2011-10-18

Aerial Rendezvous Between an Unmanned Air Vehicle and an Orbiting Target Vehicle

Mark Andrew Owen
Brigham Young University - Provo

Aerial Rendezvous Between an Unmanned Air Vehicle and an Orbiting Target Vehicle

Mark A. Owen

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Mark B. Colton, Chair
Tim W. McLain
Randy W. Beard

Department of Mechanical Engineering

Brigham Young University

December 2011

ABSTRACT

Aerial Rendezvous Between an Unmanned Air Vehicle and an Orbiting Target Vehicle

Mark A. Owen
Department of Mechanical Engineering, BYU
Master of Science

In this thesis we develop methods that facilitate an aerial rendezvous between two air vehicles. The objective of this research is to produce a method that can be used to insert a miniature air vehicle behind a rendezvous vehicle and then track that vehicle to enable a visual rendezvous. For this research we assume the rendezvous vehicle is following a relatively stable and roughly elliptical orbit. Path planners and controllers have been developed that can be used to effectively intercept the rendezvous vehicle by inserting the MAV onto the orbit of interest. A method for planning and following time-optimal Dubins airplane interception paths between a miniature air vehicle and the rendezvous vehicle is presented. We demonstrate how a vector field path following a scheme can be used for navigation along these time-optimal Dubins airplane paths. A post-orbit insertion tracking method is also presented which can be used to track the target vehicle on an arbitrarily oriented elliptical orbit while maintaining a specified following distance. We also present controllers that can be used for disturbance rejection during the orbit-insertion and interception operations. All of these methods were implemented in simulation and with hardware. Results from these implementations are presented and analyzed.

Keywords: aerial rendezvous, vector field, path following, dubins airplane, time optimal, uav, mav

ACKNOWLEDGMENTS

I wish to express thanks to the many people who have supported me through my research. First, I wish to express gratitude to my amazing wife who provided me with her unwavering support. Without her willingness to patiently edit my thesis it may never have fully made sense. I cannot thank her enough for keeping me fed and our lives running when I was focusing all of my time and energy on completing this thesis.

It would be wrong for me not to acknowledge the vast amount of inspiration that came to me from my Heavenly Father. This divine assistance helped me find bugs in my code that were making my life miserable and come up with solutions to problems that I cannot take credit for.

The example of my parents and the upbringing that they provided me taught me the work ethic that made it possible for me to start and finish this work.

Dr. Colton was a wonderful advisor. He patiently allowed me to pursue the avenues of research that interested me most and spent much time listening to my successes and failures. He graciously sacrificed his personal time to help me complete this degree and was incredibly considerate of my personal deadlines. I would like to thank Dr. Beard and Dr. McLain for the invaluable insight they offered from their experience working with UAVs.

I would also like to thank the Aerial Recovery team. Joe Nichols, Solomon Sun, Larry Moore, Dallin Briggs and Jeff Ferrin all spent countless hours working to make this project a success. More members of the MAGICC lab helped me than I am able to thank. The knowledge that the members of the lab contain is incredible and helped me through many stages of this research.

Neil Johnson, from Procerus Technologies, provided outstanding support for the Aerial Recovery team and made implementation of my work possible. He spent many hours developing all of the special features that I needed to get my code to work.

This research was funded by the Air Force Office of Scientific Research under STTR contract FA 9550-09-C-0102 to Procerus Technologies and Brigham Young University.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

a, b	ellipse major and minor radii
A	polynomial approximation basis matrix
c_i	coefficients contained in $\mathbf{C}$
$\mathbf{C}$	polynomial coefficients vector
$G(x, y, z)$	vector field contraction strength weighting function
$H(x, y, z)$	vector field circulation vs contraction weighting factor
n	order number of polynomial approximation
m	number of sample points used in polynomial approximation
Q	orthogonal matrix from QR factorization of $A^T A$
R	upper triangular matrix from QR factorization of $A^T A$
s	arc length
$S(\bar{r}, t)$	function definition used in the Lawrence vector field method
u	vector field function output, vector of directional velocity commands
$V(\alpha_1, \alpha_2)$	vector field scalar potential function
$V_F(\bar{r})$	scalar potential function use in the Lawrence vector field method
V_g	Aircraft ground speed
V_a	Aircraft airspeed
x, y, z	position variables in each of the three dimensions
$\mathbf{y}$	vector arc length derivative sample points
$\alpha_i(x, y, z)$	vector field surface functions
γ	parameters used to tune the strength and direction of vector field
Γ	parameter used to tune the Lawrence vector field
circulation and the aircraft flight path angle	
χ	Aircraft ground referenced course angle
θ	Aircraft pitch angle
ϕ	Aircraft roll angle
ψ	Aircraft yaw angle
ρ	circular helix radius
σ	parameter describing parametric defined ellipse
$\hat{\cdot}$	Represents an approximation of the variable $(\cdot)$
∇	Gradient operator with respect to x_1, x_2, and x_3
$\times$	Standard cross-product operator
$\bar{r}$	relative position in the vector field frame
$[\,]^d$	$[\,]$ is the desired parameter value
$[\,]^c$	$[\,]$ is the commanded parameter value
$[\,]_{ff}$	$[\,]$ is a feed-forward controller input

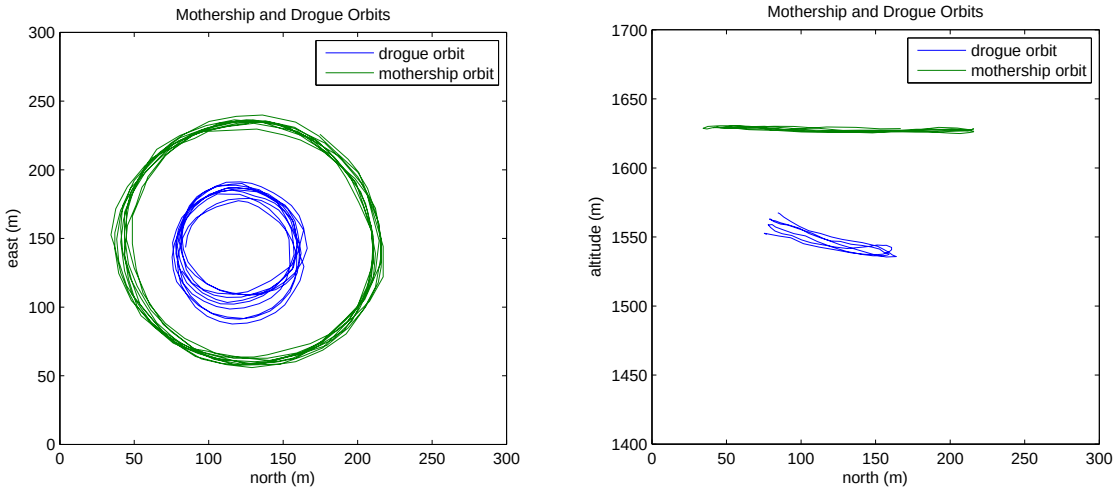
CHAPTER 1. INTRODUCTION

1.1 Problem Statement

Unmanned aerial vehicles (UAVs) have gained much attention over the past few years because they are capable of carrying out a wide variety of tasks. Small fixed-wing, unmanned aerial vehicles or micro aerial vehicles (MAVs) have the advantage of being relatively inexpensive platforms that can perform roles that previously have not been financially viable. One of the limitations of MAVs is their characteristically short flight times. In light of this limitation there has been interest in developing a method for inserting and extracting MAVs from a larger air vehicle capable of deploying and recovering the MAVs without the need for ground support. Such a capability would make it possible to deploy a large number of MAVs to an area of interest that is not accessible to ground units. Scenarios in which such a capability would be valuable include post-disaster, quick response aerial surveillance, war zone aerial reconnaissance where it is undesirable or impossible for the MAVs to return to a ground base, and any situation where the area of interest is a long distance from any ground support.

One of the difficulties involved in recovering MAVs from the air with a larger aircraft of long endurance is that their characteristic airspeeds are drastically different. We will refer to the larger vehicle performing the recovery as the mothership. Consequently, it is not feasible to perform a tail-chase of the mothership by the MAVs. A possible solution to this problem is to have the mothership reel out a high drag object on a long cable that can function as a dock. It has been shown through both simulation and experimental data that an air vehicle flying a circular orbit while towing a relatively high-drag object, which we will hence forth refer to as a drogue, will result in the drogue adopting a steady-state orbit which is smaller than that of the mothership [3]. Experimentally collected data from a mothership towed drogue system can be found in Figure 1.1. In the absence of wind, the drogue will adopt a flat circular orbit concentric to the mothership orbit.

These data have also shown that in the presence of steady wind, the drogue orbit will be roughly elliptical and rotated out of the horizontal plane.



(a) (b)

Figure 1.1: Mothership and drogue experimental results [1]

In steady-state flight both vehicles will tend to have the same angular velocity about their respective orbits. This results in the drogue traveling at a lower velocity than that of the mothership. If the cable length of the system and drag properties of the drogue are selected appropriately then the drogue will adopt an orbit that can be feasibly followed by a MAV. This would allow the drogue to achieve airspeeds lower than the maximum airspeed of the MAVs on an orbit that has a minimum turn radius greater than the minimum turn radius of the MAVs. In this way a drogue orbit can be produced that can be followed by the MAVs. With such a capability it becomes feasible to develop a scheme for recovering these MAVs via the mothership towed drogue system. The basic concept for this scheme is shown in Figure 1.2.

Simply stated, the contributions of this thesis are a method for inserting the MAV behind the drogue on an estimated drogue orbit in a near time-optimal fashion and a method for subsequently closing the distance between the two vehicles while following the estimated drogue orbit. The intent is to rely primarily on GPS to quickly intercept the drogue by inserting the MAV directly behind the drogue and continue tracking it until a final vision-based rendezvous can be carried out.

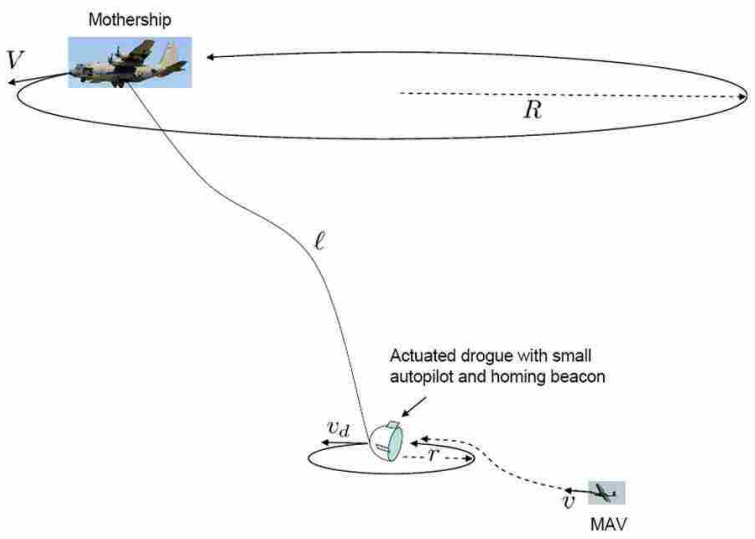


Figure 1.2: Depiction of the aerial recovery concept: the mothership pulls a towed drogue which adopts a slower interior orbit to make a rendezvous between the MAV and drogue possible.

1.2 Drogue Interception Concept

Previous efforts have developed a method for estimating the elliptical drogue orbit given GPS data of the vehicle [4]. This method has been demonstrated to effectively estimate the parameters of the drogue orbit, the major and minor radii, and the three euler angles. Armed with an estimation of the drogue orbit, we will present a method that can be used for planning and following a time-optimal interception path from the current MAV location to an appropriately selected future drogue location along the estimated elliptical orbit. The goal is to plan a time-optimal Dubins airplane path for the MAV that coordinates the arrival of the two vehicles on the drogue orbit. Planning and following time-optimal paths is a useful capability for air vehicles. This is particularly true for vehicles that, similar to MAVs, have relatively short flight times. Such a capability would increase the time efficiency of vehicle paths and would decrease the time needed to recover the MAV with a mothership towed drogue system.

Many efforts have investigated time-optimal path planning for reduced-order vehicle models [5] [6]. One publication provided a mathematically developed model for three dimensional time-optimal airplane paths. We will develop a practical method for following these paths in addition to presenting a scheme for using them to intercept a target vehicle that is traveling along a predictable path. The methods for following these time-optimal paths and pursuing the drogue

along an elliptical orbit stem from vector field path following methods. A number of efforts have been made to develop schemes for fixed-wing aircraft to be able to follow paths of interest [2] [7] [8] [9] [10]. Most notable are two works which develop a method for constructing path following based vector fields, with desirable convergence properties, around arbitrary curves. We use one of these methods to develop path following vector fields for navigation along both the time-optimal Dubins airplane paths and the elliptical orbit path. We also present a method for normalizing the vector fields to make their convergence properties less susceptible to changes in path parameters.

A robust method for planning interception paths from the MAV to the drogue along arbitrarily oriented elliptical orbits is presented. This path planning method takes into account the effects of wind. Airspeed controllers are presented that can be used to reject disturbances experienced during different stages of the rendezvous operation. These include a controller that can be used to prevent large miss distances during the interception stage and another that can be used to maintain a desired following distance between the MAV the drogue after the insertion onto the drogue orbit has been completed.

1.3 Thesis Overview

In Chapter 2, two methods for constructing path following vector fields about arbitrary curves are discussed and compared. These methods were adopted from the work presented in [7] and [9]. Low-level controllers are presented which are needed to feasibly implement these vector field path following methods. Chapter 3 uses both of the vector field construction methods to produce two stable vector field path following methods for fixed-wing aircraft navigation along arbitrarily oriented elliptical orbits. These two methods are then simulated and compared. In Chapter 4 we define time-optimal Dubins airplane paths, from the work of [6], and use one of the vector field path following methods presented in Chapter 2 to develop a vector field path following method for UAV navigation along these paths. These methods are demonstrated in simulation. Chapter 5 presents a number of algorithms and functions needed to plan the time-optimal Dubins airplane paths that can be used to intercept the drogue with a MAV. An interception root finding technique is presented as well as some airspeed controller based disturbance rejection methods. Chapter 6

presents experimental flight test results of the methods presented in this thesis. Chapter 7 presents the general conclusions and contributions of this work and discusses possibilities for future work.

CHAPTER 2. VECTOR FIELD PATH FOLLOWING

Vector field path following methods are a class of controllers that can be used to navigate a vehicle along a desired path. The basic idea is that given any point in the vector field domain a command is computed that, if followed, will result in the vehicle converging to the desired path. The objective then is to cause vehicle convergence towards the path and simultaneous progression along the path. Thus, the purpose is to travel along the path and not necessarily arrive at specific points along the path at specific times. In this way this method is distinct from trajectory tracking methods. An example vector field constructed about a circular orbit can be seen in Figure 2.1.

The advantages of these path following methods include being very intuitive to tune and relatively simple to construct and implement. One of the limitations of this method is that once the vector field has been constructed and tuned the resultant path essentially follows the vector field integral curves. Thus, there is a limited ability to manipulate the vehicle's path from its initial configuration to insertion onto the path of interest. Also, to some extent the path of interest must be known beforehand so that an appropriate vector field can be constructed to follow it.

A variety of approaches have been taken to produce vector fields. Many of these methods are case specific and cannot be applied generally for following other paths, such as a vector field designed specifically for following a circular orbit in two dimensions [10]. Other notable papers in this area provide results that are generalized and can be used for constructing vector fields that result in navigation along arbitrary curves [7] [8] [9].

Various efforts have been made to develop three dimensional vector field path following methods. One of the motivations of using these three dimensional methods compared to previous works [2] [10] is that it eliminates the need for an altitude controller. This is of particular benefit when following paths that have neither a constant altitude nor a constant climb rate, such as the case of an inclined ellipse. Also, the convergence proofs are valid in $\mathbb{R}^3$, rather than being limited

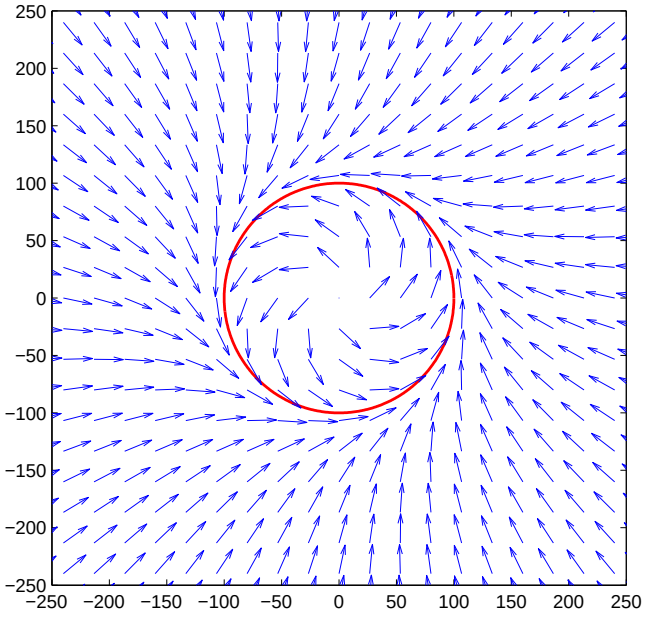


Figure 2.1: Two-dimensional view of an example path following vector field with normalized velocity commands.

to $\mathbb{R}^2$, and increase our confidence that these path following convergence properties will hold when implemented on hardware capable of motion in three dimensions.

In this thesis we discuss two notable papers, [7] and [9], that produce general results for constructing vector fields along three dimensional paths of interest and have desirable, proven convergence properties. The approach taken for constructing these three dimensional vector fields is to divide the problem into two complementary and intuitive commands: contraction and circulation. When appropriately combined these forces work together to provide convergence toward the curve which produces commands feasible for implementation with physical systems. Both [7] and [9] accomplish this by constructing a Lyapunov-like potential function such that the zero-level set contains only the curve of interest, the function is positive everywhere else, and is radially unbounded. Taking the negative gradient of this scalar potential function provides a function that produces commands that always converge towards the desired path. This command component is what we will refer to as the convergence term.

These papers, [7] and [9], also make use of a circulation term which is defined such that it is orthogonal to the vectors produced by the convergence term. When the resultant vectors of both of these functions are combined a desired velocity vector is produced. Generally speaking, if this command is integrated, starting from any initial condition, the integral curve will converge to and circulate about the curve of interest.

One of the attractive attributes of these two methods, for our application, is that they are well suited for use with fixed-wing aircraft. This is because the control output of these methods includes a continual non-zero velocity which coincides with the constraint of fixed-wing aircraft to always maintain a minimum forward velocity.

In the paper [7], Lawrence, et al. present a generic vector field equation and the corresponding constraints that produce velocity vector commands that, if followed, guarantee convergence to the curve. This result is then extended to produce a vector field equation that yields a desired acceleration vector which includes a feed-forward, error reduction term. In [9], Gonçalves, et al. also present a generic vector field equation that produces velocity vector commands, but this method removes some of the complexity of selecting appropriate functions. Rather than developing the potentially complex convergence and circulation functions for the specific path from scratch, the designer only needs to represent the path as the intersection of two surfaces. With this second method, it is also possible to create a vector field for an arbitrary number of dimensions along time varying curves —although such a capability will not be employed in this thesis. We will now discuss these two results in further detail.

We begin by first defining the coordinate frame and position vector that we will use in discussing these two methods. Rather than constraining our vector fields to use a north-east-down (NED) inertial coordinate frame, which is the frame typically used in aircraft applications [2], we will instead define a frame that can be arbitrarily oriented relative to a NED frame. It is important to carefully select the vector field coordinate frame when constructing the vector field equations because some frames substantially reduce the complexity of the vector field definitions. We define the vehicle position in three dimensions as $\bar{r}$, a 3×1 vector of positions in the x, y, and z directions relative to the frame origin. As previously mentioned, the outputs of these two vector field methods are a desired velocity or acceleration vector. These will be notated as $\dot{\bar{r}}^d$ and $\ddot{\bar{r}}^d$ respectively.

2.1 Three Dimensional Vector Field Construction with a Feed-Forward Correction Term

In the paper [7] a theorem is presented and proven such that by constructing appropriate functions a Lyapunov stable vector field can be constructed. One begins by defining the curve of interest $\mathcal{C} \in \mathbb{R}^3$ where $\mathcal{C}$ is the zero-level set of positive semi-definite, scalar potential function $V_F(\bar{r})$. Certain conditions must be satisfied by $V_F(\bar{r})$ for the convergence theorem in the paper to hold. These are summarized as follows: $V_F(\bar{r}) = 0 \Rightarrow \bar{r} \in \mathcal{C}$, $V_F(\bar{r})$ is radially unbounded, $V_F(\bar{r})$ is continuously differentiable, and $V_F(\bar{r})$ cannot be an explicit function of time (i.e. $\mathcal{C}$ must be a static curve). A second function $S(\bar{r})$ must be selected which satisfies

$$\frac{\partial V_F}{\partial \bar{r}} S(\bar{r}) = 0 \tag{2.1}$$

for any $\bar{r}$ in the vector field domain. It is this constraint which enforces orthogonality between the two vector field terms.

The preliminary result that is produced from these equations is a vector field equation that provides a desired velocity vector, $\dot{\bar{r}}_d$, for any point in the function domain. This equation is

$$\dot{\bar{r}}_d = -\left[\frac{\partial V_F}{\partial \bar{r}} \Gamma(\bar{r},t)\right]^T + S(\bar{r},t), \tag{2.2}$$

where $\Gamma(\bar{r},t)$ is a positive definite function that can be used to control the relative strength of the two other functions in the equation. The proof that this vector field equation causes convergence to the desired path shows that the system is Lyapunov stable. This can be shown by taking the time derivative of $V_F(\bar{r})$ which yields

$$\dot{V}_F(\bar{r}) = \frac{\partial V_f}{\partial t} + \frac{\partial V_F}{\partial \bar{r}} \dot{\bar{r}} = \frac{\partial V_F}{\partial \bar{r}} \dot{\bar{r}}^d. \tag{2.3}$$

Substituting from (2.2) and drawing upon the constraint in (2.1) provides us with the negative definite function,

$$\dot{V}_F(\bar{r},t) = -\left(\frac{\partial V_F}{\partial \bar{r}}\right)\Gamma(\bar{r},t)\left(\frac{\partial V_F}{\partial \bar{r}}\right)^T, \tag{2.4}$$

which leads directly to the vector field commands being Lyapunov stable.

Thus, by satisfying the conditions on the scalar potential function and the constraint in (2.1) we can produce a three dimensional vector field. This has the desirable effect of separating (2.2) into distinct convergence and circulation terms. The first term in (2.2) causes convergence towards $\mathcal{C}$, and the second, by nature of the constraint, produces circulation about it. The proof in the paper shows that all integral curves of these vector fields, that have initial conditions within a given domain, converge to the set $\mathcal{C}$. Due to the nature of the potential function, when the vehicle is on or near the curve the velocity command is dominated by the circulation term, and when it is far from the curve the convergence term dominates. Another attribute of these vector fields is that a non-isotropic $\Gamma(\bar{r},t)$ can be used to vary the contraction normal to the ellipse relative to contraction in the plane of the ellipse.

These results have the capability of working for arbitrary curves if constructed correctly. The challenge for the designer is to select a scalar potential function $V_F(\bar{r})$ that satisfies the above mentioned conditions and then a function $S(\bar{r})$ that satisfies (2.1). In [7] an example vector field equation is produced for navigation about a circular orbit. A method for normalizing the vector fields to produce a desired velocity magnitude command and a desired orbit direction is demonstrated for the specific case of the circular orbit, but no generally applicable method is presented. Thus, it is left up to the designer to derive an appropriate normalization term for any other curve.

The velocity based vector field equation, from (2.2), was extended to produce a vector field that includes a feed-forward correction term. An acceleration control law was derived from the Lyapunov vector field equation that produces commanded accelerations in the vector field frame. A proof was provided which also guarantees convergence of the integral curves of these commands. This generalized equation for producing acceleration commands provides the possibility of error reduction, but it also adds a significant amount of complexity to construction and implementation. The generalized equation is

$$\ddot{\bar{r}}^d = -P(\bar{r},\dot{\bar{r}}) + \ddot{\bar{r}}_{ff}, \tag{2.5}$$

where $P(\bar{r}, \dot{\bar{r}})$ is another designer-selected proportional tracking term that must satisfy further constraints and $\ddot{\bar{r}}_{ff}$ is a derived feed-forward acceleration term. For the sake of simplicity we will not discuss the requisite constraints on $P(\bar{r}, \dot{\bar{r}})$. Rather, we present the candidate function that we adopted for our work:

$$P(\bar{r}, \dot{\bar{r}}) = \beta(\dot{\bar{r}}^d - \dot{\bar{r}}), \quad \beta > 0. \tag{2.6}$$

The proportional term produces accelerations that direct the vehicle toward the desired velocity vector as specified by (2.2). The feed-forward term, which is the derivative of the vector field equation, contributes acceleration commands that anticipate the vector field turn rate. We define this as

$$\ddot{\bar{r}}_{ff} \triangleq \frac{d}{dt}\dot{\bar{r}}^d, \tag{2.7}$$

where $\dot{\bar{r}}_d$ is the result from (2.2).

Inclusion of the feed-forward term provides an anticipatory control component that in theory should increases the ability of the controller to accurately track the curve of interest. One of the drawbacks of this method is that converting desired acceleration into tractable UAV commands is a challenging problem. For our purposes we adopted a slightly modified version of polar converting logic [2] [11] which will be further discussed later in this chapter (Section 2.4.2). One of the limitations of this controller is that currently it does not take into account aircraft flight constraints such as minimum turn radius, maximum climb rate, and maximum airspeed.

2.2 Explicit Vector Field Construction For Arbitrary Curves in Three Dimensions

The second work we will discuss here, by Gonçalves, et al., produced a relatively simple method for constructing these three dimensional vector fields along arbitrary curves [9]. When compared with the the work of [7], by Lawrence, et al., the results of [9] provide a generalized method for producing these vector fields that is much less involved to design and more intuitive to tune.

The primary result in $\mathbb{R}^3$ is that an explicit vector field equation can be produced if the curve of interest, $\mathcal{C}$, can be mathematically described as the intersection of two surfaces. One

substantial contribution of this work to the path following field is that it is not limited to static curves. All of these results are applicable to time-varying curves. For our present application we only require the use of static curves. In the cases we investigated, generally speaking, for all initial conditions the integral curves converged to the desired path. The input to the vector field function is the vehicle's position vector expressed in the same frame as the surface functions. The output is a desired velocity vector. A summary of the results for producing vector fields about static curves follows.

To construct the vector field about a curve of interest we must first describe it as the intersection of two surfaces. We must define these surfaces mathematically as zero-level surface functions, $\alpha_i(x,y,z)$, which are defined such that the function is equal to zero only when on the surface and non-zero otherwise. For example, if we wanted to define a cylinder in $\mathbb{R}^3$, which can be written as $r = x^2 + y^2$, the zero-level function would be $\alpha(x,y,z) = x^2 + y^2 - r$. Once both zero-level surface functions have been defined, a Lyapunov-like potential function must be created. The function that we adopted was

$$V(x,y,z) = \frac{1}{2}\alpha_1^2 + \frac{1}{2}\alpha_2^2. \tag{2.8}$$

With these functions defined, we can use the proceeding explicit equation to easily produce the path following vector field in three dimensions:

$$u = \underbrace{-G\nabla V - \overbrace{\begin{pmatrix} \nabla\alpha_1 \\ \nabla\alpha_2 \\ \nabla\alpha_1 \times \nabla\alpha_2 \end{pmatrix}^{-1} \begin{pmatrix} \frac{\partial \alpha_1}{\partial t} \\ \frac{\partial \alpha_2}{\partial t} \\ 0 \end{pmatrix}}^{\text{Time varying correction}}}_{\text{Convergence}} + \underbrace{H(\nabla\alpha_1 \times \nabla\alpha_2)}_{\text{Circulation}}, \tag{2.9}$$

where G is a positive definite, diagonal matrix which can be used to tune the relative strength of the various directions of the contraction term and H is a strictly positive or strictly negative definite, diagonal matrix. As we only have need of static curves, we will henceforth eliminate any time dependence in our function definitions.

A vector field convergence proof for the result of (2.9) is presented in the paper. The proof is analogous to the one discussed in Section 2.1 but due to the additional complexity associated

with following n-th dimension, time varying curves we will not discuss it in detail here. When this vector field method is applied for following static curves in three dimensions it has many similarities to the preceding method. Yet, while there are similarities, the Gonçalves method appears to have some distinct advantages when compared to the Lawrence vector field construction method.

Some of the advantages of the Gonçalves method include design simplicity, intuitive tuning, and simple implementation. For this method the designer must only select the two surface functions to describe the desired path and adopt a scalar potential function. At the present, to the best of the author's knowledge, this method has not been extended to construct a vector field which produces stable acceleration vector commands; thus, there may be a reason to accept the increased complexity of the Lawrence vector field construction method for the benefit of the feed-forward, error reduction acceleration term.

2.3 Comparison

These two vector field construction methods are quite similar in a number of respects. In fact, the work of Gonçalves is somewhat of an extension of the Lawrence results. Both of these works provide methods for constructing three dimensional vector fields along arbitrary curves of interest. The most significant difference between these two methods is the process for designing the vector fields. The resultant velocity command based vector fields of the two methods could be identical, but the processes of creating them are quite distinct. One of the most significant contributions of [9] is a very simple method for designing these vector fields. The designer only needs to describe the curve of interest as the intersection of two surfaces and select a scalar potential function, which is made significantly easier with the surface definitions available. This makes the process of producing a stable vector field much less complex than the method presented by Lawrence.

The other primary difference is that the Lawrence paper further extends the vector field method for use as an acceleration control law that includes the feed-forward term. This extension is much more complex to derive, loses much of the intuitive properties of the velocity control vector field laws, is difficult to tune and constrain to behave feasibly, and is significantly more complex to implement.

Consequently, the Gonçalves method is superior in simplicity for the designer as well as in the intuitive nature of manipulating and tuning the resultant vector fields. The Lawrence method includes the extension of the acceleration control law, with the feed-forward term. This term offers the possibility of reducing the tracking error in a physical system implementation where some steady-state error in tracking will assuredly be present. The drawback of the acceleration-based method is a substantial increase in design and implementation complexity. We will further compare the implementation and performance of these two vector field construction methods for use in tracking elliptical orbits in Chapter 3.

2.4 Implementation

We have adopted a system architecture that is well suited for use with vector field path following functions. A high-level illustration of the concept is depicted in Figure 2.2. This architecture divides the UAV navigation problem into four distinct problems: path planning, path management, path following, and low-level autopilot control and state estimation. The first three of these problems we will discuss in detail throughout this thesis. For the fourth block, the low-level autopilot control and state estimation problem, it is assumed that such a solution already exists and is available for implementation on UAV hardware. The state feedback information that will be required consists of estimates of inertial position, course angle, airspeed and the wind vector. A detailed method for this fourth block is developed and presented in [2]. We also assume that the UAV, on which the algorithms presented later in this paper are to be implemented, is equipped with a course angle hold loop, a pitch angle hold loop, an airspeed hold loop, a pitch rate hold loop, and a roll angle hold loop.

Throughout this thesis we rely on the standard fixed-wing aircraft Euler angles, roll (ϕ), pitch (θ), and yaw (ψ) that relate what we will define as the aircraft body frame and the inertial frame. These angles and frames are considered standardized in aircraft literature and many resources can be found describing them in detail [2] [12]. We will present these frame definitions and corresponding transformations throughout this paper in greater detail as needed.

The role of the path planner is to plan current or future vehicle paths that can then be communicated to the path manager. In this work we will investigate path planning schemes for

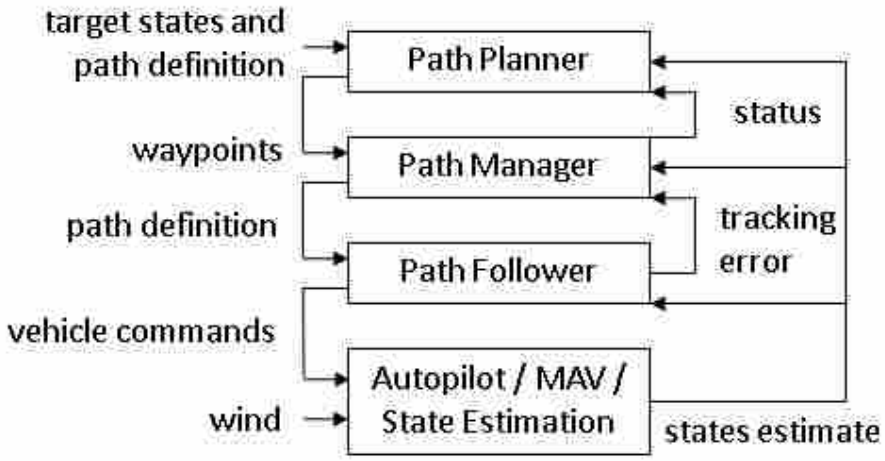


Figure 2.2: A high-level depiction of the system architecture used for the implementation of the vector field path following methods

interception of an aerial target vehicle. These path planning algorithms will be discussed in Chapter 5.

Path parameters are sent to the path manager from the path planner. In cases where a planned path consists of multiple legs that require the use of difference path following methods, the path manager is necessary. The purpose of this block is to determine which leg of the complete path the vehicle is currently on. It then communicates to the path follower the appropriate path definitions necessary for the vector field path following methods. The path management algorithm employed in this work for navigation along Dubins airplane paths is an extension of the path manager described in [2]. It was extended for use in three dimensions. This modified algorithm can be found in Appendix A.

The path follower implements the vector field path following methods that we have discussed in this chapter. It receives the path definition from the path manager, a flag that specifies which vector field is to be used (e.g., circular helix, inclined line, or elliptical orbit) and the necessary vehicle states. Depending on the method being employed, the outputs of the vector field equations are either a desired velocity vector or a desired acceleration vector. These commands are converted into the inertial frame and then converted into commands tractable by the low-level control loops with which we assumed the UAV is equipped.

This process of converting the resultant vector field commands into appropriate UAV commands is different for the two vector field methods we have discussed in this chapter. For the Lawrence method we will convert the acceleration commands into pitch rate, roll angle and airspeed commands using a modified polar converting logic. The desired velocity commands produced by the Gonçalves method will be converted into course angle, flight path angle, and airspeed commands. We now present two methods that produce the requisite vector field command conversions.

2.4.1 UAV Velocity Vector Controller

For fixed-wing aircraft there are basic kinematic equations that relate inertial velocities, the angles of travel in the horizontal and vertical directions, and what we define as ground speed. The angle between the north direction and the direction of travel in the horizontal plane we define as the course angle (χ). The angle between the magnitude of the vertical velocity and horizontal velocity in the inertial frame is defined as the flight path angle (γ). This relationship is presented mathematically as

$$\mathbf{V}_g = \begin{pmatrix} \cos\chi & -\sin\chi & 0 \\ \sin\chi & \cos\chi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\gamma & 0 & -\sin\gamma \\ 0 & 1 & 0 \\ \sin\gamma & 0 & \cos\gamma \end{pmatrix} \begin{pmatrix} V_g \\ 0 \\ 0 \end{pmatrix} = V_g \begin{pmatrix} \cos\chi\cos\gamma \\ \sin\chi\cos\gamma \\ -\sin\gamma \end{pmatrix} = \begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{pmatrix} \tag{2.10}$$

Using this relationship we can solve for equivalent ground speed, course angle and flight path angle commands. These are calculated as

$$V_g^d = \sqrt{(\dot{p}_n^d)^2 + (\dot{p}_e^d)^2 + (\dot{p}_d^d)^2} \tag{2.11}$$

$$\gamma^d = \arcsin\left(-\frac{\dot{p}_d^d}{V_g^d}\right) \tag{2.12}$$

$$\chi^d = \arctan 2(\dot{p}_e^d, \dot{p}_n^d) \tag{2.13}$$

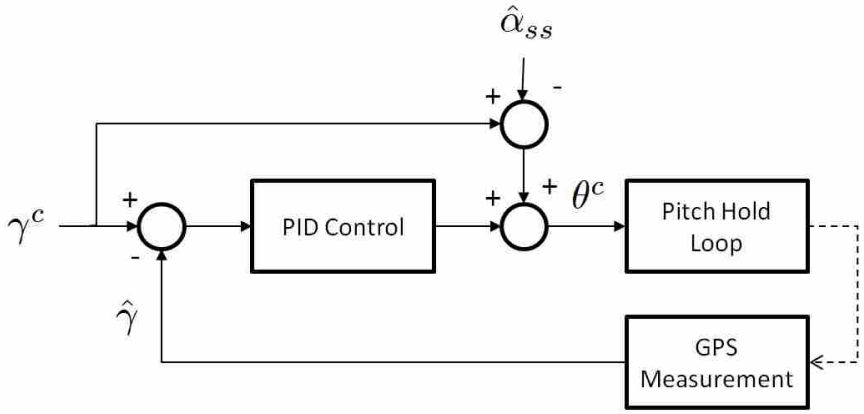


Figure 2.3: Flight path angle controller block diagram

Prior to this work the low cost MAV hardware that we employed did not have a flight path angle hold loop. Rather, the previous control schemes relied primarily on altitude control for vehicle navigation in the vertical direction. The flight path angle tracking scheme that we propose implementing is a standard proportional, integrator, derivative (PID) loop where the control output is a pitch angle command (θ_c). This flight path angle loop is wrapped around a pitch angle hold loop.

To improve the tracking capabilities of the controller we make use of the relationship between the flight path angle (γ), pitch angle(θ) and what is defined as the angle of attack (α). As shown in [2], in the absence of wind, this relationship is given by

$$\theta = \gamma + \alpha. \tag{2.14}$$

Since the angle of attack is not a constant angle and is difficult to estimate on a MAV, we will use a steady-state approximation, $\hat{\alpha}_{ss}$. Our feed-forward term then becomes

$$\theta_{ff}^c = \gamma^d + \hat{\alpha}_{ss}. \tag{2.15}$$

Figure 2.3 provides a block diagram representation of the flight path angle hold loop implementation.

2.4.2 UAV Acceleration Vector Controller - Modified Polar Converting Logic

Polar converting logic is a method of producing pitch rate and roll angle commands from vehicle acceleration commands. The basic logic of this method is to align the vehicle lift force with the combined desired lateral and longitudinal acceleration commands. Detailed descriptions of polar converting logic can be found in [2] [11]. We begin by defining a number of air vehicle reference frames.

The aircraft body frame has the the x-axis aligned out the vehicle nose, the y-axis out the right wing, and the z-axis out the bottom of the plane. The origin is at the center of mass of the vehicle. This frame is reached from the inertial frame with a 3-2-1 (yaw (ψ), pitch (θ) then roll (ϕ)) Euler rotation:

$$\mathfrak{R}_i^b(\psi,\theta,\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.16}$$

The notation is such that the subscript of $\mathfrak{R}$ refers to the frame of a vector prior to the transformation and the superscript specifies the frame after the transformation.

The vehicle-2 frame is nearly the same as the body frame with the exception of the one Euler rotation. It is reached using a 3-2 Euler rotation. The origin is also at the vehicle center of mass. The effect is that the frame is yawed and pitched with the vehicle but not rolled. This rotation can be represented as

$$\mathfrak{R}_i^{v2}(\psi,\theta) = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.17}$$

Polar converting logic references the vehicle acceleration commands in the vehicle-2 frame. We represent the desired acceleration commands in the inertial frame as

$$\ddot{\vec{r}}^{d} = \begin{pmatrix} a_x^i \\ a_y^i \\ a_z^i \end{pmatrix}. \tag{2.18}$$

These commands can be converted into the vehicle-2 frame using the transformation presented in (2.17). The result of this transformation is

$$\begin{pmatrix} a_x^{v2} \\ a_y^{v2} \\ a_z^{v2} \end{pmatrix} = \Re_i^{v2} \begin{pmatrix} a_x^i \\ a_y^i \\ a_z^i \end{pmatrix}. \tag{2.19}$$

This method of converting acceleration commands into a roll command and a pitch rate was designed for bank-to-turn missile controllers. One of the limitations of polar converting logic that makes it impractical for implementation for small UAVs is a discontinuity in the commanded roll angle as the vertical component of desired acceleration vector changes sign. For vehicles with relatively high roll rates, such as missiles, this may not be a limitation but is not practical for use with small, UAVs as they are not capable of achieving relatively high roll rates. Another limitation of this method is that it does not consider the effects of the gravity force. This is an assumption that may be valid if the vehicle lift force is always much larger than the gravity force, but for small UAVs this is generally not the case.

The standard polar converting algorithm is

$$\phi^c = \tan^{-1}\left(\frac{a_y^{v2}}{-a_z^{v2}}\right) \tag{2.20}$$

$$\dot{\theta}^c = -\text{sign}(a_z^{v2})\sqrt{(a_y^{v2})^2 + (a_z^{v2})^2}\frac{1}{V_g^d}. \tag{2.21}$$

The modified algorithm we employed that used the absolute value function is

$$\phi^c = \tan^{-1}\left(\frac{a_y^{v2}}{abs(a_z^{v2})}\right) \tag{2.22}$$

$$\dot{\theta}^c = -\text{sign}(a_z^{v2})\sqrt{(a_y^{v2})^2 + (a_z^{v2})^2}\frac{1}{V_g^d}. \tag{2.23}$$

This modification was motivated to make the polar converting logic scheme suitable for use with fixed-wing UAVs by eliminating the command of overly aggressive maneuvers. The only difference between the two algorithms is in the evaluation of the roll command, ϕ^c. In the modified (2.23), the sign of longitudinal acceleration term was ignored by evaluating the absolute value of this term. This has the desirable effect of making the direction of the roll angle only dependent on the sign of the desired lateral acceleration term.

The logic that motivated this modification is illustrated by considering a fixed-wing UAV attempting to navigate along a clockwise, flat, circular orbit, with an achievable radius. For the UAV to track such a path in near-steady-state conditions it seems intuitive that the vehicle would be rolled to the right at a near-constant angle to produce the appropriate centripetal acceleration. While maintaining this roll angle the vehicle would appropriately pitch up or down to maintain the desired altitude. This is the type of vehicle commands that we attempted to mimic with the modified polar converting logic scheme.

Considering the standard polar converting logic in the same scenario illustrates how it is not appropriate for our application. When the vehicle is below the desired orbit altitude the longitudinal acceleration command would be positive and the resultant roll and pitch commands would be identical to the above situation. The problem arises when the vehicle is above the desired orbit altitude causing a negative longitudinal acceleration commands. In this situation the vehicle would reverse the sign of the commanded roll angle and pitch down. Thus, whenever the vehicle would fluctuate above and below the desired altitude it would result in an aggressive UAV command being produced and cause significant, undesirable dynamic effects. The consequence of such a scheme would likely be to increase overall tracking error in comparison to a more conservative control commands. Simulation results of the standard polar converting logic method lead to the discovery of the algorithm infeasibility for use with a fixed-wing vehicle. These results also led to the development of the modified polar converting logic algorithm.

In this thesis, the modified polar converting logic algorithm is subsequently used only in simulation results for tracking elliptical orbits in Chapter 3. The acceleration command based vector field method presented in Section 2.1 was not selected for hardware implementation. As this algorithm is not significant to the final conclusions presented in this thesis we provide no further discussion of the modified algorithm. Nevertheless, if converting acceleration commands into tractable UAV commands is of interest, further study and analysis of the modified polar converting logic algorithm are recommend.

2.4.3 Feed-Forward Roll Command

One of the limitations of this Gonçalves based vector field path following methods is that it does have an anticipatory vector field component when following curves paths. A method has been proposed for producing feed-forward roll angle commands that based on the coordinated turn assumption, should significantly reduce the steady-state error when tracking curved paths [13]. This is accomplished by computing a nominal roll angle command using this relationship:

$$\phi_{nom} = \frac{\dot{\chi} V_g}{g}, \tag{2.24}$$

where $\dot{\chi}$ is the appropriate course angle rate, V_g is the vehicle ground speed, and g is the acceleration due to gravity. This nominal roll angle can be added to the roll angle command produced by the course angle hold loop as depicted in Figure 2.4.

Calculating $\dot{\chi}$ varies based upon the curve being followed. In this paper we develop vector fields for following two different paths using the Gonçalves methods: circular helices and elliptical orbits.

We approximate the appropriate course rate angle of the circular helix by that of a flat circular orbit. This can be calculated as

$$\dot{\chi} = \frac{V_g}{\rho}, \tag{2.25}$$

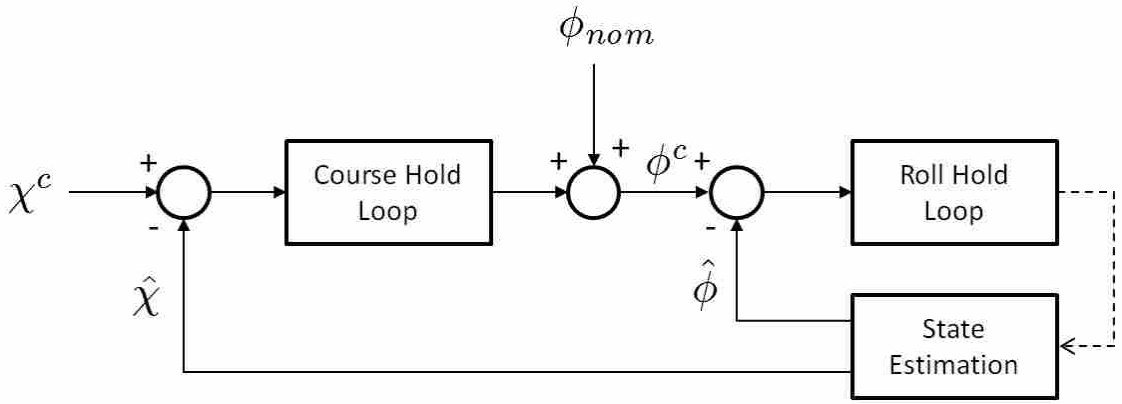


Figure 2.4: Block diagram illustrating the method for implementing the feed-forward roll commands.

where ρ_{circ} is the radius of the circular helix and $\rho_{ellipse}$ is the effective radius of the elliptical orbit that is to be followed. The effective radius of an elliptical orbit can be computed with the following equation:

$$\rho_{ellipse} = \frac{(a^2\sigma^2 + b^2\sigma^2)^{\frac{3}{2}}}{ab} \tag{2.26}$$

where a and b are the ellipse major and minor radii and σ is the parametric variable used to describe the angular position of the vehicle on the ellipse. This parametric relationship for an ellipse is represented as

$$x = a\cos\sigma \tag{2.27}$$

$$y = b\sin\sigma \tag{2.28}$$

and can be used to solve for σ give the vehicle position relative to the ellipse center.

2.5 Conclusions

In this chapter we presented two vector field construction methods which we refered to as the Lawrence method and the Gonçalves method. These methods can be used to produce path following vector fields about arbitrary curves. The output of the Lawrence method was a vector of inertial acceleration commands. The output of the Gonçalves method was a vector of inertial velocity commands. We presented methods for converting the output from both of these vector field equations into commands that can be tracked using lower-level control loops. We also presented low-level controllers that can be used to implement these path following methods onboard a MAV. In Chapters 3 and 4 we use the path following methods that we presented in this chapter to develop vector fields for following elliptical orbits and time-optimal Dubins airplane paths which are defined in Chapter 4.

CHAPTER 3. ELLIPTICAL ORBIT TRACKING

In this chapter we use the the Lawerence and Gonçalves vector field construction methods presented in the previous chapter to produced vector fields for following an arbitrarily oriented ellipse. Both of these elliptical orbit tracking methods are simulated. A comparison of the two methods is provided. We also present a scheme for nondimensionalizing the Gonçalves based vector field that allows a single set of vector field gains to be used much more generally. These elliptical orbit path following methods make it possible to track the drogue after the MAV has been inserted onto the drogue orbit.

The principle role of this research is to produce a method for inserting the MAV onto the drogue orbit and then, while navigating along the drogue orbit, gradually decrease the distance between the two vehicles to within 10 meters. The goal is to use the method presented in this research to facilitate a final visual rendezvous. This chapter presents post-insertion, orbit tracking methods. The vector field path following methods discussed in the previous chapter are well suited for solving this problem. Using these methods we develop two elliptical orbit path following vector fields and compare them in simulation. The methodology of the insertion scheme is presented in Chapter 5.

To develop these drogue orbit following methods, estimates of the drogue orbit are needed. As has been previously demonstrated in [1], when the mothership flies a circular orbit while towing the drogue in steady-state flight, the drogue will adopt an interior orbit; in the presence of constant wind this orbit will be roughly elliptical and rotated out of the horizontal plane. Prior research developed a numerical method for approximating the elliptical drogue orbit using the drogue GPS measurements [4]. This estimation method provides us with the necessary estimates of the drogue elliptical orbit parameters for an elliptical orbit tracking scheme: the orbit center, the major and minor radii, and the 3-1-3 Euler angles that relate the ellipse frame and the inertial frame.

One factor that must be considered when using this method is that the accuracy is limited primarily by the resolution of GPS, our primary positioning sensor. These GPS measurements do not provide sufficient accuracy to directly facilitate a rendezvous between the two vehicles. For this reason we have adopted the ellipse-following scheme as a mechanism for enabling a final visual rendezvous and docking.

With the estimated elliptical orbit parameters it is possible to create a vector field path following scheme that can be used to track the drogue. Before developing the two vector field equations we begin with definitions common to both methods. We define the elliptical orbit vector field frame with the x and y axes aligned with the major and minor ellipse axes respectively and the z axis oriented perpendicular to the plane containing the ellipse. The origin of the ellipse frame is located at the orbit center. The combined 3-1-3 Euler rotation matrix for converting a vector in the north, east, down (NED) inertial frame to the ellipse frame is written as

$$\mathfrak{R}_i^e = \begin{pmatrix} \cos\psi_2 & \sin\psi_2 & 0 \\ -\sin\psi_2 & \cos\psi_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_1 & 0 & -\sin\theta_1 \\ 0 & 1 & 0 \\ \sin\theta_1 & 0 & \cos\theta_1 \end{pmatrix} \begin{pmatrix} \cos\psi_1 & \sin\psi_1 & 0 \\ -\sin\psi_1 & \cos\psi_1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (3.1)$$

where the subscript i specifies that the rotation begins in the inertial frame and ends in the ellipse frame, indicated by the superscript e. We define

$$\mathbf{c} = \begin{pmatrix} c_n \\ c_e \\ c_d \end{pmatrix} \quad (3.2)$$

as the relative location of the ellipse center in the inertial frame expressed in NED coordinates. With this definition we can present the complete transformation of an inertial position vector,

$$\mathbf{p} = \begin{pmatrix} p_n \\ p_e \\ p_d \end{pmatrix}, \tag{3.3}$$

in the inertial frame to the ellipse vector field frame where we denote the position vector as $\bar{r}$ (see vector field definitions in Chapter 2). This transformation is

$$\bar{r} = \mathfrak{R}_i^e (\mathbf{p} - \mathbf{c}). \tag{3.4}$$

We will begin by constructing the vector field that produces an acceleration vector, as described in Section 2.1, followed by a simpler method that yields velocity vector commands (see Section 2.2) and then we will compare the simulation results of the two.

3.1 Vector Field with Desired Acceleration Vector

A vector field path following scheme for tracking arbitrarily oriented elliptical orbits is presented in this section. This method was developed by drawing upon the principles presented by Lawrence, et al. in [7], which were covered in Chapter 2. We will refer to this vector field method as the Lawrence method. Using these methods, a stable vector field for navigation along arbitrary elliptical orbits can produced. For the sake of simplicity we have opted to use the notation presented in [7] when constructing the elliptical orbit, acceleration command based vector field.

Three unit vectors are used to describe the ellipse frame axes. They are defined as $\hat{n}_1$, $\hat{n}_2$, and $\hat{n}_3$ where these vectors give the directions of the major axis, the minor axis, and the normal to the ellipse plane as a right-handed frame, respectively.

From the equations

$$r_1 = \hat{n}_1^T \bar{r}, \quad r_2 = \hat{n}_2^T \bar{r}, \quad \textit{and} \quad r_3 = \hat{n}_3^T \bar{r} \tag{3.5}$$

we produce the scalar values of $\bar{r}$ in the directions of the ellipse frame axes. We define a scalar potential function $V_F(\bar{r})$, to be used in (2.1) and (2.2), where $\mathcal{C}$ is the zero-level set of the otherwise positive function as

$$V_F(\bar{r}) = \frac{1}{2}r_3^2 + \frac{1}{2}\left(\left(\frac{r_1}{a}\right)^2 + \left(\frac{r_2}{b}\right)^2 - 1\right)^2. \tag{3.6}$$

In this equation, a and b are respectively the major and minor radii of the desired loiter ellipse. From (3.6) we can derive:

$$\frac{\partial V_F}{\partial \bar{r}} = r_3\hat{n}_3^T + 2\left(\left(\frac{r_1}{a}\right)^2 + \left(\frac{r_2}{b}\right)^2 - 1\right)\left(\frac{r_1\hat{n}_1^T}{a^2} + \frac{r_2\hat{n}_2^T}{b^2}\right). \tag{3.7}$$

We then define a function $T(\bar{r})$ that generates the unit tangent vector to an ellipse that is parallel to the desired loiter ellipse at the vehicle location described by $\bar{r}$:

$$T(\bar{r}) = -\left(\frac{ar_2}{b}\right)\hat{n}_1 + \left(\frac{br_1}{a}\right)\hat{n}_2. \tag{3.8}$$

The Lyapunov vector field (2.2) is then obtained with the following definitions:

$$\Gamma(\bar{r},t) = \frac{1}{\alpha(\bar{r},t)}\begin{bmatrix} g_1 & 0 & 0 \\ 0 & g_2 & 0 \\ 0 & 0 & g_3 \end{bmatrix} \tag{3.9}$$

and

$$S(\bar{r},t) = \gamma(t)\frac{T(\bar{r})}{\alpha(\bar{r},t)}, \tag{3.10}$$

where $\gamma(t) > 0$ and the speed normalization is given by:

$$\begin{aligned}\alpha(\bar{r},t) = \frac{1}{v^d}\Bigg(&\hat{n}_1\left(\frac{2g_1r_1^3}{a^4}+\frac{2g_1r_1r_2^2}{a^2b^2}-\frac{2g_1r_1}{a^2}+\frac{\gamma a r_2}{b}\right)^2\\&+\hat{n}_2\left(\frac{2g_2r_2^3}{b^4}+\frac{2g_2r_2r_1^2}{a^2b^2}-\frac{2g_2r_2}{b^2}+\frac{\gamma a r_1}{b}\right)^2\\&+\hat{n}_3\left(g_3r_3\right)^2\Bigg)^{\frac{1}{2}}.\end{aligned} \tag{3.11}$$

Combining (2.2), (3.7), (3.8), (3.9), (3.10), and (3.11) yields the normalized vector field equation:

$$\dot{\bar{r}}^d(\bar{r}) = -v^d\frac{h(\bar{r})}{|h(\bar{r})|}, \tag{3.12}$$

where

$$\begin{aligned}h(\bar{r}) =&\hat{n}_1\left(\frac{2g_1r_1^3}{a^4}+\frac{2g_1r_1r_2^2}{a^2b^2}-\frac{2g_1r_1}{a^2}+\frac{\gamma a r_2}{b}\right)\\&+\hat{n}_2\left(\frac{2g_2r_2^3}{a^4}+\frac{2g_2r_1^2r_2}{a^2b^2}-\frac{2g_2r_2}{b^2}+\frac{\gamma b r_1}{a}\right)\\&+\hat{n}_3\left(g_3r_3\right).\end{aligned} \tag{3.13}$$

The vector field is characterized by two terms: a circulation term and a contraction term. These terms are described in Chapter 2 but for simplicity the vector field equation was combined and simplified. The circulation term is represented by all of the terms in (3.13) that contain the circulation gain γ. All of the other terms cause contraction toward the elliptical orbit. A single gain, γ, controls the relative strengths of contraction toward the orbit and circulation about the orbit. This vector field is undefined about the axis through the center of the ellipse but behaves well everywhere else. An anisotropic Γ, as used in (3.9), can be used to vary the relative strengths of the contraction terms in different directions. An example velocity command based vector field produced using this method can be seen in Figure 3.1. For instance, one could use the ellipse parameters to produce the rotation matrix between the ellipse frame and the inertial frame to then

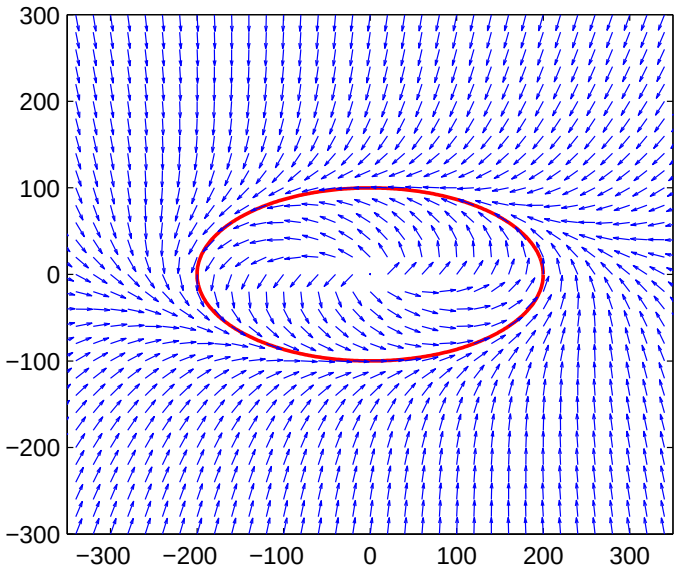


Figure 3.1: An example velocity command based vector field for an elliptical orbit. This vector field was developed using the Lawerence vector field construction method.

reduce the contraction term in the vertical direction to be within the climb rate performance parameters of the vehicle for which this scheme is being implemented.

An acceleration control law was derived from the Lyapunov vector field equation (see (2.5)) that produces an acceleration vector in the ellipse frame. As discussed in detail in Chapter 2, this control law is made up of a proportional tracking term and a feed-forward term, $\ddot{\bar{r}}_d = -P + \ddot{\bar{r}}_{ff}$, where the proportional term was selected to be

$$P = \beta(\dot{\bar{r}}_d - \dot{\bar{r}}). \tag{3.14}$$

Thus, the vector field equation for commanded acceleration can be written as

$$\ddot{\bar{r}}_d(\dot{\bar{r}}, \bar{r}) = -\beta(\dot{\bar{r}}_d - \dot{\bar{r}}) + \ddot{\bar{r}}_{ff}. \tag{3.15}$$

Deriving the feed-forward term can be somewhat involved. Using the simplified vector field equation (see (3.12)) makes this derivation less formidable. To produce the feed-forward term in (2.7) we must take the time derivative of (3.12). Doing so produces

$$\ddot{\vec{r}}_{ff} = -v^d \left(\frac{|h| \frac{d}{dt} h - h \frac{d}{dt} |h|}{|h|^2} \right), \tag{3.16}$$

where

$$\begin{aligned} \frac{d}{dt} h =& \hat{n}_1 \left(\frac{6 g_1 r_1^2 \dot{r}_1}{a^4} + \frac{2 g_1 r_2^2 \dot{r}_1}{a^2 b^2} + \frac{4 g_1 r_1 r_2 \dot{r}_2}{a^2 b^2} - \frac{2 g_1 \dot{r}_1}{a^2} + \frac{\gamma a \dot{r}_2}{b} \right) \\ &+ \hat{n}_2 \left(\frac{6 g_2 r_2^2 \dot{r}_2}{b^4} + \frac{2 g_2 r_1^2 \dot{r}_2}{a^2 b^2} + \frac{4 g_2 r_1 r_2 \dot{r}_1}{a^2 b^2} - \frac{2 g_2 \dot{r}_2}{b^2} - \frac{\gamma b \dot{r}_1}{a} \right) \\ &+ \hat{n}_3 \left(g_3 \dot{r}_3 \right) \end{aligned} \tag{3.17}$$

and

$$\frac{d}{dt} |h| = \frac{h \cdot \frac{d}{dt} h}{|h|}. \tag{3.18}$$

The proportional term, which was defined in (3.14), produces accelerations that direct the vehicle toward the desired velocity vector as specified by the Lyapunov vector field. The feed-forward term, which is the derivative of the vector field equation, contributes acceleration commands that anticipate the vector field turn rate. The intent of including feed-forward term is to increase the ability of the controller to accurately track the elliptical orbit. The commands of this control law are guaranteed to converge to the desired orbit in finite time. One of the limitations of this controller is that it does not take into account aircraft flight constraints; due to the complexity of the acceleration command control law, it is not obvious how one could constrain the relative rates of acceleration.

3.2 Vector Field with Desired Velocity Vector

The Gonçalves vector field construction method (see Section 2.2) is quite simple when compared to the Lawrence method (Section 2.1). As has already been discussed, to construct a vector field about a curve of interest using the Gonçalves method, the designer must select a scalar potential function and define the path as a zero-level surface function. In Section 2.2 we presented the potential function which we adopted for this work. As we already have presented a suitable potential function, we only need to define the surface functions and evaluate the vector field construction equation.

Our selection of the ellipse frame simplifies the zero-level surface functions needed to define the ellipse we wish to follow. We describe the ellipse of interest with the appropriate elliptical cylinder and the plane containing the x and y axes. These zero-level surface functions can be expressed as

$$\alpha_1(x,y,z) = \left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 - 1 \tag{3.19}$$

$$\alpha_2(x,y,z) = z. \tag{3.20}$$

To complete the vector field construction we must evaluate the explicit vector field equation presented in (2.2). Substituting these surface functions into (2.9) and evaluating yields

$$\dot{\bar{r}}^d = -G \begin{pmatrix} \frac{2x}{a^2}\left(\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 - 1\right) \\ \frac{2y}{b^2}\left(\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 - 1\right) \\ z \end{pmatrix} + H \begin{pmatrix} -\frac{2y}{b^2} \\ \frac{2x}{a^2} \\ 0 \end{pmatrix}. \tag{3.21}$$

This vector field equation provides a velocity vector command in the ellipse frame. Similar to the case present in the previous section, the velocity command is made up of a circulation force and a contraction force. These forces can be tuned with a single gain H which must be a strictly positive or strictly negative diagonal matrix. The sign of H determines the direction of circulation (e.g., clockwise or counterclockwise). The matrix G can be tuned to control the relative strengths

of the contraction force in the directions of the ellipse frame or inertial frame axes, but it must be a positive definite matrix. This matrix can be used to limit the commanded vertical velocity to be within the MAV performance constraints. We found that using an isotropic H made tuning the vector field to achieve the desired rate of convergence very intuitive.

3.3 Vector Field Nondimensionalization

When deriving these vector field path following equations it is intuitive to use an inertial reference frame with units of meters, as derived above. One of the consequences of developing the equations in this manner is that the relative strengths of the two zero-level surface functions are dependent upon the ellipse radii, a and b. These surface functions, which are used to derive the vector field (3.21), each add energy to the energy potential function in (2.8) based upon how far the vehicle is from the zero-level curve. Consequently, the relative strength of the energy contributions to the potential function from the two zero-level surface functions is dependent upon the curve parameters mentioned above. This has the undesirable effect of causing appropriate vector field gains to vary substantially based on the curve parameters. For example, a gain that was appropriate for relatively small ellipse radii may perform very poorly for larger radii. Thus, it would be useful to eliminate the vector field gain dependence on the curve parameters to eliminate the need for a gain adaptation scheme.

We now present a simple method that can be used to mitigate this problem for each of the vector field equations presented in this paper produced using the vector field construction method of Goncavles, et al. The method entails eliminating the curve parameters from the vector field equations. This can be accomplished by nondimensionalizing some of the variables in the vector field equations. For the ellipse case where the vector field equations are developed in the right handed ellipse frame, we now define a relative position vector which is measured relative to the ellipse parameters a and b. The components of this vector are defined as follows:

$$x_{rel} \triangleq \frac{x}{a}, \quad y_{rel} \triangleq \frac{y}{b}, \quad z_{rel} \triangleq z. \tag{3.22}$$

Thus, for a constant elliptical orbit,

$$\dot{x} = a\dot{x}_{rel}, \quad \dot{y} = b\dot{y}_{rel}, \quad \dot{z} = \dot{z}_{rel}. \tag{3.23}$$

We use this result to produce a conversion from the relative vector field commands back to the units of a and b:

$$\dot{\bar{r}}^d = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{pmatrix} \dot{\bar{r}}^d_{rel}. \tag{3.24}$$

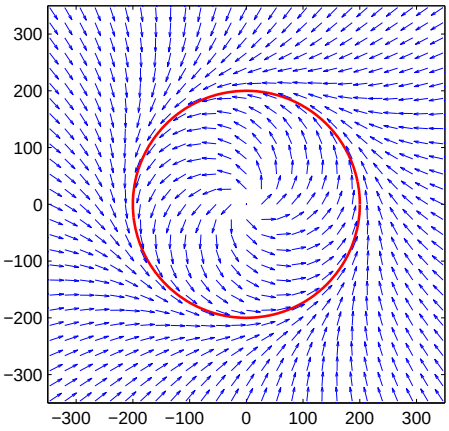
Substituting these results of (3.23) into the ellipse zero-level surface functions from (3.20) and then evaluating (2.9) yields the following non-dimensionalized field equation:

$$\dot{\bar{r}}^d_{rel} = -G \begin{pmatrix} 2x_{rel}(x^2_{rel} + y^2_{rel} - 1) \\ 2y_{rel}(x^2_{rel} + y^2_{rel} - 1) \\ z_{rel} \end{pmatrix} + H \begin{pmatrix} -2y_{rel} \\ 2x_{rel} \\ 0 \end{pmatrix}. \tag{3.25}$$

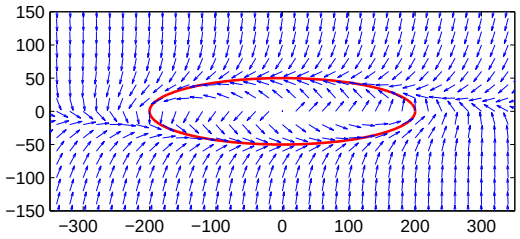
Essentially we are converting the units of the x and y directions to be non-dimensionalized and relative to the system parameters for use with the vector field equations. After producing the vector field commands we convert back to the appropriate units. Using the vector field equations we produce a desired velocity vector in these relative units. This provides the benefit of being able to use a single set of gains for more than a single set of ellipse parameters.

We present a simple visual comparison of the standard and non-dimensionalized vector field approaches to illustrate the benefits. The primary benefit can be easily visualized while ignoring the vertical dimension. As seen in Figure 3.2, for a well tuned vector field about a circular orbit (major and minor radii are equal), the standard approach and non-dimensionalized approach are identical if they have the same gains.

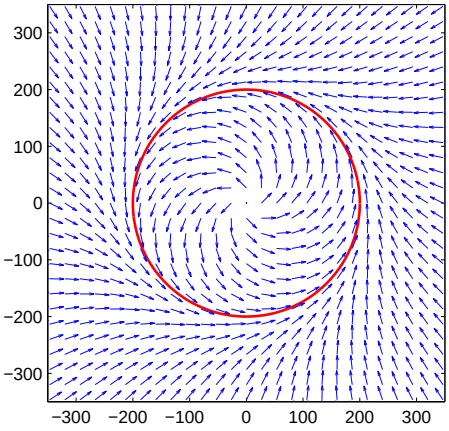
When we consider these two vector fields about an elongated ellipse, it is apparent that the vector fields behave quite differently for the two methods. The non-dimensionalized vector field appears to have the same desirable convergence properties as the circular orbit vector field. On the other hand, the standard vector field approach is dominated by the vector field contraction force in the direction of the ellipse minor axis. This is because the potential function is dependent upon the ellipse parameters. The relationship is such that the convergence force will always be



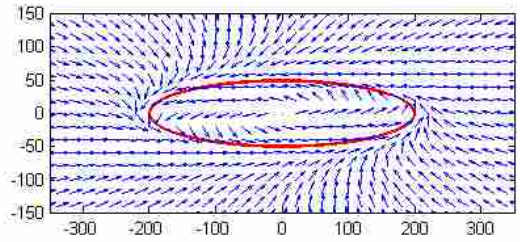
(a) Standard vector field: circular orbit



(b) Standard vector field: elongated ellipse



(c) Non-dimensionalized vector field: circular orbit



(d) Non-dimensionalized vector field: elongated ellipse

Figure 3.2: Comparison of standard vector field and non-dimensionalized vector field approaches with identical gains and orbit parameters.

dominated by the direction of the minor axis if the minor and major radii are unequal. This is also evident in Figure 3.2. Consequently, the standard vector field causes rapid convergence toward the major axis. The convergence properties of the standard vector field about the elongated ellipse are also substantially more aggressive than the circular orbit case, which is undesirable. It would be preferable to have similar convergence properties for a circular orbit and an elongated ellipse for a single set of gains.

The effect of using the vector field nondimensionalization technique is that the convergence properties are maintained for a given set of gains between distinct orbits. One of the limitations of

this method is that using the same gains for orbits of significantly different size may not perform well if the vector field integral curves violate the performance constraints of the vehicle.

To illustrate this point we compare two parallel ellipses (same ratio between the major and minor radii). For a single set of gains, an ellipse with $a = 100$ and $b = 50$ will have exactly the same non-dimensionalized vector field as an ellipse where $a = 50$ and $b = 25$. Thus, in the dimensionless space they will perform identically. When viewing the integral curves of these vector fields in their appropriate units their performance will be the same to a scaling factor.

We now consider the behavior of a vehicle with a minimum turn radius constraint. For a given set of gains, the vehicle's convergence characteristics toward the large ellipse may be quite desirable in part because the minimum turn radius of the vector field integral curves does not violate the vehicle's minimum turn radius. If this vehicle were to attempt to navigate around the smaller ellipse, this equivalent set of gains may not be appropriate because the integral curves will have half the minimum turn radius compared to those of the larger ellipse.

Even though the aircraft may be dynamically capable of navigating along the ellipse, the gains may be overly aggressive and result in undesirable oscillations back and forth over the desired elliptical orbit. Thus, this nondimensionalization scheme is good for eliminating the dependence of the vector field convergence properties on the ellipse parameters and not necessarily for preventing tuning from being needed between ellipses of various sizes. There is still a need to prevent the vector field properties from being overly aggressive for the dynamic constraints of the vehicle. Nevertheless, the variation in ideal gains between ellipses of different sizes varies much less with the dimensionless vector fields in comparison to the standard vector fields.

It is important when applying this technique to other types of paths that consideration is given to whether the nondimensionalized parameters could be normalized to infinity. This would occur for the ellipse case if either the major or minor radii were equal to zero. Even if the radii were only near zero it could still introduce substantial numerical error. Because it is impossible for a fixed-wing aircraft to navigate along an orbit with a radius of zero, for this specific case it is not a concern but it may be for other types of paths.

3.4 Simulation Results

Simulation results of both of the elliptical vector field path following methods and the subsequent methods presented in this thesis were produced using a second-order, six-degree-of-freedom, MAV model. This dynamic model uses realistic aerodynamic and propulsion models as well as a the Dryden wind gust model [12]. These can be found in Appendix B. The Matlab, Simulink simulation environment and the ode45 numerical differential equation solver were used to accurately solve the aircraft dynamic equations for our simulation purposes. The aircraft dynamic models and equations used were adopted from [2].

Low-level autopilot controllers were also developed in [2] which we adopted to use. These consisted of an airspeed controller, pitch angle controller, and a course angle hold loop which relied on inner roll angle loop controller. For the acceleration command method the low-level controllers required were an airspeed controller, a roll angle controller and a pitch rate controller. The gains for each of the loops were tuned according to the methodology presented in the above mentioned work. These controllers make up the low-level autopilot.

Since an investigation of state estimation methods is not one of the primary objectives of this work, we elected to use true state feedback with the vector field path following equations and the low-level autopilot loops. State estimation techniques would certainly introduce errors and system variability that would make analysis of the vector field path following methods presented here significantly more complicated. Thus, the motivation of this decision was to increase the visibility of the performance characteristics of the vector field path following methods.

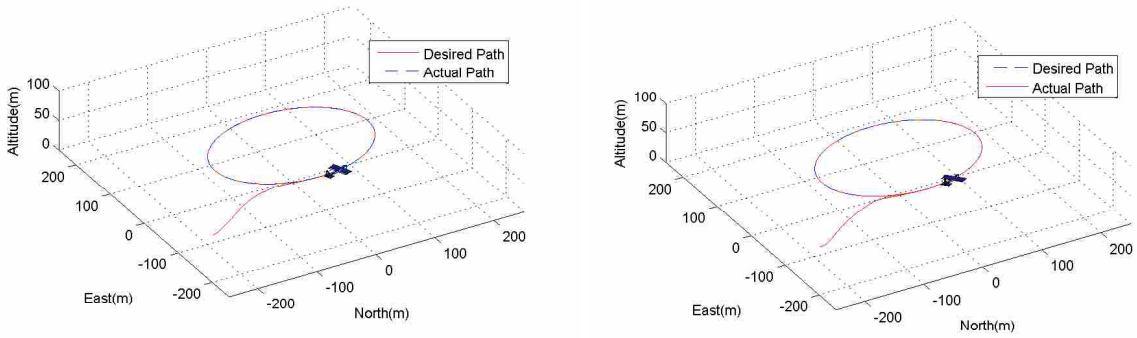
As such, the simulation results we present are intended to demonstrate the general behavior of these vector field path following methods. Experimental results collected from hardware implementations of these algorithms will be used to validate the results presented in this and subsequent simulation results sections (see Chapter 6).

We used a reduced version of the system architecture model which was presented in Figure 2.2. The blocks used for the simulation presented in this section made use of the path following and autopilot blocks. The vector field path following equations were implemented in the path following block. The inputs into the path following block were those specifying the ellipse parameters. The low-level control loops that we used corresponded to those appropriate for the particular vector field method being employed as presented in Sections 2.4.1 and 2.4.2.

Table 3.1: Initial conditions and parameters used in the simulation of the two ellipse following methods.

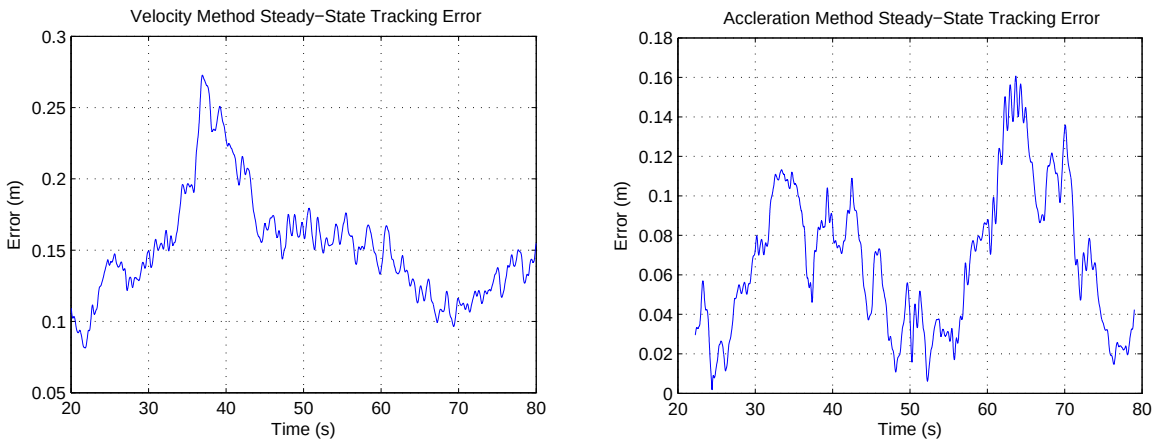
Vehicle Initial Conditions	
p_n (m)	-205
p_e (m)	-125
h (m)	0
Airspeed (m/s)	14
Ellipse Parameters	
n_0 (m)	0
e_0 (m)	0
h_0 (m)	60
ψ_1 ($^\circ$)	20
θ ($^\circ$)	5
ψ_2 ($^\circ$)	0
a (m)	150
b (m)	100
Wind Initial Conditions	
north (m/s)	1
east (m/s)	-3
down (m/s)	0

Each of the elliptical vector field path following methods were simulated using the same initial conditions, ellipse parameters, constant wind and wind gusts. These can be found in Table 3.4. The wind gusts were produced using the Dryden model with simulated white noise. Identical white noise function seeds were used for the two simulations.



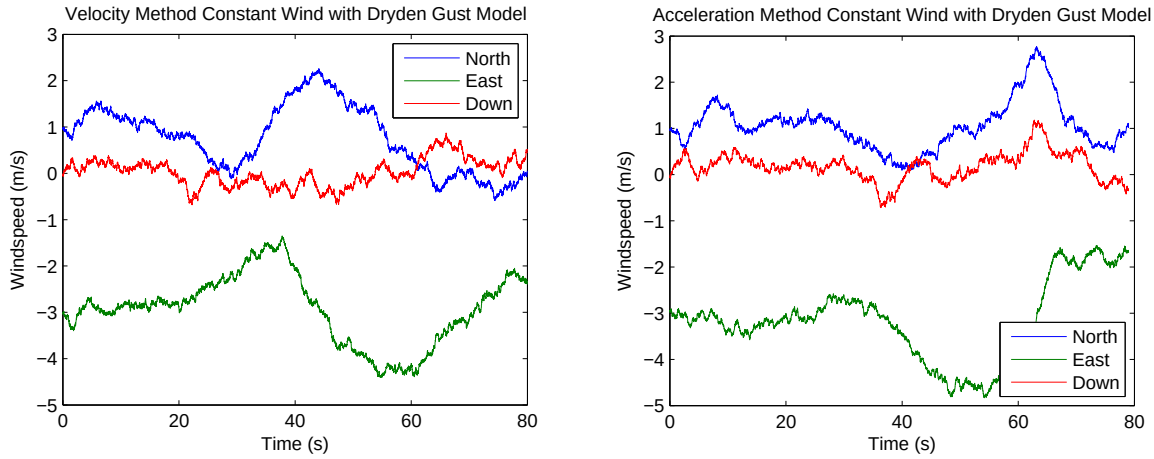
(a) Velocity method (b) Acceleration method

Figure 3.3: Plots of the actual 3D path flown for the two ellipse following simulations.



(a) Velocity method (b) Acceleration method

Figure 3.4: Plots of the steady-state tracking error of the two ellipse following methods.



(a) Velocity method (b) Acceleration method

Figure 3.5: Plots of the north-east-down wind components for the two ellipse following methods.

Plots of the north-east-down components of the simulation wind can be found in Figure 3.5. Since the wind gust model presents the gusts in the body frame of the vehicle rather than the inertial frame, the wind magnitude plots are not identical. In these simulations the magnitude of the wind was approximately 30% of the commanded airspeed. The presence of wind contributes to the steady-state error of the tracking methods as the vehicles must respond to the wind by crabbing at the appropriate angle to maintain the appropriate course angle.

Table 3.2: Summary of performance characteristics of the two elliptical orbit path following methods.

	Gonçalves Method	Lawrence Accleration Method
Max Steady-State Error (m)	0.273	0.161
Max γ ($^\circ$)	21.68	34.72
Min γ ($^\circ$)	-4.68	-6.10
Max $\dot{\chi}$ (rad/s)	1.12	1.23
Min $\dot{\chi}$ (rad/s)	-0.765	-0.338
Max $\dot{\theta}$ (rad/s)	1.54	0.662
Min $\dot{\theta}$ (rad/s)	-0.559	-0.196
Min Turn Radius (m)	157.0	156.3

Review of the performance characteristics of the two methods, found in Table 3.4, shows that the two vector field methods behaved similarly and within the dynamic limitations typical of a comparable MAV; these realistic performance characteristics were only achieved for the acceleration method after a significant amount of time was spent tuning the vector field gains and artificially constraining them. The velocity based vector field method required minimal tuning of the vector fields to produce the desirable convergence behavior presented here.

Inspection of Figure 3.3 shows that both methods effectively tracked the desired ellipse. The only visible difference on the plots appear when the vehicles are converging towards the ellipse. Figure 3.4 displays the steady-state, ellipse tracking error of the two methods. It is apparent that the acceleration method performed slightly better than the velocity method. We attribute this reduction in the steady-state error to the the feed-forward term which produces the centripetal acceleration commands necessary for the vehicle to maintain the appropriate turn rate. This steady-state error is a function of the response time of the lower-level autopilot loops and is dependent on the performance characteristics of these loops. These results show that there is some benefit to using the acceleration method over the velocity method, but this potential benefit comes at the cost of a large increase in system complexity.

The complexity difference between these two vector field methods is significant. We found implementation of the velocity method to be relatively simple. Minimal time was spent tuning gains since they produce intuitive results. With well tuned lower-level autopilot loops, we found the vector field equation to be easily tuned using the gains G and H in the vector field equation,

(see (3.21)). Each of these gains could be modified to increase the rate of convergence until the commands became overly aggressive and detrimental oscillations began to occur. These gains were then decreased slightly until an acceptable amount of oscillation occurred along the elliptical orbit.

Tuning the acceleration vector field gains was quite challenging. One of the challenges was to prevent the feed-forward term from dominating the acceleration commands with highly aggressive acceleration commands. Multiple constraints were needed to prevent the vehicle from following impractical commands with the vector field equations as well as with the implementation of polar converting logic. We feel that the results from the proceeding simulation of this method do not provide justification for using this, the Lawrence method, over the Gonçalves velocity method. Thus we will not discuss all of the tweaking and tuning of the vector field equations and the polar converting logic function that went into making this method work for one particular set of ellipse parameters. We will discuss some of the performance characteristics that make this method difficult to implement.

One of the undesirable characteristics of the acceleration method is that it consistently produces unrealistic acceleration commands that violate the dynamic constraints of fixed-wing MAVs. For instance, if the vehicle is oriented opposite the velocity commands from the proportional term (see (3.15)) the general result of the acceleration commands is to decrease the vehicle velocity to near zero and attempt a hairpin turn. There is no adequate method presented in [7] for constraining the acceleration method to produce airspeed commands that are reasonable for a MAV. To overcome this, the airspeed commands produced by the acceleration commands were overridden to get acceptable vehicle behavior. In most of the simulations we performed, the feed-forward term was overly strong and resulted in the vehicle flying an orbit interior to the desired ellipse path. Only when the acceleration commands are dominated by the proportional term are desirable results obtained. Preventing un-achievable vehicle climb rates that resulted in poor path tracking performance also required the artificial constraints to be placed on the vector field equations.

While the gains that we selected and used in simulation for following the ellipse were acceptable, these gains do not perform well for other ellipses with a significant variation of ellipse parameters. Our purpose in developing an elliptical orbit path following method is to be able to follow ellipses in general and not a single ellipse with predetermined parameters. Thus, we have selected the Gonçalves velocity based path following scheme for hardware implementation of the

elliptical orbit following. We believe that the potential ellipse tracking error reduction offered by the acceleration method does not compensate sufficiently for the substantial increase of complexity in the design and implementation process. We have also adopted the velocity command based vector field method to develop a scheme for following time-optimal Dubins airplane paths as found in the next chapter.

CHAPTER 4. TIME-OPTIMAL DUBINS AIRPLANE PATHS

In this chapter we define time-optimal Dubins airplane paths. The Gonçalves vector field construction method, discussed in Chapter 2, is used to produce a vector field path following based method for following these time-optimal paths. We present a scheme for nondimensionalizing one of the resultant vector field equations which makes tuning of the vector field gains much simpler. Simulation results for following these time-optimal Dubins airplane paths are presented and analyzed. In Chapter 5 these time-optimal paths will be used to plan and follow interception paths from the MAV to the drogue.

4.1 Path Definition

Various research has investigated the development of time-optimal paths for kinematic models of interest. One notable contribution involved the development of time-optimal paths in two dimensions for a kinematic model commonly described as a Dubins car [5]. The kinematics of the model represent a first-order approximation of a vehicle traveling at a constant velocity with a bounded turn rate. It can be written mathematically as follows:

$$\begin{aligned} \dot{x} &= V\cos\theta \\ \dot{y} &= V\sin\theta, \qquad V>0 \\ \dot{\theta} &= u, \qquad u:[a,\infty). \end{aligned}$$

This kinematic model has been used to describe time-optimal paths for various vehicles that have a lower bound on their turn rate. One example of such a vehicle is a fixed-wing aircraft that is flying at a constant altitude. While a fixed-wing aircraft is certainly not a first-order system, when flying at a constant velocity and in near-level flight the vehicle has a minimum turn-rate.

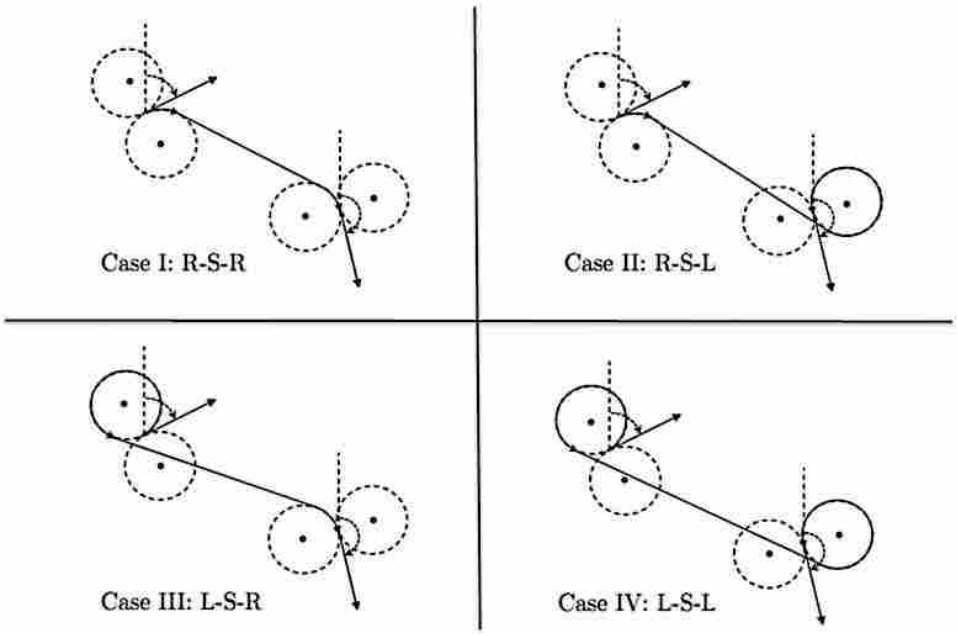


Figure 4.1: Four possible Dubins car path cases between an example start and end configuration where "R" signifies a right or clockwise turn and "L" a left or counter-clockwise turn. [2]

The Dubins car model is a simple approximation that can be used to model all paths achievable by the vehicle of interest. Furthermore, Dubins developed a closed-form solution for the time-optimal path of such a vehicle between two points in the two dimensional plane that have an initial and final heading constraint. Thus, from the work of Dubins we can plan time-optimal paths for a fixed-wing aircraft that closely represent paths that such a vehicle can feasibly follow. The result of this work is that one can solve for the time-optimal path of a fixed-wing aircraft between an initial configuration, a position with a heading constraint, and a final configuration in two dimensional space.

These time-optimal paths have been shown to consist of three unique legs. When the Euclidean distance separating the initial and final point is greater than three minimum turn radii, the time-optimal path always consists of an arc of minimum turn radius, followed by a straight segment, and is concluded by another arc of minimum turn radius. There are four possible configurations of such a path depending on the direction of travel on the two arcs of minimum turn radius (i.e., clockwise or counter-clockwise). An example of each of the four possible configurations is seen in Figure 4.1. To determine which path is the time-optimal path, each of their lengths must be calculated and the path of the shortest length is the time-optimal one.

Since fixed-wing UAVs are not restricted to traveling in the horizontal plane it becomes useful to investigate an extension of these time-optimal, two dimensional Dubins car paths to three dimensions. Chitzas, et al. developed such an extension for a kinematic model, which they referred to as a Dubins airplane [6]. The kinematic model of a Dubins airplane is nearly identical to the Dubins car with the addition of an altitude velocity that has an upper and lower bound. The kinematic model can be written as

$$\begin{aligned} \dot{x} &= V\cos\theta \\ \dot{y} &= V\sin\theta, \qquad V>0 \\ \dot{z} &= v, \qquad v:[-b,b] \\ \dot{\theta} &= u, \qquad u:[a,\infty). \end{aligned}$$

The result of [6] showed that, for the Dubins airplane kinematic model, the time-optimal path can be separated into three cases described as low goal altitude, medium goal altitude, and high goal altitude. The low goal altitude case consists of scenarios where the difference in altitude between the initial and final point is achievable while flying along the two dimensional, time-optimal Dubins car path but with the appropriate altitude velocity. Thus, for the low-goal altitude, if the time-optimal path were to be projected onto a horizontal plane it would be identical to the two dimensional, time-optimal path of a Dubins car. In three dimensions this path consists of two constant pitch, constant radii circular helices and an inclined straight line. The other cases are more involved and will not be discussed here in detail. Nevertheless, the method that is being proposed for following time-optimal, Dubins airplane paths is applicable to all three goal altitude cases. To the best of our knowledge, there is no presently published work that presents a method for following these Dubins airplane paths. Hereafter we present a method for following time-optimal, Dubins airplane paths for the low-goal altitude case.

4.2 Vector Field Construction

A number of schemes have been developed for following two dimensional Dubins paths [2] [14]. Among these are vector field path following methods. The basic idea is that the fixed-wing vehicle is commanded to fly at the altitude of the Dubins path while receiving course commands from a vector field. The vector field approach is simply a function that, for every point in $\mathbb{R}^2$, a velocity vector is produced from which we can solve for the course angle command. A gain is used to tune the aggressiveness with which the vector field integral lines approach the desired path. To use such a scheme a path following method must be developed for both the minimum turn-radius arcs and the straight line segments. Using these two vector field methods in conjunction with what we call a path manager, it becomes possible to follow these time-optimal, two dimensional Dubins car paths. A path manager function, that is used to determine which leg of the path the vehicle is on and switches between vector field methods at an appropriate time, was adopted from the two dimensional approach in [2] and adapted for use with time-optimal Dubins airplane paths (see Appendix A). We now propose a scheme for fixed-wing aircraft navigation along the time-optimal, three dimensional paths of a Dubins airplane.

To produce the vector fields necessary for following the time-optimal Dubins airplane paths we need to define the requisite surface functions. As was previously stated, the Dubins airplane paths for the low-goal altitude case consist of constant radii, constant pitch circular helices, and inclined straight lines. Using the parametric equation of a constant radius, constant pitch circular helix we can derive two surface functions for the vector field construction. Using these equations

$$x(t) = \rho \cos(t) \tag{4.1}$$

$$y(t) = \rho \sin(t) \tag{4.2}$$

$$z(t) = \beta t \tag{4.3}$$

we produce the zero-level surface functions

$$\alpha_1(x,y,z) = \left(\frac{x}{\rho}\right)^2 + \left(\frac{y}{\rho}\right)^2 - 1 \tag{4.4}$$

$$\alpha_2(x,y,z) = \beta \tan^{-1}\left(\frac{y}{x}\right) - z, \tag{4.5}$$

which are depicted in Figure 4.2.

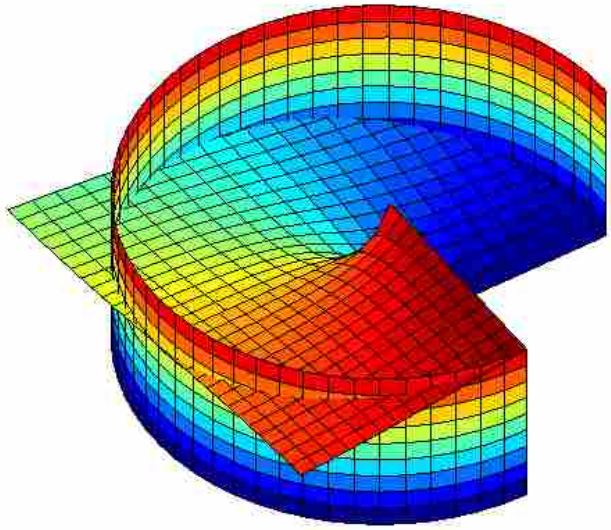


Figure 4.2: Plot of helical curve zero-level surface definitions

These two surface functions yield the following vector field equation:

$$\dot{\vec{r}}^{d} = -G \begin{pmatrix} \frac{2x}{\rho^2}\left(\left(\frac{x}{\rho}\right)^2+\left(\frac{y}{\rho}\right)^2-1\right)+\beta y(z-\beta\varphi)\left(x^2\left(\left(\frac{y}{x}\right)^2+1\right)\right)^{-1} \\ \frac{2y}{\rho^2}\left(\left(\frac{x}{\rho}\right)^2+\left(\frac{y}{\rho}\right)^2-1\right)-\beta(z-\beta\varphi)\left(x\left(\left(\frac{y}{x}\right)^2+1\right)\right)^{-1} \\ z-\beta\varphi \end{pmatrix} + H \begin{pmatrix} -\frac{2y}{\rho^2} \\ \frac{2x}{\rho^2} \\ \frac{2\beta}{\rho^2} \end{pmatrix} \tag{4.6}$$

where

$$\varphi = |\mathrm{mod}(\arctan 2(y,x)+2\pi\lambda, 2\pi\lambda)|, \quad \lambda \in \{-1,1\}. \tag{4.7}$$

We define λ as the parameter that specifies the direction of circulation about the circular helix, where values of $\lambda = 1$ and $\lambda = -1$ coincide when viewed from above with clockwise and counter-clockwise orbits respectively. The intent of setting up the vector field this way is to make the base height of the helix always occur at an angular position of zero radians and to prevent any discontinuities in the helix height for either orbit direction. By constructing the vector field in this way we found that development of a path management function was less complex in comparison to other surface definitions or wrapping points.

The inclined straight line vector field frame was defined with a 3-2 Euler rotation that aligns the x axis of the vector field frame with the inclined line of interest. A unit vector, q, that points in the direction of desired travel was used to describe the direction of the line from its start point in the inertial frame. This vector was used to solve for the angles necessary for the 3-2 Euler rotation. The rotation matrixes can be described as

$$R_i^f = \begin{pmatrix} \cos\theta_l & 0 & -\sin\theta_l \\ 0 & 1 & 0 \\ \sin\theta_l & 0 & \cos\theta_l \end{pmatrix} \begin{pmatrix} \cos\psi_l & \sin\psi_l & 0 \\ -\sin\psi_l & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.8}$$

where

$$\psi_1 = \arctan 2(q_2, q_1) \tag{4.9}$$

and

$$\theta_1 = \arcsin(q_3/|q|). \tag{4.10}$$

The surface functions that intersect along the inclined line in the vector field frame are

$$\alpha_1(x,y,z) = z \tag{4.11}$$

$$\alpha_2(x,y,z) = y. \tag{4.12}$$

The vector field equation is obtained by substituting (4.11) and (4.12) into (2.9) and evaluating to produce

$$\dot{r}^d = -G \begin{pmatrix} 0 \\ y \\ z \end{pmatrix} + H \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}. \tag{4.13}$$

4.3 Circular Helix Vector Field Nondimensionalization

The same gain normalization method presented for use with the elliptical orbit vector field, as developed in Section 3.3, can be applied to the circular helix case. There is no need to perform such a substitution for the inclined line vector field as the vector field frame for the line case was intentionally defined to avoid this problem.

The substitutions for the helical path case surface functions are

$$x_{rel} \triangleq \frac{x}{\rho}, \quad y_{rel} \triangleq \frac{y}{\rho}, \quad z_{rel} \triangleq \frac{z}{\beta}. \tag{4.14}$$

These substitutions produce the following surface functions:

$$\alpha_1(x_{rel}, y_{rel}, z_{rel}) = x_{rel}^2 + y_{rel}^2 - 1 \tag{4.15}$$

$$\alpha_2(x_{rel}, y_{rel}, z_{rel}) = \arctan 2\,(y_{rel}, x_{rel}) - z_{rel}. \tag{4.16}$$

When the surface functions are substituted into (2.9) they yield the vector field equations

$$\begin{aligned} \dot{\bar{r}}_{rel}^{d} = &-G \begin{pmatrix} 2x_{rel}(x_{rel}^2 + y_{rel}^2 - 1) + y_{rel}(z_{rel} - \varphi)(x_{rel}^2 + y_{rel}^2)^{-1} \\ 2y_{rel}(x_{rel}^2 + y_{rel}^2 - 1) - x_{rel}(z_{rel} - \varphi)(x_{rel}^2 + y_{rel}^2)^{-1} \\ z_{rel} - \varphi \end{pmatrix} \\ &+ H \begin{pmatrix} -2y_{rel} \\ 2x_{rel} \\ 2\lambda \end{pmatrix}. \end{aligned} \tag{4.17}$$

Due to the way we have defined φ and the effects of the nondimensionalization, the inclusion of λ in the circulation term is necessary to have the appropriate sign. The conversion back to the units of ρ and β is accomplished by

$$\dot{\bar{r}}^{d} = \begin{pmatrix} \rho & 0 & 0 \\ 0 & \rho & 0 \\ 0 & 0 & \beta \end{pmatrix} \dot{\bar{r}}_{rel}^{d}. \tag{4.18}$$

One of the limitations of this method is that in some instances β will equal zero, or be very near zero, and cause the vector field equation to go undefined. To prevent this from happening we define a second set of substitutions which must be used when β is small (we selected a cutoff point of $\beta = 0.1$).

The small β substitutions for the helical path case surface functions are

$$x_{rel} \triangleq \frac{x}{\rho}, \quad y_{rel} \triangleq \frac{y}{\rho}, \quad z_{rel} \triangleq z. \tag{4.19}$$

These substitutions produce the following surface functions:

$$\alpha_1(x_{rel}, y_{rel}, z_{rel}) = x_{rel}^2 + y_{rel}^2 - 1 \tag{4.20}$$

$$\alpha_2(x_{rel}, y_{rel}, z_{rel}) = \beta \arctan 2\,(y_{rel}, x_{rel}) - z_{rel}. \tag{4.21}$$

When the small β surface functions are substituted into (2.9) they yield the vector field equations

$$\begin{aligned} \dot{\vec{r}}_{rel}^{\,d} = &-G \begin{pmatrix} 2x_{rel}(x_{rel}^2 + y_{rel}^2 - 1) + \beta y_{rel}(z_{rel} - \beta\varphi)(x_{rel}^2 + y_{rel}^2)^{-1} \\ 2y_{rel}(x_{rel}^2 + y_{rel}^2 - 1) - \beta x_{rel}(z_{rel} - \beta\varphi)(x_{rel}^2 + y_{rel}^2)^{-1} \\ z_{rel} - \beta\varphi \end{pmatrix} \\ &+ H \begin{pmatrix} -2y_{rel} \\ 2x_{rel} \\ 2\beta \end{pmatrix}. \end{aligned} \tag{4.22}$$

For the small β case, the conversion back to the units of ρ is then accomplished by

$$\dot{\vec{r}}^{\,d} = \begin{pmatrix} \rho & 0 & 0 \\ 0 & \rho & 0 \\ 0 & 0 & 1 \end{pmatrix} \dot{\vec{r}}_{rel}^{\,d}. \tag{4.23}$$

4.4 Implementation

To implement these Dubins airplane path following methods on a simulated or a physical system, a supplementary function is needed. The purpose of this function is to manage a lower-level path following function containing these two vector field path following methods. This path management function needs to accomplish two primary tasks: determine which leg of the time-

Table 4.1: Simulation gains for the line following and helix following vector field path following methods.

Line Following	
k_{XY} =	0.1
k_Z =	2.0
Helix Following	
k_{XY} =	1.5
k_Z =	2.5

optimal Dubins airplane path the vehicle is on at an given time and send the appropriate parameters that describe the paths of interest to the path following function.

An effective path management scheme for navigation along two dimensional Dubins car paths has been presented by Beard and McLain [2]. We base our work on this path management method and extend it for use with three dimensional Dubins airplane paths. Another lower-level function is needed that can calculate the lengths of each of the four Dubins airplane path cases and produce parameters that mathematically communicate the path that should be followed between two points to the path following function. This path management scheme is laid out and described in detail in Appendix A in addition to a function for calculating Dubins airplane path parameters. We use this scheme for our Dubins airplane path following results, for the results of the interception path planning scheme and for our hardware implementation of these methods.

4.5 Simulation Results

The same simulation environment was used for the Dubins airplane path tracking results as was used for the elliptical orbit tracking methods (see Section 3.4). The MAV was commanded to follow two Dubins airplane paths between two consecutive waypoints. As was done with elliptical orbit simulation, a constant wind was commanded with the addition of the Dryden wind gust model. The simulation employed the vector field path following equations (4.17), (4.22), and (4.13) as well as the path management scheme presented in Appendix A.

We found it intuitive to formulate the gains such that a larger value always increased the strength of the vector field convergence. We also defined the gains such that one gain controlled

the strength of convergence in the XY plane of the vector field frame and the other in the direction of the Z axis in the same frame. For the simulation, these gains we defined as

$$H \triangleq \frac{\lambda}{k_{XY}} \tag{4.24}$$

and

$$G \triangleq \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & k_Z \end{pmatrix} \tag{4.25}$$

in (4.17) and (4.13). The gain values used in the simulation are shown in Table 4.1. In (4.24), λ is a variable that defines the direction of circulation along the path of interest. For the circular helix case, $\lambda = 1$ and $\lambda = -1$ coincide with a clockwise and counter-clockwise orbit direction respectively. For the straight line path following, $\lambda = 1$ for all cases because of the way the vector field was defined. A minimum turn radius of 40 meters was specified for the Dubins airplane path construction. This minimum turn radius estimate was produced by commanding the vehicle to fly with a constant roll angle of 30° in simulation.

The results of the simulation can be seen in Figure 4.3. Simulation tracking error was less than 2.0 meters throughout the entire simulation and only surpassed 1.0 meter when transitioning from straight paths to circular helix paths. The tracking error plot shows responses characteristic of a second-order system. We see that the vector field method is effective at tracking the desired paths within reasonable error tolerances. The plot displaying vehicle and wind heading data provides insight as to what caused the approximately 2.0 meter deviation near 40 seconds in simulation time. At that point in time the vehicle heading is nearly aligned with the wind vector which indicates that the vehicle is flying with a tail wind. The magnitude of the wind was approximately 30% of the commanded MAV airspeed (14 m/s). Generally speaking, tail winds increase the vehicle's ground speed and make it more difficult for the MAV to maneuver to follow a ground-referenced path.

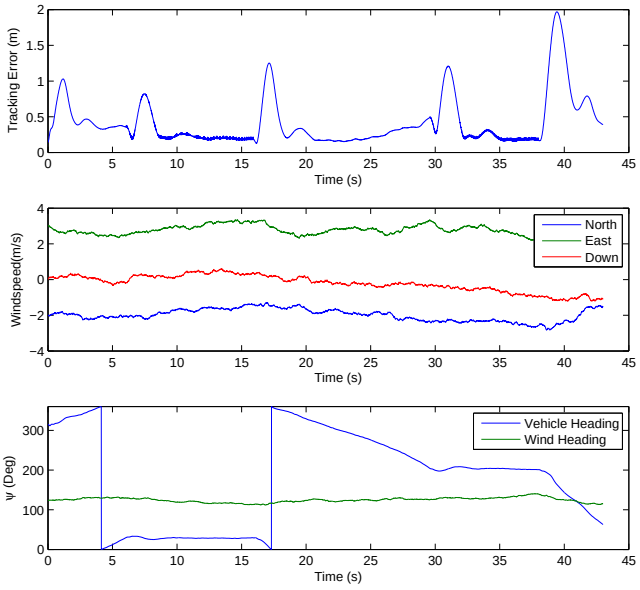
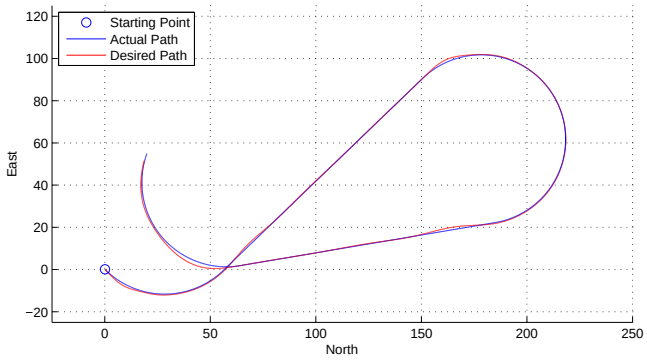


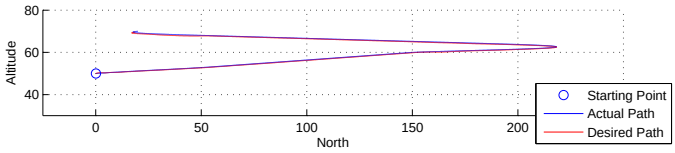
Figure 4.3: Path tracking error, corresponding wind magnitudes, and heading data of the wind and MAV from the Dubins airplane path following simulation.

Still, in the presence of this significant wind we see that this path following method is effective at rejecting the disturbance and converging to the desired path.

The results presented here are consistent with those we obtained by performing a variety of Dubins airplane path following simulations. We observed that in this test there is a steady-state error associated with the vector field path following method which is less than 0.5 meters. From these results, we conclude that the vector field path following method presented in this chapter provides a feasible method for vehicle navigation along Dubins airplane paths.



(a) Top view



(b) Side vide

Figure 4.4: Desired and actual path followed in the Dubins airplane path following simulation.

CHAPTER 5. ORBIT INSERTION AND DROGUE INTERCEPTION ALONG TIME-OPTIMAL, DUBINS AIRPLANE PATHS

5.1 Introduction

In this chapter we present a method for planning time-optimal, Dubins airplane interception paths from the current MAV position to a future drogue position. This method depends on the ability to plan and follow Dubins airplane paths, as presented in the previous chapter, as well as the drogue orbit estimation scheme that was previously developed [4]. This orbit estimate allows us to predict the future position of the drogue and is essential for the development of an estimated time of arrival (ETA) of the drogue to any future position. An estimated time to arrival function is also needed to predict how much time it will take the MAV to travel along a Dubins airplane path to a proposed interception point. Methods needed to produce the time to arrival estimates for the MAV and drogue are developed in this chapter. These include the development of an elliptical arc length approximation function and an explicit solution of ground speed from the wind triangle (see (2.10)). A robust root finding method is presented which can be used to plan time-optimal, Dubins airplane interception paths between the MAV and the drogue. We also present an interception, disturbance rejection airspeed controller and a post-orbit insertion airspeed controller which can be used to achieve a desired separation distance between the MAV and drogue. Finally, we conclude by simulating these methods and analyzing the results.

The intent of the methods presented in this chapter is to have the MAV intercept the drogue on its elliptical orbit in a near time-optimal fashion. This is accomplished by planning a time-optimal Dubins airplane path to a future drogue location that, based on the estimated time to arrival (ETA) of the two vehicles to that point, will result in a coordinated arrival to the proposed interception point. These interception paths effectively insert the MAV onto the drogue orbit and make it simple to transition to following the drogue along the elliptical orbit. An example of such an interception path is shown in Figure 5.1. If desired, the interception path can be planned such

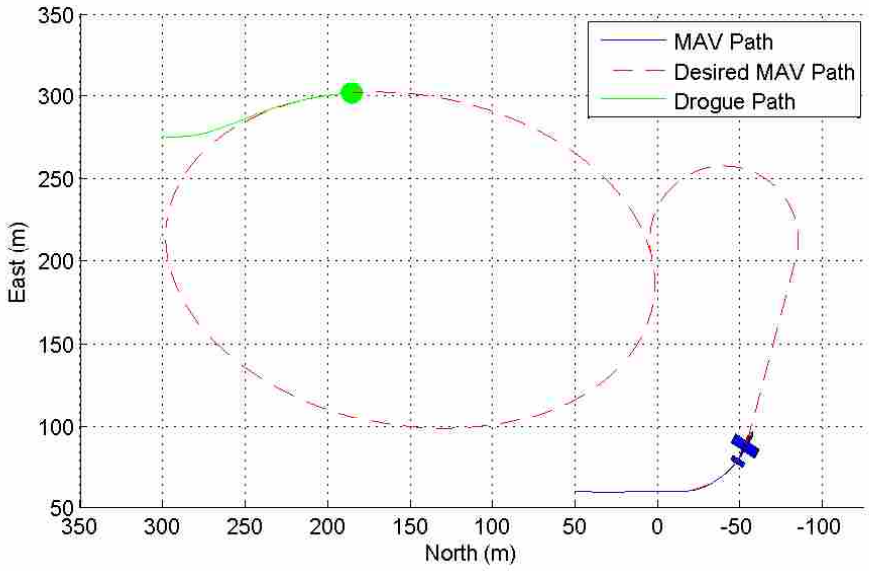


Figure 5.1: Top view plot of an example time-optimal Dubins airplane interception path. Developing the ability to plan such an interception path is the primary objective of this chapter.

that the vehicles will arrive with a specified separation distance. The purpose of these interception paths is not to perform the physical rendezvous of the two vehicles, rather it is to facilitate a visual rendezvous.

The elliptical orbit tracking method we developed and demonstrated in Chapter 3 provides us with the ability to follow the drogue along its orbit and gradually close the distance between the two vehicles until a visual rendezvous can occur. One of the limitations of this method is that the vector field path following method does not provide us with the capability of manipulating the MAV insertion point onto the drogue orbit. Rather, the insertion point is determined by the initial position and orientation of the MAV and the relative strength of the vector field contraction and circulation terms. Generally speaking, when the MAV is far from the ellipse it will head toward the closest point on the ellipse while gradually making forward progression. Once the MAV is near the ellipse, the desired velocity vector will be nearly tangent to the ellipse since it will be dominated by the circulation term. Consequently, even if we were to change the vector field gains while the MAV was approaching the ellipse, it would offer very limited control of the vehicle insertion point onto

the ellipse by increasing or decreasing the amount of forward progression that the MAV makes while converging toward the ellipse.

Our goal is to be able to insert the MAV onto the drogue orbit directly behind the drogue to decrease the time required for the aerial recovery operation. When using the elliptical orbit tracking method there is no guarantee that that the MAV will insert itself behind the drogue. Rather, the entry point of the MAV onto the drogue ellipse will be somewhat random. In some instances the MAV would insert itself on the side of the ellipse opposite from the drogue. In this situation it could take a significant amount of time to close the distance between them sufficiently, especially if their airspeeds were close in magnitude and the ellipse were relatively large. In fact, for an ellipse with major and minor radii of 150 m and 125 m, with a drogue ground speed of 12 m/s, and MAV ground speed of 18 m/s it would take over 72 seconds for the MAV to intercept the drogue. This would be in addition to the time that the MAV needed to converge to the drogue ellipse. A method that would decrease the time needed for an aerial recovery operation, compared to the less ideal scenario, would be much more desirable.

We propose a MAV orbit insertion scheme for intercepting the drogue on its orbit that decreases the overall time needed to initiate the visual rendezvous algorithms. To do this we make use of the time-optimal Dubins airplane path following scheme presented in Chapter 4. We present a drogue interception path planning algorithm that can be used to plan one of these time-optimal Dubins airplane paths. The product of this algorithm is a time-optimal Dubins airplane path that, based on the current airspeeds and positions of the two vehicles and the current wind vector estimate, is estimated to coordinate the arrival of the MAV and the drogue by a specified difference in arrival time upon completion of said path. This algorithm includes a number of components that were developed for the interception path planner and are presented here as well.

To plan such an interception path we need estimates of the drogue and MAV time-to-arrival to any point on the drogue orbit. These time to arrival functions for the MAV and drogue take the wind vector estimate into consideration and offer improved accuracy over previously developed estimates (presented in our previous work [15]) which neglected to account for the effects of wind. To produce these time-to-arrival estimates in the presence of wind, we present an explicit solution for calculating the ground speed of the two vehicles based on knowledge of their individual air-speeds, course angle, flight path angle and the current wind vector estimate. This equation was

developed from the standard wind triangle relationship presented in [2]. We also present an elliptical arc length, polynomial approximation that is used to accurately approximate the arc length of the drogue orbit for a given region on the ellipse. This elliptical arc length approximation prevents the need to use the elliptical integral equations and is sufficiently accurate for our purposes.

An interception path planner is presented that makes use of the time to arrival estimation techniques and can be used to plan an interception path for the MAV that inserts it onto the drogue orbit, behind the drogue, and within a specified time-to-arrival difference tolerance. This interception path planner poses the problem as four, one dimensional, nearly unconstrained optimization problems. The false position root finding method is used to solve for interception paths. Lastly, a function is presented that can be used to reject external disturbances and compensate for the minor errors induced in the estimation functions by coordinating the MAV and drogue arrival by varying the MAV airspeed.

5.2 Elliptical Arc Length Approximation

We begin by presenting an elliptical arc length approximation method that is numerically inexpensive to compute and sufficiently accurate for our purposes. This approximation is preliminary to producing a time-to-arrival estimate of the drogue from an initial location on the ellipse to any arbitrary point along the estimated ellipse orbit. The drogue time to arrival estimation method, which is presented in the following section, requires a scheme for calculating the arc length between two points on an ellipse. For paths with arc length functions that have analytical solutions this is not an issue, such as computing the angular position of a vehicle on a circular orbit given an arc length. One feature of the interception path planning scheme presented in this section is that it can be used to plan interception paths to any predefined vehicle path for which an accurate time-to-arrival estimate can be calculated to any future point on that path.

To produce such a time-to-arrival estimate requires a method for calculating arc lengths between any points on the path of interest. This may not be a problem in some cases which have nice, analytically produced arc length equations, but this is not possible for all types of paths. When planning an interception path, the path arc length function is used potentially hundreds or thousands of times depending on the optimization algorithm parameters. So it would be undesirable to use relatively expensive numerical methods for solving the arc length integral equations.

Various numerical integration methods are available for solving the arc length integral equations that have no analytical solutions, but they are significantly more computationally expensive than the method we present here.

To mitigate the computational expense of calculating elliptical arc lengths we present a computationally inexpensive method that can be used for any flight path that can be represented by a parametric function and whose arc length can be approximated sufficiently well with a least squares polynomial approximation. In this thesis the method shown for reducing the real-time computational expense of future vehicle location prediction was implemented for interception of a vehicle along an elliptical orbit.

The arc length equation for a parametrically defined ellipse is

$$s = \int_{\sigma_0}^{\sigma_f} \sqrt{(-a\sin\sigma)^2 + (b\cos\sigma)^2}\, d\sigma, \quad 0 \leq \sigma < 2\pi, \tag{5.1}$$

where a and b are the major and minor radii of the ellipse, respectively. There is no convenient analytical solution to this equation. To find an interception path on such an orbit we use a numerical approximation of the arc length function. This is needed for predicting the future location of the target vehicle after receiving an estimated distance to interception. To decrease the computational expense required to find an optimum interception path on an elliptical orbit, the elliptical arc length function was approximated using an 8th order polynomial least squares approximation of the arc length derivative function

$$\frac{ds}{d\sigma} = \sqrt{(-a\sin\sigma)^2 + (b\cos\sigma)^2}, \quad 0 \leq \sigma < 2\pi. \tag{5.2}$$

One of the major advantages to this method is that the majority of the computations required to solve for the polynomial coefficients can be precomputed. Due to the ill-conditioned nature of the general basis set for a polynomial least squares curve approximation, QR decomposition in conjunction with back substitution was used to solve the least squares problem without inverting the grammian matrix. The QR decomposition can be precomputed and used to solve for the polynomial coefficients by matrix multiplication and back substitution.

We defined the basis matrix as

$$A = \begin{pmatrix} 1 & \sigma & \dots & \sigma^n \\ 1 & \sigma & \dots & \sigma^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \sigma & \dots & \sigma^n \end{pmatrix} \quad ,$$

where A is an m x $(n+1)$ matrix, n is the polynomial order, and m is the number of sample points used. The least squares solution is found by solving the equation

$$A^T A\mathbf{C} = A^T \mathbf{y}, \tag{5.3}$$

where

$$\mathbf{y} = \sqrt{(-a\sin\sigma[k])^2 + (b\cos\sigma[k])^2}, \quad \sigma[1] = 0, \sigma[2] = \delta, \sigma[3] = 2\delta \cdots, \sigma[m] = \pi, \tag{5.4}$$

and $\mathbf{C}$ is a vector containing the $n+1$ coefficients of the polynomial approximation. To avoid inverting the poorly conditioned grammian matrix we used QR factorization. Such a factorization of $A^T A$ yields $QR = A^T A$, where Q is an orthogonal matrix and R is an upper triangular matrix [16]. The equation can then be rewritten as

$$R\mathbf{C} = Q^T A^T \mathbf{y}. \tag{5.5}$$

If the polynomial approximation order, n, and the number of samples, m, are predetermined for a problem of interest then the QR factorization can be precomputed as can $Q^T A^T$. Once the ellipse parameters a and b are known, the entire right-hand side of (5.5) can be computed. The computation then required to solve for the polynomial approximation coefficients, $\mathbf{C}$, is an $n+1$ dimension back-substitution. We describe this approximation as

$$\widehat{\frac{ds}{d\sigma}} = c_1 + c_2\sigma + c_3\sigma^2 + \cdots + c_n\sigma^{n-1} + c_{n+1}\sigma^n. \tag{5.6}$$

By integrating this polynomial approximation of (5.2) we have a polynomial approximation of the elliptical arc length function which is found to be

$$\hat{s} = c_1\sigma + \frac{c_2}{2}\sigma^2 + \frac{c_3}{3}\sigma^3 + \cdots + \frac{c_n}{n}\sigma^n + \frac{c_{n+1}}{n+1}\sigma^{n+1}. \tag{5.7}$$

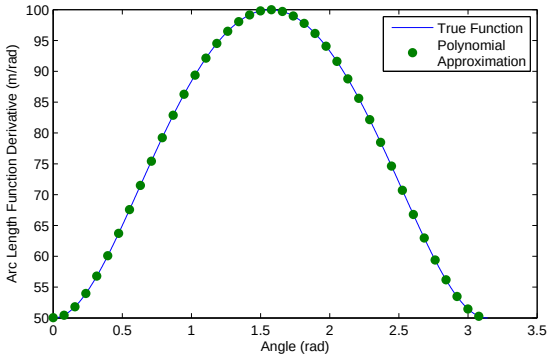


Figure 5.2: Polynomial approximation of the elliptical arc length function derivative ($a = 100$, $b = 50$) displaying discrete points of approximation along the true curve.

A plot of $\widehat{\frac{ds}{d\sigma}}$ and $\frac{ds}{d\sigma}$ is found in Figure 5.2. All polynomial approximations were found using $n = 8$ and $m = 200$. Approximation error was measured by calculating the error at each sample point and then integrating this error using the trapezoidal method. Elliptical arc length derivative functions of ellipses with various eccentricities were approximated to demonstrate the accuracy of this method (see Table 5.1).

In this thesis, the purpose of the arc length function is to provide elliptical arc length estimates which are used to estimate the time-of-arrival of the target vehicle to a interception point candidate. Once the polynomial coefficients have been found, the only computations necessary to calculate the arc length between two points are calculating the difference between two evaluations of (5.7).

Table 5.1: Elliptical arc length derivative function approximation error.

a	b	$\frac{a}{b}$	$\lvert s-\hat{s}\rvert$	$\hat{s}(\pi)$	$\frac{\lvert s-\hat{s}\rvert}{\hat{s}(\pi)}$
100	25	4	5.65e-2	214.5	2.63e-4
100	50	2	3.48e-2	242.2	1.43e-4
100	75	1.33	9.82e-4	276.3	3.48e-6
100	100	1	3.75e-9	314.2	1.19e-11

In some situations it may be necessary to estimate where the vehicle will be along the path after traveling a specified distance. This requires the use of a root finding method on the path arc length function. Due to the fact that the polynomial approximation of the arc length has a known derivative, and that for an ellipse there is only one root between 0 and π, it is well suited for use with the Newton-Raphson root finding method. In all tested cases the Newton-Raphson method was able to efficiently find the desired root. Simulation results showed that when $\pi/2$ was used as the initial guess of root location, root convergence to within 1.00e-3 accuracy occurred in four iterations or less.

5.3 Explicit Ground Speed Solution

One of the challenges associated with controlling air vehicles is the sensor measurements that are available for state estimation. Standard sensor readings include a pitot tube based airspeed measurement and GPS referenced course and flight path angle estimates.

The aerodynamic and propulsion forces that an air vehicle experiences are functions of the relative motion of the vehicle to the body of air that it is traveling through. One of the effects of this direct relationship is that the general practice in the aviation community is to command an air referenced vehicle speed, appropriately referred to as airspeed.

Throughout this paper we have discussed the use of vector field path following methods which seek to follow ground reference ellipses, inclined lines, and circular helices. We have also defined the vehicle course angle, χ, and flight path angle, γ, in (2.10). For any point along a ground reference path, a course angle and flight path angle corresponding to the direction of travel along that path can be computed. Airspeed is easily controlled using feedback from the airspeed sensor.

We have also presented a method for controlling flight path angle and assume that a course angle controller is readily available.

When the MAV is navigating along a ground reference path the vector field path following method provides course and flight path angle commands that result in convergence toward and progression along the desired path. While navigating along these paths it is desirable to command a constant airspeed. To estimate the time-to-arrival of a fixed-wing aircraft we need to determine how fast the vehicle is traveling along the path. As has already been mentioned, vehicle airspeed and ground speed are only equivalent in the absence of wind. If wind estimates are available, the wind triangle presents a relationship between airspeed and ground speed, although it does not provide us with an explicit relationship between the two. One method for estimating the wind vector is presented in [2]; we assume that such an estimate is available.

The wind triangle relationship is an implicit expression that relates the vehicle ground speed, airspeed, heading angle, ψ, air referenced flight path angle, γ_a, course angle, χ, flight path angle, γ, and the wind vector. It is desirable to have an explicit solution for estimating ground speed as a function of these variables. The inputs which are available to be used in such a function are the desired course angle and flight path angle, an estimate of the wind vector, and the airspeed measurement. With estimates of these variables, the vehicle ground speed can be calculated. Manipulation of the wind triangle equation provides us with such a relationship.

We begin by presenting the standard wind triangle equation:

$$V_g \cos\chi \cos\gamma = V_a \cos\psi \cos\gamma_a + w_n \tag{5.8}$$

$$V_g \sin\chi \cos\gamma = V_a \sin\psi \cos\gamma_a + w_e \tag{5.9}$$

$$-V_g \sin\gamma = -V_a \sin\gamma_a + w_d. \tag{5.10}$$

Solving for γ_a in (5.10) yields

$$\gamma_a = \sin^{-1}\left(\frac{V_g \sin\gamma + w_d}{V_a}\right). \tag{5.11}$$

Substituting (5.11) into (5.9) and solving for ψ produces

$$\psi = \tan^{-1}\left(\frac{V_g \sin\chi\cos\gamma - w_e}{V_g\cos\chi\cos\gamma - w_n}\right). \tag{5.12}$$

Solving (5.8) for γ_a and setting it equal to (5.11) forms

$$\sin^{-1}\left(-\frac{-V_g\sin\gamma - w_d}{V_a}\right) = \cos^{-1}\left(\frac{V_g\cos\chi\cos\gamma}{V_a\cos\psi}\right), \tag{5.13}$$

which is equivalent to

$$\left(1-\left(\frac{-V_g\sin\gamma - w_d}{V_a}\right)^2\right)^{\frac{1}{2}} = \frac{V_g\cos\chi\cos\gamma - w_n}{V_a\cos\psi}. \tag{5.14}$$

We now substitute (5.12) into (5.13). With some manipulation we can produce the following:

$$\begin{aligned} 0 =& V_g^2 - 2V_g(\cos\chi\cos\gamma w_n + \sin\chi\cos\gamma w_e - \sin\gamma w_d) \\ &+ (w_n^2 + w_e^2 + w_d^2 - V_a^2). \end{aligned} \tag{5.15}$$

Solving for the realistic root, which is associated with the larger, generally positive value, provides the following equation for ground speed:

$$\begin{aligned} V_g =& w_n\cos\chi\cos\gamma + w_e\sin\chi\cos\gamma - w_d\sin\gamma \\ &+ \sqrt{(w_n\cos\chi\cos\gamma + w_e\sin\chi\cos\gamma - w_d\sin\gamma)^2 - (w_n^2 + w_e^2 + w_d^2 - V_a^2)}. \end{aligned} \tag{5.16}$$

The other root can be described by the situation where the vehicle is traveling in the opposite direction of which it is oriented. With the result of (5.16), we can now evaluate (5.12) and (5.11).

5.4 Time to Arrival Estimation

The interception path planning scheme that we have developed searches for points on the ellipse at which the estimated arrival time of the two vehicles is within a specified tolerance. Thus, this scheme requires the use of time-to-arrival functions for the MAV along the Dubins airplane paths as well as a similar function for estimating the drogue progression along its estimated elliptical orbit. For both of these methods we assume a constant airspeed and wind field as well as ideal navigation along the desired paths (i.e., a Dubins airplane path or an ellipse path). The estimated time-to-arrival functions we present are numerical approximations. These approximations make use of the explicit ground speed solution presented in (5.16), which rely on the current wind estimate. We assume that these wind estimates are valid in the vicinity of the mission theater.

The ETA algorithms we present are functions of the specified airspeed, wind vector, and Dubins path parameters. Thus, these methods require use of the function that parameterizes the Dubins airplane paths which can be found found in Appendix A. The approach taken for producing this estimate is to solve for the vehicle ground speed at various points along the path. The arc length between each of these ground speed estimates is also computed. Dividing the arc length from a current point on the path to the subsequent point by the current ground speed estimate provides a reasonable approximation of the time it will take to travel between these two points along the path of interest. This process is repeated for a specified number of steps and the individual travel time estimates are summed to provide an ETA between the start and end points of the complete path.

For the sake of clarity and simplicity we present these two time-to-arrival estimation methods in algorithm format.

5.4.1 MAV ETA

The MAV estimation algorithm fulfills two roles. It is used in the computation of the time-optimal, Dubins airplane interception paths. This provides us an ETA for a proposed interception

path. In this chapter we also present an airspeed controller that is used while the MAV is in the process of following the interception path. It uses the MAV ETA function to vary the vehicle airspeed while traveling along the path to maintain the appropriate arrival time at the interception point. One of the inputs to this function is a flag that is sent from the path manager, which is presented in Appendix A. This flag indicates which leg of the Dubins airplane path the vehicle is currently following. The MAV ETA function also requires knowledge of the Dubins path parameters. A function which produces these path parameters is also presented in Appendix A. Algorithm 1 is used to calculate the ETA for the circular helix legs. This is accomplished by partitioning the circular helix into a specified number of segments. The MAV ground speed is calculated at the start of each of these segments with (5.16). Based on the ground speed calculation the travel time along each segment is calculated and these are summed to produce the total ETA for the circular helix.

The Calculate MAV Circular Leg ETA is used in Algorithm 2 which provides the total ETA for the uncompleted legs of the total path. One of the inputs to this function specifies on which leg of the Dubins airplane path the MAV is traveling. The Algorithm uses the current MAV location and solves for the nearest point on the Dubins airplane path that the MAV is currently following. The travel time along the circular helix portions of the path are calculated using Algorithm 1. The ground speed along the straight portion of the path is calculated and used to provide the travel time along that leg. These travel time estimates are summed to provide the total ETA for the path.

5.4.2 Drogue ETA

The Calculate Drogue ETA algorithm is similar to the Calculate MAV Circular Leg ETA algorithm. It runs a specified number of iterations along the segment between the current drogue position and the proposed interception point on the ellipse. For each iteration it estimates the ground speed based upon estimates of the drogue course angle, flight path angle, and airspeed and the wind vector estimate. Using the elliptical arc length approximation function from (5.7), an arc length approximation is made for each iteration. This arc length approximation is the length of the individual segment being analyzed. Dividing the arc length segment approximation by the segment ground speed provides an ETA for the specific segment. The segment estimated times-to-arrival are summed to provide the total ETA from the current drogue position to the interception point. Algorithm 3 presents the Calculate Drogue ETA function in detail.

Algorithm 1 Calculate MAV Circular Leg ETA:
$t_{LEG} = estimateMavEta(\theta_s, \theta_e, \rho, \lambda, \gamma, V_a, w_n, w_e, w_d, n)$

Input: Start and end angular position θ_s, θ_e, Orbit radius ρ, Orbit direction λ, Orbit flight path angle γ, Airspeed V_a, Wind (w_n,w_e,w_d), Number of iterations n

$\theta_s \leftarrow \text{mod}(\theta_s + 2\pi\lambda, 2\pi\lambda)$
$\theta_e \leftarrow \text{mod}(\theta_e + 2\pi\lambda, 2\pi\lambda)$
if $\lambda\theta_e > \theta_0$ **then**
    $\theta_{total} \leftarrow \lambda(\theta_e - theta_s)$
else
    $\theta_{total} \leftarrow 2\pi + \lambda(\theta_e - \theta_s)$
end if
$\theta_\Delta \leftarrow \frac{\theta_{total}}{n}$
$\theta_i \leftarrow \theta_0$
for $i = 1 \rightarrow n$ **do**
    $\theta_{i+1} \leftarrow \text{mod}(\theta_i + i\lambda\theta_\Delta + 2\pi\lambda, 2\pi\lambda)$
    $\chi \leftarrow \theta_i + \lambda\frac{\pi}{2}$
    V_g: Compute ground speed using (5.16)
    $t_{LEG} \leftarrow t_{LEG} + \frac{\theta_\Delta \rho}{V_g \cos\gamma}$
    $\theta_i \leftarrow \theta_{i+1}$
end for
return t_{LEG}

5.5 Interception Path Planner

Further investigation of a previously proposed drogue interception path planner, presented in our earlier work [15], showed that improvements could be made. We have developed an interception path planner that is more efficient and robust to the discontinuities characteristic of the system. The previous scheme that we proposed was similarly posed, but it was particularly susceptible to failure when it encountered discontinuities characteristic of the system.

When planning an interception path, we assume knowledge of the current drogue and MAV positions and directions of travel, the drogue ellipse estimate, and the current wind vector estimate. Simply stated, we want to find a point on the estimated drogue orbit to which the MAV can follow a Dubins airplane, time-optimal path such that it arrives at the same moment as the drogue. It is also a simple extension to plan an interception path in which the MAV is planned to arrive to the intercept point at a specified distance behind the drogue.

The algorithm previously employed had significant limitations and was computationally more expensive. To initialize the algorithm, we assumed that the interception point would be the

Algorithm 2 Calculate MAV ETA:
$t_{MAV} = estimateMavEta(, V_a$,state,$w_n, w_e, w_d, \mathbf{p}_1, \mathbf{p}_2, \mathbf{c}_1, \mathbf{c}_2, \rho, \mathbf{w}_1, \mathbf{w}^2, \gamma, n)$

Input: MAV position $\mathbf{p} = (p_n, p_e, p_d)^T$), MAV airspeed V_a, Current path leg (*state*), Wind w_n, w_e, w_d, Start and end points $\mathbf{p}_1 = (p_{1n}, p_{1e}, p_{1d})^T$, $\mathbf{p}_2 = (p_{2n}, p_{2e}, p_{2d})^T$, Start orbit center and direction ($\mathbf{c}_1 = (c_{1n}, c_{1e}, c_{1d})^T$,$\lambda_1$, End orbit center and direction, $\mathbf{c}_2(c_{2n}, c_{2e}, c_{2d})$, λ_2, Orbit radius ρ, Line start and end points $\mathbf{w}_1 = (w_{1n}, w_{1e}, w_{1d})$, $\mathbf{w}_2 = (w_{2n}, w_{2e}), w_{2d}$, Path flight path angle γ, Circular leg number of iterations n

$t_{MAV} \leftarrow 0$
$\theta_{1s} \leftarrow \arctan 2(p_{1e} - c_{1e}, p_{1n} - c_{1n})$
$\theta_{1e} \leftarrow \arctan 2(w_{1e} - c_{1e}, w_{1n} - c_{1n})$
$\theta_{2s} \leftarrow \arctan 2(w_{2e} - c_{2e}, w_{2n} - c_{2n})$
$\theta_{2e} \leftarrow \arctan 2(p_{2e} - c_{2e}, p_{2n} - c_{2n})$
$lineEnd \leftarrow \mathbf{w}_2$
if $state = 1$ OR $state = 2$ **then**
    $\theta_{1s} \leftarrow \arctan 2(p_e - c_{1e}, p_n - c_{1n})$
    $lineStart \leftarrow \mathbf{w}_1$
    $startIndex \leftarrow 1$
else if $state = 3$ **then**
    $X1 \leftarrow \mathbf{w}_1$
    $X2 \leftarrow \mathbf{w}_2$
    $X0 \leftarrow \mathbf{p}$
    $\sigma \leftarrow -\frac{(X1-X0)^T(X2-X1)}{||X2-X1||^2}$
    $lineStart \leftarrow X1 + (X2 - X1)\sigma$
    $startIndex \leftarrow 2$
else
    $\theta_{2s} \leftarrow \arctan 2(p_e - c_{2e}, p_n - c_{2n})$
    $lineStart \leftarrow \mathbf{w}_2$
    $startIndex \leftarrow 3$
end if
if $startIndex = 1$ **then**
    $t_{MAV} \leftarrow t_{MAV} + circularLegETA(\theta_{1s}, \theta_{1e}, \rho, \lambda_1, \gamma, V_a, w_n, w_e, w_d)$
end if
if $startIndex \leq 2$ **then**
    $\chi \leftarrow \arctan 2(w_{2e} - w_{1e}, w_{2n} - w_{1n})$
    V_g: Compute ground speed using (5.16)
    $t_{MAV} \leftarrow t_{MAV} + \frac{||lineEnd - lineStart||}{V_g}$
end if
if $startIndex \leq 3$ **then**
    $t_{MAV} \leftarrow t_{MAV} + circularLegETA(\theta_{2s}, \theta_{2e}, \rho, \lambda_2, \gamma, V_a, w_n, w_e, w_d)$
end if
return t_{MAV}

Algorithm 3 Calculate Drogue ETA:
$t_{drogue} = estimateMavEta(\mathbf{p}, V_a, \bar{p}_f, a, b, \mathbf{c}, \mathfrak{R}_i^e, \mathbf{C}, s_{\frac{1}{2}}, \lambda, w_n, w_e, w_d, n)$

Input: Drogue position in inertial frame $\mathbf{p} = (p_n, p_e, p_d)^T$), Drogue airspeed V_a, Ellipse radii a, b, Ellipse center in inertial frame $\mathbf{c} = (c_n, c_e, c_d)^t$, Inertial to ellipse frame rotation matrix $\mathfrak{R}_i^e$, Arc length approximation coefficients $\mathbf{C}$, Ellipse half perimeter $s_{1/2}$, Drogue orbit direction λ, Wind w_n, w_e, w_d, Number of iterations n

$t_{drogue} \leftarrow 0$
$(x, y, z)^T \leftarrow \mathfrak{R}_i^e(\mathbf{p} - \mathbf{c})$
$\theta_s \leftarrow \text{mod}(\arctan 2(y, x) + 2\pi, 2\pi)$
$(x_f, y_f, z_f)^T \leftarrow \mathfrak{R}_i^e(\mathbf{p}_f - \mathbf{c})$
$\theta_f \leftarrow \text{mod}(\arctan 2(y_f, x_f) + 2\pi, 2\pi)$
if $\lambda\theta_f > \theta_s$ **then**
    $\theta_{tot} \leftarrow \lambda(\theta_f - \theta_s)$
else
    $\theta_{tot} \leftarrow 2\pi + \lambda(\theta_f - \theta_s)$
end if
$\theta_\Delta \leftarrow \frac{\theta_{tot}}{n}$
$\theta_i \leftarrow \theta_s$
for $i = 1 \rightarrow n$ **do**
    $\theta_{i+1} \leftarrow \text{mod}(\theta_i + i\lambda\theta_\Delta + 2\pi\lambda, 2\pi\lambda)$
    $(\dot{x}, \dot{y}, \dot{z})^T \leftarrow \mathfrak{R}_i^e(-\lambda a \sin\theta_i, \lambda b \cos\theta_i, 0)^T$
    $\chi \leftarrow \arctan 2(\dot{y}, \dot{x})$
    $\gamma \leftarrow \arcsin(\dot{z}/||(\dot{x}, \dot{y}, \dot{z})^T||)$
    V_g: Compute ground speed using (5.16)
    s_i: Compute ellipse step arc length using (5.7)
    $t_{drogue} \leftarrow t_{drogue} + \frac{s_i}{V_g}$
    $\theta_i \leftarrow \theta_{i+1}$
end for
return t_{drogue}

drogue's current location. A path was planned from the MAV to this point. Once this path was planned the MAV's estimated time to arrival (ETA) could be calculated. Using the MAV's ETA, we solved the elliptical arc length approximation function to estimate the future location of the drogue when the MAV arrived to the drogue's previous location.

There were a number of significant limitations to this method. First, assuming that the drogue's current location is close to an actual interception point was generally not accurate. Second, predicting where the drogue would end up after the MAV completed its path to the approximated interception point involved solving for the roots of the elliptical arc length approximations that we developed. Third, a fixed-step solver was adopted to increment the proposed location of

interception around the ellipse until an adequate interception point was found. This fixed-step method was the first attempt at solving this problem and is numerically inefficient compared to the method we have now adopted. Also the fixed-step solver would often fail due to the possible discontinuities between consecutive MAV paths to subsequent estimated points of interception. These are some of the problems with the initial solution we proposed to this problem, all of which have been mitigated by developing a similar but superior scheme.

The new interception path planning scheme makes use of the assumption that the interception point must lie along the current drogue elliptical orbit estimate. Because the domain containing the ellipse is limited enough to be rapidly searched in its entirety, we can effectively search the entire range where the solutions may lie. That is to say that, for a parametrically defined ellipse, the interception point must lie on the ellipse between $\sigma = [0, 2\pi)$. The key is to efficiently search for all possible interception points on the ellipse. This can be accomplished by splitting the ellipse into k regions of equal angular length θ, where $\theta = \frac{2\pi}{k}$.

We want to determine whether or not a potential interception point lies in each of these regions. We begin by calculating the drogue ETA from its current configuration, θ_{D0}, to each of the k boundary points of the search regions using Algorithm 3. One advantage of this search region method compared to our previous approach is that it is numerically less expensive than estimating where the drogue will be after a specified amount of time. Now that we know the drogue ETA to the boundary points of the k ellipse segments, we need to determine the difference in arrival time between the MAV and drogue to each of those points.

To intercept the drogue with the MAV in a near, time-optimal fashion, we use the time-optimal Dubins airplane paths to navigate the MAV from an initial position to the interception point. Using Algorithm 2 provides the ability to estimate the MAV ETA from any initial position to any point on the ellipse for each of the four path cases.

The path planning algorithm is run essentially four times. Once for each of the path cases. Each time the algorithm is run the best interception path that was found for the specific case is returned. These are compared for the four path cases and the case with the shortest estimated time to interception (ETI) is selected.

For each run of the path planning algorithm the MAV ETA to each of the ellipse segment boundary points is calculated for the specific path case. Since the drogue ETA to the same points

has already been calculated, we can now estimate the difference in arrival time at each of these points.

Together the MAV and drogue ETA function provide us with the estimated difference in time-to-arrival (EDTA) to any point on the ellipse. This is a powerful tool because interception points are the roots of the EDTA function. We have constructed the drogue interception problem such that it can be posed as a one dimensional root finding problem that is run four separate times.

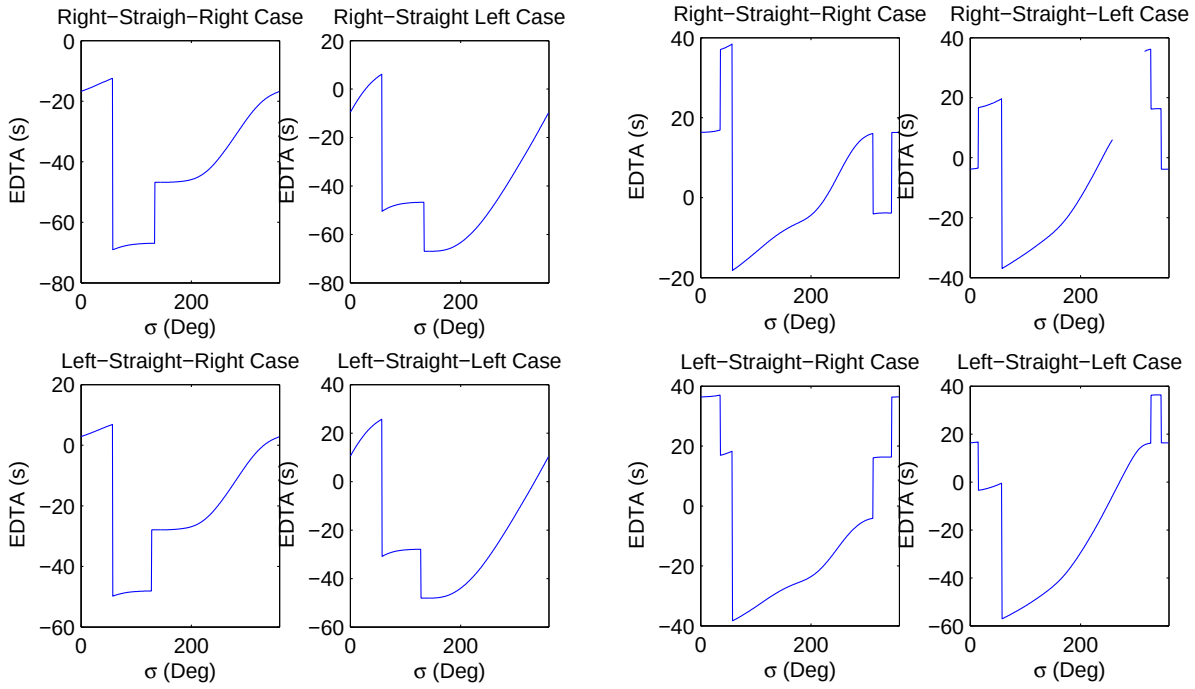
5.5.1 False Position Root Finding

When selecting a root finding algorithm, it is important to understand some of the properties of the function whose zeros we wish to find. Two plots of the EDTA function data for the four path cases can be found in Figure 5.3. These data provide valuable insight into the EDTA function characteristics and are representative of the function behavior we saw in all our simulations. It is apparent in reviewing this figure that the each of the cases always contain at least one discontinuity. This is to be expected because, at some point on the ellipse, the angular length of the final helix of the MAV path will transition from being 2π radians to a angular length of 0 radians. The other discontinuities are due to the combined effects of the MAV and drogue ETA functions.

It is important to note that not all cases have an interception path, such as with the Right-Straight-Right Case of Figure 5.3 (a). We also notice that when the initial separation distance is small there can be regions where there is is no feasible path for a given case. This is an attribute of the Right-Straight-Left and Left-Straight-Right cases when the separation distance between the path start and end point is less than three minimum turn radii apart. Other than the above mentioned characteristics, the EDTA function appears to be well behaved. Also, over larger regions of the function it progresses monotonically and nearly linearly. In all practical cases tested at least one root was present in one of the four cases.

From this we conclude that if the ellipse segment size is sufficiently small there is a low probability that it will contain a root and a discontinuity. It is important to avoid this situation because many root finding methods cannot guarantee convergence if there is a discontinuity near the root of interest.

We have selected the false position method for searching for interception paths on the drogue ellipse. The false position method is a simple, root finding scheme that guarantees conver-



(a) Large initial separation

(b) Small initial separation

Figure 5.3: Two sets of empirical EDTA data. One where the initial separation distance between the MAV and the ellipse was relatively large when the path planner was run and the other when it was relatively small.

gence if certain conditions are met. Two boundary points with opposite signs must be provided to use the false position method. Convergence to a function root is conditional on the function being continuous between the two boundary points. The false position method use a linear approximation from the boundary points and corresponding function values to estimate the location of the root. The function value at that root estimate must then be evaluated. At this point, the current boundary point with the same sign as the root estimate is replaced by the root estimate. In this way, just like the bisection method, the algorithm keeps the root contained between the boundary points. These boundary points continue to enclose the root using the linear approximation until the user specifies that the algorithm should stop.

This function is well suited for root finding of the EDTA function because the function behavior is somewhat linear and it is unlikely that a region containing a root will also contain a

discontinuity. The probability of a region containing a root and a discontinuity is largely governed by the ellipse angular segment size that is selected.

With the EDTA function evaluated at all boundary points of the k ellipse segments, determining which segments contain a root, or possibly a discontinuity, is simple. It is accomplished by checking for a difference in sign of the EDTA function value between the two neighboring boundary points. The segments which have boundary points with opposite signs, which we specify as root candidate segments, will generally contain only a root or a discontinuity, as long as the segment size is sufficiently small.

In all of our tests, when the false position method is used on a root candidate segment that actually contains a root, the algorithm converged rapidly. For an ellipse with major and minor radii $a = 150$ and $b = 100$ with $k = 100$, the false position method converged within 0.01 s of the EDTA root in a single iteration. That means that the interception path is estimated to have a difference in arrival time of 0.01 s, which is more than adequately accurate for our purposes.

For the root candidate segments that actually contain a discontinuity, this discontinuity can be effectively recognized by selecting an appropriate value for the maximum allowable difference in EDTA values between the two boundary points. Thus, if the difference in EDTA values of the boundary points is larger than typical for a segment that actually contains a root, it can be assumed that this segment contains a discontinuity. Identifying the root candidate segments with discontinuities prevents wasting computation resources in searching for a root in a region that contains none.

The limitations of the proposed interception point, root finding method are relatively minimal. If there is a discontinuity and a root contained in an ellipse segment, there is no guarantee that this root will be found. This can be avoided by not using overly large segment sizes. It is also true that if more than one root is contained in a region it will only find and return a single root. In these situations, it is possible that a superior interception path exists that was not found, but the probability of this occurring is low enough for our comfort. Again, increasing k decreases the probability of either of these events occurring. The primary limitation is that it performs a one dimensional search. More complicated methods could be used for a multi-dimensional search that varies the airspeed and Dubins airplane path turn radius that takes vehicle constraints into consideration.

5.6 Interception Disturbance Rejection Airspeed Controller

When tracking the MAV interception paths, there are a variety of disturbances that can cause undesired changes in the difference in arrival time of the MAV and drogue. We propose a scheme that can be used to modify the airspeed command of the MAV to maintain the EDTA of the two vehicles. The way that Algorithm 2, the Calculate MAV ETA function, was written makes it possible to use this function while the MAV is following an interception path. The Calculate Drogue ETA function can also be used while the MAV is following an interception path to update the drogue ETA to the interception point. Thus, while both vehicles are approaching the interception point, the EDTA function is still valid.

The disturbance rejection strategy only differs from the interception path planning method in the selection of the parameter to be varied. This method maintains the desired interception point and updates the MAV airspeed command to maintain the desired EDTA. The false position root finding method was employed to find the appropriate airspeed command. To use the false position root finding method we must provide an upper and lower bound on airspeed for it to begin. This can be done by selecting the upper and lower bound of airspeed over which the vehicle of interest flies well. In our situation, the MAV we use for hardware testing is typically flown at 14 m/s, but it performs relatively well from 10-18 m/s. Thus, we select the boundary conditions of the false position method as 10 m/s and 18 m/s.

One of the nice attributes of using the airspeed as the design variable is that the corresponding EDTA function is monotonic and continuous through the entire range of feasible airspeeds for a fixed-wing aircraft. Thus, using this method is guaranteed to find the EDTA root as long as it is within the specified boundaries. If it is not contained within these boundaries, then interception of the drogue at the proposed interception point is no longer feasible along the current interception path. One desirable attribute of this method is that it provides an indication of whether interception along the current path is feasible. This can be used as an indicator that the interception path planner should be run again.

5.7 Post-Insertion Airspeed Controller

Once the MAV has completed following the Dubins airplane interception path, the MAV has been effectively inserted the estimated drogue orbit. In the ideal situation, the interception path has placed the MAV the desired distance behind the drogue such that that the drogue is in the field of view (FOV) of the front facing camera on the MAV. For the MAV and camera lens the desired following distance was 15 to 30 meters behind the drogue. The visual tracking method that was being used has to be initiated by the user selecting the desired target, in our case the drogue, in the video feed from the MAV that has been transmitted to the ground. Designating the desired target takes a 5 to 15 seconds and is made less difficult if the drogue is maintaining a relatively stable position in the camera FOV.

We used a simple controller to maintain the desired following distance between the two vehicles. This controller calculates the error of the actual distance between the two vehicles compared to the desired separation distance between them. Based upon The angular position of the two vehicles on the estimated drogue orbit and the orbit direction we can determine if the MAV is in front of or behind the the drogue. This indicates whether the MAV must speed up or slow down. A proportional gain is used to select the aggressiveness of a correctional airspeed term which is added to, a feed-forward airspeed term, the current drogue airspeed measurement. This controller can be written as

$$V_{aMAV}^{d} = V_{aDrogue} + k_p(||\mathbf{p}_{MAV} - \mathbf{p}_{Drogue}|| - d_{sep}), \tag{5.17}$$

where k_p is a proportional gain whose sign is determined by whether the MAV is in front of or behind the drogue and d_{sep} is the user specified, desired separation distance. Essentially we are using a proportional controller around the following error to vary the desired airspeed of the MAV.

5.8 Simulation Results

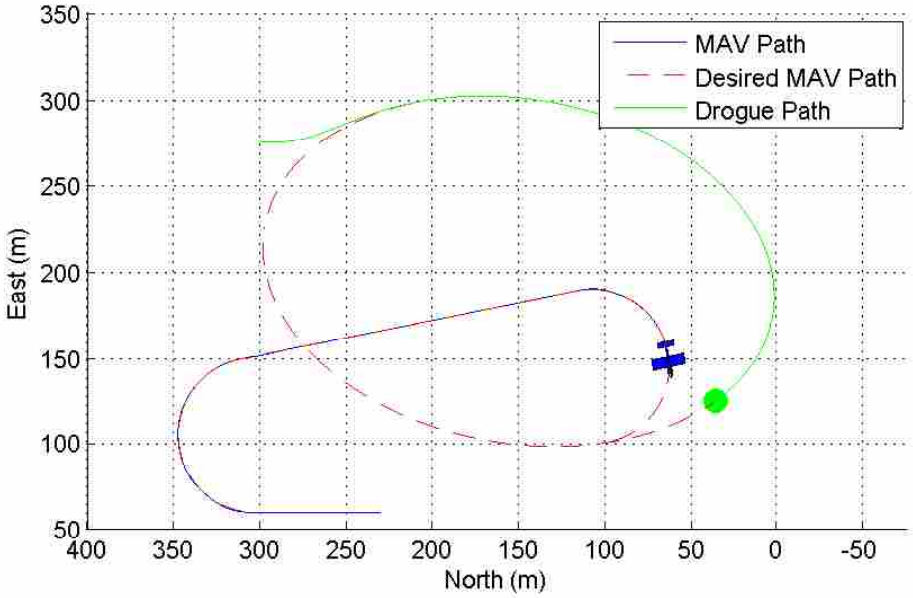
The methods presented in this chapter were tested in the same type of simulation used for the preceeding chapters but with the addition of a simulated drogue. The drogue dynamic model used was a first-order integrator with a constant airspeed. To have the drogue follow the desired

orbit, an elliptical orbit path following vector field was used as developed in Section 3.2. This vector field equation produced the desired velocity vector for the drogue. Based on the commanded drogue airspeed, current wind vector, and desired velocity vector, the explicit ground speed equation, (5.16), was used to solve for the drogue ground speed. This ground speed was used as the magnitude of the desired velocity vector which was then numerically integrated by the drogue dynamics to produce the motion of the simulated drogue. The Dubins airplane path following methods presented in the previous chapter were used to navigate the MAV along the desired interception path.

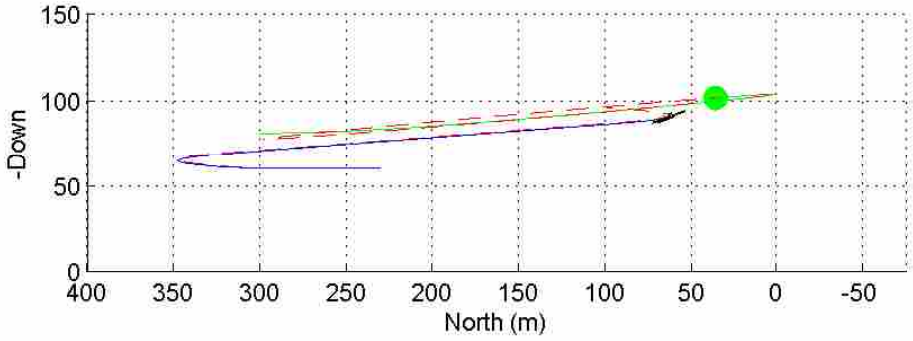
To demonstrate the effectiveness of the interception path following methods and the disturbance rejection airspeed controllers, we present the results from two simulations. In the first, the path planner was used to find an interception path and the MAV was commanded to fly with the constant airspeed that was used in the EDTA calculations. Due to the fact that there is some tracking error when following the Dubins airplane interception paths and due to the variation in the simulated wind and other errors, we expect this method to have error in the separation distance at the time of arrival.

The second simulation had the same initial conditions as the first and experienced the same simulated wind. In this simulation, the interception, disturbance rejection airspeed controller, from Section 5.6, was used to correct the commanded MAV airspeed to maintain the desired separation distance between the MAV and drogue upon arrival at the proposed interception point. After the MAV had completed navigation along the Dubins airplane interception path, it was commanded to follow the ellipse using the Gonçalves based elliptical orbit path following vector field method presented in Section 3.2. The post-insertion, airspeed controller was used to maintain the specified following distance between the two vehicles.

Results from these two simulations can be seen in Figures 5.4 and 5.5. In both of these simulations the desired separation between the MAV and drogue was specified to be zero meters. Investigation of the two separation distance plots shows that both of the interception path planning and following schemes achieve the desired results. The results from the simulation with the constant airspeed command show that the interception airspeed controllers are effective at compensating for disturbances and maintaining the desired separation distance at the point of interception. One important result is the measure of the MAV ground speed over the course of the interception



(a) Top View

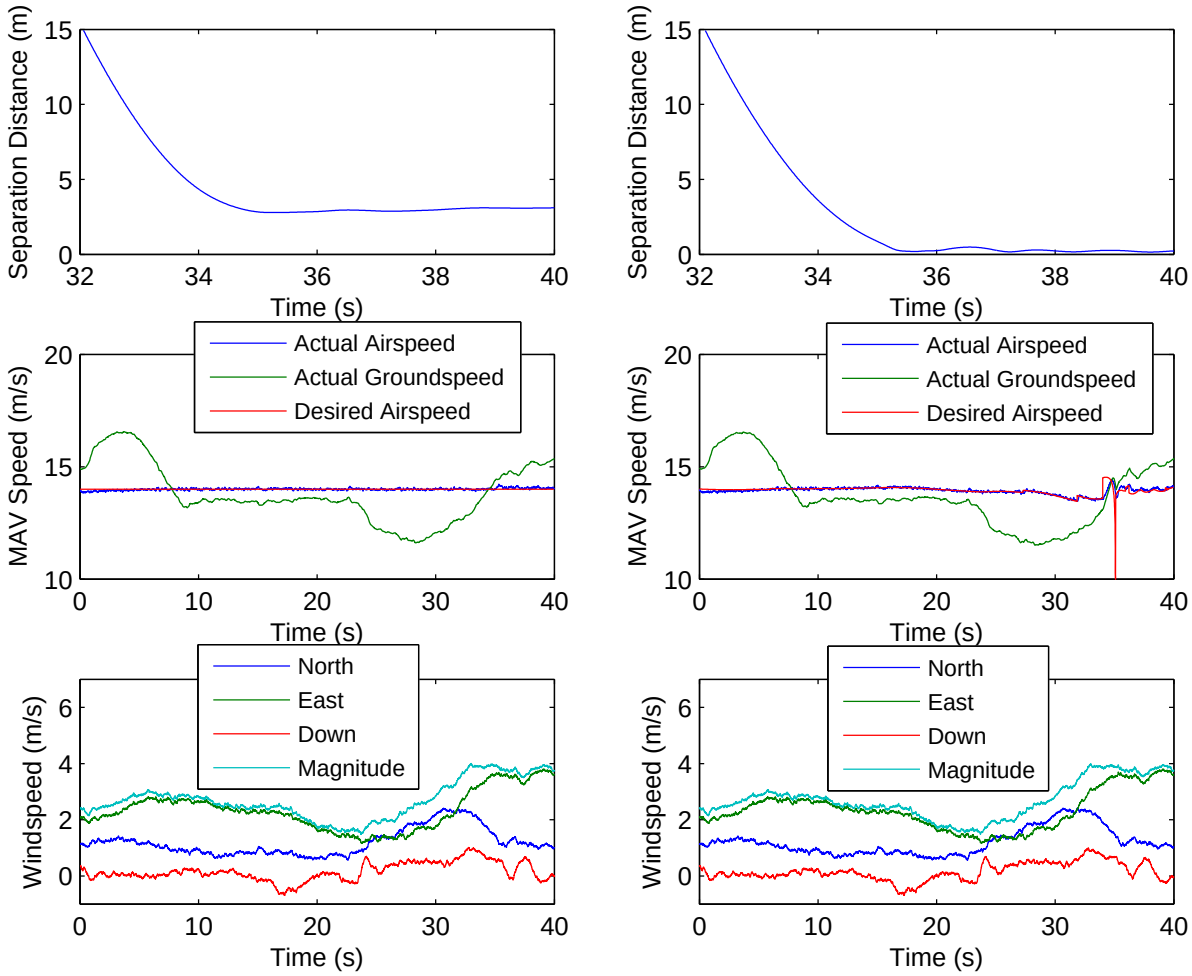


(b) Side View

Figure 5.4: Plots of the three dimensional simulated MAV path, drogue path, and the desired MAV path just prior to the drogue interception. Because these results are visually indistinguishable, we present only a single set of these plots.

operation. It should be apparent when looking at the magnitude of the actual vehicle ground speed that, if no compensation for wind was used when planning and following these interception paths, the wind would introduce large errors in the ETAs of the two vehicles.

We see that even with variation in the simulated wind, the interception path planner was effective at selecting interception paths that accounted for wind. This is made evident by the fact that even with very significant deviations in vehicle ground speed, by commanding the MAV to fly



(a) Constant Airspeed Command

(b) Interception Airspeed Controller

Figure 5.5: Data results from simulated interception path planning and path following. These two sets of data only differ in the use of the interception airspeed controller.

at a constant airspeed it was still able to close the distance between the two vehicles. The approximately 3.5 meter miss distance in the first simulation is the result of two effects: the effect that significant winds have on the vehicle while flying and an increase in airspeed resulting from the transition between following the inclined Dubins airplane interception and a declined ellipse. We have demonstrated that the interception path planner is effective even in the presence of significant deviations in actual vehicle ground speed. While the MAV functioned relatively well when fol-

lowing a constant airspeed command improved performance can be achieved by using the airspeed distance rejection controller.

For both cases the MAV ETA to the proposed interception point was calculated to be 35.39 seconds. The miss distance of the constant airspeed command at the ETA to the proposed interception point was 2.79 meters. For the case with the airspeed controllers it was 0.237 meters. This error is attributed to the transition to following the declined elliptical orbit after navigation along the inclined interception orbit had been completed. Subsequent tracking using the post-orbit-insertion airspeed controller reduced the separation distance between the two vehicles to 0.154 meters.

5.9 Conclusions

In this chapter, we have presented a number of tools that can be used to plan interception paths with a second air vehicle that is following an elliptical orbit. These tools consist of an elliptical arc length approximation method, an explicit equation for vehicle groundspeed, algorithms for calculating the EDTA to a proposed interception point that account for wind, a disturbance rejection airspeed controller, and a post-insertion airspeed controller. These methods for planning interception paths and following were demonstrated in simulation and were shown effective at completing a rendezvous with a desired separation distance within 0.237 meters in the presence of realistic wind. The strategy for post-orbit insertion airspeed control while following the drogue elliptical orbit was also shown to reduce the separation distance error between the two vehicles to 0.154 meters.

While there was variation in the results shown here compared to other simulations with different initial conditions, the trends were the same. In all cases, we found the results of these methods to perform satisfactorily in simulation. When the disturbance rejection airspeed controller was employed, the separation distance was always less than 1 m in the presence of reasonable wind and on elliptical orbits with an inclination angle no greater than 10 degrees.

One of the appealing characteristics of the interception path planning scheme presented in this chapter is that it is not overly difficult to implement in simulation and on hardware. The only complex mathematical function required for hardware implementation is a QR decomposition which is needed to produce the elliptical arc length approximation function.

CHAPTER 6. EXPERIMENTAL RESULTS

In this chapter we present and analyze experimental results obtained from implementation of the methods developed in this paper. We also describe the hardware that was used to obtain these results.

6.1 Hardware Platform

The main components of the hardware used for this project were two approximately one meter wingspan UAVs. Both of these vehicles were outfitted with an onboard Procerus Technologies Kestrel Autopilot. The autopilots contain a small onboard inertial measurement unit (IMU) which is used for the vehicle state estimation performed onboard. A GPS unit must be connected to the autopilot to provide GPS data. Standard RC plane hobby servos and a motor controller were used on the vehicles that were controlled by the autopilot. Various control loops are implemented on the autopilot that allow it to autonomously takeoff, land, and navigate between waypoints and circular orbits. The autopilot uses an antenna to communicate with a Procerus Technologies Commbox which is on the ground. This Commbox interfaces with a software package, also produced by Procerus Technologies, that is referred to as Virtual Cockpit 3D. This software was used to send high-level commands to the autopilot and VPU as well as receive realtime vehicle telemetry and produce plots of the vehicle navigation data. For our tests, the high-level commands allowed us to switch between tracking an estimated drogue orbit or a user specified ellipse and planing and following interception paths. Disturbance rejection airspeed controllers could also be turned on or off in the same fashion. A high-level block diagram of the system architecture can be seen in Figure 6.1. Pictures of the UAV hardware used can be see in Figure 6.2.

One of the vehicles was outfitted with a vision processing unit (VPU) which Procerus Technologies has developed for vision-based target prosecution. All of the algorithms discussed in this paper were implemented onboard the VPU. It uses an ARM processor which is running ARM

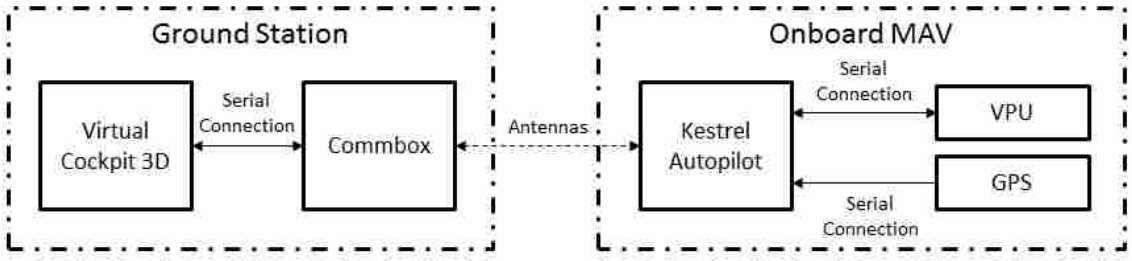


Figure 6.1: Block diagram depicting the system architecture of the hardware setup.

Linux. The VPU communicates with the Kestrel autopilot over a serial port. This allows the VPU to receive autopilot telemetry packets and to send command packets to the autopilot at 30 Hz. The VPU is capable of sending vehicle control surface and throttle commands to the autopilot, high-level commands making use of the control loops on the Kestrel, or a combination of the two. In our tests we used the autopilot pitch angle control loop, the airspeed control loop, and the roll angle control loops. Thus, when implemented onboard the VPU, our drogue interception and orbit tracking algorithms produced a desired roll angle, an airspeed, and a pitch angle commands which were then sent to the autopilot.

To obtain the data necessary to estimate the orbit of the target vehicle, and to plan interception paths, a custom communication packet was developed by Procerus. This packet made it possible for the Commbox to rebroadcast the telemetry of a target agent to the MAV autopilot. The telemetry was then made available to the VPU for the orbit estimation and target interception algorithms.

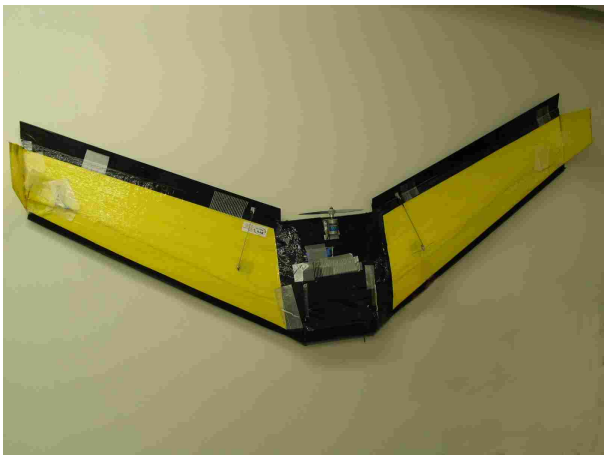
A digital camera was used in conjunction with the VPU. Visual tracking techniques have been developed which make it possible to prosecute user selected air and ground targets. The video feed is transmitted to the ground and displayed in Virtual Cockpit. The user can then select a target in the video feed that MAV should attempt to prosecute. These visual tracking techniques require preliminary methods that can be used to place and maintain the target in the field of view (FOV) of the camera. The research that has been presented in this thesis is intended to fulfill this role and efficiently place the drogue in the FOV of the VPU camera so a visual rendezvous between the two vehicles can occur.

For the algorithms demonstrated in simulation to be implemented on board the VPU, it was necessary to convert them from Matlab code to C++ code. A C++ matrix manipulation library

(a) Procerus Technologies Kestrel Autopilot

(b) Procerus Technologies VPU with Digital Camera

(c) Flight Test Target Vehicle



(d) Flight Test MAV: Zagi Flying Wing

Figure 6.2: Pictures of hardware platforms used to obtain the experimental results presented in this chapter.

called Eigen was adopted to make the conversion process less burdensome. The Eigen library has a very convenient application programming interface (API) that resembles the syntax of Matlab code. It also has functions for computing the singular value decomposition (SVD) and QR decomposition. The SVD is needed when estimating the elliptical orbits of the drogue [4]. This orbit estimation scheme was also converted from Matlab to C++ for implementation onboard the VPU. QR decomposition was used in the computation of the elliptical arc length approximation (see Section 5.2). All of this code was developed in an x86 Linux environment and cross compiled for the VPU arm processor.

Procerus has produced a method for hardware in the loop (HIL) testing of the VPU code. This HIL simulation can be used to produce relatively high fidelity results of the code being tested. To run an HIL test the Kestrel autopilot needs to be connected to the VPU and simultaneously communicating with Virtual Cockpit 3D. A MAV dynamics model is used to provide the autopilot with simulated sensor inputs. The dynamics model is a separate application, which is called Aviones, that communicates with the autopilot through Virtual Cockpit. While this HIL simulator was not used to produce any of the results presented in this section, the simulator was used extensively to debug the more than 3000 lines of code that were needed to implement our algorithms on the VPU; the results presented in this chapter could not have been produced without it.

6.2 Flight Path Angle Estimation and Control

One of the challenges experienced in implementation of the vector field path following controllers was being able to track flight path angle commands. To have closed loop control of the flight path angle, a flight path angle estimate was required. This is not one of the state variables estimated onboard the Kestrel autopilot, so it was necessary for us to produce such an estimate.

The method that was used to produce the flight path angle estimate relied on the GPS sensor measurements. One of the measurements produced by the GPS sensor is of the inertial velocity vector of the vehicle. Using this velocity vector, it is simple to compute the corresponding flight path angle using (2.12). The primary limitations of this method are the relatively poor resolution of the GPS in the vertical direction and the slow measurement update rate of approximately 2 Hz. This results in some significant noise being present in the flight path angle estimate and a delay in the production of the vehicle pitch command from the flight path angle controller (see Section 2.4.1). Nevertheless, we found that this method worked in most situations. The results, and subsequent analysis, we present in this chapter illustrate and discuss some of the weaknesses of the flight path angle estimation and control scheme that we employed.

6.3 Flight Test Data

Due to the complexity of performing flight tests with three vehicles in the air simultaneously, particularly when one of these is a towed vehicle, we decided to demonstrate the interception

path planning and following and the orbit tracking methods with two vehicles. We wish to demonstrate that these methods are capable of planning and following Dubins airplane interception paths from a MAV to a target vehicle. In many cases the drogue orbit is relatively circular and flat. Thus, we feel that demonstration of our methods with an air vehicle flying an approximately circular path is sufficient for demonstrating their effectiveness. One of the challenges with towing a drogue with our current hardware is that the vehicle used to tow the drogue is relatively small. Consequently, the effect of towing a drogue with this vehicle is a decrease in achievable flight time and a decrease in the ability of the mothership to follow commanded orbits. For these reasons, we decided to represent the drogue with another air vehicle which we will refer to in this chapter as the target vehicle.

We performed two types of flight tests. The first had a MAV flying and implementing the control algorithms while the MAV received data from an autopilot in a simulated flight mode. The second had both vehicles simultaneously airborne. The MAV used the forwarded data from the target vehicle in realtime to estimate the target vehicle orbit, to plan interception paths, and to track the target vehicle.

Due to vehicle flight time constraints, we first tested these algorithms with a simulated agent. This simulation method used a Kestrel autopilot that was operating with simulated sensor data. The benefit of this method was that the Kestrel autopilot still communicates with Virtual Cockpit in the same way as an autopilot that is actually being flown. Thus, the telemetry from the simulated agent can be forwarded to the MAV in exactly the same way as true flight telemetry. This provides the capability of testing the MAV interception and elliptical orbit tracking algorithms in real flight, with realistic data, but without actually having a second vehicle in the air.

After these preliminary flight tests with a simulated agent, the algorithms were also tested with both vehicles simultaneously airborne. In this chapter, we present results from both types of flight tests. The benefit of obtaining results using the simulated agent is a decrease in the errors introduced in our results by occasional erratic behavior of the target vehicle. Thus, using the simulated agent helped make the characteristics of the interception algorithm performance more visible.

For all of the results presented in this chapter, the data between the MAV and target vehicle were synchronized using coordinated universal time (UTC) stamps. The MAV data was stored

on the VPU at 30 Hz. The target vehicle data had to be transmitted to the ground through the Commbox and saved in a data log via Virtual Cockpit. Depending on the level of packet loss and bandwidth constraints in communication, these data points were available between 2-8 Hz.

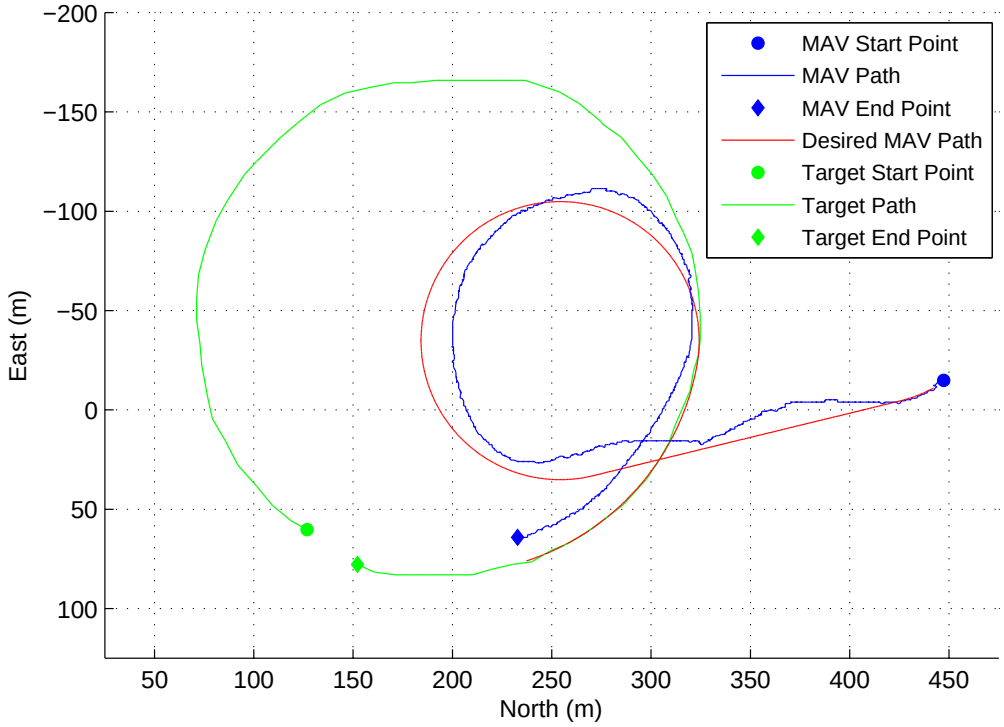
6.3.1 Interception Path Planning and Following

In this section we will present the results for the interception path planner and path follower. We performed a variety of tests and present two cases which illustrate the effectiveness of these methods and also demonstrate some of the current weaknesses. In addition to the standard vehicle telemetry, each time a path was planned the initial configuration of the vehicles, the current orbit, and the Dubins airplane interception path parameters were saved to a text file on the VPU. Using these data we were able to plot the desired path of the vehicle which begins at the interception path start point and includes a portion of path following along the elliptical orbit.

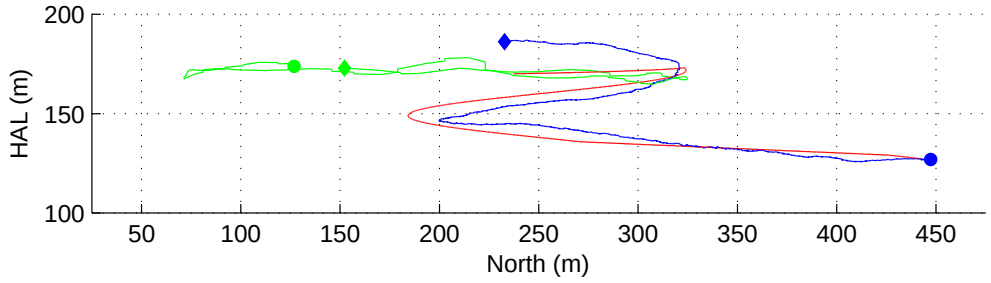
6.3.2 Airborne Target Vehicle Interception

The first test results shown used two airborne vehicles. The target vehicle was commanded to fly a circular orbit at an altitude of 175 meters. In this test, the MAV was initially flying a circular orbit approximately 100 meters north of the commanded orbit of the target agent at an altitude of 125 meters. An interception path was planned which the MAV proceeded to follow. The data shown starts at the point which the MAV began to follow the Dubins airplane interception path and ends after the target vehicle has nearly completed one revolution. Over the course of the flight the wind speed was roughly 2 m/s. During the interception path following operation the MAV was commanded to fly at a constant airspeed of 14 m/s.

Various data plots that correspond to this flight and that we found insightful are included in this section. Figure 6.3 shows a top view and side view of the actual paths of the two vehicles from the vehicle GPS data as well as the desired interception path. Figure 6.4 displays the separation distance between the two vehicles during the entire operation. Figure 6.5 (a) shows the error between the vehicle and the desired path in various planes and (b) shows the realtime estimated components of the wind. The wind estimation method onboard the Kestrel autopilot assumes that there is no wind in the vertical direction, so we did not include this component of the wind in the



(a) Top view



(b) Side vide

Figure 6.3: Plots of the desired MAV interception path, the actual MAV path, and the target vehicle path from GPS data.

plots. Lastly, Figure 6.6 shows the desired and measured values of the MAV airspeed, pitch, roll and throttle which provide insight into the vehicle behavior over the flight test. We will refer to the data in these figures during our analysis of the flight test.

Inspection of Figure 6.3 shows that while there was some deviation in the MAV flight path from the desired interception path, the MAV followed the path in general. Following the Dubins

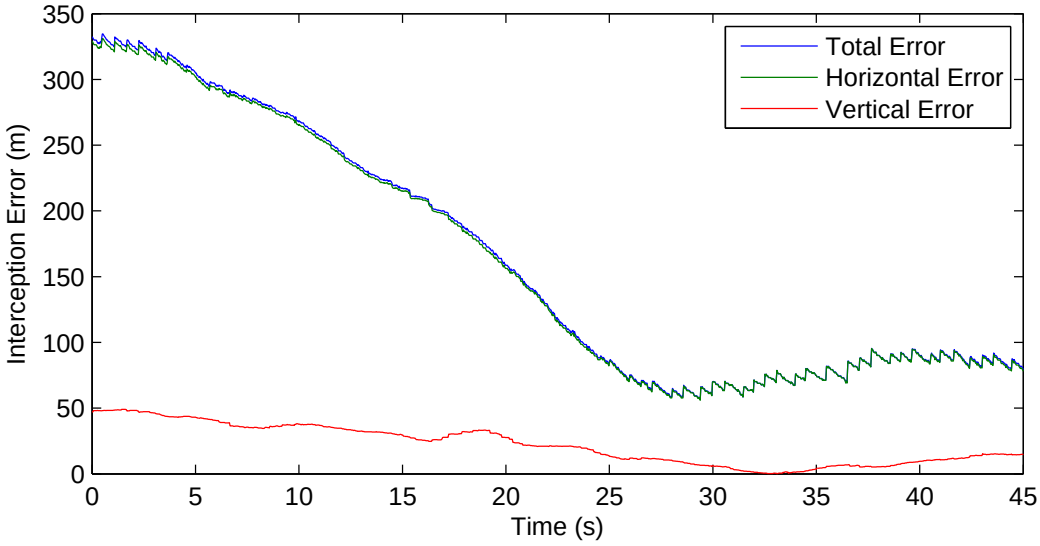
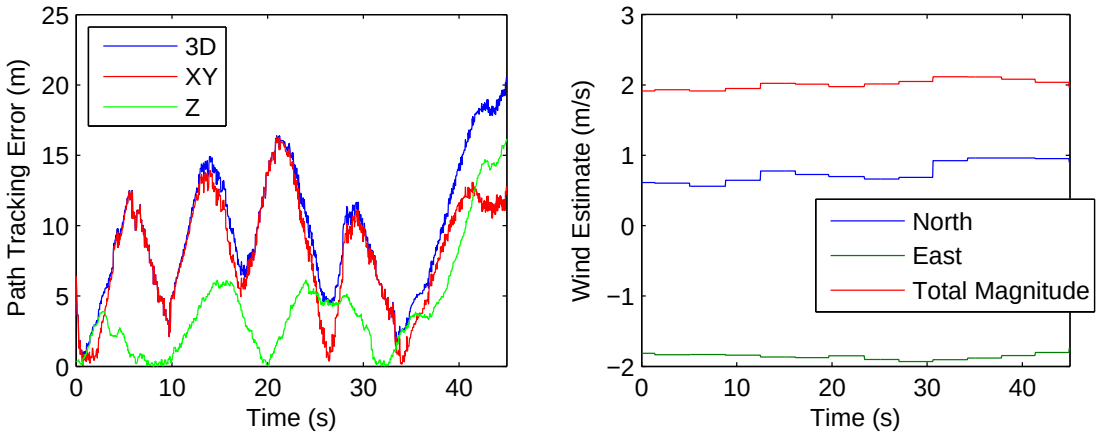


Figure 6.4: Distance between the target vehicle and the MAV while the MAV follows the Dubins airplane interception path and the elliptical orbit of the target vehicle.



(a) Path Tracking Error

(b) Wind Data

Figure 6.5: Plot of the path tracking error between the desired and actual MAV path and the realtime wind estimates.

airplane interception path took the MAV approximately 26 seconds, at which point it transitioned to following the estimated target vehicle orbit. Figure 6.5 shows that the interception path tracking error was generally less than 5 meters in the vertical direction and less than 16 meters over all. While decreasing this error would be desirable to improve the accuracy of the interception pro-

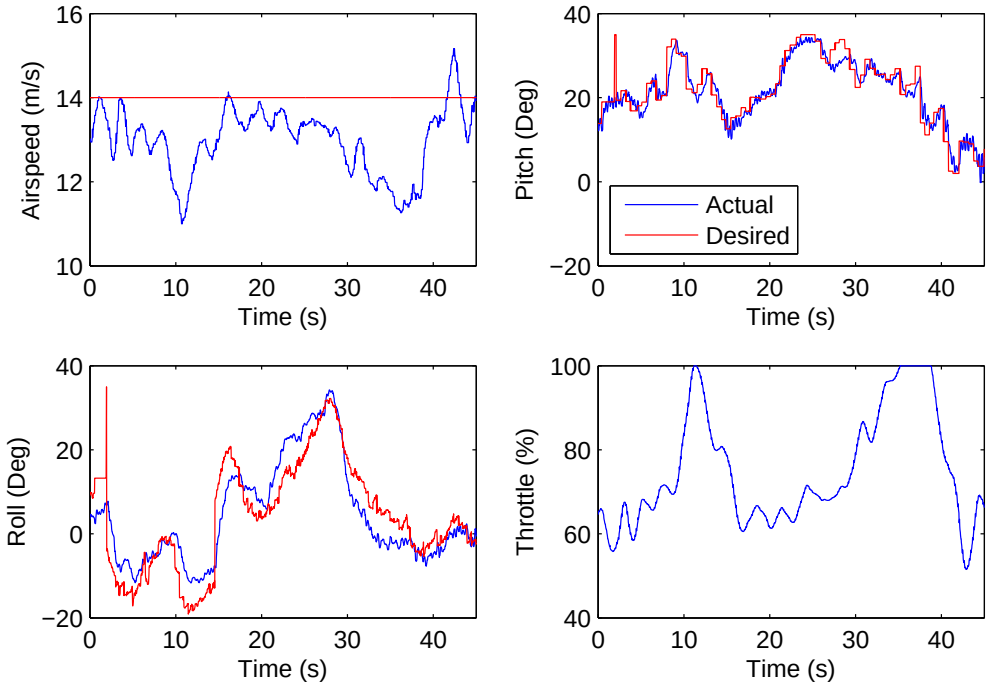


Figure 6.6: Relevant data from the MAV over the course of the interception path following operation.

cedure, the error is low enough to adequately demonstrate the Dubins airplane, vector field path following method.

The interception path tracking errors seen in Figure 6.4 are due to a number of effects. The first is seen during a transition from a circular path leg to a straight path segment or vice versa. Prior to following the interception path, the MAV had been following a flat circular orbit at a constant altitude of 125 meters. The interception path began with a short, clockwise circular helix path. As the MAV began to follow this path it transitioned well from constant altitude flight to the appropriate climb rate. The transition from the first circular leg to the straight segment had some characteristic overshoot, but, again, the MAV converged to the desired path. The second source of tracking error, which can be seen during tracking of the second circular leg, is due in part to the tail wind that the MAV experienced at this point.

After the MAV completed following the Dubins airplane interception path, it transitioned to following the estimated target orbit. At this point there was an overshoot in altitude, partially due to the headwind which was compensated for by a commanded decrease in pitch. Analysis of

the flight test data suggests that the path tracking performance could be significantly improved by increasing the horizontal vector field contraction gains and, potentially, the course angle hold loop integrator gain.

As the MAV transitioned from completing the last leg of the interception path to the target orbit it switched from a following a relatively tight orbit to a large orbit. There was some expected overshoot during this transition, but the MAV was slow to correct in the horizontal plane. It is probable that the elliptical orbit, horizontal vector field gain and the course angle hold loop integrator gain could be increased to improve the overall vehicle performance.

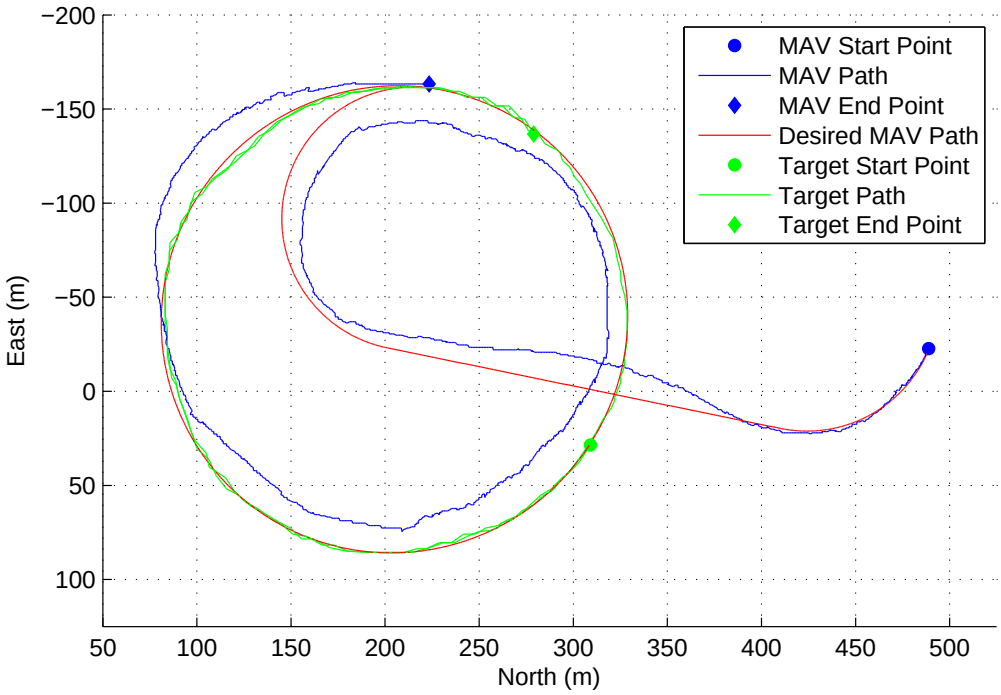
Based on the current hardware setup, tracking error plots can only be produced through post processing. Thus, tuning the vector field and low-level controller gains can only be performed visually from the vehicle GPS data plotted in Virtual Cockpit. One of the challenges experienced when doing this testing was the inability to plot the interception paths in Virtual Cockpit. Consequently, we cannot obtain any visual or numerical feedback that indicates how well the path following methods were working until we post process the data. Currently, we are only capable of plotting the desired interception paths after the fact. There is a possibility of adding such a plotting capability to Virtual Cockpit. If this avenue were pursued, it would make tuning of the vector field gains significantly easier and should improve the tracking performance of these methods.

The target vehicle interception performance was acceptable. After completing the interception path the separation distance between the two vehicles was approximately between 60 and 80 meters as seen in Figure 6.4. This equates to difference in arrival time of roughly 4 seconds. In this test, the MAV was commanded to fly at a constant velocity of 14 m/s. The miss distance of 4 seconds was primarily due to the MAV not maintaining a constant airspeed of 14 m/s as evident in Figure 6.6.

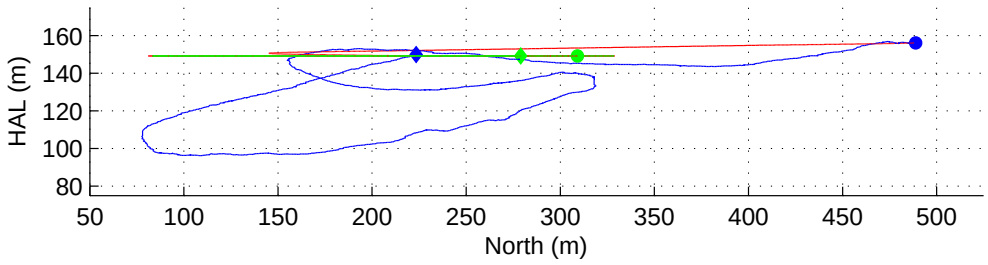
6.3.3 GPS Simulated Target Agent Interception and Post-Insertion Tracking

The test results we now present were produced using a simulated target vehicle that was commanded to fly a constant altitude, circular orbit. In this test, the MAV was initially flying a circular orbit approximately 100 meters north of the commanded orbit of the simulated drogue agent. An interception path was planned which the MAV proceeded to follow. The data shown starts at the point in time when the MAV began to follow the Dubins airplane interception path;

It ends after the MAV has completed the interception path and followed the target vehicle along the estimated orbit for nearly one revolution. Over the course of the flight, the wind speed was roughly 3 m/s. During the interception path following operation, the MAV was commanded to fly at a constant airspeed of 14 m/s. After the MAV had completed following the interception path, the post-insertion airspeed controller was turned on. With this controller, the desired separation distance between the two vehicles was specified to be 15 meters.



(a) Top view



(b) Side vide

Figure 6.7: Plots of the desired MAV interception path, the actual MAV path, and the target vehicle path from GPS data.

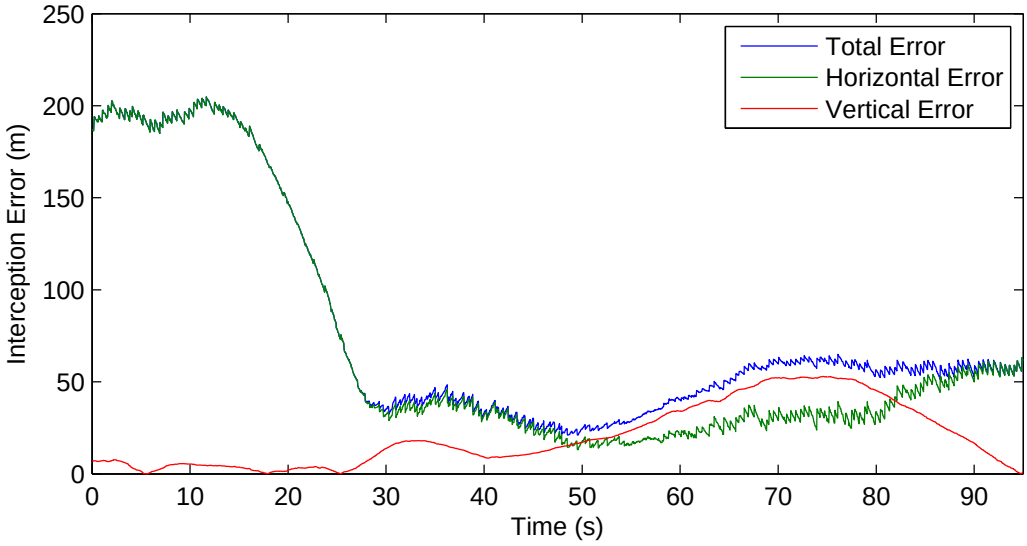
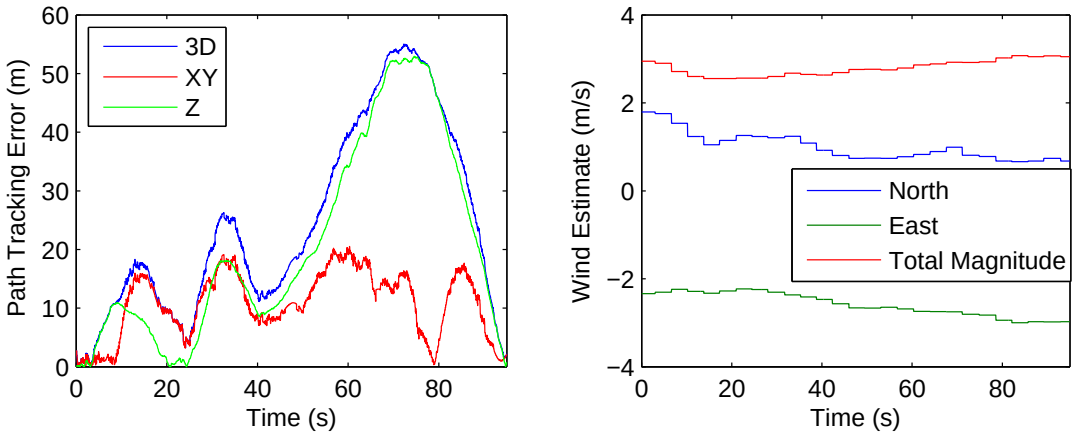


Figure 6.8: Distance between the target vehicle and the MAV while the MAV follows the Dubins airplane interception path and the elliptical orbit of the target vehicle.



(a) Path Tracking Error

(b) Wind Data

Figure 6.9: Plot of the path tracking error between the desired and actual MAV path and the realtime wind estimates.

Again, data plots similar to those shown for the previous flight test are presented for this test. Figure 6.7 shows plots of the GPS data from the MAV, the simulated data from the target agent, and the desired interception path. Figure 6.8 displays the separation distance between the two vehicles during the entire operation. Figure 6.9 (a) shows the error between the vehicle and

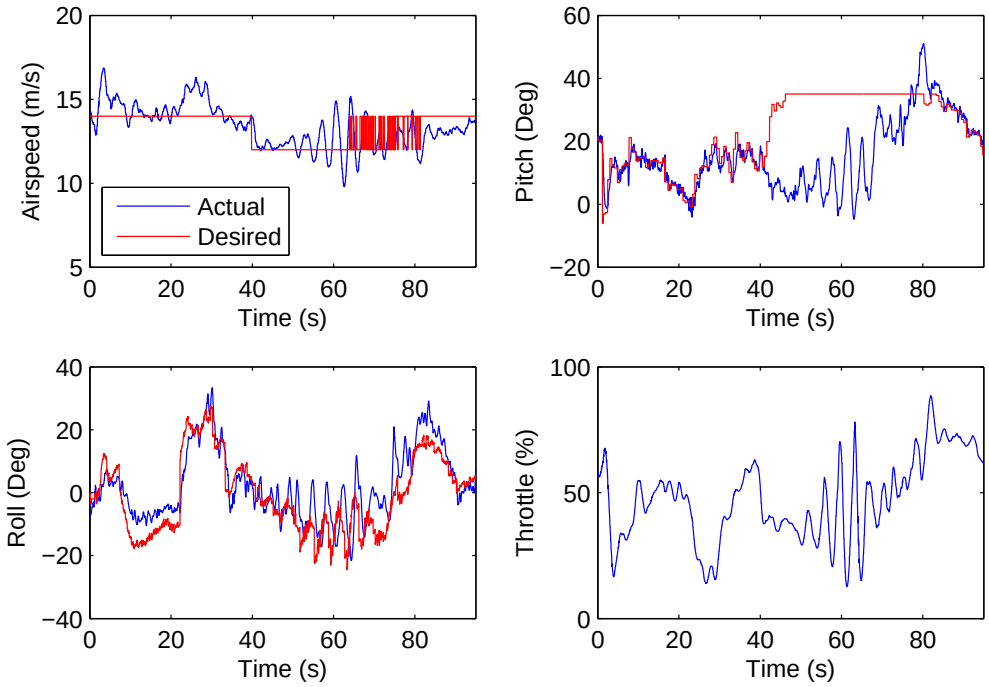


Figure 6.10: Relevant data from the MAV over the course of the interception path following operation.

the desired path in various directions, and (b) shows the estimated components of the wind. Lastly, Figure 6.10 shows the desired and measured values of the MAV airspeed, pitch, roll, and throttle. We will refer to the data in these figures during our analysis of the flight test.

We now provide a narrative of the second flight test. It is apparent that the MAV initially followed the desired interception path with minimal error. The turn on the first leg of the interception path, as expected, causes the MAV to lose some altitude. As the MAV transitions from the first circular leg to the straight path there is some overshoot and it gradually climbs back to altitude and converges toward the straight leg of the interception path. We see this in the path tracking error shown in Figure 6.9. At about 10 seconds into the flight, the MAV deviates nearly 20 meters from the interception path but then converges back to within 5 meters of the path, at which point it transitions to the second circular leg of the path. It is helpful to point out that the general direction of the wind during this flight is toward the northwest, which is toward the upper right corner of the top view of the GPS data plot (see Figure 6.7).

As the MAV begins to follow the second circular leg of the interception path, it has a relatively strong tailwind that results in the MAV losing 20 meters in altitude and a similar deviation in the horizontal direction. We see that the convergence back to the path in both the horizontal and vertical direction is relatively slow, but the vehicle is indeed converging toward the desired interception path. At approximately 40 seconds into the flight, the MAV effectively completes the interception path and switches to the post-insertion, elliptical orbit tracking mode. Largely due to the MAV having flown a path interior to the desired interception path, it arrived to the proposed interception point approximately 30 meters ahead of the target vehicle (see Figure 6.8). Based on the MAV airspeed, that equates to an error in arrival time of approximately 2 seconds.

After completing navigation of the interception path, the MAV transitions to tracking the estimated ellipse. The post-insertion airspeed controller, as described in Section 5.7, was implemented on this flight. This controller varies the commanded airspeed of the MAV in an attempt to maintain a 15 meter separation distance between the two vehicles.

Due to the fact that the MAV arrived to the proposed interception point in front of the target vehicle with a separation distance of approximately 30 meters, the post-insertion airspeed controller reduced the commanded airspeed to 12 m/s (see Figure 6.10). This occurred about 40 seconds into the flight. The MAV had been converging toward the desired path until this change in airspeed command. Rather than reducing the separation distance between the two vehicles, this airspeed controller caused a large loss of altitude for the MAV.

These results demonstrate one of the weaknesses of the lower-level control scheme being employed for these vector field path following methods which we now explain. When the reduced airspeed command was received, the vehicle immediately reduced the throttle. This reduction in throttle and resultant loss of airspeed caused the pitch angle of the vehicle to decrease. At this point, the higher level vector field path following commands increased the pitch angle command to the vehicle. This suggests that the path following methods were indeed providing good commands. The effect of this decrease in pitch angle was that the vehicle began to lose altitude. To maintain the airspeed command while losing altitude the throttle was decreased. The pitch controller attempted to pull up the vehicle, which the resulted in the pitch, roll and airspeed controllers going unstable.

The instability is likely due to the coupling between the roll angle, pitch angle, and airspeed combined with the rather slow rate of the flight path angle estimate from the GPS. This estimate

is used in the flight path angle hold loop, which produces the pitch command. In the hardware implementation of this code, the flight path angle controller runs at 2 Hz because that is the rate of flight path angle estimation. As a result of this slow update rate, a significant amount of delay from the flight path angle estimation is introduced into the pitch controller. We believe that largely due to this lag, the airspeed and pitch controllers competed with each other and caused this approximately 50 meter loss of altitude over roughly 20 seconds (see Figure 6.9). After this period of altitude loss, stability was again achieved, and the MAV proceeded to converge to within 5 meters of the desired orbit at a distance of roughly 50 meters behind the target vehicle (see Figure 6.8).

This flight test makes it clear that in some scenarios the coupling between the pitch and airspeed controllers can cause the system to go unstable and fail to track both the pitch and airspeed controllers. We saw evidence of this effect in a number of flights. The controller instability tended to occur when there was a large change in commanded airspeed. When this occurred, the airspeed controller decreased the throttle. If the delay in the pitch controller prevented a sufficiently fast response, the vehicle began to lose altitude which inherently increased the airspeed of the vehicle and caused the throttle to be further decreased. We believe that, in this situation, the lag in the pitch controller effectively caused the two controllers to fight each other and fail in tracking the pitch or airspeed commands for a time.

Once the vehicle returned to the desired altitude and converged to the elliptical orbit, the post-insertion airspeed controller was effective at closing the distance between the two vehicles. At the closest point, the MAV was approximately 25 meters from the simulated target vehicle (see Figure 6.4). We will provide further results from flights employing the post-orbit insertion method and more detailed analysis later in this chapter.

These results are representative of the general performance and flight behavior we saw in all flights. From these results we make a number of observations. While following the interception paths, the maximum path tracking error was approximately 25 meters (see Figure 6.9). It is probable that deviation was primarily due to the disturbance of a tail wind. When the MAV transitioned between the circular and straight path legs, deviations from the path occurred between 18 and 25 meters. From the Figure 6.9 it is evident that, after these spikes in path tracking error at the transition points, the MAV converged back toward the interception path to within 5 or 10 me-

ters. We believe that further tuning of the vector field gains, to increase the aggressiveness of the convergence in both the vertical and horizontal directions, would reduce the path tracking error.

We conclude, from these results, that the time-optimal Dubins airplane path, vector field path following method is effective at tracking these paths. These data also demonstrate that the interception path planner can be used to plan interception paths to the target vehicle on a slightly elliptical orbit. The separation distance between the two vehicles at the interception point is primarily due to the error in tracking the desired interception path. Thus, a reduction of the interception path tracking error would also result in a reduction of the separation between the two vehicles at the proposed interception point.

Although it cannot be definitively concluded, we believe that the primary cause of the path tracking error in this scenario resulted from the period of instability in lower-level control loops. We believe this instability is due to a lag in the desired pitch angle commands, which is the result of the current flight path angle estimation scheme. We propose an estimation technique to reduce the flight path angle lag introduced by the relatively slow GPS measurement update rate. The flight path angle is currently being estimated only when new GPS inertial velocity data is available, which is at a rate of approximately 2 Hz.

This improved flight path angle estimation makes use of the accelerometer data available from the autopilot IMU. When these accelerometer measurements are rotated into the inertial frame and integrated, they provide the inertial velocity estimates. These estimates are inherently noisy, but they can be used to supplement the GPS inertial velocity estimates. One of the primary benefits of using these measurements is that they are available at 30 Hz rather than just 2 Hz. A low pass filter can be applied to the accelerometer measurements before integrating them to produce the updated inertial velocity estimates. When the filtered accelerometer data is integrated, it is added directly to the current inertial velocity estimates.

The GPS inertial velocity estimate can be filtered into the previous inertial velocity estimate using a low pass filter each time a new measurement is received. This scheme provides an inertial velocity estimate which can be used to provide a flight path angle estimate at the same rate. This increased rate of flight path angle estimation will allow the flight path angle controller to be run at a faster rate and should reduce the lag in the resultant pitch angle command. If the lag in the pitch command is in fact the cause of the MAV low-level loop instability then this strategy

should mitigate the problem and potentially eliminate the situation in which instability occurs. If the accelerometer integration technique described here is not adequate we recommend using a Kalman filter to combine the sensor data to provide an improved estimate of the flight path angle.

It is possible that the lag is due to lag inherent in the inertial velocity estimation scheme being employed on the GPS unit. To determine whether this is the problem, we recommend first implementing the above mentioned GPS and accelerometer fusion scheme and performing further tests to assess how this impacts the system instability.

6.3.4 Airborne Target Post-Insertion Tracking

In this test, we demonstrated the post-insertion airspeed controller. For the plots to clearly depict the flight behavior, we decided to show these post-insertion tracking results separate from the interception results. These flight test results were obtained with both the MAV and target vehicles airborne.

The target vehicle was commanded to fly a flat 125 meter radius circular orbit at an altitude of 150 meters. The figures containing the data that correspond to this flight test are Figures 6.11, 6.12, 6.13, and 6.14. These contain plots very similar to those shown with the flight test results already presented in this chapter.

At the point in which the data from this flight begins, the MAV had just completed following an interception path which inserted the MAV behind the target vehicle with a separation distance of approximately 70 meters which (shown in Figure 6.12). At this point in time, the post-insertion airspeed controller was turned on, which can be seen in the airspeed data plot in Figure 6.14. In the data presented here, the MAV proceeded to follow the target vehicle for about one and a half orbit revolutions. In this flight, the wind speed was approximately 2 m/s and directed to the northwest, which is the upper left-hand corner of Figure 6.11.

The purpose of the post-insertion airspeed controller is to vary the commanded airspeed of the MAV to maintain a desired following distance. In this flight, the desired following distance was selected to be 15 meters. Similar to the results of the previous two tests there were significant deviations when following the desired orbit with the MAV. These can be explained in a manner similar to the previously discussed flight test results. In the presence of a tailwind, the MAV does not track as well and the opposite effect occurs in the presence of the wind. To decrease these

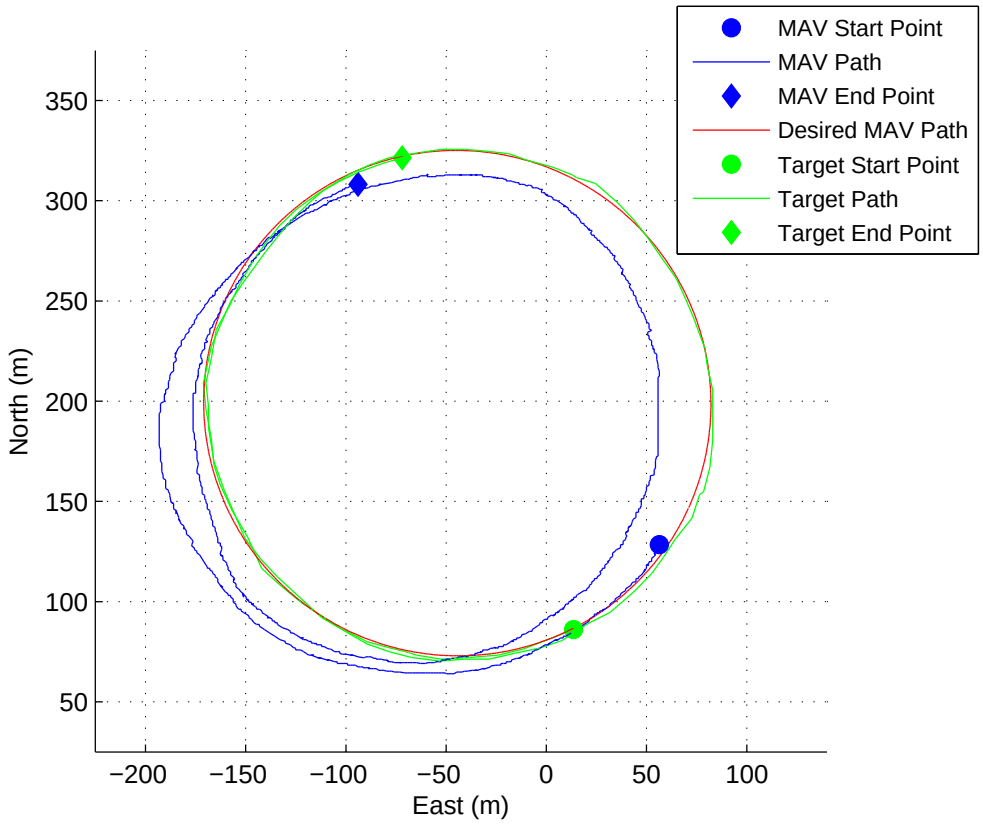


Figure 6.11: Plot of the estimated target vehicle orbit, the target vehicle path, and the actual MAV path during post-insertion tracking.

tracking errors, we again recommend further tuning of the vector field gains and the course angle controller integrator gain. Doing so should substantially reduce this error.

Figure 6.12 shows the separation distance between the two vehicles during this flight. These results show that post-insertion airspeed controller decreased the separation distance between the vehicles to between 15 and 30 meters for the more than 40 seconds depicted in that figure. This desirable behavior indicates that this controller is effective at maintaining a following distance of 15 meters.

One of the issues that was made apparent when implementing this scheme was that this controller can cause some undesired behavior when the MAV deviates significantly from the target orbit. The objective of this airspeed controller was to maintain the target vehicle in the camera FOV while following the target orbit. The weakness of this controller can be seen when viewing the interception error plot in Figure 6.12 between 46 to 50 seconds in the flight. At this point in

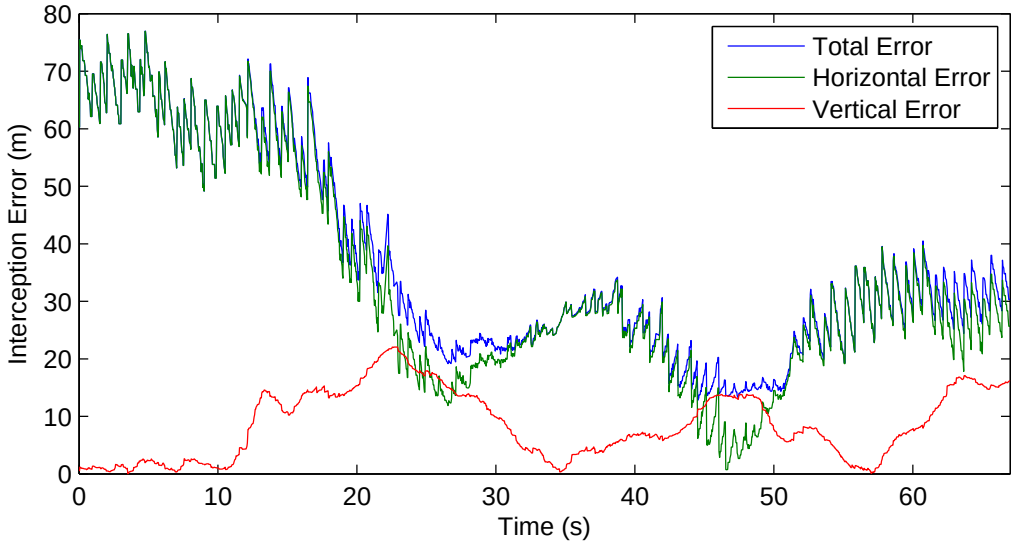
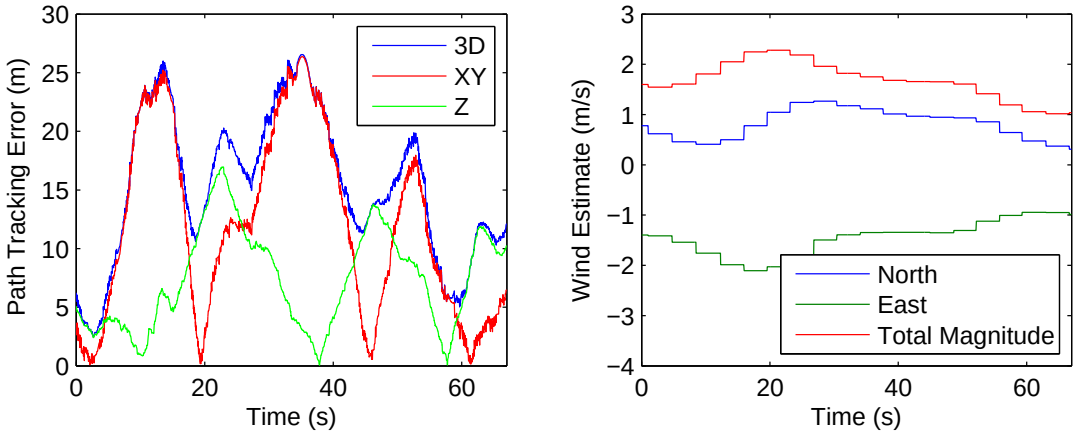


Figure 6.12: Distance between the target vehicle and the MAV during post-insertion tracking.



(a) Path Tracking Error

(b) Wind Data

Figure 6.13: Plot of the path tracking error between the estimate target vehicle orbit and the actual MAV path in addition to the realtime wind estimates during post-insertion tracking.

time, the separation distance between the two vehicles in the horizontal plane dropped very close to 0 meters, but the vertical error was approximately 15 meters. While it is true that the airspeed controller achieved a separation of 15 meters, the MAV was positioned almost directly below the target vehicle and out of the FOV of the front facing camera on the MAV. When the airspeed controller has achieved the desired separation distance between the two vehicles, it commands the

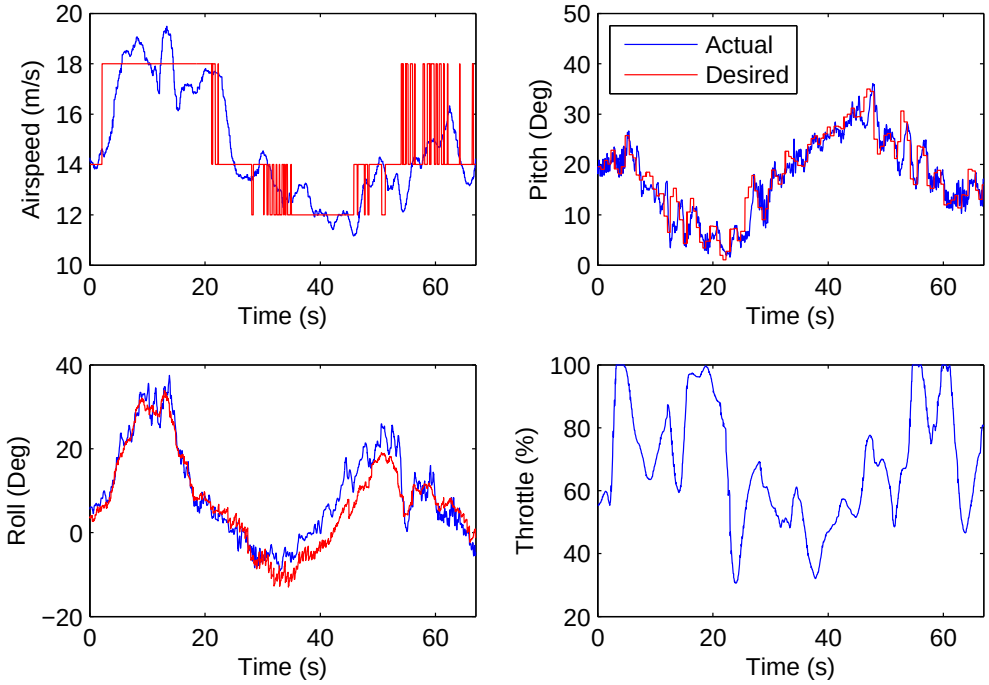


Figure 6.14: Relevant data from the MAV during the post-insertion tracking operation.

MAV to fly at the same airspeed as the target vehicle. Thus, in the case just discussed the controller would attempt to maintain the position directly below the target vehicle. Since our objective was to have the vehicle positioned 15 meters in front the MAV, and not in an arbitrary direction, we recommend a modification to this scheme.

To overcome this weakness, we propose a new definition for the separation distance. Rather than use the Euclidean distance between the vehicles to calculate the effective tracking error, we wish to define and use the forward separation distance. The forward separation distance is the projection of the Euclidean distance vector onto the vector that describes the current MAV direction of travel. If the forward separation distance is used rather than the Euclidean distance, it should mitigate the limitations of the current method.

In the scenario above, where the MAV was located almost directly below the target vehicle, the forward separation distance airspeed controller should produce much more desirable results. In that situation, the forward separation distance airspeed control would register a forward separation distance between the two vehicles near zero. As a result, it would command the MAV to decrease airspeed until the forward distance was 15 meters. With this modified airspeed controller, even if

the altitude difference between the vehicles were still 15 meters, the MAV would have increased the forward separation distance, and it is probable that the target vehicle would now be in the camera FOV.

6.3.5 Interception Disturbance Rejection Airspeed Controller

Some results were obtained from the interception airspeed controller presented in Section 5.6, but, due to flight test time constraints, they were somewhat limited and we do not believe that they merit a detailed analysis. The basic result was that, when the MAV was accurately tracking the desired interception path, the airspeed controller worked well. The vehicle airspeed was varied appropriately to maintain the desired difference in arrival time. This is the same result as shown in simulation results of this controller in Section 5.8.

The current weakness of this controller is that, when there is a significant deviation from the desired interception path, the controller compensates for this by increasing the commanded airspeed. Due to the coupling between the appropriate airspeed, roll angle, and pitch angle commands, any significant change in airspeed also requires changes to be made to the roll and pitch commands. Thus, we typically see that significant changes in airspeed, changes of roughly 1 m/s or more, cause an increase in the path tracking error until the roll and pitch controllers are able to respond appropriately. This is particularly apparent when the airspeed increases. Consequently, attempting to substantially increase the vehicle airspeed, to compensate for tracking errors, tends to exacerbate the problem and further increase these path tracking errors.

To mitigate this problem, we suggest selecting a tracking error window within which the interception, disturbance rejection airspeed controller is to be used. Thus, if the tracking error is less than a user specified value, the airspeed command from the controller would be used. One potential problem with this scheme is that it would introduce a discontinuity in the airspeed command when the vehicle transitions from using the constant airspeed command to the controller based command. A possible solution to this would be to proportionally transition between the two values. When the vehicle has zero tracking error, the commanded airspeed would be equal to be the commanded airspeed produced by the controller. When the vehicle is at the boundary of the selected tracking error window, or with a greater tracking error, the constant airspeed com-

mand would be used. Anywhere between these two tracking error values, a proportional weighting scheme, or some other variant, could be used to prevent discontinuities in the commanded airspeed.

6.4 Chapter Summary

In this chapter we presented experimentally obtained flight test results. These results illustrated the methods and algorithms presented in this thesis. We analyzed the flight test data to determine the strengths and weakness of target interception and tracking methods. Generally speaking, we found that the methods presented in the paper are effective at achieving their objectives. A number of weaknesses were discussed and schemes for mitigating the errors introduced by the weaknesses were presented.

We believe that the experimental results we presented here adequately demonstrate the aerial interception and orbit tracking methods. Nevertheless, we believe that these errors produced in our results could be substantially reduced through subsequent flight tests if the vector field and control loop gains were further tuned.

CHAPTER 7. CONCLUSIONS

7.1 Conclusions

In this thesis, a method was produced for intercepting a target air vehicle flying a roughly elliptical orbit with an autonomous MAV preparatory to a visual rendezvous. The algorithms necessary to plan an interception path that accounts for the effects of wind were also developed and presented. A scheme for following the Dubins airplane interception paths as well as elliptical orbits was proposed. An airspeed controller was also developed that can be used to reject disturbances and maintain a coordinated arrival time between the two vehicles to a proposed interception point. After the vehicle has completed following an interception path, another airspeed controller was proposed which can be used to maintain a desired following distance between the MAV and target vehicle, which is necessary to facilitate a final visual rendezvous. These methods were demonstrated in simulation and experimentally with actual flight tests. The contributions of this research are summarized as follows:

- Vector field path following schemes, that have proven convergence properties, for following time-optimal Dubins airplane paths and arbitrarily oriented elliptical orbits. These methods were demonstrated with simulated and experimental results.

- A simple, robust solution for planning time-optimal Dubins airplane interception paths between a MAV and a target vehicle that accounts for the effects of wind. These interception paths can be planned from an arbitrary set of initial conditions and result in the interception, or coordinated arrival, of the MAV and a target vehicle. The current scheme is established for a target vehicle following an elliptical orbit, but, in general, it is applicable to any type of path for which an estimated time to arrival function can be produced for the target vehicle between any two points on the vehicle path. This interception scheme was demonstrated in simulation and with experimental results.

- A conference paper which was presented at and published in the proceedings of the AIAA Guidance, Navigation, and Controls Conference.
- It is anticipated that a journal paper that includes a significant portion of the work presented in this thesis will be shortly submitted for publication.

The primary contributions of this research are dependent upon a number of lower-level contributions. These contributions can be summarized as follows:

- Development of a simple flight path angle estimation scheme and a closed loop controller that is needed to follow the commands produced by the vector field path following methods.
- A method for the nondimensionalization of the elliptical orbit and Dubins airplane path following vector fields that allow much more general use of a given set of vector field gains.
- Development of a numerically inexpensive method that can be used to approximate elliptical arc length functions. This method also has potential application for use with other functions which can be expressed parametrically.
- An explicit solution for aircraft ground speed from the wind triangle. This result makes it possible to estimate the future ground speed along a path based upon the wind estimate and other readily available information.
- A method for producing disturbance rejection airspeed commands during an interception operation that can be used to maintain the coordinated arrival of the MAV and target vehicle to a proposed interception point.
- A post-orbit-insertion airspeed controller that can be used to maintain a desired separation distance between the MAV and target vehicle. This controller facilitates the initialization of the visual tracking and target prosecution algorithms.

7.2 Future Work

There are many areas of research from the thesis which could be further explored. Further research in these areas would potentially offer improved performance of the drogue interception in

addition to providing contributions to the research area in general. Some recommended areas of further research and extensions to the methods presented in this thesis include:

- Modification of the post-orbit-insertion airspeed controller to use the forward separation distance when computing the appropriate airspeed commands. This would prevent the controller from being influenced unnecessarily by lateral and vertical tracking errors. See Section 6.3.4.

- Developing an improved flight path angle estimation method that decreases the lag in the MAV pitch angle commands. This could potentially be an implementation of the scheme discussed in Section 6.3.3.

- Improving the feed-forward roll angle command with a model that does not assume horizontal flight.

- Further tuning of the vector field gains would potentially offer improved tracking performance.

- Developing a wind based time-optimal Dubins airplane paths. Such a result has been developed for horizontal flight in [17]. This could be extended for use with the time-optimal Dubins airplane paths. The path following scheme developed in this paper could be applied to follow these time-optimal paths that include the effects of wind. Development of a path following scheme for these time-optimal paths which are valid in the presence of wind could offer a significant performance improvement for the current drogue interception scheme.

- Currently, the time-optimal Dubins airplane path following scheme has only been implemented for paths consisting of an arc, a straight segment, and concluded by another arc. These are only guaranteed time-optimal when the initial and final point are separated by a distance of at least three minimum turn radii. The other case consists of paths made up of three minimum turn radius arcs. Developing the capability of following and planning interception paths with this type of Dubins airplane path would increase the functionality of the drogue interception scheme.

- Developing a scheme that solves for a modified elliptical orbit that would maintain the drogue in the FOV of the MAV camera even when the MAV has a significant crab angle due to the presence of wind.

REFERENCES

[1] Sun, L., Beard, R. W., and Colton, M. B., 2010. "Motion planning and control for mothership-cable-drogue systems in aerial recovery of micro air vehicles." In *Proc. American Control Conf. (ACC)*, pp. 2101–2106. vii, 2, 25

[2] Beard, R. W., and McLain, T. W., 2011. *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press. vii, 4, 7, 9, 12, 15, 16, 18, 19, 37, 44, 46, 52, 60, 65, 111, 117, 120

[3] Sun, L., Beard, R. W., Colton, M. B., and McLain, T. W., 2009. "Dynamics and control of cable-drogue system in aerial recovery of Micro Air Vehicles based on Gauss's principle." In *Proc. ACC '09. American Control Conf*, pp. 4729–4734. 1

[4] Carlson, D. C., and Colton, M. B., 2010. "Out-of-plane orbit estimation and tracking for aerial recovery of micro air vehicles." In *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, pp. 1130–1135. 3, 25, 57, 85

[5] Dubins, L. E., 1957. "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents." *American Journal of Mathematics,* **79**(3), pp. 497–516. 3, 43

[6] Chitsaz, H., and LaValle, S. M., 2007. "Time-optimal paths for a dubins airplane." In *Proc. 46th IEEE Conference on Decision and Control*, pp. 2379–2384. 3, 4, 45

[7] Lawrence, D. A., Frew, E. W., and Pisano, W. J., 2008. "Lyapunov vector fields for autonomous unmanned aircraft flight control." *Journal of Guidance, Control, and Dyanmics,* **31**(5), Septermber-October, pp. 1220–1229. 4, 7, 8, 9, 10, 11, 12, 27, 41

[8] Frew, E. W., Lawrence, D. A., Dixon, C., Elston, J., and Pisano, W. J., 2007. "Lyapunov guidance vector fields for unmanned aircraft applications." In *Proc. American Control Conf. ACC '07*, pp. 371–376. 4, 7

[9] Goncalves, V. M., Pimenta, L. C. A., Maia, C. A., Dutra, B. C. O., and Pereira, G. A. S., 2010. "Vector fields for robot navigation along time-varying curves in *n*-dimensions." pp. 647–659. 4, 7, 8, 9, 12, 14

[10] Nelson, D. R., Barber, D. B., McLain, T. W., and Beard, R. W., 2007. "Vector field path following for miniature air vehicles." pp. 519–529. 4, 7

[11] Lee, J. G., Han, H. S., and Kim, Y. J., 1999. "Guidance performance analysis of bank-to-turn missiles." In *Proc. IEEE Int Control Applications Conf*, Vol. 2, pp. 991–996. 12, 19

[12] Stengel, R. F., 2004. *Flight Dynamics*. Princeton University Press. 15, 37

[13] Barber, D. B., Griffiths, S. R., McLain, T. W., and Beard, R. W., 2007. "Autonomous landing of miniature aerial vehicles." *AIAA Journal of Aerospace computing, Information, and Communication,* **4**, pp. 70–84. 22

[14] Nelson, D. R., Barber, D. B., McLain, T. W., and Beard, R. W., 2006. "Vector field path following for small unmanned air vehicles." In *Proc. American Control Conf.* 46

[15] Owen, M., Nichols, J., and Colton, M. B., 2011. "Cooperative aerial tracking and rendezvous along time-optimal 3-dimensional curves." In *AIAA Guidance, Navigation and Control Conference*. 59, 69

[16] Moon, T. K., and Stirling, W. C., 2000. *Mathematical Methods and Algorithms for Signal Processing*. Prentice Hall. 62

[17] Techy, L., and Woolsey, C. A., 2009. "Minimum-time path planning for unmanned aerial vehicles in steady uniform winds." *Journal of Guidance, Control, and Dyanmics,* **32(6)**, pp. 1736–1746. 107

APPENDIX A. DUBINS AIRPLANE PATH MANAGEMENT ALGORITHMS

In this section we present algorithms that were developed for use with the path planning and path following of the time-optimal Dubins airplane paths which were presented in Chapter 4. This is an extension of the algorithms presented in [2] from two dimensions to three dimensions.

A.1 Path Definition Parameters

We now present an algorithm, Algorithm 4, which can be used to produce the parameters used to describe and implement the time-optimal Dubins airplane paths. This method relies the following definitions:

$$\langle \vartheta \rangle \triangleq \vartheta \mod 2\pi, \tag{A.1}$$

$$H_{12} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{A.2}$$

and

$$H_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{A.3}$$

In the following equations we use numbers in the subscripts of a vector to reference a single element of the vector. The vector subscripts denote whether the helix is right ($\mathbf{c}_r$) or left ($\mathbf{c}_l$) orbit and a start ($\mathbf{c}_s$) or end orbit ($\mathbf{c}_e$). For example,

$$\mathbf{c}_{rs} \triangleq \begin{pmatrix} c_{rs1} \\ c_{rs2} \\ c_{rs3} \end{pmatrix}. \tag{A.4}$$

This notation was adopted for the sake of simplicity in the following equations. A number of equations are needed to compute the lengths of the path segments. The equations for the right, straight, right case are

$$\theta_2 = \arctan 2(c_{re2} - c_{rs2}, c_{re1} - c_{rs1}) \tag{A.5}$$

$$L_{1c1} = R\langle 2\pi + \langle \theta_1 - \frac{\pi}{2} \rangle - \langle \chi_s - \frac{\pi}{2} \rangle \rangle \tag{A.6}$$

$$L_{1d} = ||\mathbf{c}_{re} - \mathbf{c}_{rs}|| \tag{A.7}$$

$$L_{1c2} = R\langle 2\pi + \langle \chi_e - \frac{\pi}{2} \rangle - \langle \theta_1 - \frac{\pi}{2} \rangle \rangle \tag{A.8}$$

$$L_1 = L_{1c1} + L_{1d} + L_{1c2}. \tag{A.9}$$

The equations for the right, straight, left case are

$$\ell = ||H_{12}(\mathbf{c}_{le} - \mathbf{c}_{rs})|| \tag{A.10}$$

$$\theta_2 = \arctan 2(c_{le2} - c_{rs2}, c_{le1} - c_{rs1}) \tag{A.11}$$

$$\theta_{22} = \theta_2 - \cos^{-1} \frac{2R}{\ell} \tag{A.12}$$

$$L_{2c1} = R\langle 2\pi + \langle \theta_2 2 \rangle - \langle \chi_s - \frac{\pi}{2} \rangle \rangle \tag{A.13}$$

$$L_{2d} = \sqrt{\ell^2 - 4R^2} \tag{A.14}$$

$$L_{2c2} = R\langle 2\pi + \langle \theta_{22} + \pi \rangle - \langle \chi_e + \frac{\pi}{2} \rangle \rangle \tag{A.15}$$

$$L_2 = L_{2c1} + L_{2d} + L_{2c2}. \tag{A.16}$$

The equations for the left, straight, right case are

$$\ell = ||H_{12}(\mathbf{c}_{re} - \mathbf{c}_{ls})|| \tag{A.17}$$

$$\theta_3 = \arctan 2(c_{re2} - c_{ls2}, c_{re1} - c_{ls1}) \tag{A.18}$$

$$\theta_{32} = \theta_3 - \cos^{-1} \frac{2R}{\ell} \tag{A.19}$$

$$L_{3c1} = R\langle 2\pi + \langle \chi_s + \frac{\pi}{2} \rangle - \langle \theta_3 + \theta_{32} \rangle \rangle \tag{A.20}$$

$$L_{3d} = \sqrt{\ell^2 - 4R^2} \tag{A.21}$$

$$L_{3c2} = R\langle 2\pi + \langle \chi_e - \frac{\pi}{2} \rangle - \langle \theta_3 + \theta_{32} - \pi \rangle \rangle \tag{A.22}$$

$$L_3 = L_{3c1} + L_{3d} + L_{3c2}. \tag{A.23}$$

The equations for the left, straight, left case are

$$\theta_4 = \arctan 2(c_{le2} - c_{ls2}, c_{le1} - c_{ls1}) \tag{A.24}$$

$$L_{4c1} = R\langle 2\pi + \langle \chi_s + \frac{\pi}{2} \rangle - \langle \theta_4 + \frac{\pi}{2} \rangle \rangle \tag{A.25}$$

$$L_{4d} = ||\mathbf{c}_{le} - \mathbf{c}_{ls}|| \tag{A.26}$$

$$L_{4c2} = R\langle 2\pi + \langle \theta_4 + \frac{\pi}{2} \rangle - \langle \chi_e + \frac{\pi}{2} \rangle \rangle \tag{A.27}$$

$$L_1 = L_{1c1} + L_{1d} + L_{1c2}. \tag{A.28}$$

With these equations and definitions we now present Algorithm 4 for calculating the Dubins airplane path parameters.

Algorithm 4 Find Dubins Airplane Path Parameters:
$(L, \mathbf{c}_s, \lambda_s, \mathbf{c}_e, \lambda_e, \mathbf{w}_1, \mathbf{q}_1, q_{13}, \mathbf{w}_2, \mathbf{w}_3, \mathbf{q}_3) = find3dDubinsParameters(\mathbf{p}_s, \chi_s, \mathbf{p}_e, \chi_e, R)$

Input: Start configuration $(\mathbf{p}_s, \chi_s)$, End configuration $(\mathbf{p}_e, \chi_e)$, Radius R)

$H_{12} = \text{diag}(1,1,0), H_3 = \text{diag}(0,0,1)$
$\mathbf{e}_1 \leftarrow (1,0,0)^T, \mathbf{e}_3 \leftarrow (0,0,1)^T$
$\mathbf{c}_{rs} \leftarrow \mathbf{p}_s + R\mathcal{R}_z(\frac{\pi}{2})(\cos\chi_s, \sin\chi_s, 0)^T, \mathbf{c}_{ls} \leftarrow \mathbf{p}_s + R\mathcal{R}_z(-\frac{\pi}{2})(\cos\chi_s, \sin\chi_s, 0)^T$
$\mathbf{c}_{re} \leftarrow \mathbf{p}_e + R\mathcal{R}_z(\frac{\pi}{2})(\cos\chi_s, \sin\chi_s, 0)^T, \mathbf{c}_{ls} \leftarrow \mathbf{p}_e + R\mathcal{R}_z(-\frac{\pi}{2})(\cos\chi_s, \sin\chi_s, 0)^T$
Compute Equations (A.5) - (A.28)
if arg min$\{L_1, L_2, L_3, L_4\}$ = 1 **then**
    $L \leftarrow L_1, \mathbf{c}_s \leftarrow \mathbf{c}_{rs}, \lambda_s \leftarrow +1, \mathbf{c}_e \leftarrow \mathbf{c}_{re} - q_{13}L_{c2}\mathbf{e}_3, \lambda_e \leftarrow +1$
    $\mathbf{q}_1 \leftarrow \frac{H_{12}(\mathbf{c}_e - \mathbf{c}_s)}{||H_{12}(\mathbf{c}_e - \mathbf{c}_s)||}, q_{13} \leftarrow \frac{(p_{e3} - p_{s3}}{L}, \mathbf{w}_1 \leftarrow \mathbf{c}_s + R\mathcal{R}_z(-\frac{\pi}{2})\mathbf{q}_1 + L_{c1}q_{13}\mathbf{e}_3, \mathbf{w}_2 \leftarrow \mathbf{c}_e + R\mathcal{R}_z(-\frac{\pi}{2})\mathbf{q}_2$
else if arg min$\{L_1, L_2, L_3, L_4\}$ = 2 **then**
    $L \leftarrow L_2, \mathbf{c}_s \leftarrow \mathbf{c}_{rs}, \lambda_s \leftarrow +1, \mathbf{c}_e \leftarrow \mathbf{c}_{le} - L_{c2}q_{13}\mathbf{e}_3, \lambda_e \leftarrow -1$
    $\ell \leftarrow ||\mathbf{c}_e - \mathbf{c}_s||, \vartheta \leftarrow \text{angle}(\mathbf{c}_e - \mathbf{c}_s), \vartheta_2 \leftarrow \vartheta - \frac{\pi}{2} + \sin^{-1}\frac{2R}{\ell}$
    $\mathbf{q}_1 \leftarrow \mathcal{R}_z(\vartheta_2 + \frac{\pi}{2})\mathbf{e}_1, q_{13} \leftarrow \frac{p_{e3} - p_{s3}}{L}, \mathbf{w}_1 \leftarrow \mathbf{c}_s + R\mathcal{R}_z(\vartheta_2)\mathbf{e}_1 + L_{c1}q_{13}\mathbf{e}_3, \mathbf{w}_2 \leftarrow \mathbf{c}_s + R\mathcal{R}_z(\vartheta_2 + \pi)\mathbf{e}_1$
else if arg min$\{L_1, L_2, L_3, L_4\}$ = 3 **then**
    $L \leftarrow L_3, \mathbf{c}_s \leftarrow \mathbf{c}_{ls}, \lambda_s \leftarrow -1, \mathbf{c}_e \leftarrow \mathbf{c}_{re} - L_{c2}q_{13}\mathbf{e}_3, \lambda_e \leftarrow 1$
    $\ell \leftarrow ||\mathbf{c}_e - \mathbf{c}_s||, \vartheta \leftarrow \text{angle}(\mathbf{c}_e - \mathbf{c}_s), \vartheta_2 \leftarrow \cos^{-1}\frac{2R}{\ell}$
    $\mathbf{q}_1 \leftarrow \mathcal{R}_z(\vartheta + \vartheta_2 - \frac{\pi}{2})\mathbf{e}_1, q_{13} \leftarrow \frac{p_{e3} - p_{s3}}{L}, \mathbf{w}_1 \leftarrow \mathbf{c}_s + R\mathcal{R}_z(\vartheta + \vartheta_2)\mathbf{e}_1 + L_{c1}q_{13}\mathbf{e}_3, \mathbf{w}_2 \leftarrow \mathbf{c}_e + R\mathcal{R}_z(\vartheta + \vartheta_2 - \pi)\mathbf{e}_1,$
else if arg min$\{L_1, L_2, L_3, L_4\}$ = 4 **then**
    $L \leftarrow L_4, \mathbf{c}_s \leftarrow \mathbf{c}_{ls}, \lambda_s \leftarrow -1, \mathbf{c}_e \leftarrow \mathbf{c}_{le} - L_{c2}q_{13}\mathbf{e}_e, \lambda_e \leftarrow -1$
    $\mathbf{q}_1 \leftarrow \frac{\mathbf{c}_e - \mathbf{c}_s}{||\mathbf{c}_e - \mathbf{c}_s||}, q_{13} \leftarrow \frac{p_{e3} - p_{s3}}{L}, \mathbf{w}_1 \leftarrow \mathbf{c}_s + R\mathcal{R}_z(\frac{\pi}{2})\mathbf{q}_1 + L_{c1}q_{13}\mathbf{e}_3, \mathbf{w}_2 \leftarrow \mathbf{c}_e + R\mathcal{R}_z(\frac{\pi}{2})\mathbf{q}_2$
end if
$\mathbf{w}_3 \leftarrow \mathbf{p}_e$
$\mathbf{q}_3 \leftarrow \mathcal{R}_z(\chi_e)\mathbf{e}_1$

A.2 Path Manager

In this section we present the path manager in Algorithm A.2.1. This function determines which leg of the Dubins airplane path the vehicle is currently on and sets the appropriate parameters for use with the vector field path following functions. The vector field path following functions which are called in this algorithm can be produced using (3.25), (4.17), and (4.13). Within the path following functions the path parameters and position vector need to be converted into the appropriate vector field frames.

Algorithm A.2.1 Dubins Airplane Path Manager:
$(V_a^c, \chi^c, \gamma^c, \phi_{nom}) = followDubinsAirplanePath(\mathbf{p}_s, \chi_s, \mathbf{p}_e, \chi_e, \mathbf{p}, R, a, b, \lambda_{ellipse}, \mathbf{c}_{ellipse}, \psi_1, \theta, \psi_2)$

Input: Start configuration $(\mathbf{p}_s, \chi_s)$, End configuration $(\mathbf{p}_e, \chi_e)$, Radius R, Current position $\mathbf{p}$

if New path configuration received **then**
  Initialized state machine: state $\leftarrow 1$
  First time in state boolean: firstTimeInState $\leftarrow 1$
end if
$(L, \mathbf{c}_s, \lambda_s, \mathbf{w}_1, \mathbf{q}_1, q_{13}, \mathbf{w}_2, \mathbf{w}_3, textbf{q}_3) \leftarrow find3dDubinsParameters(\mathbf{p}_s, \chi_s, \mathbf{p}_e, \chi_e, R)$
if state = 1 **then**
  $\mathbf{c} \leftarrow \mathbf{c}_s, \rho \leftarrow R, \lambda \leftarrow \lambda_s$
  $\mathbf{q} \leftarrow (L\mathbf{q}_1 + \mathbf{e}_3(p_{e3} - p_{s3}))/||L\mathbf{q}_1 + \mathbf{e}_3(p_{e3} - p_{s3})||$
  if $(\mathbf{p} - \mathbf{w}_1)^T \mathbf{q}_1 \geq 0$ and firstTimeInState = 1 **then**
    state $\leftarrow 2$, firstTimeInState $\leftarrow 1$
  else if $(\mathbf{p} - \mathbf{w}_1)^T \mathbf{q}_1 \geq 0$ **then**
    state $\leftarrow 3$, firstTimeInState $\leftarrow 1$
  else
    firstTimeInState $\leftarrow 0$
  end if
  $(V_a^c, \chi^c, \gamma^c, \phi_{nom}) \leftarrow helixVectorFieldEquation(\mathbf{p}, \mathbf{c}, \rho, \lambda, \mathbf{q})$
else if state = 2 **then**
  $\mathbf{c} \leftarrow \mathbf{c}_s, \rho \leftarrow R, \lambda \leftarrow \lambda_s$
  $\mathbf{q} \leftarrow (L\mathbf{q}_1 + \mathbf{e}_3(p_{e3} - p_{s3}))/||L\mathbf{q}_1 + \mathbf{e}_3(p_{e3} - p_{s3})||$
  if $(\mathbf{p} - \mathbf{w}_1)^T \mathbf{q}_1 < 0$ **then**
    state $\leftarrow 1$, firstTimeInState $\leftarrow 1$
  else
    firstTimeInState $\leftarrow 0$
  end if
  $(V_a^c, \chi^c, \gamma^c, \phi_{nom}) \leftarrow helixVectorFieldEquation(\mathbf{p}, \mathbf{c}, \rho, \lambda, \mathbf{q})$
else if state = 3 **then**
  $\mathbf{c} \leftarrow \mathbf{0}, \mathbf{r} \leftarrow \mathbf{w}_1, \lambda \leftarrow 1$
  $\mathbf{q} \leftarrow (L\mathbf{q}_1 + \mathbf{e}_3(p_{e3} - p_{s3}))/||L\mathbf{q}_1 + \mathbf{e}_3(p_{e3} - p_{s3})||$
  if $(\mathbf{p} - \mathbf{w}_2)^T \mathbf{q}_1 \geq 0$ **then**
    state = 4, firstTimeinState $\leftarrow 1$
  else
    firstTimeInState $\leftarrow 0$
  end if

  $(V_a^c, \chi^c, \gamma^c, \phi_{nom}) \leftarrow inclinedLineVectorFieldEquation(\mathbf{p}, \mathbf{r}, \mathbf{q}, \lambda)$
else if state = 4 **then**
  $\mathbf{c} \leftarrow \mathbf{c}_e, \rho \leftarrow R, \lambda \leftarrow \lambda_e$
  $\mathbf{q} \leftarrow (L\mathbf{q}_3 + \mathbf{e}_3(p_{e3} - p_{s3}))/||L\mathbf{q}_1 + \mathbf{e}_3(p_{e3} - p_{s3})||$
  if $(\mathbf{p} - \mathbf{w}_3)^T \mathbf{q}_3 \geq 0$ and firstTimeInState = 1 **then**
    state $\leftarrow 5$, firstTimeInState $\leftarrow 1$
  else if $(\mathbf{p} - \mathbf{w}_3)^T \mathbf{q}_3 \geq 0$ **then**

42: state $\leftarrow 6$, firstTimeInState $\leftarrow 1$
43: **else**
44: firstTimeInState $\leftarrow 0$
45: **end if**
46: $(V_a^c, \chi^c, \gamma^c, \phi_{nom}) \leftarrow helixVectorFieldEquation(\mathbf{p}, \mathbf{c}, \rho, \lambda, \mathbf{q})$
47: **else if** state = 5 **then**
48: $\mathbf{c} \leftarrow \mathbf{c}_e, \rho \leftarrow R, \lambda \leftarrow \lambda_s$
49: $\mathbf{q} \leftarrow (L\mathbf{q}_1 + \mathbf{e}_3(p_{e3} - p_{s3}))/||L\mathbf{q}_1 + \mathbf{e}_3(p_{e3} - p_{s3})||$
50: **if** $(\mathbf{p} - \mathbf{w}_3)^T \mathbf{q}_3 < 0$ **then**
51: state $\leftarrow 4$, firstTimeInState $\leftarrow 1$
52: **else**
53: firstTimeInState $\leftarrow 0$
54: **end if**
55: $(V_a^c, \chi^c, \gamma^c, \phi_{nom}) \leftarrow helixVectorFieldEquation(\mathbf{p}, \mathbf{c}, \rho, \lambda, \mathbf{q})$
56: **else if** state = 6 **then**
57: $(V_a^c, \chi^c, \gamma^c, \phi_{nom}) \leftarrow ellipseVectorFieldEquation(\mathbf{p}, a, b, \psi_1, \theta, \psi_2, \mathbf{c}_{ellipse}, \lambda_{ellipse})$
58: **end if**

APPENDIX B. SIMULATION MODELS

The nonlinear equations of motions used to produce the simulation results in Chapters 3 and 4 were adopted from [2]. We do not discuss these equations in detail but provide them to illustrate the models that were used to produce the simulation results. We also present the simulation parameters that were used in conjunction with the equations of motion.

B.1 Summary of Nonlinear Equations of Motion

The six-degree-of-freedom, 12-state equations of motion with quasi-linear aerodynamic and propulsion models are presented in this section. The drag terms were modeled as nonlinear function of the angle of attack, and the propeller thrust was modeled as a nonlinear function of the throttle command. Incorporating the aerodynamic and propulsion models described in [2] we get the following equations of motion:

$$\dot{p}_n = (\cos\theta\cos\psi)u + (\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi)v + (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)w \quad \text{(B.1)}$$

$$\dot{p}_e = (\cos\theta\sin\psi)u + (\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi)v + (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)w \quad \text{(B.2)}$$

$$\dot{h} = u\sin\theta - v\sin\phi\cos\theta - w\cos\phi\cos\theta \quad \text{(B.3)}$$

$$\dot{u} = rv - qw - g\sin\theta + \frac{\rho V_a^2 S}{2\text{m}}\left[C_X(\alpha) + C_{X_q}(\alpha)\frac{cq}{2V_a} + C_{X_{\delta_e}}(\alpha)\delta_e\right] + \frac{\rho S_{\text{prop}} C_{\text{prop}}}{2\text{m}}\left[(k_{\text{motor}}\delta_t)^2 - V_a^2\right] \quad \text{(B.4)}$$

$$\dot{v} = pw - ru + g\cos\theta\sin\phi + \frac{\rho V_a^2 S}{2\text{m}}\left[C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{bp}{2V_a} + C_{Y_r}\frac{br}{2V_a} + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r\right] \quad \text{(B.5)}$$

$$\dot{w} = qu - pv + g\cos\theta\cos\phi + \frac{\rho V_a^2 S}{2\text{m}}\left[C_Z(\alpha) + C_{Z_q}(\alpha)\frac{cq}{2V_a} + C_{Z_{\delta_e}}(\alpha)\delta_e\right] \quad \text{(B.6)}$$

$$\dot{\phi} = p + q\sin\phi\tan\theta + r\cos\phi\tan\theta \tag{B.7}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{B.8}$$

$$\dot{\psi} = q\sin\phi\sec\theta + r\cos\phi\sec\theta \tag{B.9}$$

$$\dot{p} = \Gamma_1 pq - \Gamma_2 qr + \frac{1}{2}\rho V_a^2 Sb\left[C_{p_0} + C_{p_\beta}\beta + C_{p_p}\frac{bp}{2V_a} + C_{p_r}\frac{br}{2V_a} + C_{p_{\delta_a}}\delta_a + C_{p_{\delta_r}}\delta_r\right] \tag{B.10}$$

$$\dot{q} = \Gamma_5 pr - \Gamma_6(p^2 - r^2) + \frac{\rho V_a^2 Sc}{2J_y}\left[C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{cq}{2V_a} + C_{m_{\delta_e}}\delta_e\right] \tag{B.11}$$

$$\dot{r} = \Gamma_7 pq - \Gamma_1 qr + \frac{1}{2}\rho V_a^2 Sb\left[C_{r_0} + C_{r_\beta}\beta + C_{r_p}\frac{bp}{2V_a} + C_{r_r}\frac{br}{2V_a} + C_{r_{\delta_a}}\delta_a + C_{r_{\delta_r}}\delta_r\right], \tag{B.12}$$

where the inertia parameters are specified by

$$\begin{aligned}
\Gamma &\triangleq J_x J_z - J_{xz}^2 \\
\Gamma_1 &= \frac{J_{xz}(J_x - J_y + J_z)}{\Gamma} \\
\Gamma_2 &= \frac{J_z(J_z - J_y) + J_{xz}^2}{\Gamma} \\
\Gamma_3 &= \frac{J_z}{\Gamma} \\
\Gamma_4 &= \frac{J_{xz}}{\Gamma} \\
\Gamma_5 &= \frac{J_z - J_x}{J_y} \\
\Gamma_6 &= \frac{J_{xz}}{J_y} \\
\Gamma_7 &= \frac{(J_x - J_y)J_x + J_{xz}^2}{\Gamma} \\
\Gamma_8 &= \frac{J_x}{\Gamma},
\end{aligned} \tag{B.13}$$

where $h = -p_d$ is the altitude, and

$$
\begin{aligned}
C_{p_0} &= \Gamma_3 C_{l_0} + \Gamma_4 C_{n_0} \\
C_{p_\beta} &= \Gamma_3 C_{l_\beta} + \Gamma_4 C_{n_\beta} \\
C_{p_p} &= \Gamma_3 C_{l_p} + \Gamma_4 C_{n_p} \\
C_{p_r} &= \Gamma_3 C_{l_r} + \Gamma_4 C_{n_r} \\
C_{p_{\delta_a}} &= \Gamma_3 C_{l_{\delta_a}} + \Gamma_4 C_{n_{\delta_a}} \\
C_{p_{\delta_r}} &= \Gamma_3 C_{l_{\delta_r}} + \Gamma_4 C_{n_{\delta_r}} \\
C_{r_0} &= \Gamma_4 C_{l_0} + \Gamma_8 C_{n_0} \\
C_{r_\beta} &= \Gamma_4 C_{l_\beta} + \Gamma_8 C_{n_\beta} \\
C_{r_p} &= \Gamma_4 C_{l_p} + \Gamma_8 C_{n_p} \\
C_{r_r} &= \Gamma_4 C_{l_r} + \Gamma_8 C_{n_r} \\
C_{r_{\delta_a}} &= \Gamma_4 C_{l_{\delta_a}} + \Gamma_8 C_{n_{\delta_a}} \\
C_{r_{\delta_r}} &= \Gamma_4 C_{l_{\delta_r}} + \Gamma_8 C_{n_{\delta_r}}.
\end{aligned}
$$

The aerodynamic force coefficients in the X and Z directions are nonlinear functions of the angle of attack (α). They can be stated as

$$
\begin{aligned}
C_X(\alpha) &\stackrel{\triangle}{=} -C_D(\alpha)\cos\alpha + C_L(\alpha)\sin\alpha \\
C_{X_q}(\alpha) &\stackrel{\triangle}{=} -C_{D_q}\cos\alpha + C_{L_q}\sin\alpha \\
C_{X_{\delta_e}}(\alpha) &\stackrel{\triangle}{=} -C_{D_{\delta_e}}\cos\alpha + C_{L_{\delta_e}}\sin\alpha \\
C_Z(\alpha) &\stackrel{\triangle}{=} -C_D(\alpha)\sin\alpha - C_L(\alpha)\cos\alpha \\
C_{Z_q}(\alpha) &\stackrel{\triangle}{=} -C_{D_q}\sin\alpha - C_{L_q}\cos\alpha \\
C_{Z_{\delta_e}}(\alpha) &\stackrel{\triangle}{=} -C_{D_{\delta_e}}\sin\alpha - C_{L_{\delta_e}}\cos\alpha.
\end{aligned}
$$

The effects of stall into the lift coefficient were modeled as

$$C_L(\alpha) = (1-\sigma(\alpha))\left[C_{L_0} + C_{L_\alpha}\alpha\right] + \sigma(\alpha)\left[2\text{sign}(\alpha)\sin^2\alpha\cos\alpha\right],$$

where

$$\sigma(\alpha) = \frac{1 + e^{-M(\alpha-\alpha_0)} + e^{M(\alpha+\alpha_0)}}{\left(1+e^{-M(\alpha-\alpha_0)}\right)\left(1+e^{M(\alpha+\alpha_0)}\right)},$$

and M and α_0 are positive constants. Drag was modeled as a nonlinear quadratic function of lift as

$$C_D(\alpha) = C_{D_p} + \frac{(C_{L_0} + C_{L_\alpha}\alpha)^2}{\pi e AR},$$

where e is the Oswald efficiency factor and AR is the aspect ratio of the wing.

The equations provided in this section completely describe the dynamic behavior of a MAV in response to inputs from the throttle and the aerodynamic control surfaces (ailerons, elevator, and rudder). These equations are the core of the MAV simulation environment that was used to produce the results of Chapters 3 and 4.

B.2 Zagi Flying Wing Simulation Vehicle Parameters

The vehicle parameters used in the simulation are presented in Table B.1. These parameters were adopted from [2] and were developed to provide a reasonable model for the MAV zagi flying wing that was used to produce the experimental results in Chapter 6.

Table B.1: Parameters for a Zagi flying wing.

Parameter	**Value**	**Longitudinal Coef.**	**Value**	**Lateral Coef.**	**Value**
m	1.56 kg	C_{L_0}	0.09167	C_{Y_0}	0
J_x	0.1147 kg m^2	C_{D_0}	0.01631	C_{l_0}	0
J_y	0.0576 kg m^2	C_{m_0}	-0.02338	C_{n_0}	0
J_z	0.1712 kg m^2	C_{L_α}	3.5016	C_{Y_β}	-0.07359
J_{xz}	0.0015 kg m^2	C_{D_α}	0.2108	C_{l_β}	-0.02854
S	0.2589 m^2	C_{m_α}	-0.5675	C_{n_β}	-0.00040
b	1.4224 m	C_{L_q}	2.8932	C_{Y_p}	0
c	0.3302 m	C_{D_q}	0	C_{l_p}	-0.3209
S_{prop}	0.0314 m^2	C_{m_q}	-1.3990	C_{n_p}	-0.01297
ρ	1.2682 kg/m^3	$C_{L_{\delta_e}}$	0.2724	C_{Y_r}	0
k_{motor}	20	$C_{D_{\delta_e}}$	0.3045	C_{l_r}	0.03066
k_{T_p}	0	$C_{m_{\delta_e}}$	-0.3254	C_{n_r}	-0.00434
k_Ω	0	C_{prop}	1.0	$C_{Y_{\delta_a}}$	0
e	0.9	M	50	$C_{l_{\delta_a}}$	0.1682
		α_0	0.4712	$C_{n_{\delta_a}}$	-0.00328
		ε	0.1592		
		C_{D_p}	0.0254		