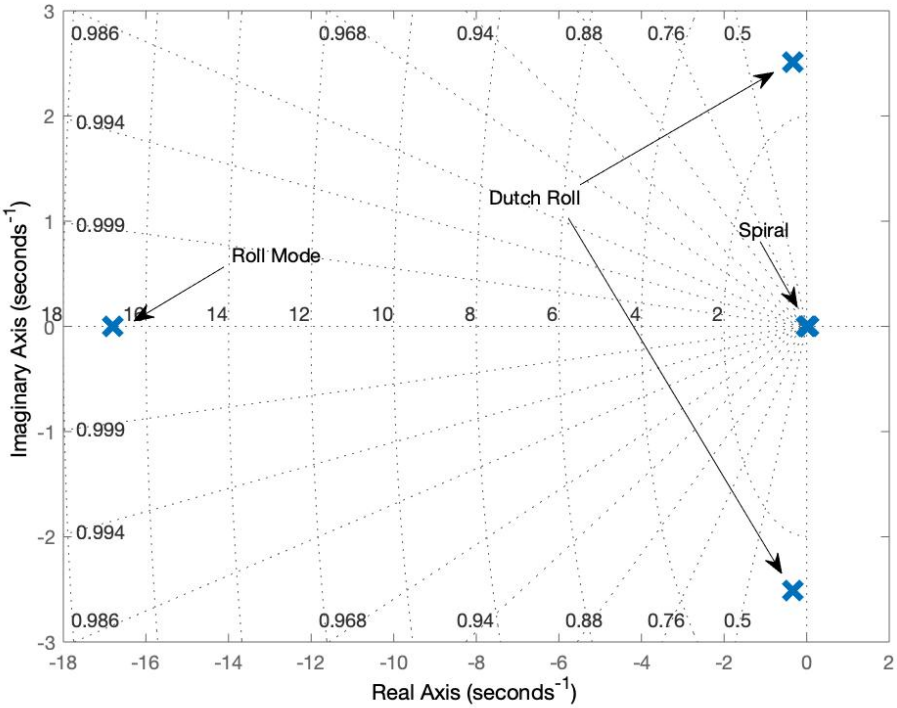


Figure 3.5. Pole-zero map of B-52 lateral dynamics.

Though the response to the rudder doublet is relatively benign, the system response will be stiffened via implementation of a stabilizing PID inner loop. If rudder commands are commanded in the terminal phase of a CARP mission, any roll or yaw rate ought be minimized, as at the point of release the weapon would pick up those states as initial conditions of the corresponding ballistic flight, thus inevitably increasing the miss distance.

Next, the aileron effectiveness was tested with a 25-degree aileron step command. The model of the B-52 utilized asymmetric aileron deflection, or the deflection of a single

Table 3.2. Eigenvalues, natural frequency, and damping ratio of lateral modes.

	$\lambda\ (sec^{-1})$	$\omega_n\ (rad/sec)$	ζ	$T(sec)$
Dutch Roll	$-0.324 \pm 2.51i$	2.53	0.128	3.08
Roll Mode	-16.8	16.8	1.0	0.0595

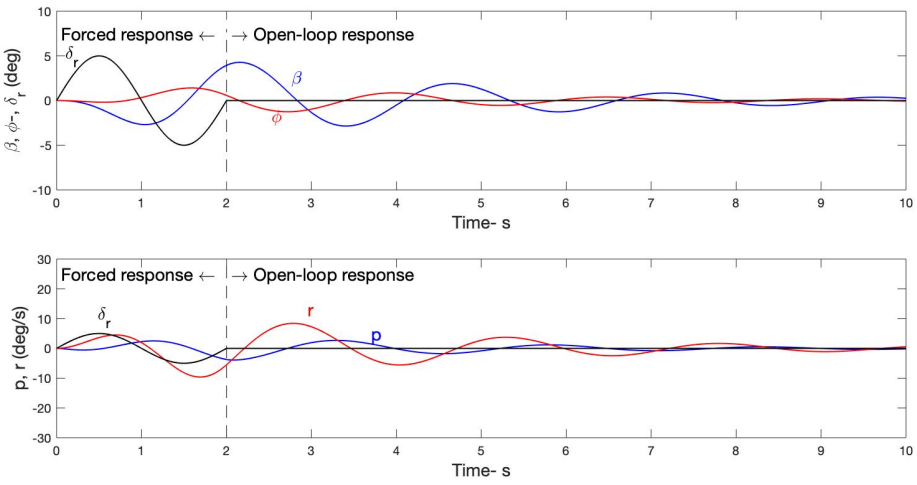


Figure 3.6. System response to rudder doublet command.

aileron vice traditional deflection of both ailerons in concert. Because the approximation was made that asymmetric aileron deflection would adequately model the spoilerons of the B-52, it must be verified that the resulting roll moment incited by the actuators is adequate for flight. Figure 3.7 displays the results of the simulation.

As seen in Figure 3.7, the maximum roll rate achieved is 35.6 degrees per second. The sluggish roll rate is confirmation that the model achieves the objective of modeling a system which has roll actuators such as the B-52. Another takeaway from Figure 3.7 is that the B-52 displays a large adverse yaw given aileron deflection. However, it is in the opposite sense of the yawing moment a spoileron would incite. The adverse yaw behavior of the model is a consequence of using an inviscid solver to compute the aerodynamic characteristics, as inviscid solvers cannot model separation drag.

In conclusion, after evaluating the aerodynamic characteristics of the vehicle, the flight stability analysis was deemed satisfactory, and the model was integrated into Simulink to tune the GNC parameters.

3.3 Guidance, Navigation, and Control

Having confidence in the model of vehicle dynamics, the aerodynamic coefficients were integrated with a Simulink GNC model. The Simulink Drone Reference App [28] includes a

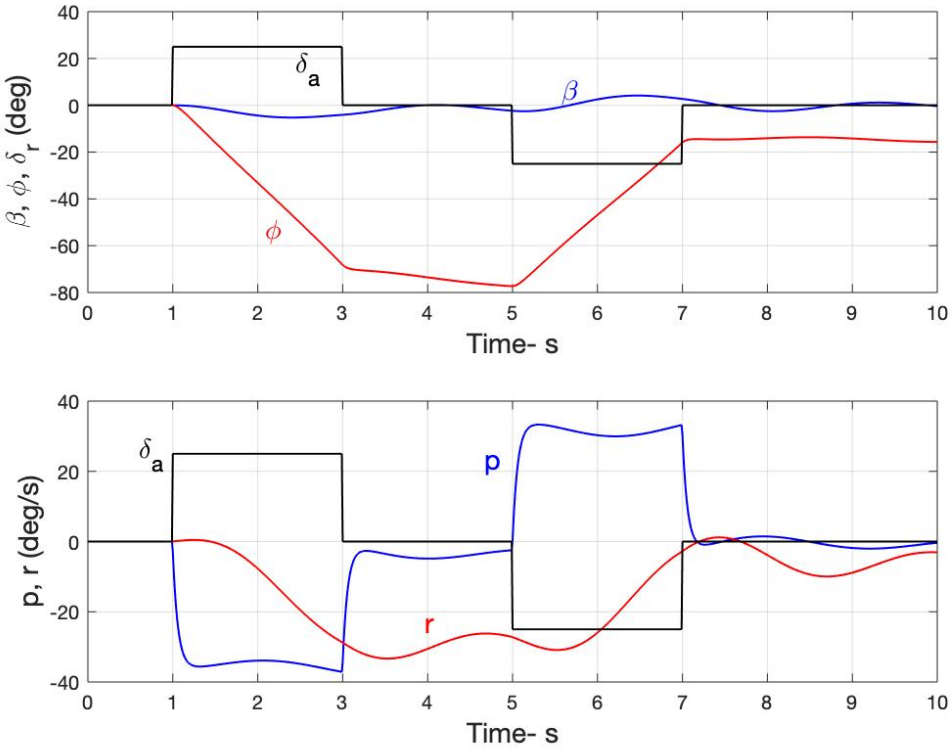


Figure 3.7. System response to step aileron command.

6-DOF dynamics block for computation of the vehicle dynamics, and includes four actuator channels (throttle, ailerons, rudder, and elevator). Additionally, it includes an autopilot model, which includes the GNC components. The Simulink Drone Reference App also interfaces with QGC via the MavLink communication protocol. The first step in modifying the Simulink Drone Reference App to serve as a test interface for the autopilot study was to tune the inner loop autopilot for stabilized flight.

3.3.1 Control

The framework of the longitudinal inner loop flight controller is shown in Figure 3.8.

The highlighted boxes a, b, c represent the PID architecture shown in Figure 1.2, and the gains are specified externally. Block (a) in Figure 3.8 represents a transformation from the velocity error to the throttle command, or the dynamics of speed due to throttle input. Similarly, block (b) represents a transformation from height error to a pitch command,

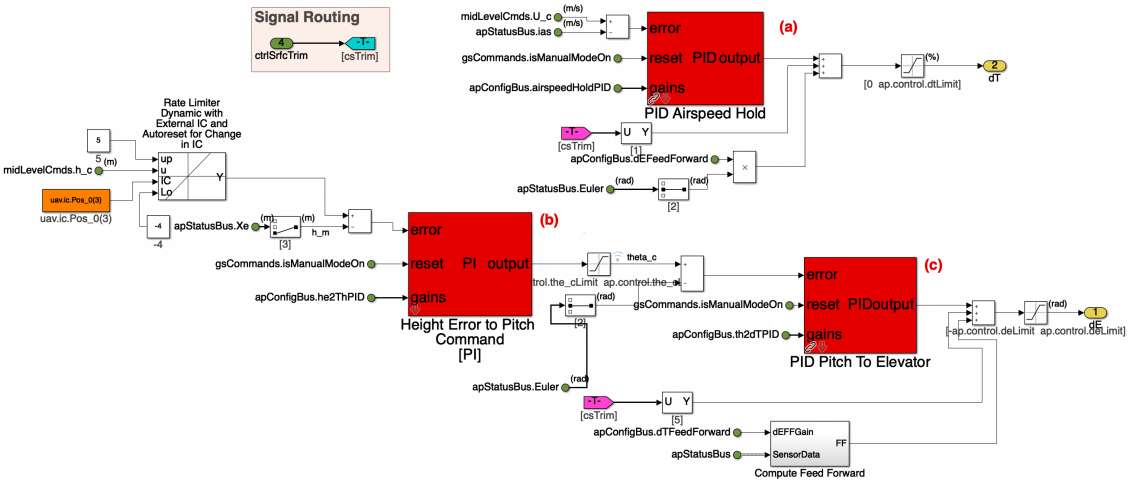


Figure 3.8. Longitudinal inner loop.

and block (c) transforms pitch error to elevator deflection. As mentioned previously, there are many methods to tune a PID controller, including pseudo self-tuning approaches within Simulink. A manual tuning approach based on the classical Ziegler-Nichols [29] approach was taken to tune the longitudinal inner loop gains, informed by the stability analysis done previously. When tuning a real RC aircraft, the first step it to identify the trim condition, or the state about which to linearize. Tuning the longitudinal inner loop to follow a constant reference in altitude and height mimics the process a pilot follows when establishing steady, wings level, unaccelerated flight (SLUF). Tuning lateral gains before establishing longitudinal trim is not best practice, as the lateral tuning will have to be redone to minimize changes in airspeed and altitude given a command in the lateral channel.

The functionality of the Simulink Drone Reference App to specify whether the vehicle ought to follow waypoints or follow mid-level commands on states, specified by the user, aided in the tuning of the inner loop. Figure 3.9 displays the simulated ground station commands.

The guidance mode was set to mid-level commands, and the longitudinal inner loop was first tuned in velocity. First the derivative and integral gains were set to zero, and the proportional gain was adjusted such that the vehicle velocity error oscillated along the reference. Then, the proportional gain was decreased, and the integral gain was increased to cancel steady state error. As mentioned, this methodology approximates Ziegler Nichols tuning, however, the resulting gains provided a magnitude for further fine tuning by trial and error. Once the

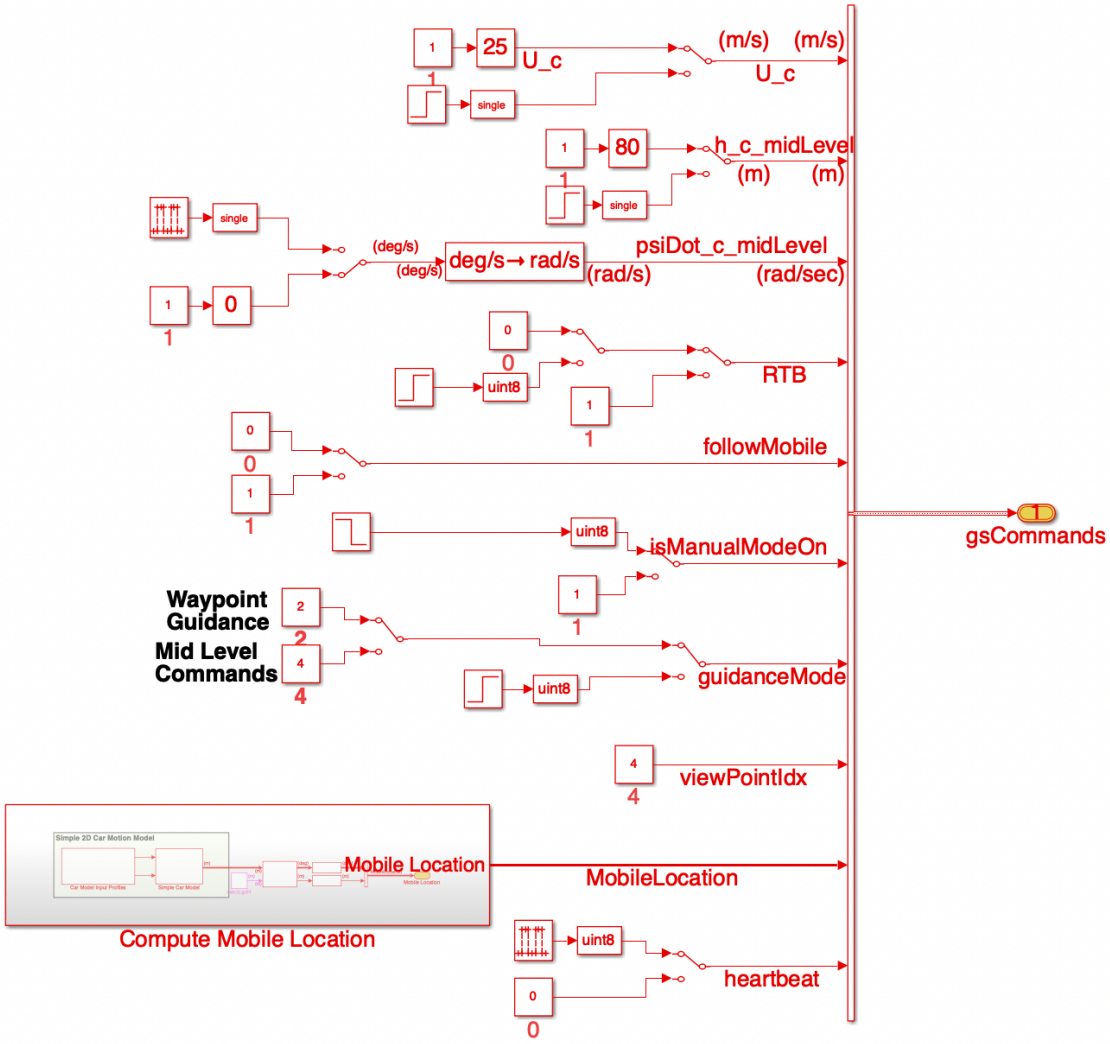


Figure 3.9. Implementation of logical switches including the guidance mode.

combination of PID gains were effective at reference following, the focus transitioned to the transformation from pitch angle error to elevator deflection, concluding with height error to pitch angle. Once the longitudinal gains were tuned for reference following, the lateral gains were tuned in the same manner. Finally, the guidance mode was switched to waypoint guidance along a path, and the gains were augmented to identify and isolate to the greatest extent coupling of controls.

3.3.2 Guidance

The guidance algorithm of the model follows the method put forth in [30]. The method is similar to the L1 methodology of [19] with a notable departure being the previously defined L1 vector, which defined the lookahead distance, is replaced with a lookahead time. The computation of the lookahead vector is shown in Figure 3.10.

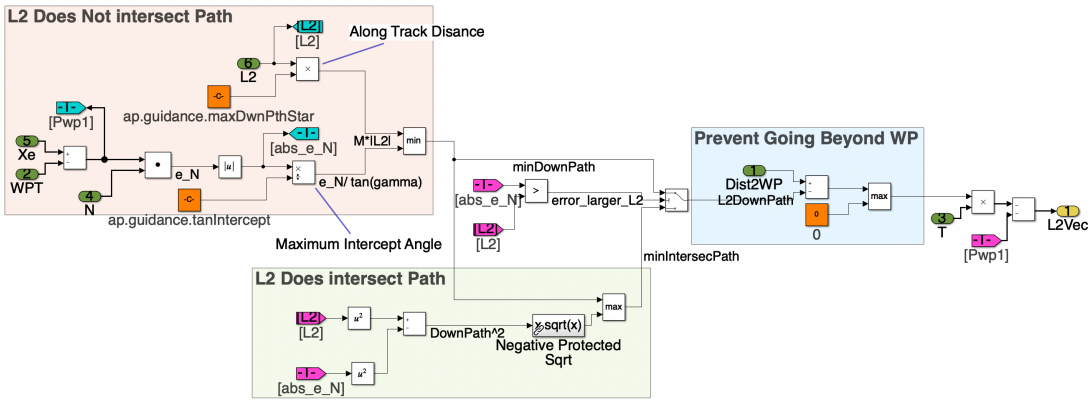


Figure 3.10. Computation of model lookahead vector.

In the same manner described by Figure 2.7, the vector to the reference point is transformed to a lateral acceleration, which in turn provides a bank angle command to the flight controller. Similarly, the mid-level commands of velocity and turn rate are transformed into a bank angle command to be sent to the flight controller.

3.3.3 Navigation

Lastly, the navigation architecture included in the Simulink Drone Reference App provides a model of the UAV sensors. A model of sensor noise, which implements additive band-limited white noise with controllable characteristics, will be used to evaluate the impact of sensor quality on the mission effectiveness of the B-52. The noise model implementation is shown in Figure 3.11.

For each sensor, the amplitude of the noise power spectrum density, which is related to the variance of the signal, is supplied. A sensor noise profile will be developed which approximates the sensor quality of the COTS Pixhawk 1, and henceforth the sensor noise will be varied to explore whether high-quality sensors provide a boost in performance proportional to their cost.

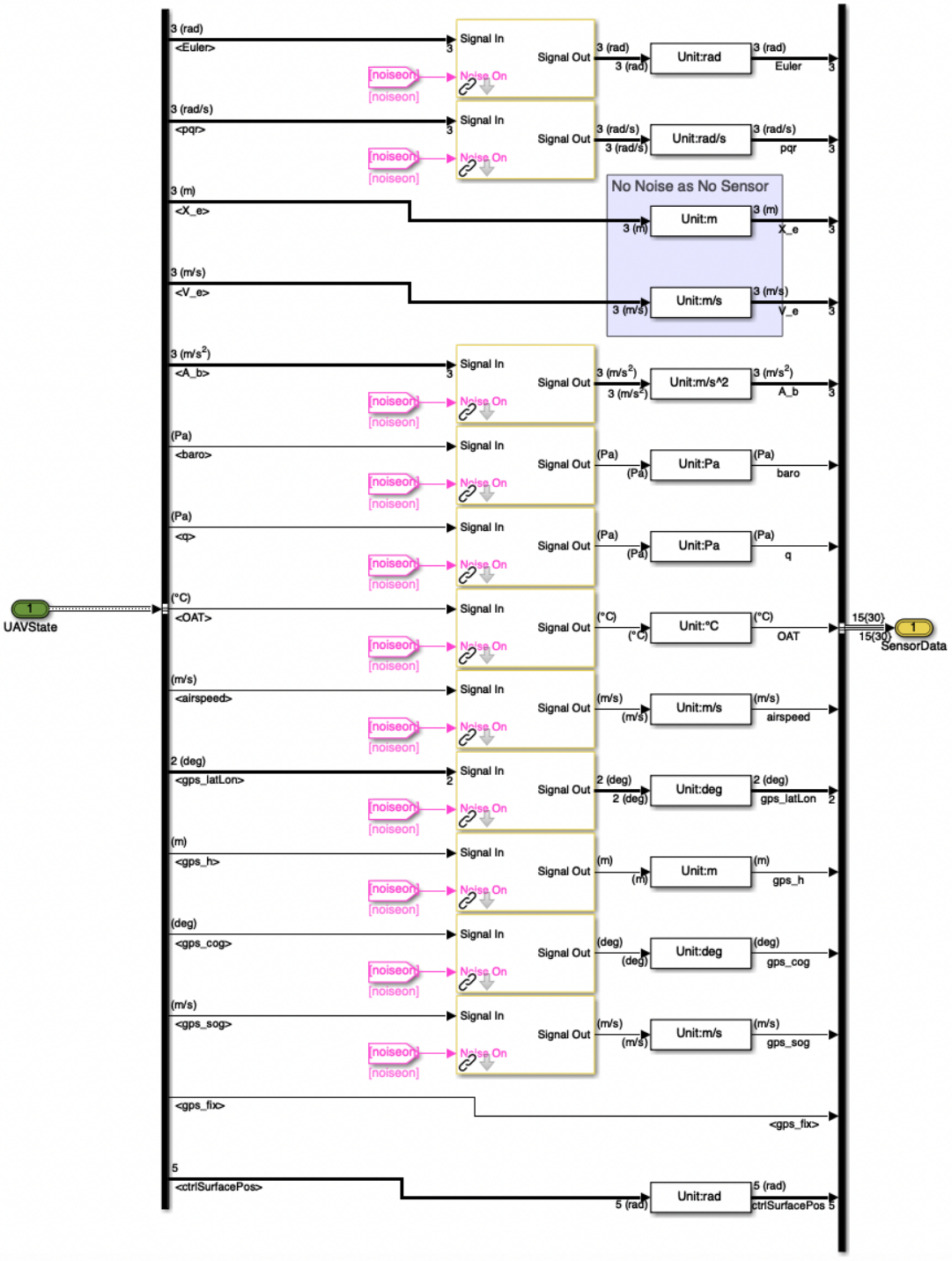


Figure 3.11. Addition of Gaussian noise to sensor signal.

Regarding GPS signals, the maximum navigation update rate is also modeled by implementing a zero order hold on successive signals. The GPS signals are stored for the interval specified by the frequency of the update rate value.

3.3.4 Ground Station Connectivity

The Simulink Drone Reference App offers a co-simulation mode in which the commands to the flight controller are sent via MavLink to Simulink [28], and the states are sent to the ground station from the 6-DOF model. The co-simulation environment abstracts the plant model from the user, allowing for waypoint generation in the QGC environment. The co-simulation feature will be used to evaluate the challenge posed by implementing a UAV for the CARP mission as posed to the user in-situ. The tasks associated with managing the mission from the ground station include creating and uploading a waypoint path, uploading the UAV gains and navigation parameters, and monitoring the UAV along the simulated flight.

In conclusion, the model of the B-52 vehicle dynamics, and GNC components combine to create a viable testing platform to evaluate the technological level required to accomplish the CARP material delivery mission. In subsequent testing, the sensor models will be augmented to reflect different technology levels, as well as different wind velocities. With the previously described model, the minimum level of sophistication will be evaluated to complete a CARP mission, and the connectivity with QGC will be used to evaluate the workload of the end user.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Results and Discussion

This chapter describes and summarizes the results of tests carried out on the B-52 UAV model, specifically to understand the effect of sensor quality and environmental conditions on metrics associated with a CARP mission. As such, the focal point of the following discussion is the performance of its navigation loop; in simple terms the path following performance. GPS update rate was identified as a key parameter shaping mission performance, and knowledge of wind conditions with respect to path planning was identified as a prerequisite skill necessary for robust system performance in all conditions.

4.1 Test Procedure

In order to test the ability of the UAV to successfully reach a CARP in a waypoint mission, a track consisting of three waypoints was proposed in order to evaluate the ability of the UAV to settle upon a new heading after a tight turn. The proposed test trajectory has the UAV settle its cross track error in 650 meters. Figure 4.1 displays the test trajectory, for reference.

In order to verify the suitability of this scenario for evaluation of the UAV, completing a CARP mission, a Monte Carlo simulation was undertaken. The independent variables in the Monte Carlo simulation were initial position and initial body velocity. Noise and wind were not considered in this preliminary evaluation. The initial conditions were varied according to a gaussian random number generator, and the extreme values in x and y position were ± 100 meters, with height being bounded between 80 and 120 meters, about the mean of 100 meters. The body velocities were similarly varied, with the u component body velocity bounded between 20 and 30 meters per second, about the mean of 25. The v and w components of the body velocity were bounded by $\pm 5 m/s$. The goal of varying the initial conditions was to simulate a cross track error in the UAV's traversal from a preceding waypoint. Additionally, the bounds of the varied initial conditions were grossly extended beyond expected cross track error in order to test the robustness of the inner loop and navigation algorithms. As a reference, a simulation was run with initial conditions set to the mean of the states that were randomly generated. The miss distance for this reference

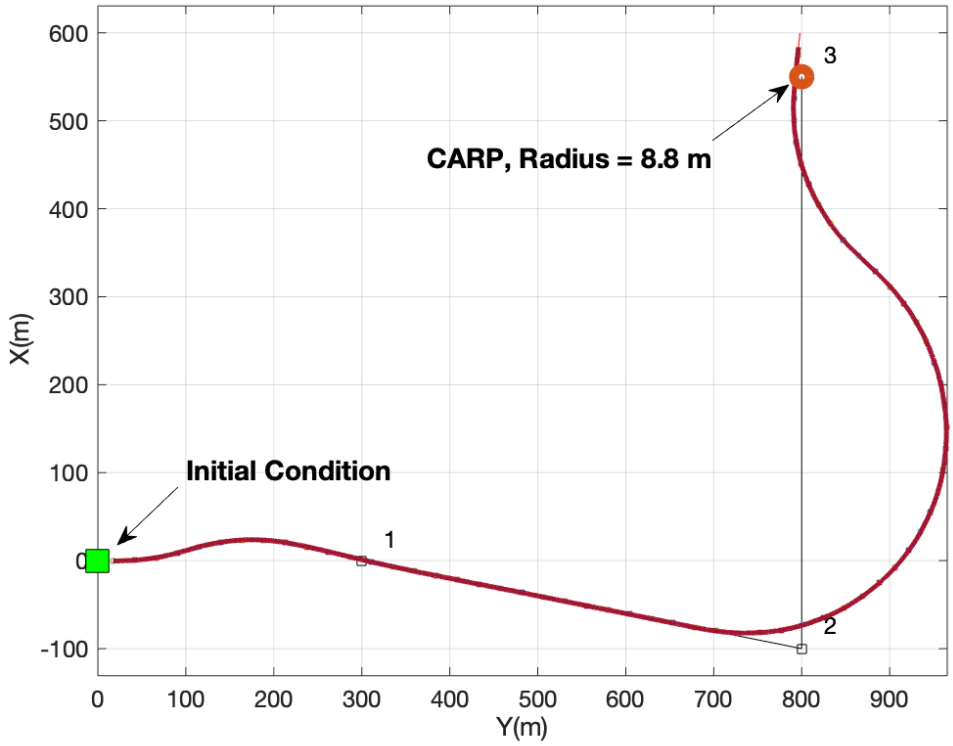


Figure 4.1. Proposed trajectory for test.

case was 7.53 meters.

The Monte Carlo simulation was run for 1000 iterations, and the results are presented in Figure 4.2.

The maximum of the Rayleigh fit to the miss distance data corresponds to a miss distance of 7.53 meters, which matched the miss distance of the reference. Therefore, it was validated that the three-waypoint track was an effective means of simulating the UAV approach to the CARP. In addition to validating the track for testing, the results of the Monte Carlo simulation provide a benchmark for accuracy in the way of the cumulative probability function. The cumulative probability function is shown in Figure 4.3.

The Rayleigh cumulative probability function intersects 50% at a miss distance of 8.8 meters. Simply, given the wide range of boundary conditions that were tested, for any given initial condition there was a 50% chance of intersecting a sphere with a radius of 8.8 meters

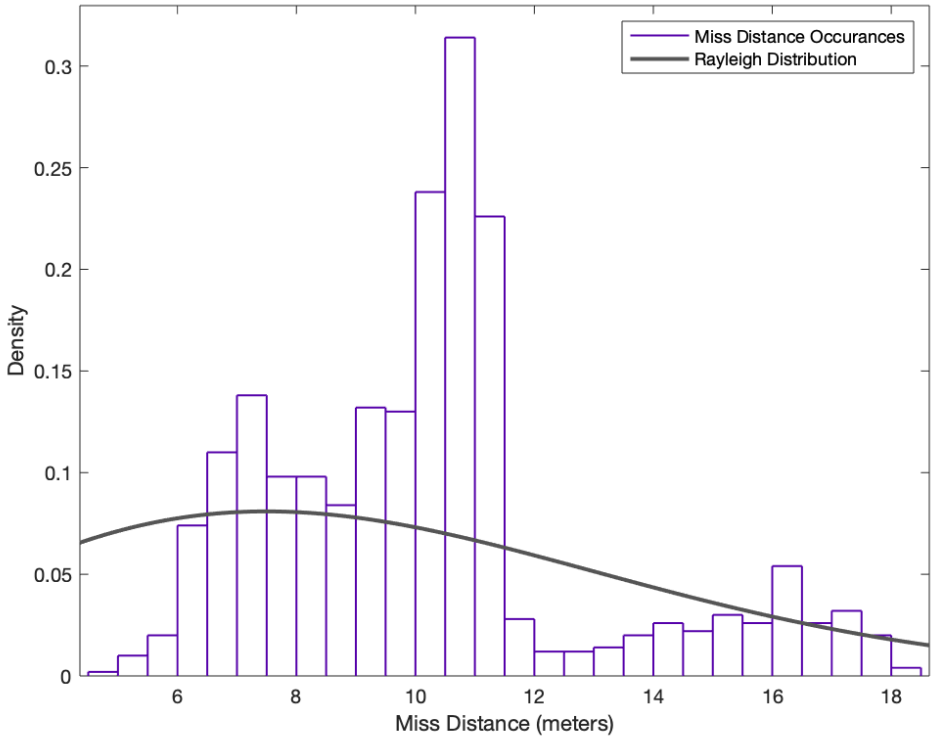


Figure 4.2. Probability density function of miss distance data.

from the CARP. This spherical error probable (SEP) will serve as the minimum bar of performance for the subsequent evaluation, given sensor noise and wind disturbances.

4.2 Gyro and Accelerometer Noise

The gyro and accelerometer noise were initialized to values one would find in a COTS unit, with noise specifications from the L3GD20H gyro [31] and MPU-60X0 [32] data sheets, respectively. Preliminary studies in which noise values were set to zero with the exception of the sensor to be examined proved to have no effect on miss distance. The complimentary filtering, which compensates for noisy signals with those that are less noisy by fusing sensor readings, makes identifying the impact of noise in individual sensors problematic. Therefore, the noise in the accelerometer and gyroscope were altered in concert with each other. The other sensors, including the air data sensors, magnetometer, and GPS were initialized to conservative values one may find on a COTS unit. The method of altering the

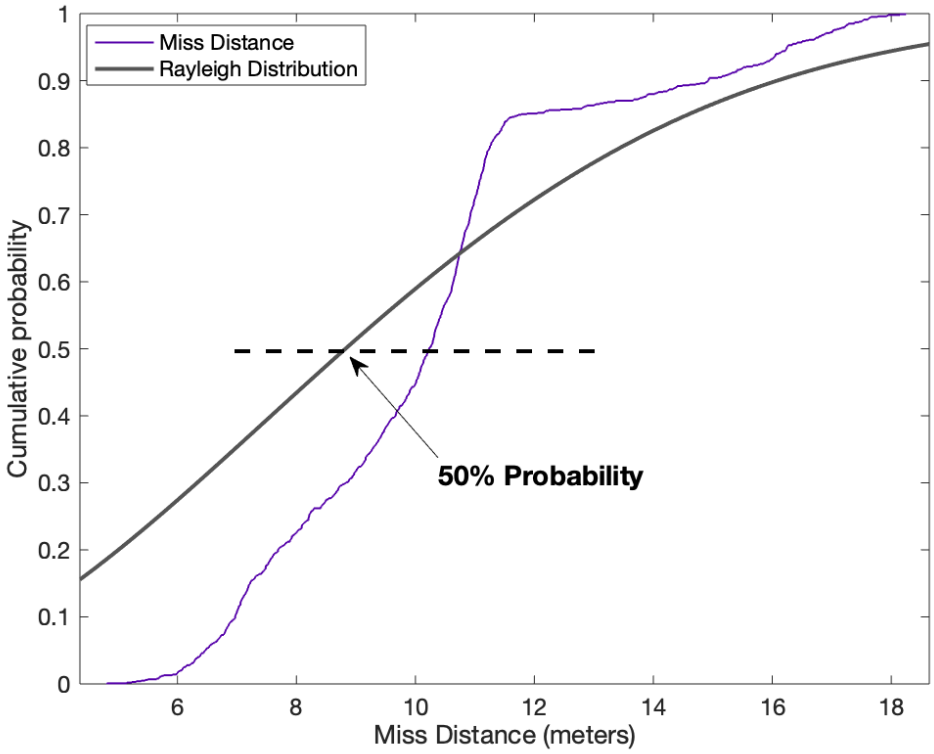


Figure 4.3. Cumulative probability function of miss distance data.

noise level involved multiplying the additive noise by a gain, which ranged from 0 to 3 in intervals of 0.05. Figure 4.4 displays the miss distance for increasing noise gain.

As can be seen in Figure 4.4, despite increasing the gain of the noise signal to 300% of the values provided on the respective data sheets, the miss distance does not follow a clear trend with noise gain. To clarify this result, the data was plotted in a histogram and a normal distribution applied. Figure 4.5 displays the histogram and accompanying distribution.

The mean value of the normal distribution was a miss distance of 7.73 meters, with a standard deviation of 4.9 centimeters. Therefore, the conclusion was reached that with a nominal level of GPS noise and a maximum navigation update rate of 5 Hz, the "true" state data obtained from the GPS receiver was obtained at a sufficient rate to limit the effect of noise in IMU constituent sensors. Therefore, the GPS noise parameters were augmented with the nominal value of accelerometer and gyroscope noise kept constant at the level

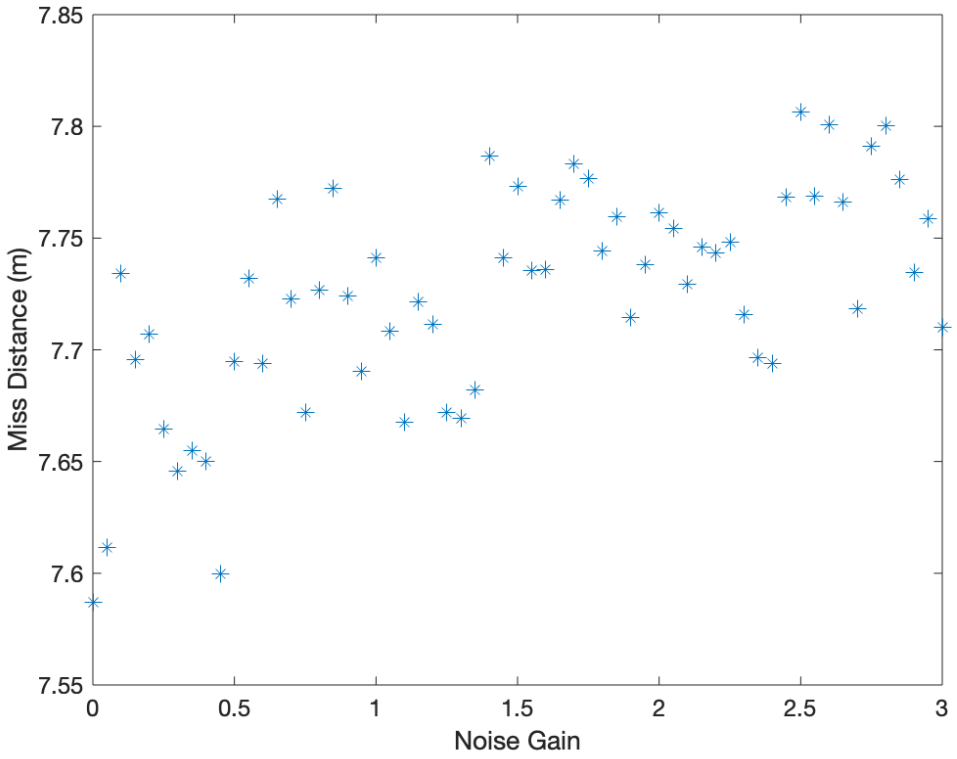


Figure 4.4. Miss distance for increasing gyroscope and accelerometer noise.

specified by the manufacturer.

4.3 GPS Navigation Update Rate

Acknowledging that noise in IMU measurements had minor effect on the miss distance of the UAV to the CARP when paired with a capable GPS, the effect of navigation update rate was tested. Navigation update rates of 1 to 10 Hz were tested to evaluate the effect on the overall miss distance. Figure 4.6 displays the trend of miss distance for a given update rate.

In Figure 4.6, a two-term exponential function was overlaid with the experimental data. Therefore, miss distance decreases exponentially with increasing update rate. Transitioning from a nominal navigation update rate of 5 Hz, which was used in the IMU study, to an update rate of 10 Hz, the miss distance decreases below the no noise miss distance. Comparing a system with noise comparable to a COTS system and a navigation update

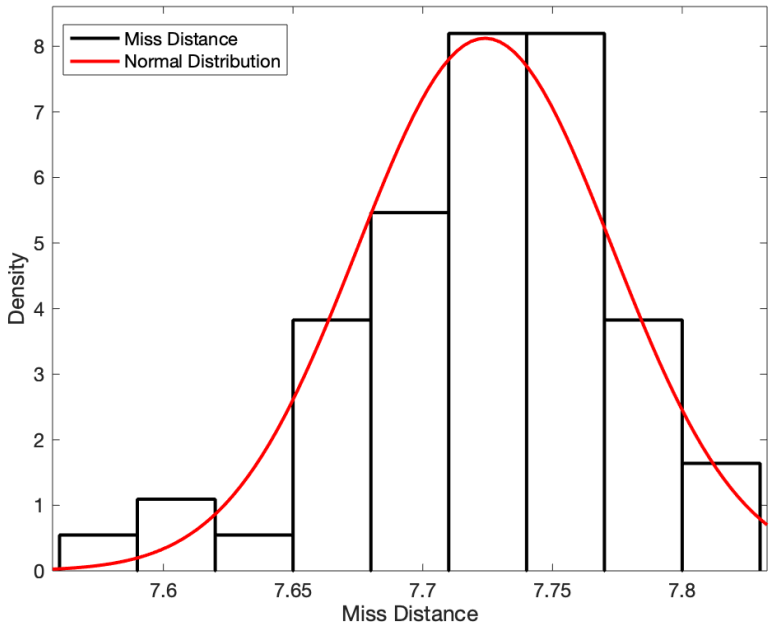


Figure 4.5. Histogram of miss distance for increasing gyroscope and accelerometer noise.

rate of 10 Hz to a system with no modeled noise and an update rate of 5 Hz, as was the reference case, the former achieves a 1.86 meter decrease in miss distance. Therefore, the successful operation of the navigation loop in path following was identified as highly dependent upon the rate at which the vehicle receives "true" position information. Related to the miss distance are the vehicle states over the CARP. Because the vehicle is challenged to settle along the commanded trajectory before the CARP, it is important to evaluate the attitude of the vehicle at the CARP, as the payload inherits the vehicle states at this point as its initial condition. The vehicle Euler angles, pitch, roll, and yaw, were captured at the minimum distance to the CARP, as well as the velocity of the UAV. Figure 4.7 shows the dependencies of states errors of the vehicle as functions of navigation update rate.

Considering the bank angle of the UAV, the angles corresponding to an update rate below 5 Hz are severe. Because the navigation algorithm commands a lateral acceleration to converge to a track, the implication is that in this instance of time the UAV has overshot the CARP and is trying to correct. The pitch angle variation is bounded by a much narrower error than the bank error, yet the trend exists that the slower GPS update rates tend towards

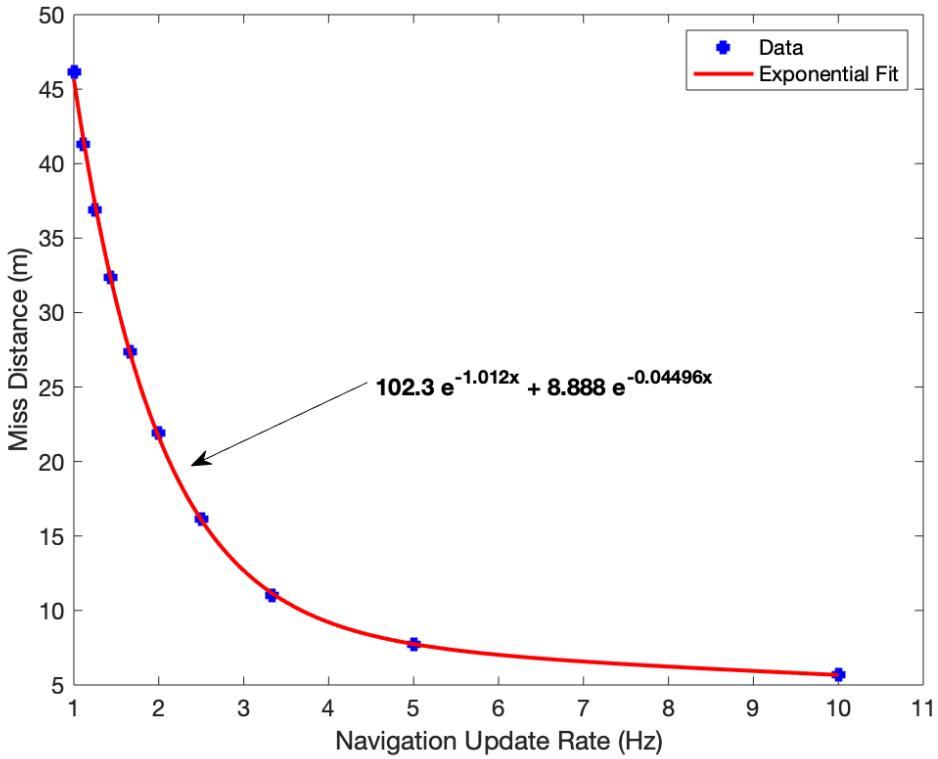


Figure 4.6. Miss distance as a function of navigation update rate.

a dive. However, the magnitude of the dive angle is also indicative of inability to reach steady state by the CARP.

Next, the heading angle was considered. Despite the slower update rates having a lower heading angle error, the roll error and distance error invalidate any implications this result would carry. For the faster rates, the implication is that the UAV is slightly abeam of the target, and is correcting a cross track error. Considering the velocity magnitude errors, they are deemed negligible. The magnitude of the velocity error for this trial is confirmation that the longitudinal speed inner loop is tuned properly. Therefore, considering the trade off in values between a higher grade IMU or a GPS with a faster update rate, the investment in a fast GPS system will play exponential dividends.

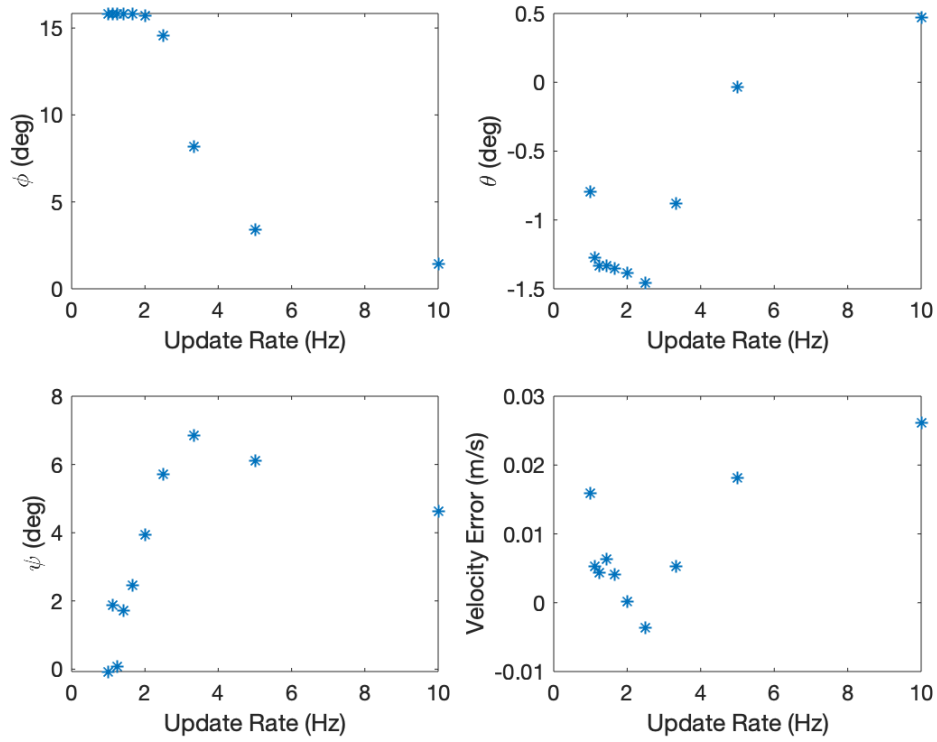


Figure 4.7. UAV states as a function of navigation update rate.

4.4 GPS Noise

A shortcoming of the experimental setup as it exists is the ability to model noise in the GPS signal. Whereas, in Section 2.2.2, the susceptibility of a GPS signal-to-noise ratio was explored in terms of the received signal power compared to received power from extraneous sources, the noise in GPS signals in the model consists of noise independently in the GPS height, course over ground, and speed over ground measurements. In previous trials, the power of the band-limited white noise was initialized as 0.0005, and this noise power was combined with a gain of 0.15 for the height and course over ground, and 0.1 for the speed over ground. From this nominal level, the gain of the noise signal was incremented over successive trials to study the effect of GPS noise. The noise gain values were varied between 0 and 1, incremented by 0.2. The test was run at the nominal update rate of 5 Hz, to both provide a conservative estimate of a COTS autopilot, as well as to be consistent with the IMU noise trials. Figure 4.8 displays the trend between miss distance and noise in the GPS measurements for incremental gain values.

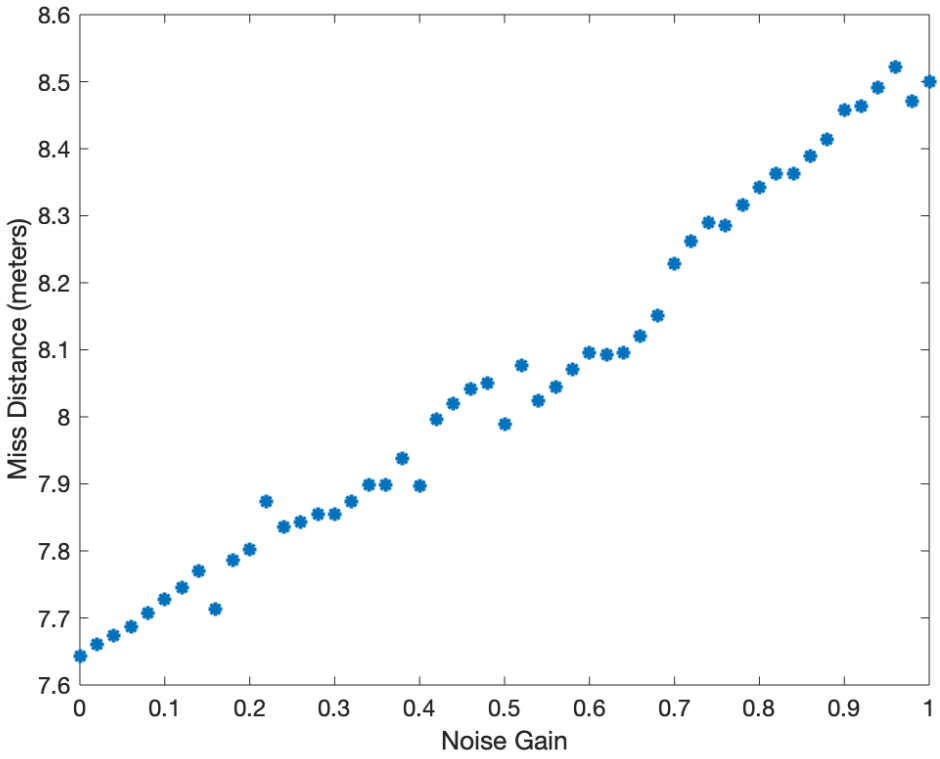


Figure 4.8. Miss distance as a function of GPS measurement noise.

Observing the trend in Figure 4.8, the miss distance has a linear relationship to the magnitude of the gain applied to the noise signal. By recognizing the maximum miss distance of the test as 8.52 meters, comparison with the SEP value of 8.8 meters revealed that despite the relatively large estimate of GPS noise, the configuration still falls within the previously established error tolerance given benign ambient conditions.

4.5 Wind Modeling

After evaluating the UAV performance subject to varying levels of sensor quality, the system response to adverse environmental conditions was modeled. In order to evaluate the effect of wind on the approach to the CARP, a constant horizontal wind component was coupled with a Dryden wind turbulence model to model unsteady wind acting on the vehicle. The constant wind component was set to a velocity of 12 m/s, or roughly half the cruise velocity of the vehicle. The wind direction was initialized by establishing the direction of the

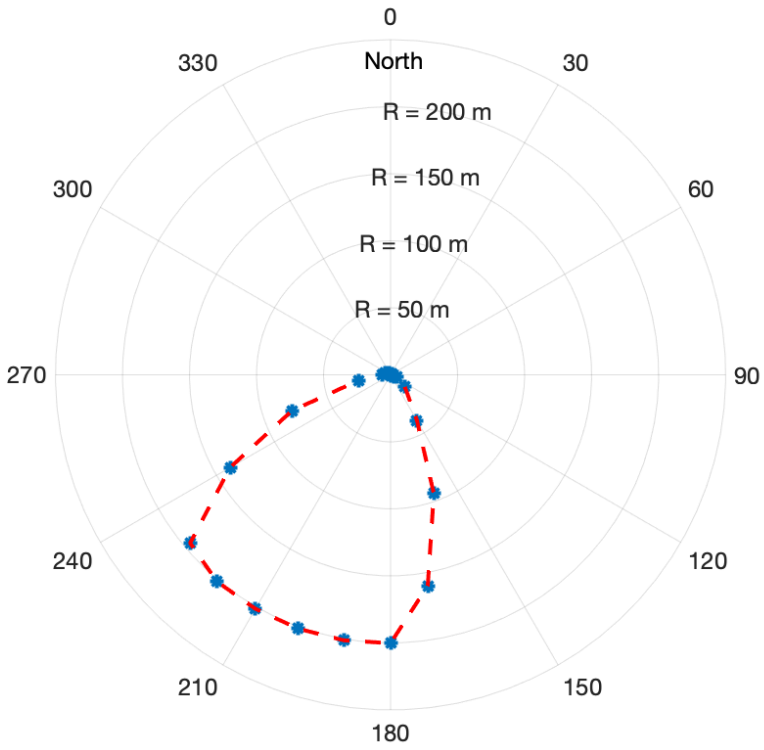


Figure 4.9. Miss distance as a function of wind direction.

constant horizontal wind component and the direction of the turbulent wind. The unsteady wind was initialized to blow 90 degrees clockwise from north. The constant horizontal wind component was varied in direction from 0 to 360 degrees clockwise from north in increments of 10 degrees. Figure 4.9 shows the miss distance as a function of the wind direction.

As observed in Figure 4.9, between 150 and 270 degrees clockwise from north, the UAV exhibits gross errors in the miss distance, up to 200 meters. However, clustered at the origin are a number of successes. Figure 4.10 shows the instances of the UAV achieving a certain miss distance, with each bin being representing a 6 meter range.

Of 37 trials, 24 resulted in a miss distance of less than 6 meters. Furthermore, 19 of the 37 trials missed the CARP by less than two meters. In the case of the UAV approaching the CARP, with the exception of wind from 130 to 260 degrees clockwise from north, the

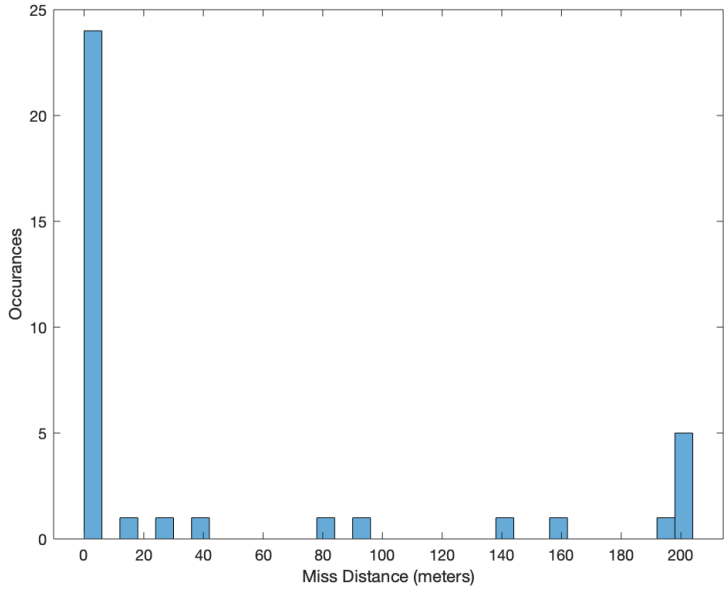


Figure 4.10. Occurrences of miss distances for wind test.

wind decreased the overshoot of the UAV in turning from waypoint 2 to waypoint 3. Noting that the miss distance was closely related to wind direction, a similar test was conducted in which the distance from waypoint 2 to the CARP was doubled. Figure 4.11 shows that the maximum miss distance decreased from 200 to 60, and the number of points grossly affected by wind direction significantly decreased.

Therefore, the conclusion was reached that the UAV is capable of effective operation in a significant wind state, relative to its cruise speed. However, in answering the question of what prerequisite knowledge is necessary for employment of a UAV for a CARP mission, it becomes apparent that operating procedures pertaining to wind speed and direction are necessary. Considering robust operation despite environmental conditions, it is noted that situating the path to the CARP with a tail wind will result in outcomes unacceptable for mission success. Therefore, experience in path planning, paying mind to wind and weather, is necessary.

4.6 Ground Station Interface

Observing the variation in UAV performance due to variation in sensor quality and environmental conditions, next the chosen UI was evaluated from the perspective of an end user, abstracted from the design process. The process of utilizing the DGS for flight operations begins in tuning the autopilot's inner, and navigation loops. As mentioned previously, this task is nontrivial and represents the largest learning curve in utilizing a COTS UAV. However, as stated in the decision-making matrix, when choosing an exemplar COTS DGS, the process is well documented and widely available [33]. Once the autopilot is tuned, whether in a mechanized or manual fashion, the parameters of the control systems must be loaded into the DGS, and this file is known as a "parameters" (.param) file. The .param file contains the inner loop gains, guidance mode, as well as the L1 vector characteristics.

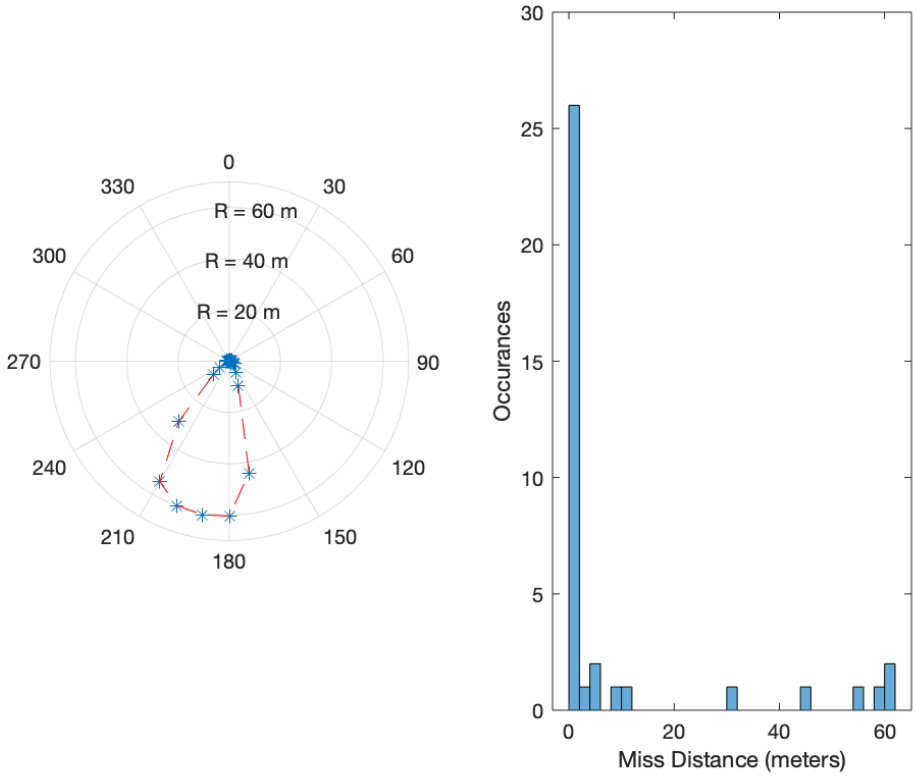


Figure 4.11. Miss distances as a function of wind direction, increased distance to CARP.

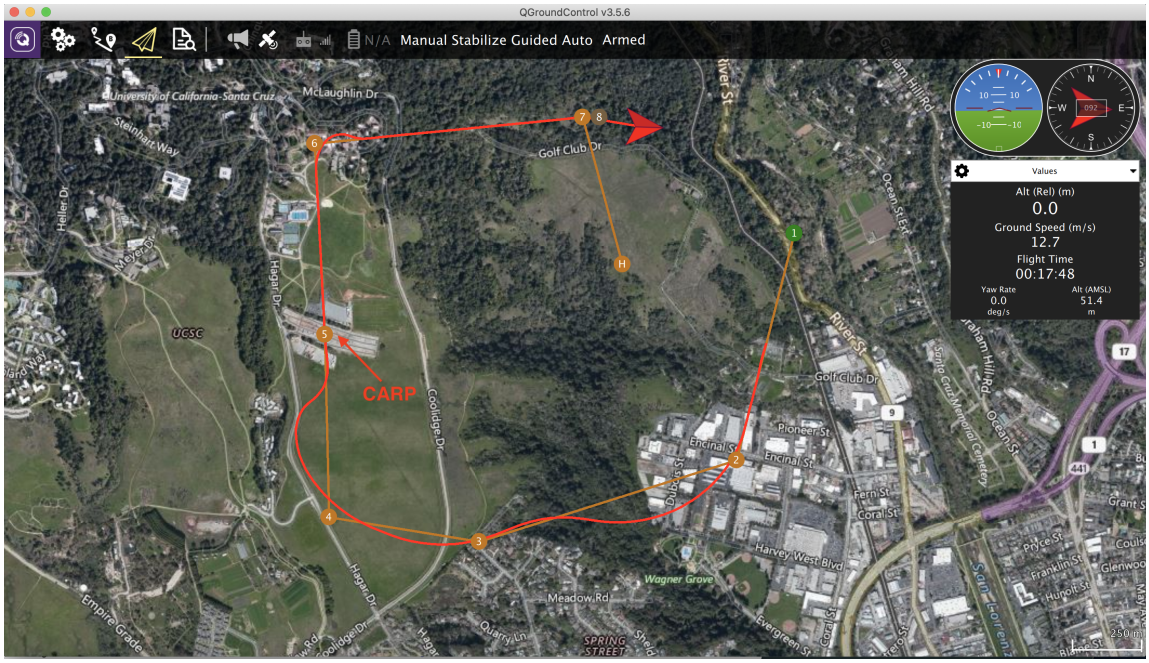


Figure 4.12. Simulated waypoint mission using QGC UI.

The next consideration is the path planning interface. The UI for path planning is exceedingly intuitive. A satellite image represents the area that is to be the operating area for the UAV. Waypoints are added, and positioned in a drag and drop fashion, and allow for specific height and velocity requirements at the commanded points. Similar to the autopilot parameters, the waypoint mission can be saved as a "plan" (.plan) file for future execution by the UAV.

Finally, when the time of execution comes for a CARP mission, the .plan and .param files are uploaded to the unit and the mission can be executed. From the DGS the mission can be monitored, assuming the distance is such that telemetry data can be received by the DGS. Simulating this process with the model previously devised, including the noise and wind levels nominally established previously, but interacting only with the QGC UI, results in the trajectory shown in Figure 4.12.

Regarding the UI of the COTS DGS software, it represents a very intuitive and user friendly interface for planning and carrying out waypoint missions. The learning curve for such software to be implemented effectively is very small, as it is designed for hobbyists. A potential malicious actor wishing to leverage this capability would find very few barriers to entry and could effectively operate a COTS UAV, with the previously described level of

sensor sophistication, for a CARP mission.

CHAPTER 5:
Conclusions and Recommendations

This paper considered common COTS autopilots, the metrics which define their quality, conceptual design and modeling of a UAV, and proposed test cases to evaluate the feasibility of a malicious actor to exploit COTS hardware and software for a nefarious purpose. As such, the tools used to evaluate the application of the B-52 for a CARP mission additionally provides a methodology for rapid design of other small UAVs.

As a result of this inquiry, it was found that in the absence of GPS denial, a COTS autopilot could deliver a payload to a ground target autonomously, and with a high degree of accuracy. The prohibitive barrier to entry to this space for a malicious actor twas identified as knowledge and resources to carry out conceptual design of an aerial vehicle. That is not to say, however, that an excitingly capable airframe could be purchased and used in the same manner as a COTS autopilot could be exploited.

Considering the weaknesses exposed for a makeshift weaponized UAV, GPS denial was identified as the most effective method of defeating the threat. By jamming the UAV GPS receiver, an effect similar to that seen in decreasing the navigation update rate of the system would be observed, and the error accumulated in the inertial systems would make the system ineffective. In practice, often times when a COTS UAV loses GPS signal, a return to base or immediate landing protocol is initiated. Triggering a return to base or landing protocol is another way to defeat the weaponized COTS UAV threat. However, as GPS jamming is indiscriminate and many rely on the capability that GPS provides, methods of identifying an incoming UAV in-situ, and implementing jamming, are recommended as topics of future research.

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] W. Hennigan. (2018, May. 31). Experts say drones pose a national security threat - and we aren't ready. *Time*. [Online]. Available: time.com/5295586/drones-threat/

[2] L. K. S. Bhattacharyya, A. Datta, *Linear Control Theory: Structure, Robustness, and Optimization*, 1st ed. Boca Raton, LA, USA: CRC Press, 2009.

[3] x-engineer. (n.d.). Open loop vs. closed loop control systems. [Image]. Available: https://x-engineer.org/graduate-engineering/signals-systems/control-systems/open-loop-vs-closed-loop-control-systems-xcos/. Accessed Nov. 19, 2019.

[4] Safari Books Online. (n.d.). Introducing the PID controller. [Image]. Available: https://www.oreilly.com/library/view/intelligent-iot-projects/9781787286429/6886ad81-fc35-4886-8929-3fda05a34156.xhtml. Accessed Nov. 21, 2019.

[5] CloudCapTechnology. (2013, Feb. 28). Piccolo Autopilot - customer UAV aircraft & systems. [YouTube video]. Available: https://www.youtube.com/watch?v=KYYsjSi0Drk. Accessed May. 20, 2020.

[6] M. Zenmiller and V. Miley, *Piccolo Setup Guide*, Cloud Cap Technology, Hood River, OR, 2012.

[7] A. D. Team. (2019). Autopilot hardware options. ArduPilot. [Online]. Available: https://ardupilot.org/copter/docs/common-autopilots.html

[8] A. D. Team. (2019). Pixhawk overview. ArduPilot. [Online]. Available: https://ardupilot.org/copter/docs/common-pixhawk-overview.html. Accessed May. 1 2020.

[9] C. Vélez S. and A. Agudelo, "Control and parameter estimation of a mini-helicopter robot using rapid prototyping tools," *WSEAS Trans. on Systems*, vol. 5, pp. 2250–2256, Oct 2006.

[10] R. J. Bunker, "Terrorist and insurgent unmanned aerial vehicles: use, potentials, and military implications," Army War College (U.S.). Strategic Studies Institute, Tech. Rep., 2015.

[11] A. Bittar and N. Oliveira, "Central processing unit for an autopilot: Description and hardware-in-the-loop simulation," *Journal of Intelligent & Robotic Systems*, vol. 70, Apr 2013.

[12] M. A. M. Robert H. Klenke, William C. Sleeman, "A high-throughput processor for flight control research using small uavs," 2006.

[13] *Piccolo II*, Collins Aerospace, accessed Apr. 25, 2020. Available: http://www.cloudcaptech.com/products/detail/piccolo-ii

[14] R. B. Langley, "GPS receiver system noise," *GPS world*, vol. 8, no. 6, pp. 40–45, 1997.

[15] Pixhawk 1 flight controller. (n.d.). PX4 Dev Team. [Online]. Available: https://docs.px4.io/v1.9.0/en/flight_controller/pixhawk.html. Accessed Apr. 12, 2020.

[16] A. Safian and A. Soleimani, "Piezoelectric energy harvesting from direct buoyancy force," in *3rd International Conference on Mechanical and Aerospace Engineering*, Tehran, Iran, Apr. 2018.

[17] M. Kok, J. D. Hol, and T. B. Schön, "Using inertial sensors for position and orientation estimation," *Foundations and Trends® in Signal Processing*, vol. 11, no. 1-2, p. 1–153, 2017. Available: http://dx.doi.org/10.1561/2000000094

[18] Wikiwand. (n.d.). Kalman filter. [Image]. Available: https://www.wikiwand.com/en/Kalman_filter#/overview. Accessed Apr. 11, 2020.

[19] S. Park, J. Deyst, and J. How, *A New Nonlinear Guidance Logic for Trajectory Tracking*. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2004-4900

[20] A. D. Team. (2019). Navigation tuning. ArduPilot. [Online]. Available: https://ardupilot.org/plane/docs/navigation-tuning.html

[21] D. Hunsaker, "A numerical lifting-line method using horseshoe vortex sheets," May 2011.

[22] M. Palaski, "Modeling flow over 3-d wing spoilers for model B-52," M.S. thesis, Naval Postgraduate School, Monterey, California, 2020.

[23] USU aero lab. (n.d.). Utah State University. [Online]. Available: http://aero.go.usu.edu. Accessed May. 20, 2020.

[24] D. H. C. Goates and J. Schoenfeld, "MachUpX," https://github.com/usuaero/MachUpX, 2020.

[25] Q. Fang, "Jsonlab: a toolbox to encode/decode json files," https://www.mathworks.com/matlabcentral/fileexchange/33381-jsonlab-a-toolbox-to-encode-decode-json-files, 2019.

[26] "Lateral and longitudinal dynamic simulation," class notes for EA413, Dept. of Aerosp. Eng., United States Naval Academy, Annapolis, MD, 2018.

[27] W. F. Phillips, *Mechanics of Flight*, 2nd ed. John Wiley & Sons, Inc, 2010.

[28] M. Lizarraga, "Simulink drone reference app," https://github.com/mathworks/simulinkDroneReferenceApp, 2020.

[29] K. Ogata, *Modern control engineering*. Boston: Prentice-Hall, 2010.

[30] R. Curry, M. Lizarraga, B. Mairs, and G. Elkaim, "L+2, an improved line of sight guidance law for uavs," June 2013, pp. 1–6.

[31] MEMS motion sensor: three-axis digital output gyroscope. (2013). STMicroelectronics. [Online]. Available: https://www.st.com/content/ccc/resource/technical/document/datasheet/35/b1/e0/39/71/57/43/01/DM00060659.pdf/files/DM00060659.pdf/jcr:content/translations/en.DM00060659.pdf

[32] "Mpu-6000 and mpu-6050 product specification revision 3.4," InvenSense Inc., 1197 Borregas Ave, Sunnyvale, CA 94089 U.S.A., Aug 2013.

[33] Tuning setup. (n.d.). Drone Code. [Online]. Available: https://docs.qgroundcontrol.com/en/SetupView/Tuning.html. Accessed May. 7, 2020.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California