NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

RANDOMIZED PATH OPTIMIZATION FOR THE MITIGATED COUNTER-DETECTION OF UAVS

by

Mitchell Heaton

June 2017

Thesis Advisor:	Ralucca Gera
Second Readers:	Levi DeVries
	Michael D.M. Kutzer

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

1. AGENCY USE ONLY *(Leave Blank)*

2. REPORT DATE
June 2017

3. REPORT TYPE AND DATES COVERED
Master's Thesis 09-01-2016 to 05-31-2017

4. TITLE AND SUBTITLE
RANDOMIZED PATH OPTIMIZATION FOR THE MITIGATED COUNTER-DETECTION OF UAVS

5. FUNDING NUMBERS

6. AUTHOR(S)
Mitchell Heaton

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

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 document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

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

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

UAVs provide exceptional capabilities and a myriad of potential mission sets, but the ability to disguise where the aircraft takes off and lands would expansively advance the abilities of UAVs. This thesis describes the development of a nonlinear estimation algorithm to predict the terminal location of an aircraft and a trajectory optimization strategy to mitigate the algorithm's success. Vehicle paths are generated using a matrix-based quadratic trajectory computation method. The paths are then tracked by recursively updating time-based observations of vehicle position using Bayesian filtering. The KL divergence is used to compare the probability density of aircraft termination to a normal distribution around the true terminal location. Results show that the optimal conditions to obfuscate path include waypoints at or beyond the vehicle terminal location, variations in velocity throughout the time of flight, and the minimal use of an aircraft's maximum potential time of flight.

14. SUBJECT TERMS
unmanned aerial vehicles, path planning, optimization, obfuscation, extrapolation, mitigated

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

THIS PAGE INTENTIONALLY LEFT BLANK

RANDOMIZED PATH OPTIMIZATION FOR THE MITIGATED COUNTER-DETECTION OF UAVS

Mitchell Heaton
Ensign, United States Navy
B.S., United States Naval Academy, 2016

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN APPLIED MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
June 2017

Approved by: Ralucca Gera
Thesis Advisor

Levi DeVries, United States Naval Academy
Second Reader

Michael D.M. Kutzer, United States Naval Academy
Second Reader

Craig Rasmussen
Chair, Department of Applied Mathematics

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

UAVs provide exceptional capabilities and a myriad of potential mission sets, but the ability to disguise where the aircraft takes off and lands would expansively advance the abilities of UAVs. This thesis describes the development of a nonlinear estimation algorithm to predict the terminal location of an aircraft and a trajectory optimization strategy to mitigate the algorithm's success. Vehicle paths are generated using a matrix-based quadratic trajectory computation method. The paths are then tracked by recursively updating time-based observations of vehicle position using Bayesian filtering. The KL divergence is used to compare the probability density of aircraft termination to a normal distribution around the true terminal location. Results show that the optimal conditions to obfuscate path include waypoints at or beyond the vehicle terminal location, variations in velocity throughout the time of flight, and the minimal use of an aircraft's maximum potential time of flight.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

C2	command and control
DDL	digital datalink
ISR	information, surveillance, and reconnaissance
KL	Kullback-Leibler
RADM	Rear Admiral
SEAL	Sea, Air, and Land
TOF	time of flight
UAV	unmanned aerial vehicle
UUV	unmanned underwater vehicle

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

Unmanned Aerial Vehicles (UAVs) have matriculated into every battlespace imaginable. From shipboard operations to launches from torpedo tubes, innovation has seen UAVs performing a myriad of mission sets with many more possible. The ability to obfuscate a vehicle's termination point would allow the user to increase its potential mission sets and improve a UAVs utility. This work focuses on the obfuscation of terminal location and further seeks to determine if the vehicle's starting point is compromised from an observed portion of its path.

This thesis describes the development of a nonlinear estimation algorithm to predict the terminal location of an aircraft and a trajectory optimization strategy to mitigate the algorithm's success. A recursive Bayesian filtering scheme is used to assimilate noisy measurements of the UAVs position to predict its terminal location. We use a blackbody radiation-based likelihood function tuned to the UAVs known endurance limitations to assimilate the position measurements. A quadratic trajectory generation method with waypoint and time variation is used to produce a parameterized family of potential aircraft trajectories. The estimation algorithm is then used to assess parameterized UAV trajectories that minimize certainty of the true terminal location. The KL divergence is used to compare the probability density of aircraft termination to a normal distribution around the true terminal location. Confidence intervals with user input values for specificity of starting point are used to indicate the susceptibility of an operator's starting point as a result of the vehicle's path. Results show that the greatest obfuscation of path directly correlates to variations in time of flight with respect to the vehicle's maximum possible flight time while the successful prediction of vehicle starting location is a function of the specificity required by the user.

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

The opportunity to attend the Naval Postgraduate School has provided me with an exceptional start to my naval career. Throughout the duration of this work, I received generous assistance and support from faculty, students, and friends. In particular, I would like to thank Professor Ralucca Gera for her incredible patience and guidance at every turn. Additionally, I extend my thanks to RADM Ellis for his support in allowing me to garner community expertise and insight as a result of travels to multiple conferences. Lastly, I would like to thank my second readers for staying on to monitor this project from its genesis at the U.S. Naval Academy to its completion at the Naval Postgraduate School.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

Unmanned aerial vehicles (UAVs) have been one of the most frequently discussed topics in the Department of Defense and in the public domain as well. Their potential abilities and missions are dynamically expanding. Only naming a few capabilities, UAVs are used in the modern battlesapce to perform information, surveillance, and reconnaissance missions (ISR), launch strikes on enemy targets, and coordinate unit communication. Although a protean platform possessing innumerable skills, the UAV carries with it a diverse set of challenges. This work looks to explore potential solutions to the difficulties presented to a UAV and its operator as a result of its vulnerability to tracking and accidental disclosure of operator locations and objectives.

1.1 Motivation

UAVs are being used extensively to expand the reaches of surveillance and the mission capabilities of military units. Deploying a UAV from a submarine provides interesting extensions of control, including increased surveillance, weapons capabilities, and communication. Currently, UAVs deployed from submarines are considered to be disposable and launching an aircraft guarantees that it will strategically crash after it has completed its mission or run out of fuel. A recoverable UAV would save space on the boat, minimize cost, increase payload capabilities, and promote independence since the vehicle does not need to communicate throughout the life of its mission, as data can be gathered physically after UAV recovery.

Though recoverable UAVs present operational advantages, the issue of recoverability also poses challenges. UAV recovery by a submarine may compromise stealth. Tracking a UAV throughout its time of flight until mission termination could potentially draw unwanted attention to the deployed vehicle. The goal of this work, through the application of path planning, is to optimize flight trajectories of UAVs while minimizing the risk of an observing entity predicting terminal location. Obfuscation of a search pattern and use of natural barriers to tracking, such as sea state, altitude, and the position of the sun, could all be used in designing a program that allows for the recovery of UAVs without compromising the location of an evader.

An additional extension of the problem of observable flight produces the issue of not only revealing the position of the operator at termination, but also at the launch of the UAV. While an observer might not be able to immediately detect the UAV upon launch, any observation of the vehicle during its flight could provide the enemy with an indication of where the operator's location currently or at some previous point in time. A study looking at the probability of an enemy determining the operator's location will indicate the magnitude of the threat presented by observable flight. In turn, this will highlight the potential need to randomize the UAV's path with preference towards obfuscating the launch location in addition to potential recovery positions as well.

1.2 Unmanned Vehicle Implementation in the Undersea Domain

From the genesis of anti-submarine warfare, aviation has been on the front lines of the effort. As vessels that traveled primarily on the surface in the early years, submarines were largely susceptible to surveillance from aircraft. With the development of technology throughout the years, submarines could be placed in danger by depth charges as well [1]. Emerging technology continued in both the undersea and air domains, one field continually attempting to match and outstrip the capabilities of the other. The U.S. Navy's capabilities in the field of maritime patrol relied primarily on the $P-3C$ Orion for the almost half of a century. Launched as a part of the Navy's fleet in 1962 [2], the $P-3$ has served as the front line for submarine detection and battlespace dominance throughout the Cold War and into the conflicts in Iraq and Afghanistan. With long-range electro-optical cameras and special imaging radar that has been continuously updated with advancing technology, the $P-3$, the Navy's main air anti-submarine warfare asset, was capable of extending its abilities to an effective range of $2,380$ nautical miles [2]. At 10.5 hours of flight time per mission provided for over fifty years [2], the $P-3$ was unmatched in its capabilities throughout its service life.

New technology, advancing rapidly and outstripping the design of the $P-3$, proved that antiquated airframes would no longer be capable of supporting the necessary mission set.

Boeing stepped in with the $737-800ERX$ airframe to fill the void and replace the aging $P-3$. Designated as the $P-8A$ Poseidon, the Navy planned to utilize the airframe as a shell that would support rapidly developing technology and its seamless integration for battlespace dominance [3]. Compared to the $P-3$ from a technical standpoint, the $P-8$ sacrifices on-board weapons capabilities for increased sensors and technology. While the $P-8$'s mission altitude is not published, it has a ceiling of $41,000$ feet, cruising speed of 490 knots, and an innovative airframe design [3]. All of these factors readily lend to the conclusion that it has a comfortable operating mission altitude much higher than the $P-3$, which operated on station at $1,500$ feet [2]. So, while able to capitalize on emerging technology, the $P-8$ still leaves vulnerabilities in the U.S. Navy's ability to achieve dominance of the undersea warfare domain. UAVs, as features in the undersea warfare battlespace, have recently started to emerge as possible solutions to fulfilling the mission areas that the $P-8$ might not be able to cover.

The Navy, as recently as 2010, began incorporating the idea of submarine-launched UAVs [4]. Airframes such as the Switchblade have been investigated as potential workhorses for the developing form of the Navy's ISR mission [5]. Former head of undersea warfare (N97), Rear Admiral Barry Bruner expressed his satisfaction with the developing technology.“You can launch it, you can control it, you can get video feedback to the submarine. It sure beats the heck out of looking out of a periscope at a range of maybe $10,000$ to $15,000$ yards on a good day. Now you're talking 20 to 40 miles” [5]. RADM Bruner's expression of the capability of a UAV back in 2012 accurately describes how valuable a UAV can be to a submarine conducting predictive path monitoring, surveillance, jamming, or any other of a host of capabilities possible with UAVs. While Switchblade was the UAV of the past, the Navy has invested in AeroVironment's Blackwing as its submarine-launched UAV of the future. Having recently signed a procurement contract with the company, the UAV provides a plethora of capabilities to support U.S. dominance in the undersea domain. The Blackwing has demonstrated it can operate in three key areas successfully that are crucial to a submarine-launched UAV's mission: acting as a Command and Control (C2) data and communication relay, use of secure digital datalink (DDL), and the ability to link with a swarm of unmanned undersea vehicles [6].

The ability of a UAV to operate as a C2 hub to link platforms and coordinate communication is vital to the success of a submarine-launched UAV. While this not only extends the

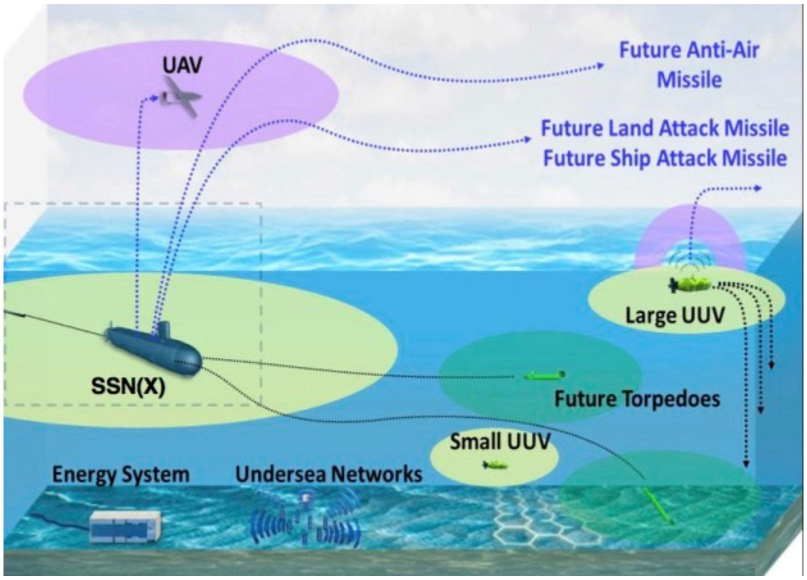


Figure 1.1. A potential future operating concept in undersea warfare demonstrating swarm networking. Source: [7].

capabilities of single submarine, this allows for an entire network of submarines, UUVs, and UAVs to be linked seamlessly in coordinated communication. With the Navy's continuing developments in data transmission through lasers and their propagation paths in the undersea domain, the swarm network's capabilities could continue to extend further and further. The success of the C2 mission, however, relies entirely upon the security of the communications and data being transmitted. The Blackwing's demonstrated capability to use the proven DDL network already in use by the Navy guarantees the security of the information and further opens up possibilities for the missions the UAV can perform. Last, but perhaps the most important of the key capabilities as it relates to future advancement, is Blackwing's ability to operate as a part of a swarm. This testing is crucial to the development of future mission sets in a coordinated undersea domain as multiple assets can be used in coordination to achieve a mission. Figure 1.1 shows a U.S. Navy concept of what a coordinated battlespace of the future might look like. While there are already four vehicles in Figure 1.1, a coordination of four vehicles could likely be on the lower end of potential swarm sizing. Dependent upon data transmission distances and the ability to avoid degraded command and control, the size of an undersea swarm could grow as large as the theater commander requires.

While the Blackwing provides fantastic capabilities to the Navy moving forward, many other potential UAVs are in development to achieve dominance of the undersea battlespace. One such example of an emerging prototype is an unmanned vehicle that has the capability to operate as both an unmanned underwater and aerial vehicle. The vehicle utilizes a multipurpose aileron allowing for controllability in flight and flipper-like function when in the water for additional propulsion and control [8]. The flying swimmer (Flimmer) would generate an entirely different mission set than previously seen in undersea warfare. Flimmer could be deployed independently with its own ISR mission or in coordination with a submarine with a potential for recovery.

The idea of recovering a deployed asset from a submarine is far from a revolutionary idea. The U.S. Navy's SEAL teams regularly coordinate operations with submarines using an attachable dry dock shelter or built-in lockout shelters. UAVs deployed to support these missions would allow the SEAL teams advanced "eyes on target" and could even reduce the amount of equipment they need to carry such as long-range communications equipment. UAVs deployed in this fashion could be recovered by the SEAL team upon rendezvous with the submarine or, if requiring greater secrecy, a submersible UAV like Flimmer could be used and recovered in a submerged condition.

UAVs offering submerged recovery could be used in a plethora of ways beyond deployment alongside a special forces unit. In its very nature, a submarine maintains its value by preserving its ability to avoid detection. A UAV that is recoverable in a submerged state could perform all ISR missions that other UAVs perform, but it would avoid the need of the submarine to maintain communication throughout its time of flight. Using a hydrogen cell, as has been tested in UAVs at the Naval Research Laboratory, a UAV could achieve an endurance in excess of 24 hours. At a cruise speed of 30 knots, this UAV could perform missions covering 720 nautical miles. To put this in perspective, a submarine could launch the UAV from the Persian Gulf with a mission in Baghdad and the potential for recovery. Further, a UAV deployed from the Persian Gulf could transit all the way to the Mediterranean and rendezvous with a ship there. While these are both hypothetical transits, they demonstrate the innumerable opportunities that an increased battery life presents in terms of potential missions and the ability of a boat to perform missions with extreme standoff ranges.

Large standoff ranges do not necessarily mean UAVs need to perform missions at such distances. Short distance missions may provide even better strategic advantage, especially with the continuing development of swarm technology associated with UAVs. Blackwing has already demonstrated capability to perform swarm work with undersea assets and the capability could readily translate to coordination with multiple aerial vehicles. Because the Blackwing only has a battery life of 60 minutes, it would be an ideal vehicle to deploy against a surface vessel as a part of a swarm. Multiple UAVs deployed against a surface vessel would be able to neutralize the vessel using non-lethal force, with an electromagnetic pulse or jamming payload, and avoid destruction by any anti-air assets because of their small, insignificant radar signature. While this is just one example of swarm tactics, the addition of this feature to the Navy's UAV squadron would strengthen an already expansive repertoire of capabilities.

CHAPTER 2:
Background

In this chapter, we examine prior research and developments in the fields of vehicle tracking, path planning, and extrapolation for intercept. This overview, based on previous work by the author [9], is presented in three distinct sections: The Observer, The Vehicle, and Extrapolation Methods.

2.1 The Observer

2.1.1 Problem Definition

Randomized path optimization is a relatively new field, particularly with its application to UAV path planning. Prior to the path optimization problem, consider the problem of predicting the terminal location of an unmanned vehicle given measurements of its position over the course of its trajectory. Let the unmanned vehicle's position as a function of time, t, be $r(t) = [x(t), y(t)]$. The time of termination is t_f, such that the terminal location in Cartesian coordinates is $r_f = [x_f, y_f]$, where $x_f = x(t_f)$, $y_f = y(t_f)$. The goal of the estimation algorithm, the third-party viewer, is to predict the terminal location with maximized probability, whereas the goal of the tracked vehicle is to obfuscate its path such that the estimation algorithm assigns minimal probability to its actual terminal location.

Although initially described to particularly apply to the deployment of a UAV from a submersible vehicle, the problem can be applied to any unit that launches a UAV and does not want it's terminal location to be easily predicted. For a submarine, compromising the terminal location can lead to the identification and tracking of the unit, failure of the mission, and endangerment of the submersible and its crew. Ground troops, for similar reasons as submarines, launch UAVs, allow them to perform their mission set, and recover them after a fixed period of time. Thus, the path obfuscation of aircraft can be applied universally to any unit that employs UAVs and wants to conceal their future location at when they recover the aircraft.

To aid in the universal application of this study to all warfare communities and units, a generic set of terms will be used to describe the components of this problem. The aircraft or UAV will be referred to as the “vehicle.” The individual or unit that launches the vehicle will be referred to as the “operative.” The third-party viewer that is attempting to track the vehicle and determination the location of the operative will be referred to as the “observer.” The vehicle will be deployed by the operative and assigned an objective to accomplish. After the accomplishment of the objective, the vehicle will then return to the operative. This process will be referred to as the “mission profile.”

Assumptions and Unknowns of the Observer

In order to define the observer and his capabilities, a series of assumptions is used to limit the scope of his knowledge. The most fundamental assumption for the observer is that he will have full view of the vehicle throughout the entire duration of its flight including lunching point. This is assumed to be a radar “view” of the vehicle such that the observer is also assumed to know the entirety of the path that the vehicle has already traveled. This prior path is therefore available to the observer for the analysis of prior behavior and can be used as data for the prediction of the vehicle’s terminal location. This data will be collected at varying time intervals and, just like a radar would use distance traveled between samplings, the observer will therefore be able to predict the velocity of the aircraft from its measurements. The final assumption for the observer is that the maximum achievable time of flight (TOF) of the vehicle is known. Thus, with calculated vehicle velocity, the observer can predict the maximum range of the vehicle. The velocity of the vehicle is not an assumption, however, so the maximum range of the vehicle could oscillate over time according to an undulating velocity profile.

While there are many assumptions made to define the role of the observer in this problem, there are still key components that are unknown. The terminal location of the vehicle, for example, is not only an unknown component of the problem, but the key parameter that this problem is focused around. There are two other unknowns that a vehicle can use to its advantage to cloak its terminal location. The most important variable that the vehicle and operator have to work with is the termination time of the mission. Regardless of whether or not the observer knows the maximum time of flight, the operator is free to terminate the vehicle’s flight at any point up to and including the maximum time of flight. Further, the path of the vehicle is made more difficult to predict in that the observer does not have any prior knowledge as to the mission profile or objective of the vehicle. Thus, the combination of a varying termination time and unpredictable flight path works to the advantage of the operator and is used in the obfuscation of the vehicle's terminal location.

2.1.2 Recursive Bayesian Estimation

From the viewpoint of the observer, the estimation problem can be likened to traditional target tracking [10]. This section provides the background on Bayes' theorem, Bayesian inference, and single target tracking with recursive Bayesian filtering. The latter is the combination of a recursive update of a likelihood function in order to predict the probabilities of the termination of the vehicle. A combination of all of these parameters leads to the analysis and solution of the target tracking problem with recursive Bayesian filtering.

Likelihood Principle

This principle is a specific version of the conditionality principle which states that inference should be solely based upon those events that actually occurred in the experiment [11]. Thus, only the observable data collected by the observer about the position of the vehicle should be used in the development of the inference as to its future position. According to Stone, the likelihood principle is only slightly more specific in that the principle holds that the likelihood function evaluated for all the observations is the entire picture of the information [11]. The likelihood function is therefore used to describe the observations of the position of the vehicle and is relied upon as the sole source of information for the prediction of the future position of the vehicle.

Bayes' Theorem

Although it bears his name, the interpretation of Bayes' theorem used in this paper was actually developed by the work of Laplace [11]. The theorem describes the method by which previous data can be resolved with current data to produce a future prediction. Prior information collected by the observer provides knowledge about the vehicle in the parameter space and the evaluation of the likelihood function provides a current set of position data for the vehicle. The combination of these two functions to produce a state set, according to Bayes' theorem, should be accomplished by multiplying the two functions together and then normalizing their result to produce a posterior probability distribution. In this manner, Bayes' theorem is utilized to assimilate the data collected by the observer in this problem.

Recursive Bayesian Estimation Algorithm

The Bayesian estimation procedure is as follows. Let $\hat{r}_f = [\hat{x}_f, \hat{y}_f]$ be the terminal location estimate and $S = [r(t_1), r(t_2), ...] = [r_1, r_2, r_3 ...]$, where r_k is the measured position at time t_k, denote the set of observations of the unmanned vehicle's position in time. The posterior probability density, $p\left(t_k, r_k\right)$, of a terminal location r_f given measurements S is updated recursively following the update rule seen in Equation (2.1).

$$p\left(t_k, r_k\right) = \frac{1}{C} L_k\left(s_k\right) p^-\left(t_{k-1}, s_{k-1}\right) \tag{2.1}$$

where p^- is the prior probability density, $L(s_k)$ is the likelihood function, and $p(t_k, r_k)$ is the resulting posterior probability density function. The constant C normalizes the posterior such that it has unit integral. Evolution of the probability density function in time is accomplished in the Bayesian framework by Equation (2.2).

$$p\left(t_k, r_k\right) = \frac{1}{C} L_k\left(r_k\right) \int p\left(t_k, t_{k-1}\right) p\left(t_{k-1}, r_{k-1}\right) dr_{k-1} \tag{2.2}$$

We define the transition probability density function, $p\left(t_k, t_{k-1}\right)$, as a Gaussian distribution corresponding to a random walk of the target. From the recursively updated probability density function, a predicted terminal location of the aircraft can be determined. The maximum of this function is used as the estimated terminal location of the aircraft. As the functions updates, therefore, so does the estimated terminal location of the aircraft as the maximum of the density function.

2.1.3 Likelihood Function Development

The utilization of Bayesian filtering implies the use of a likelihood functions; they are a necessity in the use of filtering and recursive assimilation. Various likelihood functions were developed to cover the different mission profiles of vehicles. Each likelihood function offers a different probability distribution and corresponding density function.

Cone Likelihood Function

One method that was studied to produce a likelihood function was the use of a likelihood function that predicted the vehicle would terminate at the maximum edge of its range. A cone-shaped likelihood function is the solution to this specific mission profile design because it allows for a constantly increasing probability up until the maximum termination point of the vehicle. This is the most basic likelihood function used in this paper to describe a mission profile. The cone likelihood function is described according to the Equation (2.3).

$$L = h\sqrt{(x - m)^2 + (y - n)^2} \tag{2.3}$$

In Equation (2.3), h is the slope of the cone and the cone's center is defined by $r_k = (x_k, y_k)$. Figures 2.1 and 2.2 present the graphical illustrations of both the probability distribution and the likelihood function generated by Equation (2.3). Of note is the maximum likelihood location at the maximum range of the aircraft.

The use of a function that is statically defined as describing a vehicle that travels in a straight line from its launch point to its termination point is improbable and unrealistic. In reality, vehicles have varying courses to accomplish their mission and a straight line mission profile only applies to vehicles focused on resupply and transit. Thus, while this likelihood function is a viable option, it does not accurately describe the mission profiles that are associated with this paper.

Blackbody-based Likelihood Function

This likelihood function was inspired by the decay and emission of blackbody radiation. The purpose for the development of this function was to develop a predictor that did not have zero probability at the origin nor did it make the assumption that the aircraft had a straight-line mission from launch to termination. The blackbody likelihood function models a mission profile that has a steeply increasing probability of termination up until approximately 35% of maximum range with an exponentially decreasing probability afterwards towards the maximum range of the aircraft. The blackbody likelihood function is described according to Equation (2.4).

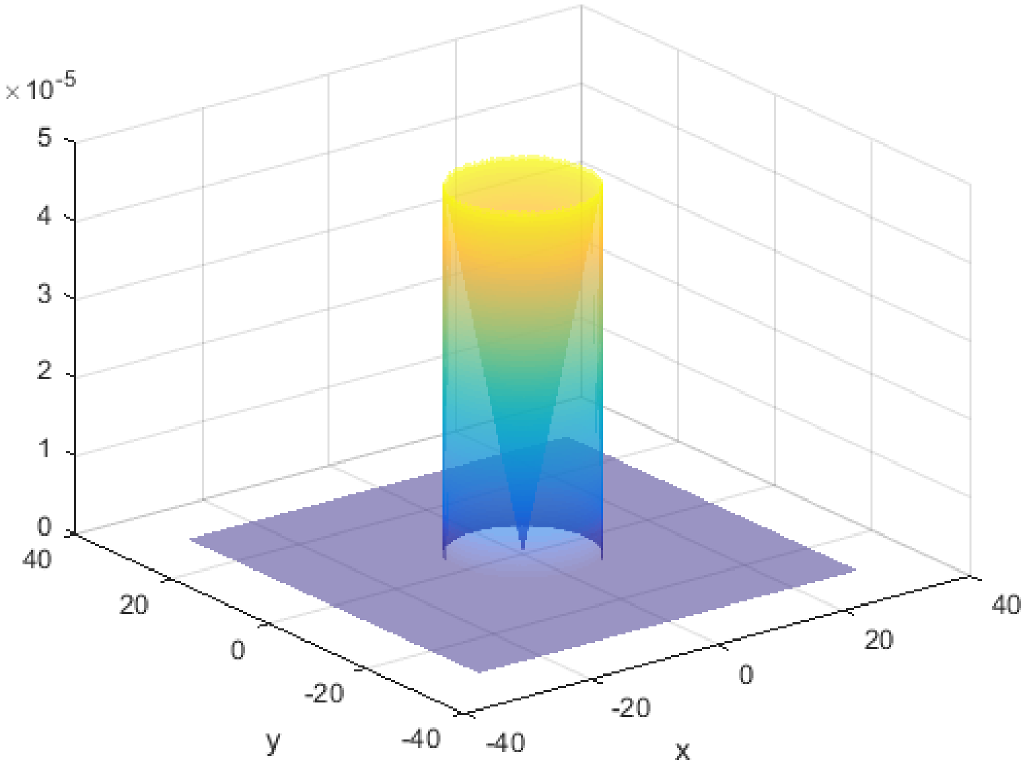


Figure 2.1. Likelihood function illustrating how the position measurements assign likelihood to spatial termination points as a function of the vehicle's maximum remaining range

$$L = \frac{1}{C}\frac{(R-a)^2}{e^{\frac{R}{R_0}}} - b \tag{2.4}$$

The likelihood function produced by Equation (2.4) is motivated by a blackbody radiation curve where R is the distance from the vehicle's current position, C is the normalizing constant such that L has integral 1, R_0 is the maximum range of the UAV, a is a constant parameter to ensure a non-zero probability density at $R = 0$, and b is a shaping parameter to ensure L exists at $R = 0$.

Figures 2.3 and 2.4 present the graphical illustrations of both the probability distribution and the likelihood function generated by Equation (2.4). Of note is the maximum probability

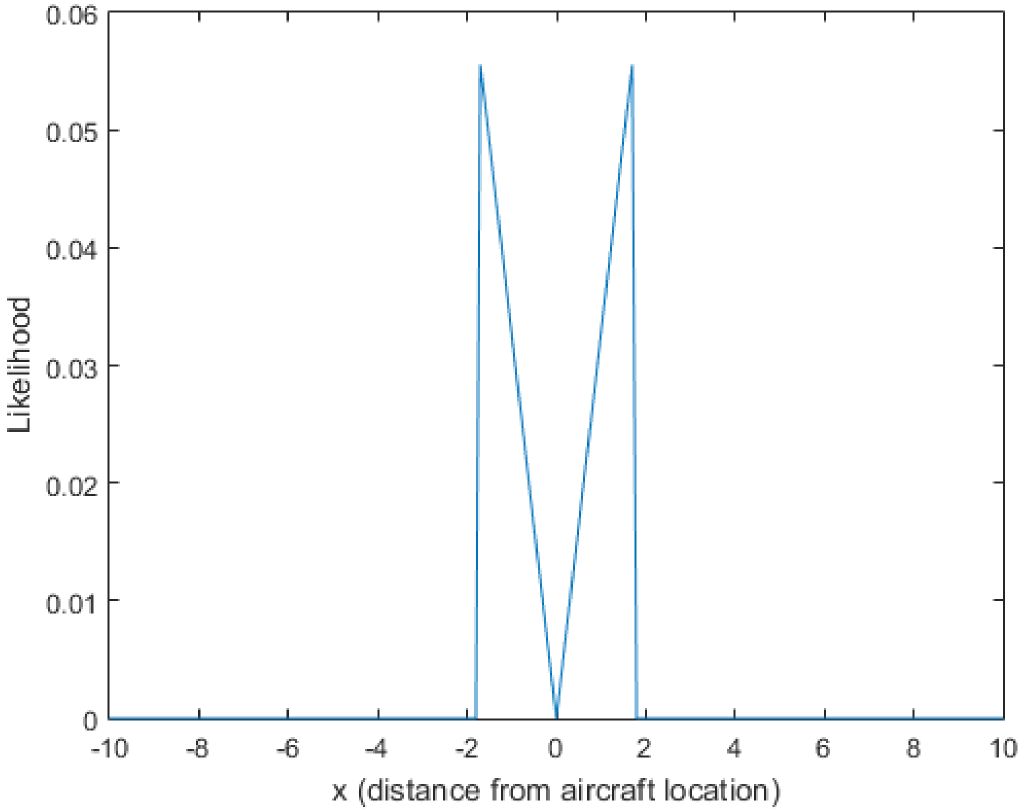


Figure 2.2. Cross-section of Figure 2.1 showing the two-dimensional spatial likelihood distribution of an aircraft terminal location

location in comparison to the maximum probability previously seen in Figure 2.1.

The use of the blackbody radiation function has ambiguous shaping characteristics and allows little modification to be made to create specificity for a mission profile. A desirable function would allow for the shaping of the likelihood function around the maximum probability point. To meet this need, the peak of the blackbody curve was derived so that it could be shifted to fit the maximum flight time of the given vehicle. Thus, the probability distribution of the blackbody function was used to generate the Driven Blackbody Function found in Equation (2.5).

$$L = \frac{1}{C}\frac{(R-a)^2}{e^{\frac{R}{K}}} \tag{2.5}$$

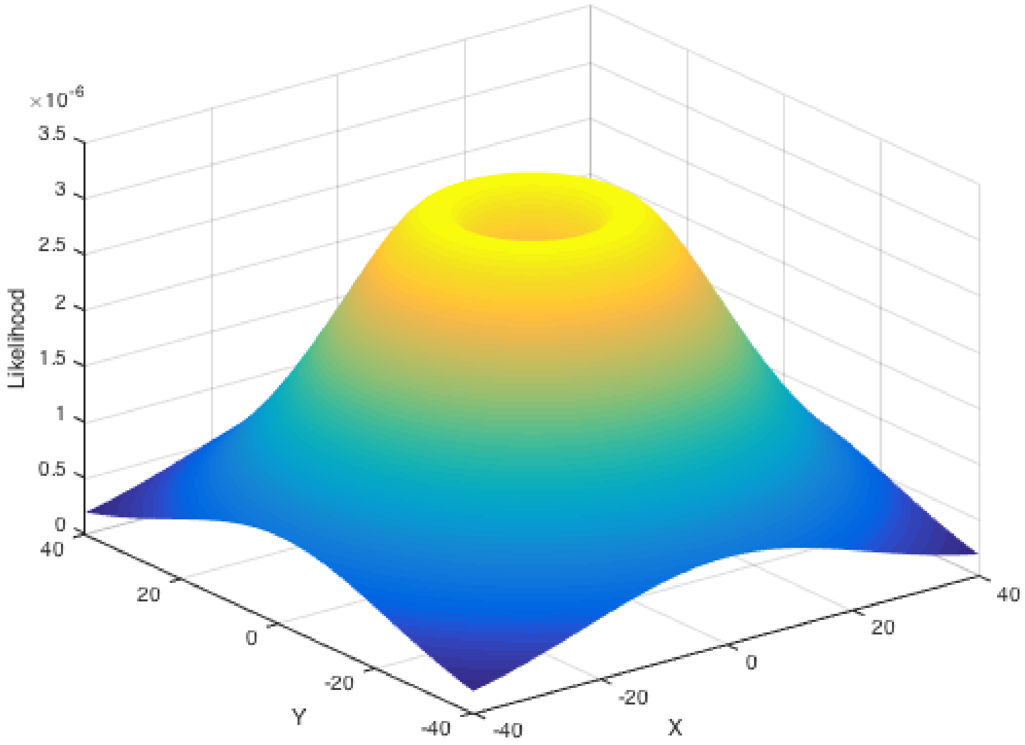


Figure 2.3. Likelihood function illustrating how the position measurements assign likelihood to vehicle termination points

In this equation, the parameter K is the shaping factor that is a function of the design point R_D. The design point can thus be modified to fit differing vehicle's mission profiles. Rearranged to include R_D, the formula becomes Equation (2.6).

$$L = \frac{1}{C}\frac{(R-a)^2}{e^{\frac{R}{\frac{R_D+a}{2}}}} \tag{2.6}$$

2.1.4 Terminal Location Prediction Results

We now discuss the different probability densities presented by the varying likelihood functions and the importance of using the blackbody likelihood function.

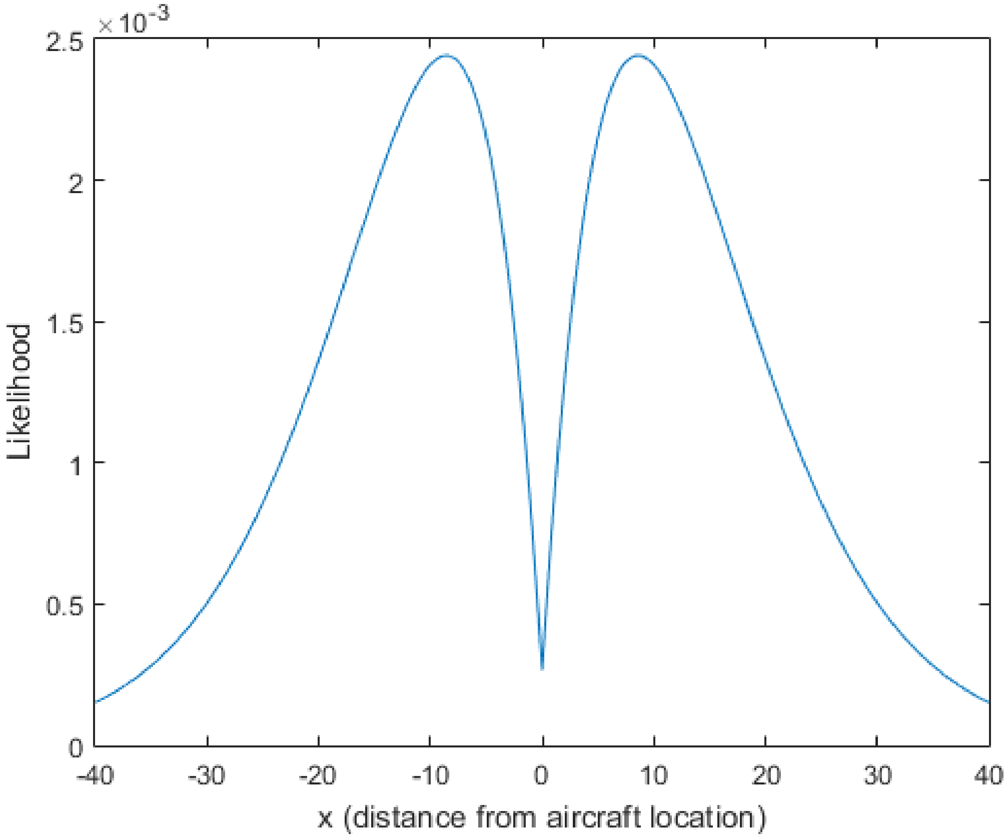


Figure 2.4. Cross-section of Figure 2.3 showing the likelihood distribution for an aircraft terminal location

Figure 2.5 illustrates simulation of a UAV traveling within an environment and an observer assimilating position measurements. Four snapshots of simulations using the likelihood function from Equation (2.3) (left column) and (2.4) (right column) illustrate differences in the terminal prediction algorithms. In each simulation, the UAV travels in a looping trajectory illustrated by the dashed white line. The color scale of each plot illustrates the spatial distribution of the probability density function predicting the terminal location of the UAV. As described by the color bars on the right-hand side of each figure, yellow corresponds to higher probability density and likelihood that the UAV will terminate its trajectory at the given location, whereas blue indicates the opposite. It is interesting to note that both likelihood functions are similar in their probability density in that they are concentrated at the termination point of the aircraft. On the other hand, it is important to take into account that the blackbody radiation model includes a high probability at the termination point of the aircraft throughout the simulation. The cone model, however, varies its likelihood greatly throughout the simulation and, shown in Figure 2.5 (b) and (c), has low probabilities at the termination point during multiple snapshots of the simulation.

The developed blackbody likelihood function allows for the prediction of vehicle termination points with an accurate distribution of probabilities across the entire mission profile. The ability to shape the likelihood function based upon past vehicle data allows for an increase in capabilities of the observer and therefore an added component of complexity and difficulty for the operator to obfuscate his path using future control algorithms.

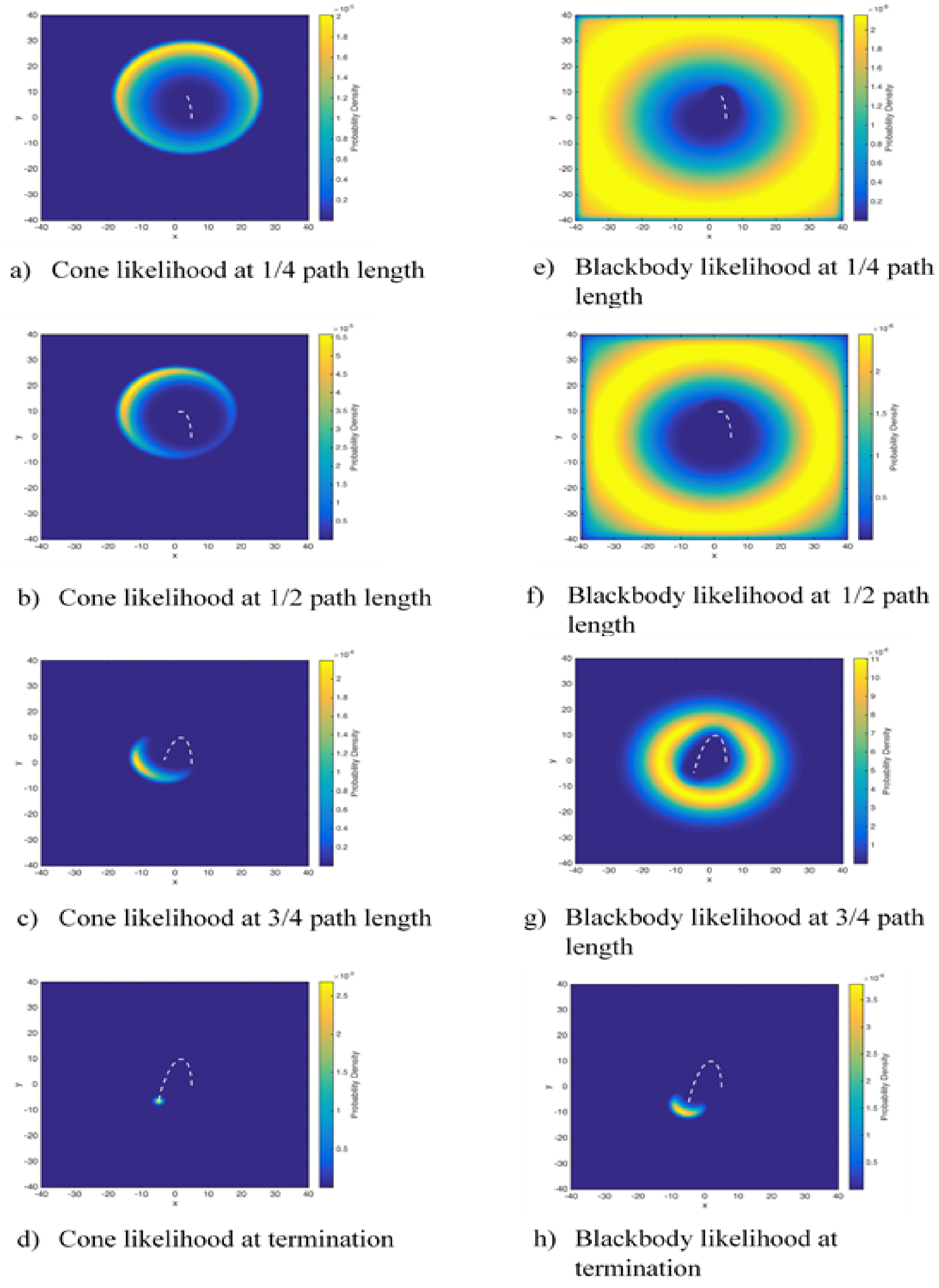
2.2 The Vehicle

2.2.1 Path Development

With the development of an algorithm that accurately predicts the termination point of a vehicle, the issue of path obfuscation can be addressed. In order to choose the path that minimizes the vehicle's chance of having its termination point predicted, one must first develop a set of potential paths. From this set of paths, the path that best mitigates detection will be chosen as the path for the aircraft to follow. For each mission profile and set of preset conditions, a specific set of paths is generated. Given that every vehicle has a start point, q_{init}, and an endpoint, q_{goal}, an infinite set of possible paths exists in between them.

In order to scale down the problem from an infinite number of paths, a polynomial path generation method was selected with the stipulation that time and speed be held constant for a given path. Generation of a path is dependent upon the specification of a waypoint, q_{wpnt}, or waypoints, q_{init}, and q_{goal}. Let each point be defined by an x-coordinate and y-coordinate where the point q_{init} is defined such that $q_{init} = [x_{init}; y_{init}]$. Let the matrix $\tau(s)$ be defined as the set of all coordinates in a generated trajectory where s is a timing function given for a specific mission profile. A polynomial generation method allows for the matrix $\tau(s) = [x(s); y(s)]$ to be a continuous map from 0 to 1. The number of points specified in the path determines the order of the polynomial.

For the purpose of this study, a second-order function was the polynomial used because three points $\left(q_{init}, q_{wpnt}, q_{goal}\right)$ were specified. Incorporated in the matrix $\tau(s)$, the three waypoints satisfy the conditions $\tau(0) = q_{init}$, $\tau(0.5) = q_{wpnt}$, $\tau(1) = q_{goal}$. The remaining



a) Cone likelihood at 1/4 path length

e) Blackbody likelihood at 1/4 path length

b) Cone likelihood at 1/2 path length

f) Blackbody likelihood at 1/2 path length

c) Cone likelihood at 3/4 path length

g) Blackbody likelihood at 3/4 path length

d) Cone likelihood at termination

h) Blackbody likelihood at termination

Figure 2.5. Evolution of the posterior probability function for vehicle position as measurements are assimilated in time

points in the x-plane are the solution of the second-order polynomial $x\left(s\right) = as^2 + bs + c$, where the coordinates for the y-plane are calculated in the same manner. The solution for matrix τ is best achieved through matrix algebra and can be seen in the development from Equations (2.7) to (2.11).

$$T\left(s\right) = \begin{bmatrix} x_{init} & x_{wpnt} & x_{goal} \\ y_{init} & y_{wpnt} & y_{goal} \end{bmatrix} \tag{2.7}$$

$$T\left(s\right) = M[S] \tag{2.8}$$

$$T\left(s\right) = M \begin{bmatrix} s_{q_{init}}^2 & s_{q_{wpnt}}^2 & s_{q_{goal}}^2 \\ s_{q_{init}}^1 & s_{q_{wpnt}}^1 & s_{q_{goal}}^1 \\ s_{q_{init}}^0 & s_{q_{wpnt}}^0 & s_{q_{goal}}^0 \end{bmatrix} \tag{2.9}$$

In Equations (2.7) and (2.9), the sizes of the matrices will vary as a function of the number of points specified in the path. If n points are predetermined, $T(s)$ will be a $2 \times n$ matrix while matrix S will be of size $n \times n$. Because the matrix dimensions will always agree, one can use matrix algebra to solve for the unknown matrix, M.

$$M = T[S]^{-1} \tag{2.10}$$

With the solution of Equation (2.10), one can then solve for the set of all trajectories τ. The final solution shown below in Equation (2.11) is a set of x and y-coordinates modified by the timing function s to dictate at what point in time the vehicle arrives at any point (x, y).

$$\tau = M \begin{bmatrix} s^2 \\ s \\ 1 \end{bmatrix} \tag{2.11}$$

2.2.2 Trajectory Implementation

In this work, a quadratic function was used to describe the trajectory and the number of waypoints tested was bounded to a set of one hundred from a 10 × 10 grid of x and y coordinates (shown in Figure 2.6). Additional simplicity was achieved in the solution and implementation of the problem by using a constant timing function, s, and therefore a constant velocity throughout the time of flight. A lack of velocity variation allows for the third-eye observer to have a better probability of predicting the aircraft's terminal location and it brings simplicity to the trajectory.

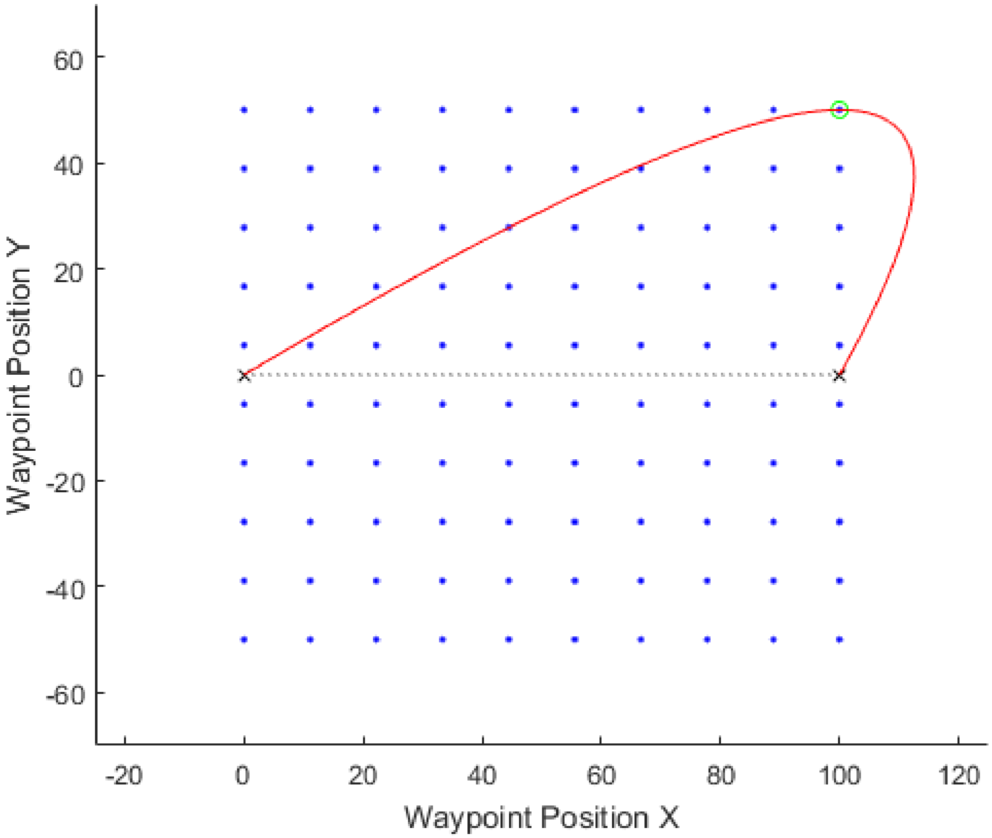


Figure 2.6. Set of all possible waypoints for the given study and an example trajectory through one waypoint

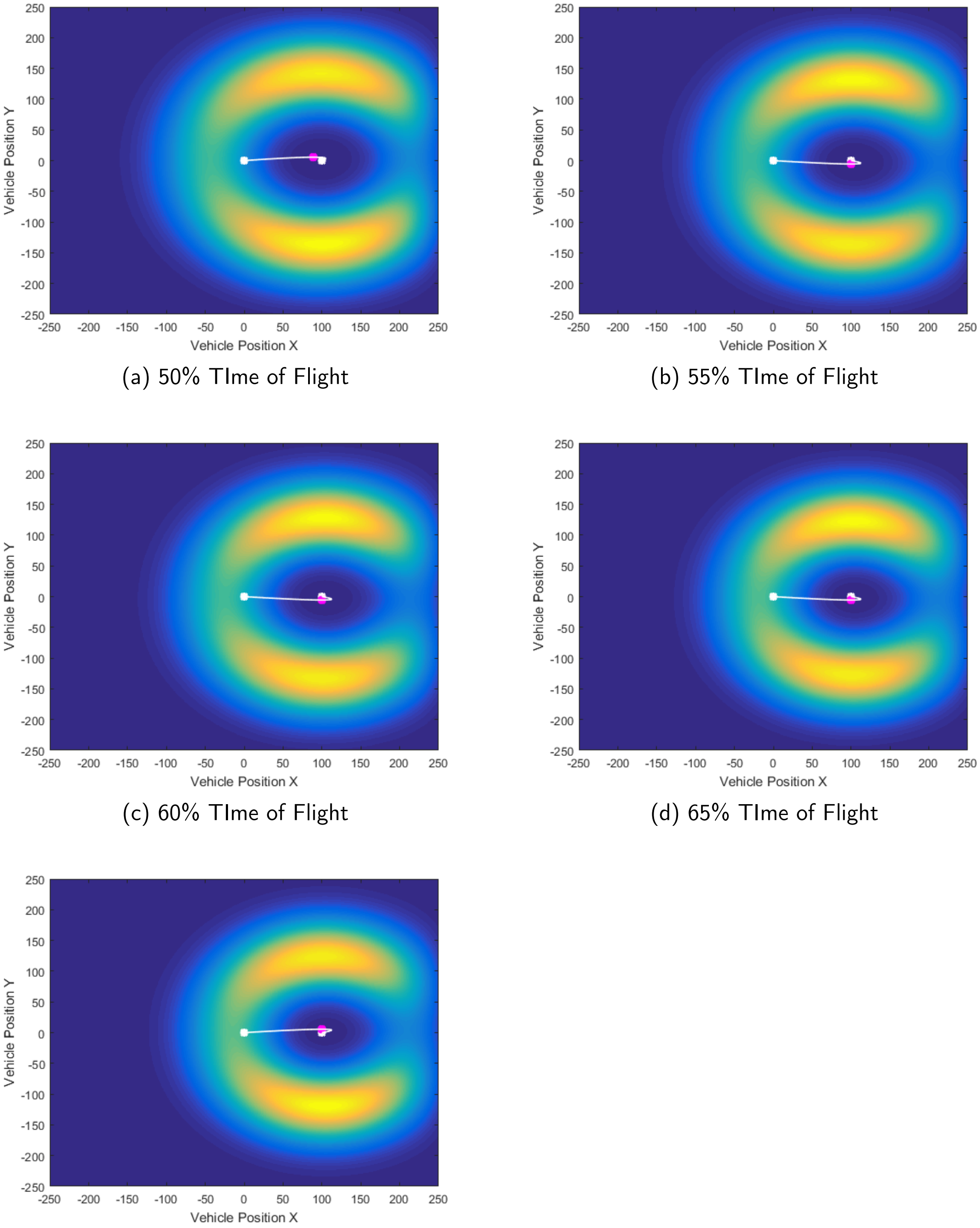
2.2.3 Path Obfuscation

The trajectory generation method presented in Subsection 2.2.1 was tested against the observer prediction method developed in Section 2.1. The key features used to “fool” the observer were the path and variation in the time of flight of the vehicle. Because the observer has no indication as to how long the vehicle will fly, variation in the maximum time of flight of the vehicle is key to the obfuscation of the vehicle’s terminal location. Five different time profiles were tested ranging from a profile where the vehicle used 50% of its maximum possible flight to a vehicle that used 70%. With 100 potential waypoints and five different time profiles, the set of infinite paths was narrowed down to a set of 500 possible paths. In order to determine which path provided the minimum probability of detection, tests were compared to one another using the Kullback-Leibler Divergence (KL Divergence). The KL Divergence is a measure of the disparity between two probability distributions. Each probability distribution generated from a given trajectory was compared to a baseline normalized Gaussian distribution for consistency of measurements. The images shown in Figure 2.7 are the best obfuscated paths for each time of flight profile. Each path was selected because it had the highest KL Divergence value from the potential waypoints at that time profile.

Figure 2.8 shows the comparison between all of the KL Divergence values for the 500 potential paths. The vertical black and pink lines show the vehicle’s initial and goal points, respectively. The vertical green line indicates the waypoint that provides the minimum probability of detection. The figure further shows that the time profile that minimizes the probability of detection is the one in which the vehicle only used 55% of its maximum possible time of flight.

2.2.4 Results

KL Divergence values indicate that mission profiles that utilize less than 60% of the vehicles maximum time of flight have the lower probabilities that their terminal location will be predicted. Trajectories with waypoints near the terminal location and a path that continues beyond it have lower probabilities that their terminal location will be predicted. Thus, the greatest path obfuscation comes from the observer’s inability to determine when a vehicle will terminate and the assumption that the vehicle will generally continue on its current trajectory and not terminate near a point that it has already passed.



(a) 50% TIme of Flight

(b) 55% TIme of Flight

(c) 60% TIme of Flight

(d) 65% TIme of Flight

(e) 70% TIme of Flight

Figure 2.7. The paths that best minimize a vehicle's probability of detection for varying time of flight profiles

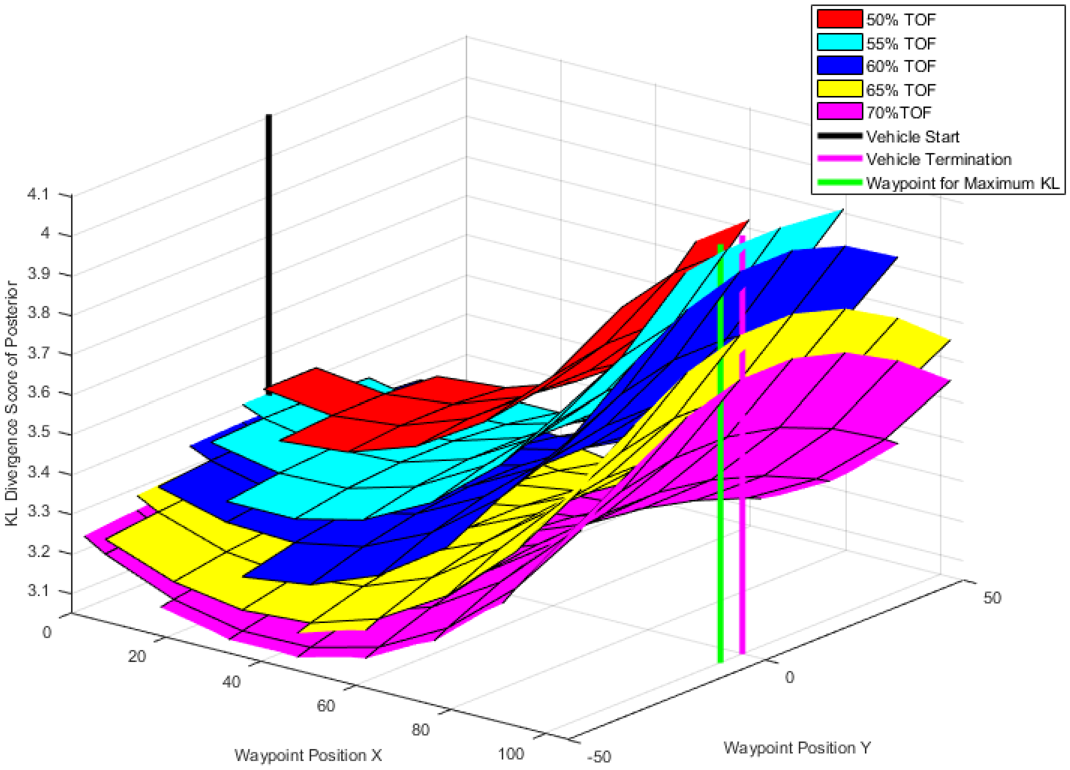


Figure 2.8. Surface plots of all KL Divergence values for each test separated by percent time of flight

2.3 Extrapolation Methods

Previous work in the field of extrapolation for the prediction of a vehicle's location has almost exclusively been done for future predicted position. Many works are available addressing numerical methods for extrapolation; some prominent works can be found in sources [12], [13], and [14]. The work in this thesis is most closely related to sources [15] and [16], but the interested reader is referred to works [12], [13], and [14] for further reading.

Soechting et al. pursued an analysis of the predictive contributions that analytical models offer to the domain of smooth pursuit [15]. Applicable to this work is their use of smooth velocity, speed, and direction changes throughout their analysis and the application of their analysis to them. The work revealed that with a constant speed but variable direction, the target motion was very difficult to predict. In particular, their use of a Kalman filter generated “a predictive component whose importance could fluctuate” [15]. This work was performed on sinusoidal paths and leaves the need for an analysis of extrapolation and prediction for polynomial-based paths.

Looking towards the field of predictive path, the previous background on Bayesian filtering largely covers this domain for extrapolation and likelihood of positioning. The work of Thorvaldsson and Bandi illustrates the implementation of the Bayesian and Kalman filtering methods in order to generate a smooth path prediction for the motion of a vehicle [16]. Their use of these methods, while successful and unique, lacks the far-forward reaching component of the prediction that this paper’s current work requires.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Methodology

In this chapter we discuss the specific issues to be addressed by this thesis and how they are going to be approached. Previously examined in Chapter 2, the terminology will remain consistent as we examine the experimental setup moving forward.

This chapter has been divided into two sections as we look at two distinct, novel approaches to advancing capabilities surrounding randomized path optimization. Chapter 3 first examines how to improve the predictor and path generation models through the incorporation of wind and a three-dimensional problem space.

Following the discussion of these improvements, we examine a unique redirection of the predictor model in its application to predicting the location of the operator in Section 3.1. Using anterior extrapolation from given vehicle location information over some span of the vehicle's flight up until its termination, the predictor will produce a probability distribution offering information as to where the operator was when he launched the vehicle.

3.1 Incorporation of Wind

Vehicle flight is so largely affected by wind that it is crucial to model it properly. This approach will seek to properly incorporate wind as a vector field so as to equally improve the predictor method and the path obfuscation component of the problem.

3.1.1 Assumptions

For the purpose of this work, simplifying assumptions were made to both decrease potential computation time and allow for universal applicability. The assumptions made are listed below:

1. Uniform, time-variant wind field.
2. Vehicle's maximum time of flight (TOF) is known.
3. Component directions of the wind add or subtract directly from the vehicle's speed. No preference is given to vehicle aerodynamics with respect to orientation relative to

the wind vector.

4. Wind only acts in the x-y plane and has no component in the z-direction.
5. Wind field does not vary as a result of vehicle altitude.

These assumptions allow for consistency across various different vehicle platforms and between the predictor and observer methods. The implementation of the above assumptions in no way reduces the legitimacy of the data that is produced as a result.

3.1.2 Approach

Previously held constant throughout flight, the modification of the vehicle's velocity will be a function of the vehicle's orientation at the previous time-step and the direction of the wind vector. As a result of our assumptions, we can directly relate the component vectors of the wind and the vehicle. These relations can be seen in Equations (3.1) and (3.2).

$$v_{x_{total}} = v_{wind} cos(\theta_{wind}) + v_{vechicle} \cos(\theta_{vehicle}) \tag{3.1}$$

$$v_{y_{total}} = v_{wind} sin(\theta_{wind}) + v_{vechicle} \sin(\theta_{vehicle}) \tag{3.2}$$

In these relations, θ_{wind} is the angle of the wind vector measured counterclockwise from the positive x-axis while $\theta_{vehicle}$ is the angle of the vehicle's vector measured counterclockwise from the positive x-axis. Through generic trigonometric identities, the velocities can therefore be related to one another in order to produce an overall vehicle velocity shown in Equation (3.3).

$$v_{total}^2 = v_{x_{total}}^2 + v_{y_{total}}^2. \tag{3.3}$$

The dynamically changing vehicle velocity adds an increased component of complexity to the predictor as it tries to predict the vehicle's terminal location. Shifting from a generic circular area of probability resulting from the multiplication of the vehicle's remaining TOF and constant velocity, varying velocity now creates an ever-changing region of potential vehicle termination positions. While wind is often the nemesis of aircraft, here it can work cohesively to further obfuscate its path. The path planning methodology, unlike the observer, benefits from the ability to select the best path prior to flight. Because of this and the wind's direction, its incorporation modifies the potential set of waypoints the vehicle can reach as a result of an increased or decreased range.

3.2 Producing a Three-Dimensional Problem Space

The factor of vehicle altitude is important to consider when dealing with aircraft. Traditional aircraft flight would dictate that the vehicle would takeoff at its minimum altitude, fly to a mission altitude, and then proceed to land at a lower altitude. Because of the complexity of this problem, varying mission altitudes, and unpredictable paths, we must define a new problem space and a means by which it can be approached.

3.2.1 Assumptions

In order to frame the problem and limit computational complexity, we first make simplifying assumptions. These assumptions are listed below:

1. The vehicle starts and terminates at an altitude of 0 feet (sea level).
2. The aircraft floor (minimum altitude) is sea level (i.e., there is no potential for a vehicle to fly below sea level).
3. The vehicle has known maximum rates of climb and descent known by the observer.
4. The vehicle has a maximum operating altitude (ceiling) known by the observer.
5. The vehicle is going to land (i.e. the vehicle must descend to an altitude of 0 at the conclusion of its flight).

As a result of these assumptions, the problem space is bounded. Without a bounding of the problem space, we could not accurately predict the effectiveness of the observer, nor could we demonstrate the best obfuscation methods available to the vehicle and the operator.

3.2.2 Approach

Unlike the incorporation of wind, the expansion of the problem into the z-dimension does not allow for similar integration in the observer and path planning domains. Thus, we will examine how to incorporate the third dimension in each.

3-D Predictor Model

In order to model the incorporation of altitude z in the predictor, we must account for variation in vehicle flight. To best achieve the prediction of the vehicle's termination and avoid false predictions, we look to use a piece-wise view of the problem space.

$$\begin{cases} Launching\ or\ Terminating & 0 \leq z \leq 0.20(VehicleCeiling) \\ Ascent\ or\ Descent\ Profile & 0.20(VehicleCeiling) < z \leq 0.50(VehicleCeiling) \\ Mission\ Space & 0.50(VehicleCeiling) < z \leq VehicleCeiling. \end{cases}$$

We use this piece-wise function to bias the predictor towards predicting vehicle termination at lower altitudes while maintaining the traditional prediction method in the "mission space" domain. We will use this function and progressively iterate over a field of potential probabilities to determine which set of parameters best biases the density function towards the proper vehicle termination location. Incorporating both the start and ending profiles of the vehicle's flight allows for the predictor's immediate incorporation in the tactics necessary for Section 3.3.

Path Planning in 3-D

Using the 3-D problem space to the advantage of the vehicle is key to obfuscation success. In our approach to this domain, we will analyze the vehicle's obfuscation as a result of three potential altitude profiles. Each profile provides different options for accomplishment of the mission and obfuscation of path through altitude. Each altitude profile will include an immediate climb to mission altitude and a period of flight at that altitude until the vehicle has passed the waypoint and accomplished the mission. Altitude profile 1 will be characterized by a flight path in which the vehicle does not descend to the termination point until it is absolutely necessary as constrained by a safe rate of descent. Altitude profile 2 will include an immediate departure from mission altitude to a 0.5 $(MissionAltitude)$ for as long as allowed by the flight before returning to mission altitude and then descending for termination. Lastly, altitude profile 3 will depart from mission altitude to 0.2 $(MissionAltitude)$. The vehicle will then fly at this altitude until rate of descent restrictions indicate that the vehicle must descend for termination. All three of these profiles will present different issues for the observer and will indicate the best flight profile for vehicle termination obfuscation.

3.3 Anterior Extrapolation for Initial Point Prediction

The extrapolation of vehicle path for the initial (launch) point can be intuitively looked at as a reapplication of the Observer problem. Given a set of known points and an upper-bound for the vehicle's time of flight, we can restrict the vehicle's possible initial positions to a set domain. To accurately predict the vehicle's launch point while accounting for variable times of flight, we must reexamine the application of the Observer algorithm and note how we can improve it for the purpose of predicting the vehicle's initial point. We now discuss the key area that needs to be explored in order to reapply the prediction algorithm properly to achieve accurate anterior extrapolation for initial point prediction.

3.3.1 Optimal Scaling Factor R_D

The utilization of the scaling factor R_D was first introduced in Chapter 2.1.3. The equation in which it is implemented is the driven blackbody likelihood function shown in Equation 2.5. The key component here is the variable K. In its components, K is a function of constant a and scaling factor R_D. This relationship can be seen in Equation 3.4.

$$K = \frac{R_D + a}{2}. \tag{3.4}$$

For larger values of R_D, the constant a becomes insignificant as $\frac{R_D}{2} \approx \frac{R_D+a}{2}$. We will therefore solely focus on the value of R_D. To properly determine the optimal value of R_D for anterior extrapolation, we must eliminate all other factors that contribute to the prediction of the vehicle's termination point. Thus, we will iterate through possible R_D values varying percentages of the maximum range with a generic vehicle path, no wind, and no altitude considerations. From this, we will examine the location of the maximum probability in the posterior function at three different points in predictive time: 50% of path seen, 75% of path seen, and 100% of path seen. The percent of R_D that produces the lowest average distance from the true initial point will be used to compute all further results. The successful implementation of R_D is crucial to the accuracy of the anterior extrapolation method and its feasibility as a reliable algorithm moving forward.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4: Results

Building off the processes outlined in Chapter 3, this chapter displays and discusses results produced from the execution of the introduced methodology. In Section 4.1, we will first look at how the incorporation of wind in the prediction and path planning algorithms modified the vehicle's potential obfuscation. Section 4.2 will then allow us to analyze how varying altitude profiles improve vehicle obfuscation and successful prediction of termination. Finally, Section 4.4 will examine the reapplication of the predictive techniques to determine a vehicle's starting location.

4.1 Wind-Influenced Results

The initial plan to incorporate the vehicle's response to wind in component form allowed for similar integration in both the prediction and path planning algorithms. When incorporated into the prediction algorithm, the affect of wind, and each of its possible components was examined. To verify the affect the wind had on the successful prediction of vehicle termination, each test was compared against a generic test through the same waypoint with no wind incorporated. The results of each comparison can be seen in Table 4.1.

Test Type	KL Divergence Value	Improvement in Prediction (Percent)
Baseline 80% TOF	8.98	N/A
Headwind of 5 kts	8.64	3.86
Tailwind of 5 kts	10.25	-14.09
Crosswind of 5 kts	9.43	-4.92

Note: A positive percentage indicates an increased probability of detection at its terminal location

Table 4.1. Comparison of wind direction and its effects on termination prediction

From this, we find that only in the headwind scenario did the predictor improve its probability of predicted termination. In terms of the predictor, this indicates that the shift in the maximum probability for the likelihood function creates an unwarranted skewing of the data towards the region in which the wind is blowing. From this, we make the determination that the incorporation of maximum range skewed by wind is overall less likely to predict vehicle termination than the simple use of the vehicle's maximum range. Thus, all of our obfuscation tests were run on the modified and more informed prediction algorithm as opposed to one improperly influenced by wind.

The incorporation of wind into the path planning aspect produced a myriad of different results when tested across a broad spectrum of possible wind directions. The author recognized that testing combinations of every possible direction and wind velocity was unrealistic, and so the test environment was constrained to include wind directions at increments of 45° and three different wind velocities to simulate light (5 kts), moderate (15 kts), and strong (30 kts) winds. The results of these tests are summarized in Figure 4.1.

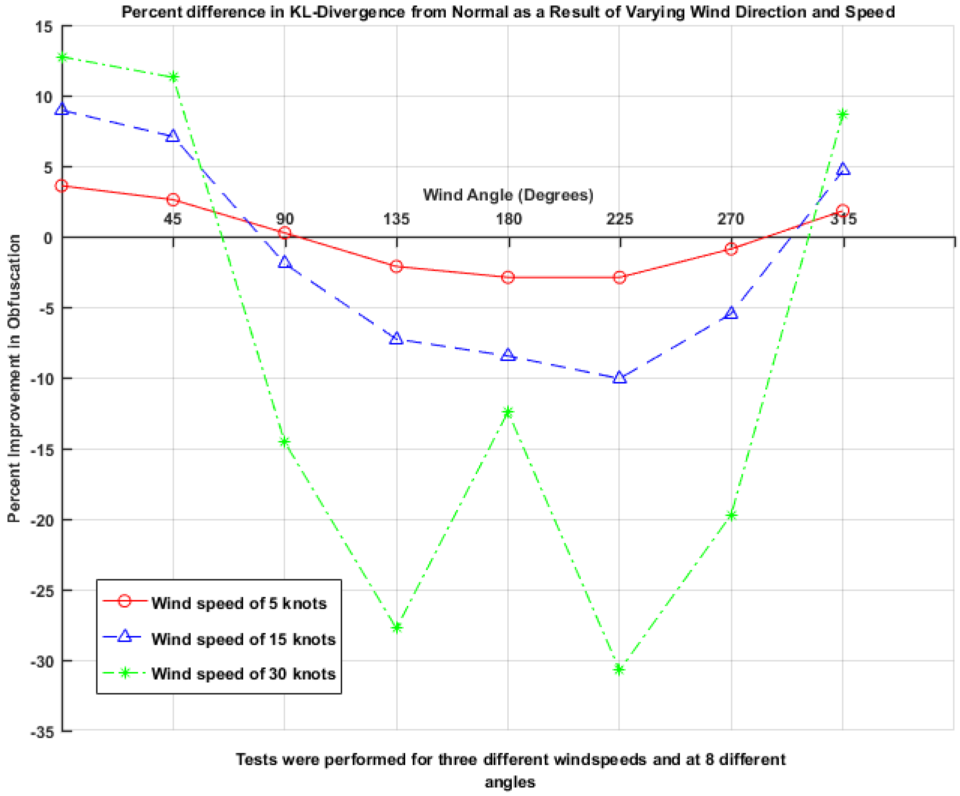


Figure 4.1. Wind results represented in percent deviation from the baseline prediction algorithm

Observing the trends from the different angles of wind, those that are offset by no more than 45° with respect to the vehicle straight line path (directly along the positive x-axis in this case) produce the greatest obfuscation in path. Conversely, outside of that window, the wind at all velocities decreased the vehicle's ability to obfuscate its path. When examining the affect of the wind velocity, one can note that an increased velocity directly correlates to a magnification in the vehicle's ability to obfuscate path. Overall, it can be stated that the maximum obfuscation that results from the wind occurs when the wind acts as a tailwind to maximally increase the vehicle's velocity in the direction of the path.

While the assumption of constant wind velocity has been made, this does not translate directly to a constant velocity for the vehicle. As the vehicle's path changes with respect to the direction of the wind, so does the vehicle's velocity profile. Figure 4.2 is the velocity profile for a vehicle with a cruise speed of 80 knots operating with the maximum obfuscation profile of a wind along the positive x-axis (0°).

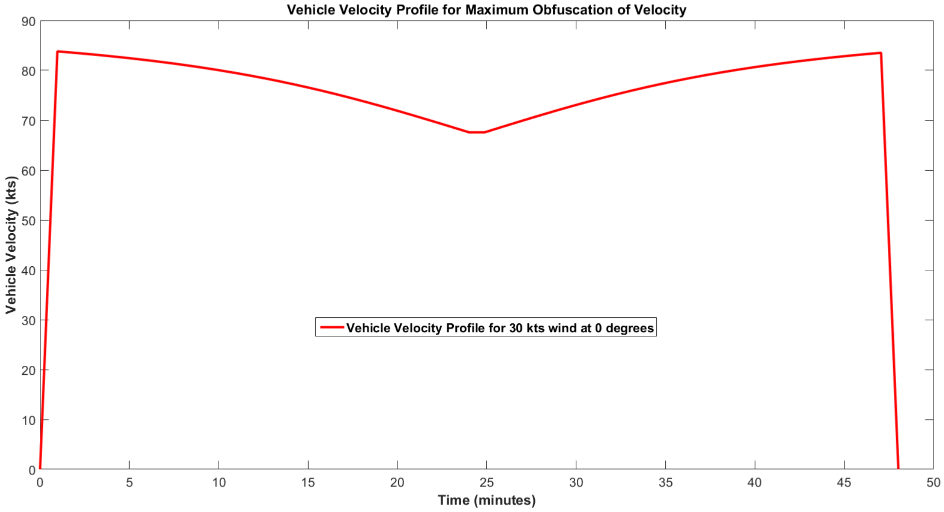


Figure 4.2. Velocity profile that corresponds to the maximum obfuscation wind direction of 0°

Note that as the vehicle changes direction throughout its flight, velocity changes similarly as a result of the new interaction between vehicle path and perceived wind. We know from our data in Figure 4.1 that headwinds produce a less obfuscated path and therefore slower velocities are correspond to poor path obfuscation. Applying this lens to Figure 4.2, we can note that although the vehicle dips below the cruise speed of 80 knots, its increase in velocity at the beginning and end of flight allow for an increase in obfuscation. Thus, a fast overall speed and increased speed near the initial and terminal points of a path correlate to increased obfuscation.

4.2 Altitude Incorporation

The incorporation of altitude proved to provide improvement to the prediction algorithm regardless of which altitude profile, seen in Chapter 3.2.2, was used. Table 4.2 indicates that the longer the vehicle spends at a lower altitude, the greater the performance of the prediction algorithm. Looking at the results from Altitude Profile 3, the vehicle's utilization of its lowest possible flight domain created the worst obfuscated path.

Test Type	KL Divergence Value	Improvement in Prediction (Percent)
Baseline 80% TOF	8.98	N/A
Altitude Profile 1	8.42	6.27
Altitude Profile 2	6.97	22.41
Altitude Profile 3	2.74	69.50

Note: A positive percentage indicates an increased probability of detection at its terminal location

Table 4.2. Comparison of wind direction and its effects on termination prediction

This information is also backed up by the KL Divergence comparisons seen in Figure 4.3. Altitude Profile 3 clearly provides the best obfuscated path over all possible waypoints.

Naturally, one can therefore conclude that the best possible obfuscation occurs when the vehicle waits until the last possible moment to descend and terminate its flight. While the other altitude profiles might not produce the best obfuscation of terminal location, their ability to obfuscate the vehicle's objective, an exploration of future work, could lend credence to an argument that balances

4.3 Obfuscation of Varying Time Profiles

With the knowledge of the best obfuscation characteristics produced by wind and altitude, we then incorporate them to determine the optimal time of flight (TOF) profile that provides the best mitigation of the terminal location. Comparing five TOF profiles (50, 60, 70, 80, and 90%) over 100 possible waypoints, the TOF profile that corresponded to the usage of 70%

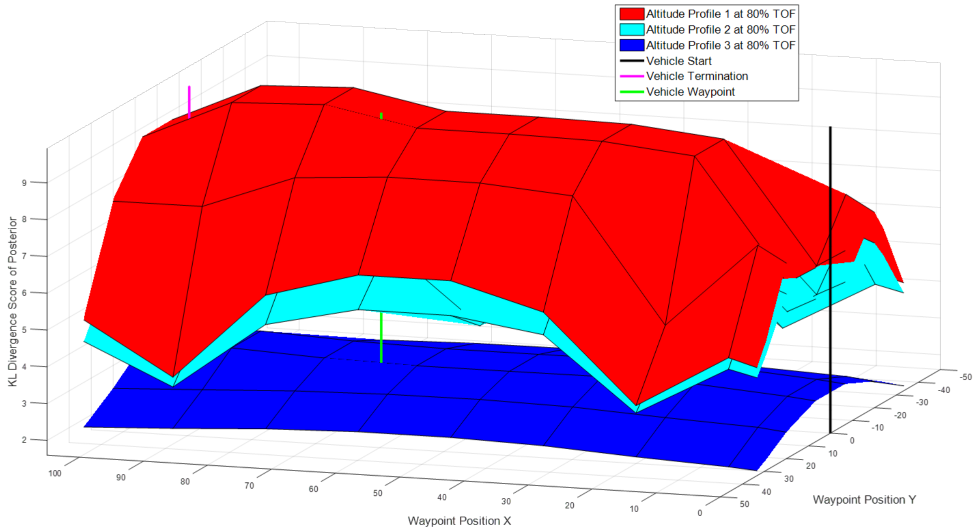


Figure 4.3. Comparison of the KL Divergence values for each altitude profile over a set of 100 waypoints

of the vehicles maximum potential time of flight produced the best possible obfuscation. Compared against the next best profile at 60% time of flight, 70% time of flight produced 14 more possible obfuscated vehicle paths. Additionally, while it did not contain the "best" obfuscated path of all possible paths, the difference between the best obfuscated path and the best path produced by 70% TOF is only 0.08%.

4.4 Anterior Extrapolation for the Initial Point

4.4.1 Selection of Optimal R_D Value

First looking at the modeling that will best produce the optimal anterior extrapolation algorithm, we analyze the ideal R_D scaling value. According to the plan outlined in Chapter 3.3.1, the values in Table 4.3 were computed. Each row is indicative of the percent of the maximum range (R_D) used to shift the peak of the likelihood function. The columns represent the distance that the maximum probability point is from the true initial point at various different percentages of the path seen by the Observer.

Looking at the values in Table 4.3, however, we seek to improve a matching of the algorithm in utilizing multiple R_D scaling values corresponding to the potential time of flight remain-

Percent of R_D	Distance from true initial point at 50% path seen (km)	Distance from true at 75% Path Seen (km)	Distance from true at 100% Path Seen (km)	Average Distance from true (km)
40% of R_D	53.97	74.00	5.39	44.45
50% of R_D	27.68	82.27	7.43	39.13
60% of R_D	36.87	81.39	9.01	42.42
70% of R_D	53.27	80.08	10.31	47.88

Table 4.3. Analysis of maximum probability distance from true initial position of the aircraft with respect to the maximum-range scaling factor

ing. So, we examine the results produced in using one R_D scaling factor while the time remaining is at least 40% of the vehicle's maximum time of flight, with the other scaling factor value (a percent of R_D) corresponding to a terminal time region. The results for these tests can be seen in Table 4.4.

Percent of R_D	Distance from true at 50% Path Seen (km)	Distance from true at 75% Path Seen (km)	Distance from true at 100% Path Seen (km)	Average Distance from true (km)
40% and 50% of R_D	53.97	82.31	7.43	47.90
50% and 30% of R_D	27.68	53.97	3.16	28.27
50% and 40% of R_D	27.68	66.50	5.22	33.13
50% and 60% of R_D	27.68	92.96	9.19	43.28
50% and 70% of R_D	27.68	97.86	10.77	45.44

Table 4.4. Analysis of maximum probability distance from true initial position of the aircraft as a result of a two-fold variation in the maximum-range scaling factor

The tests run in Table 4.4 were inspired by the results found in Table 4.3. Notably, we see that the best scaling factor for initial time of flight is 50% while a scaling factor of 40% produces the smallest separation of predicted start point and true start point. Thus, we explore that relationship and tailor subsequent tests to discern the optimal combination of scaling factors. Results indicated that the optimal combination of scaling factors was an initial scaling factor of 50% with a terminal region scaling factor of 30%. The combination of these two components ensures that the anterior extrapolation algorithm will produce an accurate prediction of the vehicle's starting point.

4.4.2 Starting Point Prediction

The reapplication of the observer algorithm to the prediction of the vehicle start point produced a myriad of results. Given that there are various different potential paths a vehicle could take, we will narrow down the set of paths to six paths that cover the general types of waypoints near the start location, terminal location, close to the centerline of the waypoint set, and substantially deviating from the centerline. The analysis of these waypoints over varying percents of path observed will indicate how successful we can expect to be in predicting the initial point of the vehicle. In order to measure the successful application of the anterior extrapolation algorithm, we will provide two metrics to compare against. We first examined the distance (in kilometers) between the predicted initial point (indicated by the maximum probability on the probability density function) and the true initial point of the vehicle. Further, we look at all of the potential initial points of the vehicle as a factor of its cruise velocity and remaining time. Comparing the prediction to this, we will label the algorithm as a success if it decreases the amount of potential starting points by at least 50%. Figure 4.4 shows the representation of this filtering of initial points visually. In particular, the regions highlighted represent according to their probability the top 30% of probable initial points (green), the top 10% of probable initial points (cyan), and the top 5% of probable initial points (magenta).

Applying this methodology to a myriad of paths and times of observed path, we find that the algorithm is incredibly successful at narrowing down our search field in every scenario. The data to support this is shown in Table **??**. Each set of rows in this table represents a different percentage of the time of flight that was available to the Observer. For example, if the vehicle has a total time of flight of only two hours and the Observer has 50% time of flight seen, only the final 60 minutes of the vehicle's path is available for use in the anterior extrapolation method. Not only were these different percentages tested, but each was compared against six paths that imitate all potential difficult paths the vehicle could use.

Regardless of the amount of time of flight observed, the data presented in Table **??** indicates that the algorithm continually achieves successful prediction within at least 50% probability.

Observer sees 50% of path

Waypoint Position	(0,25)	(25,50)	(50,25)	(50,50)	(75,50)	(100,25)
Distance from Maximum Probability (km)	58.7	18.2	53.0	11.2	14.7	109.4
Successful prediction within at least 50% Probability	Yes	Yes (95%)	Yes (70%)	Yes (95%)	Yes (95%)	Yes (90%)

Observer sees 60% of path

Waypoint Position	(0,25)	(25,50)	(50,25)	(50,50)	(75,50)	(100,25)
Distance from Maximum Probability (km)	49.4	13.8	54.4	19.8	111.5	96.4
Successful prediction within at least 50% Probability	Yes	Yes (95%)	Yes (70%)	Yes (95%)	Yes (95%)	Yes (70%)

Observer sees 70% of path

Waypoint Position	(0,25)	(25,50)	(50,25)	(50,50)	(75,50)	(100,25)
Distance from Maximum Probability (km)	39.5	16.9	55.7	33.4	90.7	84.1
Successful prediction within 50% Probability	Yes	Yes (95%)	Yes	Yes (95%)	Yes (90%)	Yes (70%)

Observer sees 80% of path

Waypoint Position	(0,25)	(25,50)	(50,25)	(50,50)	(75,50)	(100,25)
Distance from Maximum Probability (km)	32.2	38.1	56.7	49.3	72.9	69.5
Successful prediction within 50% Probability	Yes (70%)	Yes (90%)	Yes	Yes (70%)	Yes (70%)	Yes (70%)

Observer sees 90% of path

Waypoint Position	(0,25)	(25,50)	(50,25)	(50,50)	(75,50)	(100,25)
Distance from Maximum Probability (km)	53.6	22.2	57.1	32.1	41.0	58.2
Successful prediction within 50% Probability	Yes (70%)	Yes (95%)	Yes	Yes (90%)	Yes (90%)	Yes

Note: If the anterior extrapolation algorithm narrowed down the search field by more than 50%, the percentage of initial points ruled out of the search is indicated in parentheses

Table 4.5. Compilation of results seen from the utilization of the anterior extrapolation method over six different observation profiles

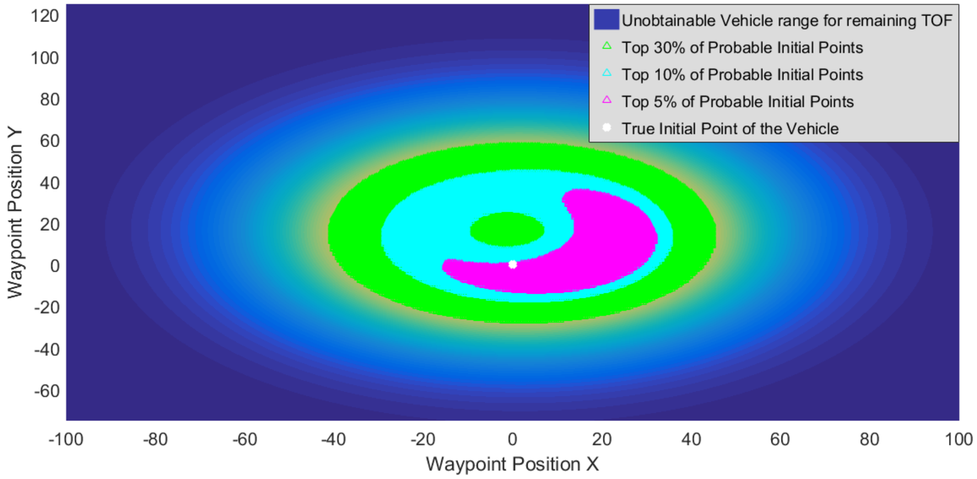


Figure 4.4. Visualization of the results computed by the anterior extrapolation algorithm with 90% of the vehicle's path observed

In other words, the anterior extrapolation algorithm produced a 100% success rate in narrowing down the field of potential starting points by at least 50%. Delving deeper into the data, we see that on many occasions the algorithm exceeded the 50% bar and eliminated from eligibility an even higher percentage of waypoints. Regarding trends in the data, we can see that the more path the anterior extrapolation algorithm is able to observe, the smaller the average distance of the maximum probability from the true initial point. However, the values produced at all observer times produced significant results and could lead to the eventual persecution of any assets at or near the initial point of the vehicle's path.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Conclusions and Future Work

As we have now discussed the results of this present work, this section will look at their importance as a whole and how they can be improved in the future. The first portion of this chapter, Section 5.1, discusses the main takeaways from Chapter 4 and their applicability to current operators and environments. Next, in Section 5.2, we look at the many possible ways that this work can be improved for more robust solutions and increased applicability.

5.1 Conclusions

In this thesis we extended previous work, by considering the addition of more variables: wind, altitude and varying time profiles. We also introduce a new methodology and studied the potential identification of the initial point that the aircraft was launched from.

As a result of the work completed, we are able to conclude that the best possible wind direction to obfuscate path occurs within 45° of the vehicle's straight line path. This is a result of the observed correlation between vehicle speed and path obfuscation. As the vehicle's speed increases, so does the probability of obfuscation. This can be contributed to both the fact that vehicle is rapidly changing direction, but also is a result of the fact that the vehicle need not spend as much time in the air allowing for observations.

The altitude profile that is best associated with vehicle obfuscation is one in which termination, or any decrease in altitude, is delayed until the last possible moment. The delay in the termination of the vehicle prohibits the premature narrowing of the prediction algorithm as a result of a higher probability of termination. Moreover, the obfuscation of vehicle path hinges on the time of flight profile and the best possible selection is the use of 70% of the vehicle's potential time of flight. While this path does not produce the single most optimal path, it does produce the largest number of obfuscated paths with minimal sacrifice (0.08%) on the upper-end of vehicle obfuscation.

In terms of the anterior extrapolation, the algorithm predicts with confidence the terminal location of the vehicle at nearly every possible percent of path observed. While the accuracy of this prediction varies, one can say with confidence that the application of the anterior extrapolation method narrows down the search field significantly and produces an increased capability by allowing the Observer to determine the starting point of most vehicles it encounters.

5.2 Future Work

While this work already stands on its own with readily applicable results, there are ways in which it can be improved to provide a better solution to both the operator and the observer. As these are problems that have been considered in the development of this work but not yet addressed, a potential method for their implementation and the benefit of their incorporation is also offered. The future work is not divided into different categories as a change to one component of the system will likely require the adaptation of the others as well. Thus, each proposed addition is suggested to improve the entire method and not just one aspect.

Considering the constraints that could be placed on paths as a result of vehicle limitations, there are many areas that could be incorporated into the program. Vehicle specific cornering velocity, turn radius, and rates of climb and descent could be used to bound the path in various different ways. The cornering velocity and turn radius figures would limit sharp changes in direction that could arise from waypoints near the initial or terminal locations. Further, vehicle aerodynamics could be considered to incorporate how much the wind will alter the vehicle's flight at varying angles relative to its path.

Considering the obfuscation of path, further consideration could be given to obfuscated the vehicle's waypoint as well. The ability to obfuscate the objective would certainly provide additional security to the operative. Additionally, the incorporation of multiple waypoints would produce many more possible paths and would likely allow for an increase in the vehicle's obfuscation potential.

Looking at the anterior extrapolation method, there is significant room for improvement. While any form of proper extrapolation is successful, the ability to only do so at 90% of path is not robust or necessarily realistic for real-world applications. Modifying the algorithm with an increased number of considerations could potentially narrow the field of view of the observer and therefore increase its probability of successful extrapolation. This could be achieved by adding constraints on where the vehicle will terminate such that it will not land within a certain distance of its initial location.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX: Path Planning Code

```
%==============================
% Script name:        RandomizedPathOptimzationMain
% Written by:         ENS Mitchell T. Heaton
% Last modified:      02 JUN 2017
% Disclaimer:         MODIFY AT YOUR OWN RISK!!!
%==============================
% The main file from which all thesis work is coordinated.
    Take a series of
% user inputs for what the desired output is and coordinate
    the proper
% execution of periphery function files.

% INPUT ARRAYS:
%     *.txt file
%     Reference for iterating through a directory
%
% OUTPUT ARRAYS:
%     TBD
%
%
%==============================
%==============================
%% Workspace Cleanup

clear
close all
clc

%% Globally used variable setup (used as inputs in at least
    two functions)
```

```
q_init = [0; 0]; % Aircraft starting point (in kilometers)
q_term = [100; 0]; % Termination point of the aircraft (
  final position) (in kilometers)
tMin = 0; % minimum time of flight of the vehicle (always =
  0)
tMax = (2)*60*60; % first number is in hours but converted
  to seconds input('What is the maximum time of flight of
  the vehicle?')    % Maximum TOF of the vehicle as a
  function of the assumption it will operate at velocity
  for best endurance
vBE = convvel(60,'kts','km/s'); % units: km/s (after
  conversion from knots) Velocity for best endurance of the
   aircraft. This will change with a given platform. Value
  is in knots converted to km/h
ObFreq = 0.01; % in Hz. In order to achieve observations per
   minute, multiply value by 60.
reqTimeInt = 1/ObFreq; % Indicates how to space the time
  intervals in the time vector
distTO = convlength(250,'ft','km'); % Aircraft takeoff
  distance (distance to rotation, but used as the
  acceleration parameter for the vehicle)
timeTO = distTO/(0.5*vBE); % Computation for the time to
  reach vBE given linear acceleration
missionAlt = convlength(1000,'ft','km'); % Aircraft height
  needed to "perform mission"
missionTime = 360; % Time in seconds that the aircraft must
  be on mission altitude around the target (1/2 before, 1/2
   after)
rateClimb = 600; % Rate of climb in feet per minute
rateClimbMetric = rateClimb.*(0.3048/(1000*60)); % Rate of
  climb in km/s
```

```
%% User modifiable components

%Establish options that the user will have as to how the
   code will run/the
%outputs it will produce
% option1 = 'Generic path test on 2-D predictor';
% option2 = 'Generic path test on 3-D predictor';
% option3 = 'Display set of all possible paths';
% option4 = 'Produce KL-Divergence analysis of predetermined
     prediction set';
% option5 = 'Full system run: produce set of possible paths
   over varying time profiles, run predictor against them,
   produce KL-Divergence, and indicate the optimal path/time
     combination';
% option6 = 'Cancel. I did not mean to click run. Exit the
   danger zone';

% Which altitude profile will be used.

profile1 = 'Stay␣at␣mission␣altitude␣the␣entire␣time␣and␣
   only␣decrease␣altitude␣when␣necessary␣for␣descent.';
profile2 = 'Climb␣immediately␣to␣mission␣altitude,␣then␣
   immediately␣dip␣down␣to␣0.5*MissionAlt␣between␣waypoint␣
   if␣possible.␣Continue␣process␣of␣climb␣and␣descend␣until␣
   final␣descent.';
profile3 = 'Climb␣immediately␣to␣mission␣altitude,␣stay␣at␣
   mission␣altitude␣until␣mission␣completed,␣dip␣down␣to␣
   0.20*MissionAlt␣and␣hold␣until␣descent␣to␣termination';

% Allow user to select option to run
% optionChoice = menu('Choose the outputs and/or tests that
    you would like to run.',option1,option2,option3,option4,
    option5,option6);
profileChoice = menu('Choose␣the␣altitude␣profile␣that␣you␣
    would␣like␣to␣test.',profile1,profile2,profile3);
prompt = {['As␣a␣percentage␣of␣the␣maximum␣possible␣time␣of␣
    flight␣(',num2str(tMax/3600),'␣hours),␣how␣much␣time␣
    would␣you␣like␣the␣vehicle␣to␣use']};
dlg_title = 'Input';
num_lines = 1;
defaultans = {'80'};
answer = inputdlg(prompt,dlg_title,num_lines,defaultans);
percTOF = str2double(cell2mat(answer(1,1)))/100;
while (percTOF <= 0) || (percTOF>1)
     warnBox = warndlg('The␣value␣must␣be␣a␣valid␣percentage␣
         greater␣than␣0␣and␣less␣than␣or␣equal␣to␣100','
         Invalid␣input!');
     waitfor(warndlg('The␣value␣must␣be␣a␣valid␣percentage␣
         greater␣than␣0␣and␣less␣than␣or␣equal␣to␣100','
         Invalid␣input!'))
     answer = inputdlg(prompt,dlg_title,num_lines,defaultans)
         ;
     percTOF = str2double(cell2mat(answer(1,1)))/100; % User
         values for percent time of flight converted to
         decimal.
     if (percTOF > 0) && (percTOF<=100)
          if exist('warnBox','var')
               delete(warnBox) % Closes unnecessary warning box
                   .
          end
          break
     end
```

```
end

prompt = {'In␣knots␣(kts),␣indicate␣the␣magnitude␣of␣wind␣in
   ␣this␣scenario.','In␣what␣direction␣is␣the␣wind␣blowing␣
   in␣degrees␣from␣the␣positive␣x−axis␣(if␣blowing␣from␣the␣
   SW␣to␣the␣NE,␣angle␣is␣45␣degrees)?'};
dlg_title = 'Wind␣components';
num_lines = 2;
defaultans = {'30','0'};
answer = inputdlg(prompt,dlg_title ,num_lines,defaultans);
windMag = str2double(cell2mat(answer(1,1)));
windDir = str2double(cell2mat(answer(2,1)));

% Validate user inputs for the wind. Sneaky little user
   tried to input the
% wrong value, but momma didn't raise no fool.
while (windMag < 0) || (windDir<0) || (windDir>360)
    warnBox = warndlg('Wind␣must␣be␣a␣value␣greater␣than␣0␣
       and␣wind␣direction␣must␣be␣a␣value␣between␣0␣and␣360␣
       in␣degrees.','Invalid␣input!');
    waitfor(warndlg('Wind␣must␣be␣a␣value␣greater␣than␣0␣and
       ␣wind␣direction␣must␣be␣a␣value␣between␣0␣and␣360␣in␣
       degrees.','Invalid␣input!'))
    answer = inputdlg(prompt,dlg_title ,num_lines,defaultans)
       ;
    windMag = str2double(cell2mat(answer(1,1)));
    windDir = str2double(cell2mat(answer(2,1)));
    if (windMag < 0) && (windDir<0) && (windDir>360)
        if exist('warnBox','var')
            delete(warnBox) % Closes unnecessary warning box
               .
        end
        break
    end
end

windMag = convvel(windMag,'kts','km/s');

% withWind = 'Yes, use wind bias in the predictor.';
% noWind = 'No, operate with a consistent maximum possible
   range.';
predictWithWindChoice = 0; %menu('Would you like the
   predictor to operate with an R_max bias from the varying
   velocity due to wind?',withWind,noWind);

withAlt = 'Yes,␣use␣altitude␣to␣bias␣the␣predictor.';
noAlt = 'No,␣operate␣without␣processing␣altitude␣positions.'
   ;
predictWithAltitude = menu('Would␣you␣like␣the␣predictor␣to␣
   operate␣with␣preferential␣termination␣time␣computed␣as␣a␣
   result␣of␣the␣vehicle␣altitude?',withAlt,noAlt);


%% Create the variables and plotting region for the Observer
   : Position Predictor

% Creating plotting region
StdX = 200; % Standard deviation in the x-direction
StdY = 200; % Standard deviation in the y-direction
GridMax = round(vBE.*tMax); % Rmax_init * 1.1 ; %
   Establishes the maximum bound of the plotting grid
xGrid = -GridMax:0.5:GridMax; % Grid bounds for the plot
yGrid = -GridMax:0.5:GridMax; % Grid bounds for the plot
[likeX,likeY] = meshgrid(xGrid,yGrid);
posterior = ones(size(likeX))/sum(sum(ones(size(likeX))));
RmaxArray = zeros(size(likeX));
```

```
% Likelihood function constants
L = 1; % Baseline value of the likelihood function for
    posterior assimilation
a = 2; % Constant for the likelihood function

% Initial position
xLast = 0; % Establishes the initial position of the a/c
yLast = 0; % Establishes the initial position of the a/c

% Figure Setup
figure(); clf
s1 = surf(likeX,likeY,posterior);
set(s1, 'FaceAlpha', 0.5, 'EdgeColor', 'none')
hold on
p1 = plot(0,0, '-k', 'LineWidth', 2.5);
xlabel('x')
ylabel('y')
hold off
drawnow
view(2)

%% Trajectory variables and setup
TOFused = percTOF.*tMax; % Units: seconds.
numTrajPoints = TOFused/reqTimeInt; % This is the bound on
    number of waypoints. This will set a bound on waypoints.
    This provides a rough estimate in order to create a
    relative bound on the potential sampling interval
    necessary.
yMax = vBE.*(1/2).*TOFused; % Establishes the maximum y-
    position of a waypoint to vary no more than 1/2 of the
    vehicle's potential TOF from the centerline
yMin = -yMax;
```

```
x = linspace(q_init(1),q_term(1),10);
y = linspace(yMin,yMax,10);
[X,Y] = meshgrid(x,y);
dataSet = 0; %#ok<NASGU>

% Compute the components of wind
windX = windMag*cosd(windDir);
windY = windMag*sind(windDir);

% Plot for current waypoint
fig = figure;
axs = axes('Parent',fig,'NextPlot','Add');
plt(1) = plot(axs,[q_init(1),q_term(1)],[q_init(2),q_term(2)
    ],'x:k');
plt(2) = plot(reshape(X,1,[]),reshape(Y,1,[]),'.b');
plt_q_wpnt = plot(axs,0,0,'og'); % Plot for (highlight)
    current waypoint being used
plt_traj   = plot(axs,0,0,'r'); % Plot for the trajectory of
     the aircraft

% Plot for current altitude profile

figure(); clf
z1 = plot(0,0,'-r');
xlabel('Time␣of␣Flight␣(minutes)')
ylabel('Vehcile␣Altitude␣(feet)')
drawnow
view(2)

%% KL Divergence setup- create gaussian to compare against.
% Declare the gaussian for comparison
mu = [100 0];
Sigma = 50*eye(2);
```

```
x1 = xGrid; x2 = yGrid;
[X1,X2] = meshgrid(x1,x2);
F = mvnpdf([X1(:) X2(:)],mu,Sigma);
F = reshape(F,length(x2),length(x1));
KLMat = zeros(size(X,1),size(X,2));
DNQ = 0;
dataSet = 0;

%% Meat of the code: trajectory generatred, predictor run
    for each path.

for i = 1:size(X,1)
    for j = 1:size(X,2)

        % Trajectory generation and plotting
        q_wpnt = [50;25]; %[X(i,j);Y(i,j)]; % Establishes
            current waypoint for this loop
        [xPath, yPath] = PositionFunction( q_init, q_term,
            q_wpnt, numTrajPoints); % Produces trajectory
            points for the path.
        [pathTimeVec, vTotalVec] = WindAdjust( vBE, windMag,
             windDir, xPath, yPath );
%           zTotalVec = zeros(1,length(pathTimeVec))+
    missionAlt;
        [zTotalVec] = altitudeFunction(rateClimbMetric,
            pathTimeVec,q_wpnt,xPath,yPath,missionTime,
            missionAlt,profileChoice);
        pathTimeVecMin = pathTimeVec./60;
        zTotalVecFeet = convlength(zTotalVec,’km’,’ft’);
        set(z1,’xData’,pathTimeVecMin,’yData’,zTotalVecFeet)
            ;
        set(plt_q_wpnt,’xData’,q_wpnt(1),’yData’,q_wpnt(2));
        set(plt_traj,’xData’,xPath,’yData’,yPath);
```

```
disp(['Plot␣of␣waypoint␣',num2str(j+ 10*(i-1)),'␣of␣
   ',num2str(size(X,2)*size(X,1)), '␣total␣waypoints
   '])
dataSet = dataSet + 1;
% The Observer termination prediction.
if (max(pathTimeVec) <= TOFused) & (~isnan(
   zTotalVec))%#ok<BDSCI> % Could be swapped for
   tMax, but this allows for the percentage scaling
   to be the upper bound. Thus, all paths x% and
   under provide us with __ KL-Div Values

    NumObs = floor(max(pathTimeVec)./reqTimeInt); %
       Number of observer samplings during the
       flight. Used floor function because if a time
        interval isn't complete and the vehicle
       lands, it will not see the vehicle anyways.
    obsTimeVec = (0:reqTimeInt:NumObs*reqTimeInt); %
        Vector for the time vector the observer will
        use to sample and see the vehicle's position
       .
    xVals = zeros(1,length(obsTimeVec));
    yVals = zeros(1,length(obsTimeVec));

    for m = 1:length(obsTimeVec)
        % Radar observation of the vehicle by the
           observer.
        % Requires interpolation to achieve the
           vehicle's position
        % in between points on the vehicle's track
           in case the time
        % samplings don't match exactly.
        tNow = obsTimeVec(m);
        nowInd = find(pathTimeVec<=tNow,1,'last');
```

```
                xInterp = interp1([pathTimeVec(nowInd),
                  pathTimeVec(nowInd+1)],[xPath(nowInd),
                  xPath(nowInd+1)],tNow);
                yInterp = interp1([pathTimeVec(nowInd),
                  pathTimeVec(nowInd+1)],[yPath(nowInd),
                  yPath(nowInd+1)],tNow);
                zInterp = interp1([pathTimeVec(nowInd),
                  pathTimeVec(nowInd+1)],[zTotalVec(nowInd)
                  ,zTotalVec(nowInd+1)],tNow);
                xVals(m) = xInterp; % This adds this
                  coordinate to those previously seen by
                  the radar/observer
                yVals(m) = yInterp; % This is the path that
                  the observer sees.
                tRemain = tMax−tNow; % Amount of possible
                  remaining time the observer knows
                altNow = zInterp;
                % Determination loop for utilization of the
                  Percent of Path
                % Prediction based on time
%                 if (tRemain > 0.15*tMax)
%                     terminationPathPerc = 0.75; %
  Percentage for the expected travel of vehicle as a
  function of its maximum allowable TOF
%                 else
%                     terminationPathPerc = 1;
%                 end
%
                % Determination statement for probability of
                   termination
                % based on vehicle altitude with respect to
                  percent of
                % mission altitude
```

```
if predictWithAltitude == 1
    if (altNow > (0.50*missionAlt)) && (
       tRemain > 0.10*tMax)
        terminationScalePerc = 1;
    elseif (altNow > 0.20*missionAlt) && (
       tRemain > 0.10*tMax)
        terminationScalePerc = 0.75;
    elseif (altNow <= 0.20*missionAlt) && (
       tRemain > 0.10*tMax) && (tNow >
       missionAlt/rateClimbMetric)
        terminationScalePerc = 0.30;
    else
        terminationScalePerc = 1;
    end
else
    terminationScalePerc = 1;
end

% Creating the Rmax array
if predictWithWindChoice == 1
    for o = 1:length(xGrid)
        for q = 1:length(yGrid)
            xDiff = xGrid(o)-xInterp;
            yDiff = yGrid(q)-yInterp;
            ratioNow = xDiff/yDiff;
            if (xDiff>0) && (yDiff>0) % Path
                between 0 and 90 degrees
                thetaPath = atand(ratioNow);
            elseif (xDiff<0) && (yDiff>0) %
              Path between 90 and 180
              degrees
                thetaPath = 180 + atand(
                   ratioNow);
```

```
elseif (xDiff<0) && (yDiff<0) %
   Path between 180 and 270
   degrees
    thetaPath = 180 + atand(
       ratioNow);
elseif (xDiff>0) && (yDiff<0) %
   Path between 270 and 0
    thetaPath = 360 + atand(
       ratioNow);
elseif (xDiff > 0) && (yDiff==0)
    % Path direction is 0
   degrees.
    thetaPath = 0;
elseif (xDiff< 0) && (yDiff==0)
   % Path direction is 180
   degrees.
    thetaPath = 180;
elseif (xDiff==0) && (yDiff>0) %
    Path direction is 90 Degrees
    thetaPath = 90;
elseif (xDiff==0) && (yDiff<0) %
    Path direction is 270
   degrees
    thetaPath = 270;
elseif (xDiff==0) && (yDiff==0)
   % Necessary for the position
   predictor.
    thetaPath = 0;
end
vxTotal = windMag*cosd(windDir)
   + vBE*cosd(thetaPath);
vyTotal = windMag*sind(windDir)
   + vBE*sind(thetaPath);
```

```
            vTotal = sqrt(vxTotal^2 +
               vyTotal^2);
            RmaxArray(o,q) = vTotal.*tRemain
               .*terminationScalePerc;
        end
    end
else
    RmaxArray = vBE*tRemain*
       terminationScalePerc;
end


% Likelihood Calculations
R = sqrt((likeX-xInterp).^2 + (likeY-yInterp
   ).^2); % Calculation of the distance
   from the current position for blackbody
% likelihood determination
K = (RmaxArray + a)./2; % Constant to shift
   the curve to have the higher likelihoods
   near an anticipated design point Rmax
L = ((R + a).^2)./exp(R./K); % The
   likelihood of the occurrence with the
   specified parameters

% Normalization
L = L/sum(sum(L));
L(L<eps) = 10*eps;

% Updating the posterior function for the
   new data
posterior = L.*posterior;
posterior = posterior/sum(sum(posterior));
Gaus = exp(-1/(2*StdX)*(-likeX).^2-1/(2*StdY
                    )*(-likeY).^2);
                Gaus = Gaus/(sum(sum(Gaus)));
                posterior = convolve2(posterior,Gaus,'same')
                    ;
                set(s1,'ZData',posterior)
                set(p1,'XData',xVals(1,1:m),'YData',yVals
                    (1,1:m))
                drawnow
            end

            sprintf('Saving␣%d',dataSet)
            save(sprintf('DataRun_%d.mat',dataSet),'
               posterior','xPath','yPath','q_wpnt','percTOF'
               ,'vBE','reqTimeInt','xGrid','yGrid')
            KLMat(i,j) = KLDiv(F,posterior);

        else
            KLMat(i,j) = NaN;
        end

        clearvars posterior L Gaus
        posterior = ones(size(likeX))/sum(sum(ones(size(
           likeX))));


    end
end

save(sprintf('KLMat.mat'),'KLMat')
disp(['The␣KL␣Divergence␣for␣this␣test␣is␣',num2str(KLMat
  (1,1))])
```

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] Torpedo Bay Navy Museum. (n.d.). Anti submarine warfare. [Online]. Available: http://navymuseum.co.nz/worldwar1/battles-operations/anti-submarine-warfare/. Accessed May 4,2017.

[2] Lockheed Martin. (n.d.). P-3 Orion. [Online]. Available: http://www.lockheedmartin.com/us/products/p3.html. Accessed May 4,2017.

[3] U.S. Navy. (n.d.). The U.S. Navy – fact file: P-8A Poseidon multi-mission maritime aircraft (MMA). [Online]. Available: http://www.navy.mil/navydata/fact_display.asp?cid=1100&tid=1300&ct=1. Accessed May 4,2017.

[4] R. Pursell. (2010, Jul.-Sep.). Trident Warrior 2010 Leads the Way in Maritime Experimentation. *CHIPS- The Department of the Navy's Information Technology Magazine*. [Online]. Available: http://www.doncio.navy.mil/CHIPS/ArticleDetails.aspx?ID=2387

[5] S. Freedberg. (2012, Oct. 26). Run silent, go deep: drone-launching subs to be Navy's 'wide receivers'. [Online]. Available: http://breakingdefense.com/2012/10/run-silent-go-deep-drone-launching-subs-to-be-navys-wide-rec/

[6] United States Navy Demonstrates Cross-Domain Communications, Command and Control via AeroVironment Blackwing Submarine-Launched UAV. (n.d.). Business Wire. [Online]. Available: http://www.businesswire.com/news/home/20160907005733/en/United-States-Navy-Demonstrates-Cross-Domain-Communications-Command. Accessed May 4, 2017.

[7] M. Eckstein. (2016, Mar. 24). Navy: Future undersea warfare will have longer reach, operate with network of unmanned vehicles. [Online]. Available: https://news.usni.org/2016/03/24/navy-future-undersea-warfare-will-have-longer-reach-operate-with-network-of-unmanned-vehicles

[8] D. Edwards. (n.d.). Flying-Swimmer (Flimmer) UAV/UUV. Laboratory for Autonomous Systems Research. [Online]. Available: https://www.nrl.navy.mil/lasr/content/flying-swimmer-flimmer-uavuuv. Accessed May 4, 2017.

[9] M. Heaton, L. DeVries and M. Kutzer, "Randomized Path Optimization for the Mitigated Counter-Detection of UAVs," in *International Conference on Unmanned Aircraft Systems*, Miami, FL, 2017, pp. 72–78.

[10] L. DeVries and A. Paley, “Observability-based optimization for flow sensing and control of an underwater vehicle in a uniform flowfield,” in *American Control Conference (ACC)*, Washington, DC, 2013, pp. 1386–1391.

[11] L. Stone, R. Streit, T. Corwin and K. Bell, *Bayesian Multiple Target Tracking*, 2nd ed. Boston, MA: Artech House, 2014.

[12] C. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*. Englewood Cliffs, NJ: Prentice Hall, 1971.

[13] R. Bulirsch and J. Stoer, “Numerical treatment of ordinary differential equations by extrapolation methods,” *Numerische Mathematik*, vol. 8, no. 1, pp. 1–13, June 1965.

[14] K. Brenan, S. Campbell and L. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, 2nd ed. North-Holland, NY: Society for Industrial and Applied Mathematics, 1996.

[15] J. Soechting, J. Juveli and H. Rao. (2010, September 3). Incorporating prediction in models for two-dimensional smooth pursuit. *PloS one*. [Online]. *5*(9). Available: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0012574

[16] A. Thorvaldsson and V. Bandi, “Reference path estimation for lateral vehicle control master’s thesis in systems, control and mechatronics,” Ph.D. dissertation, Chalmers University of Technology, Gothenburg, Sweden, 2015.

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California