Remote Terrain Navigation for Unmanned Air Vehicles

Stephen R. Griffiths
Brigham Young University - Provo

Follow this and additional works at: https://scholarsarchive.byu.edu/etd
Part of the Mechanical Engineering Commons

BYU ScholarsArchive Citation
https://scholarsarchive.byu.edu/etd/1059

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
REMOTE TERRAIN NAVIGATION FOR
UNMANNED AIR VEHICLES

by

Stephen Richard Griffiths

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

Master of Science

Department of Mechanical Engineering
Brigham Young University
April 2006
This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date

Timothy W. McLain, Chair

Date

Randal W. Beard

Date

Deryl O. Snyder
As chair of the candidate’s graduate committee, I have read the thesis of Stephen Richard Griffiths in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Timothy W. McLain
Chair, Graduate Committee

Accepted for the Department

Matthew R. Jones
Graduate Coordinator

Accepted for the College

Alan R. Parkinson
Dean, Ira A. Fulton College of Engineering and Technology
ABSTRACT

REMOTE TERRAIN NAVIGATION FOR
UNMANNED AIR VEHICLES

Stephen Richard Griffiths
Department of Mechanical Engineering
Master of Science

There are many applications for which small unmanned aerial vehicles (SUAVs) are well suited, including surveillance, reconnaissance, search and rescue, convoy support, and short-range low-altitude perimeter patrol missions. As technologies for microcontrollers and small sensors have improved, so have the capabilities of SUAVs. These improvements in SUAV performance increase the possibility for hazardous missions through mountainous and urban terrain in the successful completion of many of these missions.

The focus of this research was on remote terrain navigation and the issues faced when dealing with limited onboard processing and limited payload and power capabilities. Additional challenges associated with canyon and urban navigation missions included reactive path following, sensor noise, and flight test design and execution. The main challenge was for an SUAV to successfully navigate through a mountainous canyon by reactively altering its own preplanned path to avoid canyon walls and other stationary obstacles.

A robust path following method for SUAVs that uses a vector field approach to track functionally curved paths is presented along with flight test results. In these results,
the average tracking error for an SUAV following a variety of curved paths is 3.4 m for amplitudes ranging between 10 and 100 m and spatial periods between 125 and 500 m. Additionally, a reactive path following method is presented that allows a UAV to continually offset or bias its planned path as distance information from the left and right ranging sensors is computed. This allows the UAV to center itself between potential hazards even with imperfect waypoint path planning. Flight results of an SUAV reactively navigating through mountainous canyons experimentally verify the feasibility of this approach. In a flight test through Goshen Canyon in central Utah, an SUAV biased its planned path by 3 to 10 m to the right as it flew to center itself through the canyon and avoid the possibility of crashing into a canyon wall.
ACKNOWLEDGMENTS

I would like to thank Professors Tim McLain, Randy Beard, and Deryl Snyder for all their help in conducting this research. I also need to thank all members of the MAGICC Lab, especially those that helped in the collection of flight data, the development of the navigational methods, the design and construction of optic flow sensors, and with the construction of the UAV sensor platform including Blake Barber, Dave Johansen, Derek Nelson, Derek Kingston, Andrew Eldredge, Jon Ostler, Joseph Jackson, Nathan Knoebel, Andrew Curtis, Steve Osborne, and Neil Johnson. I am also extremely grateful to my loving wife, Anna, for her tireless support.
# Contents

## Acknowledgments

## List of Tables

## List of Figures

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Background</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Motivation</td>
<td>1</td>
</tr>
<tr>
<td>1.3</td>
<td>Literature Review</td>
<td>2</td>
</tr>
<tr>
<td>1.4</td>
<td>Contributions</td>
<td>4</td>
</tr>
<tr>
<td>1.5</td>
<td>Document Organization</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Experimental Apparatus</td>
<td>7</td>
</tr>
<tr>
<td>2.1</td>
<td>UAV Simulation Environment</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>Flight Test Hardware</td>
<td>8</td>
</tr>
<tr>
<td>2.3</td>
<td>Optic Flow Based Ranging</td>
<td>9</td>
</tr>
<tr>
<td>2.4</td>
<td>Preliminary Optic Flow Sensor</td>
<td>12</td>
</tr>
<tr>
<td>2.5</td>
<td>Lens Selection</td>
<td>12</td>
</tr>
<tr>
<td>2.6</td>
<td>Focusing Procedure</td>
<td>14</td>
</tr>
<tr>
<td>2.7</td>
<td>Sample Rate Selection</td>
<td>16</td>
</tr>
<tr>
<td>2.8</td>
<td>Optic Flow Ranging Results</td>
<td>17</td>
</tr>
<tr>
<td>2.9</td>
<td>Sensor Orientation</td>
<td>19</td>
</tr>
<tr>
<td>2.10</td>
<td>New Prototype Optic Flow Ranging Sensor</td>
<td>20</td>
</tr>
<tr>
<td>2.11</td>
<td>Summary</td>
<td>22</td>
</tr>
</tbody>
</table>
## Reactive Terrain Navigation

3.1 Introduction ........................................ 23
3.2 Reactive Terrain Navigation Using Path Biasing ............... 25
3.3 Path Biasing Implementation .................................. 26
3.4 Simulation Results ........................................ 28
3.5 Flight Test Results ........................................ 29
   3.5.1 Tree Avoidance ........................................ 29
   3.5.2 Canyon Navigation ...................................... 30

## Curved Path Following

4.1 Introduction ........................................ 35
   4.1.1 Overview ........................................ 35
   4.1.2 Literature Review ...................................... 36
4.2 Problem Description ....................................... 37
4.3 Technical Approach ....................................... 38
   4.3.1 Control Approach ...................................... 39
   4.3.2 Implementation Issues .................................. 44
   4.3.3 Effects of Model Error .................................. 47
   4.3.4 Bounding Following Error ............................... 49
4.4 Flight Test Results ....................................... 51
4.5 Conclusions ........................................ 53

## Conclusions and Future Work

5.1 Optic Flow Ranging ....................................... 57
5.2 Reactive Terrain Navigation .................................. 58
5.3 Curved Path Following ..................................... 58

Bibliography ........................................ 60
List of Tables

3.1 Steps of Path Offsetting Method ............................................ 27
List of Figures

2.1 Screenshots show the UAV simulation test environment .......... 7
2.2 Main elements of SUAV testbed .................................... 8
2.3 The bottom side of the custom designed UAV platform ........... 9
2.4 Graphical interpretation of values used to compute distance from optic flow ........................................ 11
2.5 Agilent ADNS-2610 optic flow sensor ................................ 13
2.6 Optic flow sensors with 1.2, 2.5, and 6.5 degree fov optics configurations ................................................. 14
2.7 Focusing board used to adjust the lens until the images were in focus ..................................................... 15
2.8 Resolution as function of distance .................................... 16
2.9 Flight test results using using a 1.2 degree fov lens and fixed curve resolution ............................................. 18
2.10 Optic flow range data compared to laser rangefinder data .......... 19
2.11 Lateral sensor range vs. sensor orientation angle α .................. 20
2.12 Circuit diagram for new optic flow sensor board ................... 21
2.13 New optic-flow sensor board layout .................................. 21
2.14 Custom designed and ranging sensors ................................ 22
3.1 Straight line vector field ................................................. 24
3.2 Flight results demonstrating straight line and circular path following ......................................................... 24
3.3 Path biasing diagram .................................................... 25
3.4 Continuous path biasing diagram ...................................... 26
3.5 Examples of path centering ............................................. 27
3.6 Screen shot from a terrain navigation simulation using path offsetting ..................................................... 28
3.7 Path offset simulation results ........................................... 29
3.8 SUAV used for tree avoidance tests ................................... 30
3.9 Photos of tree avoidance site and SUAV flight tests ................ 31
3.10 Results from tree avoidance flight tests ............................... 31
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.11</td>
<td>Photo of the research SUAV navigating through Goshen Canyon</td>
<td>32</td>
</tr>
<tr>
<td>3.12</td>
<td>Flight test results through Goshen Canyon</td>
<td>33</td>
</tr>
<tr>
<td>3.13</td>
<td>Path offset results through Goshen Canyon</td>
<td>33</td>
</tr>
<tr>
<td>4.1</td>
<td>Curved path vector field</td>
<td>38</td>
</tr>
<tr>
<td>4.2</td>
<td>Vector field for curved path following</td>
<td>39</td>
</tr>
<tr>
<td>4.3</td>
<td>Contour plot of $\dot{\chi}^p_{\text{max}}$ in deg/s for $V_g = 13$ m/s</td>
<td>46</td>
</tr>
<tr>
<td>4.4</td>
<td>Simulation results of SUAV following an irregular circular path</td>
<td>52</td>
</tr>
<tr>
<td>4.5</td>
<td>Flight test results for curved path following</td>
<td>54</td>
</tr>
<tr>
<td>4.6</td>
<td>Flight results comparing the numerical and analytical results for $\dot{\chi}^p$</td>
<td>55</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background

Small Unmanned Aerial Vehicles (SUAVs)\footnote{Small unmanned aerial vehicles are considered to be those with wingspans between 0.3 m to 2 m, and micro aerial vehicles to be those with wingspans under 0.3 m. Here the abbreviation SUAV denotes \textit{small} unmanned aerial vehicle while the abbreviation MAV denotes \textit{micro} aerial vehicle} have many important uses including forest fire surveillance, civilian search and rescue, border patrol, environmental and traffic monitoring, disaster relief, and military applications. The versatility and low cost of SUAVs provide a potential for a dramatic increase in their use in the public and private sectors. There are several advantages associated with using SUAVs. In addition to being less costly than larger UAVs, they are also more easily transported and can be launched even in rough terrain. In general, they are also less complex and therefore require fewer ground operators. Furthermore, they can usually be deployed more quickly than larger UAV systems. Some disadvantages of SUAVs include range, payload, and power limitations. Also, SUAVs are more adversely affected by inclement weather than large UAVs. As SUAVs overcome these limitations and become more reliable and maneuverable, their missions will become more difficult and require that they fly through more formidable terrain. Collision avoidance systems will need to be in place to achieve many of these missions.

1.2 Motivation

The focus of this research is comprised of five objectives. The first was to attempt to use an inexpensive lightweight optic flow sensor to measure distances off the left and right sides of the SUAV. The second step was to develop a terrain navigation algorithm
that would incorporate these experimental sensors to avoid obstacles even in the presence of imperfect waypoint path planning. In step three, these methods were implemented and tested in a UAV simulation environment. The fourth step was to develop a curved path following method to allow SUAVs to navigate winding canyon passages more adeptly than with the current straight line path follower. The final goal was to flight test the algorithms and sensors in a mountainous canyon.

The scenario envisioned for this technology was for an SUAV to successfully navigate through a mountainous canyon by reactively altering its own preplanned path to avoid canyon walls and other stationary obstacles. The preplanned path traverses the canyon intentionally close to canyon walls and obstacles to simulate path planning errors and to verify that the UAV would alter its waypoint path in order to successfully navigate the terrain.

1.3 Literature Review

Recent research in the area of SUAV collision avoidance has explored such applications as urban terrain navigation [5], formation flying [4], and safe landing area determination and autonomous landings of rotocraft UAVs [6]. Other papers and publications referenced throughout this thesis, provide important knowledge and concepts that were used to develop the collision avoidance methods and path following strategies discussed herein.

One group of researchers investigated the accuracy of gradient-based optic-flow ranging for potential use on UAVs to estimate range to targets. Simulation results demonstrate accurate ranging to objects 3 km away with only 5 percent error. However, the authors concluded that as pitch and yaw rates increase, the ranging accuracy degrades due to poor gradient estimation caused by high inter-frame pixel motion. In addition, as the processing frame rate increases, the variance in the range estimate decreases significantly [13].

Similarly, in [14] a simulated rotary-wing UAV was able to explore an urban-like environment without hitting obstacles by using correlation-based optic flow calculations. It also uses a biologically inspired optic flow balancing strategy to center itself between buildings.
UAV research presented in [8] combines an optic flow balancing approach for side-looking fisheye cameras with forward facing stereo cameras to detect obstacles in front of a rotary-wing UAV. Their results demonstrate that combining forward looking obstacle detection with side looking optic flow allows the UAV to navigate around sharp corners and improves the overall collision avoidance system.

A path planning strategy presented in [12], requires a UAV to fly sinusoidal paths to keep the faster flying UAV at the same forward progress along a road as the slower moving convoy.

In the area of UAV path following, one approach for curved path following of Unmanned Ground Vehicles (UGVs) uses two Fuzzy Logic Controllers (FLCs) [7]. The first FLC is used to navigate straight or slowly-varying curved path segments. The second is used to account for significant path curvature. In experimental results, this method consistently outperformed linear controllers at following circular paths.

A methodology for the designing UAV guidance and control systems is presented in [9]. The authors design the guidance and control simultaneously. This method also addresses the stability of the combined guidance and control system.

A curved path trajectory tracking approach is presented in [17]. Curved segments are followed using a controller derived from a kinematic model with the addition of an anticipatory control element that improves the tracking capability for curved paths. This method accounts for wind disturbances by the addition of an adaptive kinematic factor.

A method for UAVs to follow straight lines and arcs in the presence of wind has already been developed by Nelson, et al. [16]. Their method uses a vector field of course commands to guide the UAV on the waypoint path with minimal overshoot across a wide range of ground speeds. Adapting this vector field method for curved path following could provide a better path following scheme to navigate narrow canyon passages in connection with collision avoidance schemes.

A great deal of previous collision avoidance research was performed on rotary-wing UAVs with simulated and experimental results (e.g. [6, 8, 14]). In general, most of this previous research has required computationally prohibitive algorithms that cannot currently be run onboard fast fixed-wing SUAVs that have limited computational resources. The goal for
This research was to illustrate the feasibility of canyon and terrain navigation in unmanned missions by flight testing the algorithms onboard SUAVs. Inherent in hardware testing are a number of difficult problems including path following, sensor design and fabrication, sensor testing, sensor integration, sensor inaccuracies, construction and maintenance of a flight-worthy SUAV, and algorithm implementation into the autopilot, simulation software, and ground station software.

1.4 Contributions

The research described in this thesis presents six key contributions that extend the capabilities and expand the body of knowledge pertaining to SUAVs. These main contributions include:

1. Development of a method for reactively altering the flight paths of SUAVs to avoid hazards and center themselves between terrain walls
2. Flight test results using experimental optic flow ranging sensors to reactively avoid hazards and to demonstrate the feasibility of the path biasing strategy
3. Extension of the path following scheme for straight lines and orbits to also include vector field path following of arbitrary curved paths
4. Flight tests illustrating the effectiveness of the curved path following method and the feasibility of UAV navigation through extreme mountainous terrain
5. Design and construction of prototype optic flow distance ranging sensors using the Agilent ADNS-2610 optic flow sensor
6. Sensor board design and construction using Eagle Layout editor for a new sensor board for distance ranging that features the more advanced Agilent ADNS-3080 optic flow sensor
1.5 Document Organization

The main body of this thesis is composed of five main chapters including the introduction. Chapter 2 gives an overview of the simulation environment, autopilot, experimental hardware, and the SUAVs used for this research. It also describes the experimental optic flow sensor used for canyon navigation missions, explains how it can be used for ranging, and presents data demonstrating its capabilities. Chapter 3 describes the basic approach to terrain navigation and describes the development of a path biasing algorithm. Also presented are experimental results of SUAVs avoiding obstacles and navigating through terrain using these sensors and algorithms. Chapter 4 presents a new approach to path following which allows SUAVs to follow curved paths. It also includes the derivation, stability proof, keys for implementation, flight test results, and discusses the limitations of this approach. Finally, Chapter 5 includes a summary of conclusions from each chapter and makes recommendations for additional testing and future enhancements for the optic flow sensors and path following schemes discussed in this thesis.
Chapter 2

Experimental Apparatus

An important objective of this research was to flight test the path following and navigational methods developed for SUAVs. In this chapter an overview is given on all software and hardware used throughout this research for experimental testing and software simulations.

2.1 UAV Simulation Environment

Prior to flight testing, all algorithms and path following methods were tested in a computer simulation environment. This environment, developed at BYU, is an open source UAV flight simulation package called Aviones that allows autopilot software to be tested on the ground before it is flight tested. Virtual Cockpit software interfaces with Aviones

![Aviones UAV Simulation Environment](image1.png)

![Virtual Cockpit](image2.png)

(a) Aviones UAV Simulation Environment   (b) Virtual Cockpit

Figure 2.1: Screenshots show the UAV simulation test environment.
allowing the operator to send commands to the virtual UAV and receive telemetry, including the virtual UAVs attitude, location, and other information of interest. The simulation software also allows USGS terrain data to be loaded as a representative three-dimensional model of actual terrain.

2.2 Flight Test Hardware

BYU has also developed a reliable and robust platform for unmanned aerial vehicle testing. Figure 2.2 shows the main elements of the testbed. The first frame shows the ground station components. A laptop runs the Virtual Cockpit software that interfaces through a communication box with the SUAV over a 900 MHz wireless modem. An RC transmitter is used as a stand-by fail-safe mechanism. At any time during flight tests an RC pilot can assume control of the SUAV to help facilitate safe operations.

The second frame in Figure 2.2 shows the airframe used for most of the flight tests reported in this paper. The airframe has a 1.5 m wingspan and was constructed with an EPP foam core covered with Kevlar to stiffen the platform and provide protection to the sensor payload. This design was selected for its durability, payload capacity, ease of component installation, and flight characteristics. The airframe can carry a 12 ounce payload and can remain in flight for over 45 minutes at a time. Embedded in the airframe are the autopilot, 12.6 V lithium-polymer batteries, a 1000 mW, 900 MHz radio modem, a 12 channel GPS receiver, a small analog video camera, and a 1000 mW video transmitter. The collision
avoidance sensors embedded in the airframe include a laser rangefinder, and three optic flow sensors.

The collision avoidance and path follow methods described in this paper were implemented on the Kestrel 1.45 autopilot made by Procerus Technologies. The Kestrel autopilot is equipped with a Rabbit 3100 29 MHz processor, three-axis rate gyros, three-axis accelerometers, and absolute and differential pressure sensors. The autopilot measures $5 \times 10 \times 1$ cm and weighs 36 grams. The autopilot also serves as a data acquisition device and is able to log 175 kbytes of user-selectable telemetry at rates up to 60 Hz. The laser ranger and all three optic flow sensors interface directly with the autopilot and all the algorithms described herein are executed on-board the Rabbit processor.

![Figure 2.3: The bottom side of the custom designed UAV platform.](image)

### 2.3 Optic Flow Based Ranging

To reactively avoid obstacles while flying along a preplanned path, accurate distance information to potential hazards and terrain features is required. Sensors such as RADAR and scanning LADAR are typically too large and heavy for SUAVs. Laser rangefinders have been used on SUAVs for height above ground ranging and forward pointing obstacle detection. However, even simple laser rangefinders require a significant amount of space, weight, and power resources. These sensors can also be very expensive, costing thousands of dollars each.
In addition to size, weight, and power constraints, SUAVs are also limited by the availability of onboard computing resources. For most SUAVs, the primary computational resource is the excess processing capability of the autopilot microcontroller. Additional computational capacity can be added, but many computer systems exceed the payload capacity of SUAVs. For this reason smaller microcontrollers are typically used. This limitation in processing power prohibits the effective use of onboard computer vision methods such as optic flow and feature tracking. In an effort to save money, space, weight, and computational power, optic flow ranging using a compact and commercially available optic mouse sensor was explored as an alternative to laser ranging for use in height above ground (HAG) estimation and reactive terrain navigation on SUAVs.

Research performed by Barrows, et al. in [3] demonstrates that optic mouse sensors can be successfully used to follow terrain in low flying SUAVs. Their work also demonstrates how readings from an optic flow sensor can be used to for distance ranging of HAG [2]. Much of the research presented in this section builds upon their preliminary work.

Optic flow is a passive method used to measure distance by relating the flow of features on an imaging array to the speed the imaging array is moving relative to the surface. The number of pixels that a given object moves in the imaging plane is combined with data from the UAV’s Inertial Measurement Unit (IMU) and GPS to determine HAG according to

\[ h = \frac{V_g T_s}{\tan \left( \frac{\lambda_{\text{eff}}}{2} \right) \cos \theta \cos \phi}, \]  

(2.1)

where \( h \) represents HAG, \( \theta \) is the average pitch of the sensor over the sample period, \( \phi \) is the average roll of the sensor over the sample period, \( T_s \) is sample period, and \( V_g \) is the average groundspeed of the UAV over the sample period as reported by the GPS unit. The product of \( V_g \) and \( T_s \) represents the total distance travel by the UAV over the sample period. The effective field-of-view, \( \lambda_{\text{eff}} \), is given by

\[ \lambda_{\text{eff}} = \frac{\delta p_x \cdot \text{fov}}{p_n} - \dot{\theta} T_s, \]  

(2.2)
where $\delta p_x$ represents the average number of pixels of displacement of features across the imaging plane over the sample time in the $x$ direction as reported by the optic flow sensor, $\text{fov}$ represents the field of view of the sensor, $p_n$ is the number of pixels in the imaging array in the direction of motion of the sensor, and $\dot{\theta}$ is the average pitch rate of the sensor over the sample period.

Equation (2.1) can easily be modified to provide distance measurements to hazards on the left and right sides of the SUAV according to

$$D_{\text{left}} = \frac{V_g T_s}{2 \tan \left( \frac{\delta p_x \text{fov}}{2p_n} - \frac{\psi T_s}{2} \right)} \cos \phi \sin \alpha$$

(2.3)

$$D_{\text{right}} = \frac{V_g T_s}{2 \tan \left( \frac{\delta p_x \text{fov}}{2p_n} + \frac{\psi T_s}{2} \right)} \cos \phi \sin \alpha$$

(2.4)

where $\alpha$ is the angle describing the orientation of the sensor defined from body frame $x$-axis, $\dot{\theta}$ is replace by $\dot{\psi}$ which represents the average yaw rate of the sensor over the sample period and can be obtained by averaging the yaw-rate gyro on the IMU over the sample period. A graphical representation of values used in the computation of distance from optic flow using Equation (2.4) is shown in Figure 2.4.

![Graphical interpretation of values used to compute distance from optic flow.](Image)
2.4 Preliminary Optic Flow Sensor

Distance ranging for terrain navigation was performed using the Agilent ADNS-2610 optic flow sensor shown in Figure 2.5. This sensor can be interfaced with a 3.3 V microcontroller using a non-addressable I²C communication protocol. The ADNS-2610 measures the flow of features across an 18 by 18 pixel CMOS imaging array. It has a small form factor, measuring only 10 mm by 12.5 mm. It computes optic flow at 1500 frames per second (fps) and runs on a 24 MHz clock cycle. In order for it to register optic flow it requires a light intensity of at least 80 mW/m² at a wavelength of 639 nm or 100 mW/m² at a wavelength of 875 nm. The sensor uses a proprietary gradient based method for computing optic flow and outputs two main values, \( \delta p_x \) and \( \delta p_y \) that represent the total optic flow across the sensor’s field of view in both the \( x \) and \( y \) directions. These values are stored in a buffer that accumulates optic flow up to 128 pixels. Once the buffer is full no more optic flow readings are accumulated until the buffer has been read. Reading the buffer clears it and allows it to begin reaccumulating optic flow measurements. The buffer must be read frequently enough to clear it before filling up, at which point all further optic flow data is lost. The flow data in the camera \( y \) direction corresponds to lateral motion of the UAV and can be safely ignored. The flow data in the camera \( x \) direction can be combined with data from the IMU and GPS to determine distances according to Equations (2.1), (2.3), and (2.4). The surface quality (squal) of the image can also be read from the sensor. This represents the number of distinct features that can be distinguished by the sensor in the image. Squal can be used to determine if the surface that appears in the sensor image is sufficiently texture rich to accurately measure optic flow.

2.5 Lens Selection

One consideration in outfitting a UAV with an optic flow sensor is the selection and setup of the optics. Narrow angle lenses increase the distance at which the UAV will be able to pick up appreciable optic flow. This increases the distance at which the sensor will be able to accurately measure distance for a given groundspeed. One drawback of using narrow angle lenses is a significant increase in the length and weight of the sensor. Narrow angle lenses can also decrease the low end range of the sensor by causing overflow in the
optic flow sensor’s $\delta p_x$ register or by causing the flow rate seen by the sensor to become too fast for the sensor to register. Since the sensor only returns a single pixel translation value in the $x$ and $y$ directions, the distance measurement calculated from optic flow will essentially be the average distance of all features in the image over the sample period. Therefore, as the lens fov gets smaller and the amount of viewable surface in the image gets correspondingly smaller, the computed distance will more closely approximate a distance to a single point in space. Wide angle lenses allow for smaller longitudinal sensor size, but require that larger features be available in the environment. They also decrease the distance at which appreciable optic flow will be recorded for a given groundspeed. A particular advantage of wider angle lenses is that they are less susceptible to noise introduced by angular oscillations of the UAV in pitch, roll, and yaw.

Another important consideration when designing the sensor optics is lens diameter. This is especially important with regard to the corresponding f-stop value of the lens. The f-stop is defined as the ratio of the lens focal length to its diameter and is a measure of the amount of light that the optics allow to pass through to the imaging array. Lenses with larger f-stops allow less light to pass through. Keeping in mind that the imaging array requires a light intensity of 80 to 100 W/m$^2$ in the correct spectral range, it is important to select a lens with a low f-stop in order to increase the sensor’s operational range of environmental lighting conditions. Because the sensor automatically adjusts shutter speed to keep the light intensity at an ideal level, the only disadvantage to selecting a lens with a
lower f-stop is increased size. Another important implication of the f-stop is that selecting a lens with an increased focal length (a narrower angle lens) will not only increase the length of the sensor, but also increase the diameter of the sensor if the f-stop is to be maintained.

For the purposes of these tests, lenses were selected which yielded fields of view of 6.5, 2.5, and 1.2 degrees when mounted on the sensor. These lenses had accompanying f-stops of 2.0, 2.0, and 2.5 respectively. Each configuration is shown in Figure 2.6. Comparable results were achieved using both the 2.5 degree and 1.2 degree field-of-view setups. However, the 1.2 degree field-of-view lens offered slightly greater range while performing better over feature-poor surfaces such as roads and sidewalks. The disadvantages of the 1.2 degree field-of-view setup include its susceptibility to noise caused by pitch and roll oscillations, and its increased weight and size.

2.6 Focusing Procedure

Focusing the lens requires mounting the lens such that it is the proper distance from the pinhole of the optic flow sensor. For example, if the selected lens has a 10 mm focal length, then the lens should be mounted 10 mm from the pinhole per the lens manufacturer’s specifications. Proper verification of the focusing of the selected lens is much
more difficult. This is because the sensor only has an 18 by 18 pixel array which makes distinguishing objects difficult regardless of how well the sensor is in focus. To facilitate focusing the sensor, an “image dump” is requested over the serial line. These images can then be displayed on a monitor. The process of sending the image and displaying it is relatively slow, allowing a frame rate of only about 1 fps. A custom focusing board was developed to help focus the lenses as images from the sensors are viewed from the monitor as shown Figure [2.7]. The lens can be adjusted until the simple objects seen on the focusing board are best distinguishable. Then the fov of the lens-sensor combination can be determined according to

$$\text{fov} = 2 \arctan \left( \frac{p_n W_{\text{actual}}}{2DW_{\text{pixels}}} \right)$$

(2.5)

where $D$ is the distance between the pinhole on the sensor and the focusing board in inches, $p_n$ is the width of the CMOS pixel array in pixels, $W_{\text{pixels}}$ is the width of the feature seen in the image in pixels, and $W_{\text{actual}}$ is the measured width of the feature on the focusing
board in inches. Repeating these calculations from different distances \( D \) and averaging them together improves the estimate of the \( \text{fov} \).

### 2.7 Sample Rate Selection

Another important consideration when equipping a UAV with an optic flow sensor is the selection of sample rate. The resolution of the sensor (measured in units of distance per sensor count) as described in [1] is given by

\[
r = \lim_{\delta h \to 0} \left( \frac{\delta h \frac{\text{fov}}{2p_n}}{\tan^{-1}\left( \frac{V_t}{2(h - \delta h)} \right) - \tan^{-1}\left( \frac{V_t}{2h} \right)} \right) \tag{2.6}
\]

where \( V_g \) is groundspeed and \( \delta h \) is a perturbation in the ranged distance. For a fixed field of view, the sensor’s resolution is a function of the sample rate, groundspeed, and the measured distance. Plots of sensor resolution for a fixed field of view and varying sample rates and ground speeds are shown in Figure 2.8.

![Figure 2.8: Resolution as function of distance.](image)

Each sensor count corresponds to a relatively large distance measure for high values of \( r \). Correspondingly, the noise on the sensor measured in sensor counts is amplified by \( r \) in the HAG measurement. Therefore, large values of \( r \) are undesirable because they
can significantly decrease the sensor’s signal to noise ratio. It is possible to decrease $r$ by decreasing the sample rate as shown in Figure 2.8(a). However, this can be undesirable because it decreases the sensor’s response time and can lead to overflow in the sensor’s $\delta p_x$ register. Furthermore, for a fixed sample rate, maintaining a low value of $r$ for large distance values results in unnecessarily low values of $r$ for lower distance values. This means low sample rates are being maintained where significantly faster sample rates could be used without significant loss in signal to noise ratio. This approach results in premature sensor overflow and corresponding loss of low-end range. The solution is to use a dynamic sample rate. This approach maintains an $r$ value along a specified resolution curve regardless of groundspeed of the UAV.

Most UAV flight control systems utilize a feedback loop to control around a commanded airspeed. Even when the feedback loop is closed around groundspeed, the commanded groundspeed may not be realizable due to wind conditions. In the presence of wind, groundspeed may vary significantly depending on the changing nature of the wind and the direction of travel of the UAV with respect to the wind direction. As shown in Figure 2.8(b), variations in groundspeed can alter the resolution curve significantly. However, if the sample period is set according to

$$T_s = \frac{\tau}{V_g}, \quad (2.7)$$

where $\tau$ is a positive constant that represents the sample rate divisor, the value of $r$ will follow a fixed-resolution curve as a function of the measured distance as shown in Figure 2.8(b). This is useful because it eliminates variation in resolution due to velocity, and provides repeatable and reliable sensor resolution in spite of wind conditions or the commanded airspeed.

2.8 Optic Flow Ranging Results

Flight test results using the constant curve resolution are shown in Figure 2.9. Spikes in the optic flow measurement near sample 275 on the constant curve resolution plot occurred as the UAV flew over an asphalt road. The sensor was unable to compute
optic flow over the texture poor asphalt surface and thus returned low values for $\delta p_x$. In practice such spikes are ignored by discarding measurements for which the optic flow sensors report a low surface quality (squal).

Figure 2.9: Flight test results using a 1.2 degree fov lens and fixed curve resolution.

The accuracy of the distance measurements calculated from optic flow was tested by comparison with a laser rangefinder readings. Testing was performed by mounting three optic flow sensors and a laser rangefinder perpendicular to the motion of a vehicle driving along a freeway at varying distances from the freeway sound barrier wall. Range values were recorded for both the optic flow sensors and the laser rangefinder. The results of this test are shown in Figure 2.10. The results show close agreement and validate the readings obtained from the optic flow calculations. Taking the laser rangefinder readings to be true, the range estimates of all three optic flow sensors had less than 1.5 m of error.

Range data from the optic flow sensor in flight tests showed reliable, repeatable readings for heights above ground less than 50 m. For the purposes of estimating distances
to obstacles on the left and right sides of the UAV from optic flow, values above 50 m are clamped to 50 m. In HAG estimation, values of 50 m or more are also discarded as noise.

2.9 Sensor Orientation

The physical orientation of the sensors on the SUAV is also an important consideration when using optic flow sensors for distance measurements. Noise in the distance readings makes it important to filter the optic flow distance measurements. A simple low-pass digital alpha filter was selected to reduce measurement noise in the distance calculations. The rise time of the implemented filter is approximately 0.4 seconds. To mitigate lag from this filter and to give the SUAV a forward looking capability, the optic ranging sensors are oriented at an angle $\alpha = 60$ degrees, measured from the UAV’s body frame x-axis as shown in Figure 2.4. This angle was selected so that filter lag is mitigated when the UAV is flying at a nominal groundspeed ($V_{nom} = 13$), and at a critical lateral distance from obstacles ($d_{crit} = 10$ m). The drawback for pointing the sensor at a forward angle is a reduction in the lateral range of the sensor. Figure 2.11 shows the relationship between lateral range and sensor orientation. With the sensors mounted at this forward angle, the maximum lateral range of the sensors $D_{max}$ is effectively reduced from 50 m to 44 m. However, at a distance $D_{max}$, traveling at an airspeed $V_{nom}$ the UAV is able to sense obstacles approximately 18 m ahead. This look ahead distance translates into a 1.4 second look ahead time if the SUAV is
traveling at a groundspeed of 13 m/s. This lookahead time could give the UAV extra time to sense and avoid obstacles forward along its path.

![Lateral sensor range vs. sensor orientation angle α.](image)

Figure 2.11: Lateral sensor range vs. sensor orientation angle $\alpha$.

### 2.10 New Prototype Optic Flow Ranging Sensor

After preliminary ranging tests using the ADNS-2610 sensor demonstrated the effectiveness of optic flow ranging, a cutting-edge optic-flow sensor from Agilent was selected to be used in a custom design circuit board for ranging. In contrast to ADNS-2610 with a $18 \times 18$ pixel CMOS imaging array that computes optic flow at 1500 fps, the ADNS-3080 has a higher resolution $30 \times 30$ pixel CMOS imaging array that can process optic flow at a blazing 6400 fps, more than four times faster than the ADNS-2610. An image of the wiring schematic for this prototype sensor is shown in Figure 2.12. The new sensor board incorporates all surface mount components to miniturize the sensor and reduce weight. The board layout is depicted in Figure 2.13 where the blue represents the bottom layer and the
red represents the top layer of the two layer board. A photograph of the completed first run of the prototype sensors is shown in Figure 2.14.
The new board design allows for precision placement of the selected optics and a more compact design with a locking and keyed connector for added functionality. The added precision, improved sensor resolution, and increased frame rate reduces sensor noise and increases the range of the sensor.

2.11 Summary

In summary, optic flow sensors can be used to accurately measure distances and are a nice alternative to laser rangefinders because of their low cost, light weight, low power consumption, and accurate ranging capabilities over distances less than 50 m. However, laser rangefinders should be used to measure longer distances or when accurate ranging is required along the direction of travel.
Chapter 3

Reactive Terrain Navigation

3.1 Introduction

Applications such as rural search and rescue, urban surveillance, and perimeter monitoring require UAVs to fly down city streets or near the surface of mountainous terrain. In order to accomplish these missions, careful pre-mission path planning must take place to best assure that the desired path is clear of hazards.

Even with cautious path planning using the most current maps and terrain data, many potential hazards exists that could be overlooked. Automated path planning algorithms like the Rapidly Expanding Random Tree (RRT) depend on terrain data to select safe paths. This terrain data is not always adequate for two main reasons. First, the resolution between data points is at best 10 m and could be as low as 100 m. Terrain maps generated from this data interpolate between data points. This has the effect of rounding mountain tops and valleys, thus inaccurately representing the terrain. Second, this data frequently omits trees, buildings, towers, and other hazards which represent a significant hazard to low flying SUAVs. Therefore, while the ability to follow preplanned paths precisely is very important, the UAV must also be able to reactively alter or deviate from its planned path to avoid potential hazards, then promptly return to the preplanned path when it is safe to do so.

An accurate method of path following has already been established by Nelson in [15]. His method uses a vector field approach to closely follow paths composed of straight lines segments and arcs. Using this method, when the UAV is far away from the path segment the objective is to fly toward the path. As the UAV approaches the path, the
desired course transitions from approaching the path to flying along the path as shown in Figure 3.1. The desired course is computed based on the sign and magnitude of the tracking error $d$ which is the perpendicular distance of the UAV to the path for straight lines. Nelson’s results in [15] show that the lateral-tracking and course error of this method will approach zero asymptotically even in the presence of wind. These claims are justified
using Lyapunov stability proofs and are verified with experimental results as shown in Figure 3.2.

3.2 Reactive Terrain Navigation Using Path Biasing

A simple extension of this vector field approach allows the SUAV to offset or bias its preplanned path toward the center of a canyon or between buildings and other hazards. This allows the UAV to safely navigate extreme terrain and accomplish mission objectives even in the presence of imperfect waypoint path planning.

![Path biasing diagram.](image)

Figure 3.3: Path biasing diagram.

Suppose that a UAV is capable of estimating distances to obstacles off its left and right sides. This UAV is traveling along a straight path segment from waypoint 1 to waypoint 2. However, this path segment is not safely centered between the two walls as depicted in Figure 3.3. The amount by which the UAV should shift its path is given according to

\[ \delta = \frac{1}{2}(D_{\text{right}} - D_{\text{left}}), \]

(3.1)
where $D_{\text{right}}$ is the distance between the UAV and the right wall, and likewise $D_{\text{left}}$ is the distance between the UAV and left wall. Shifting the desired path by this offset $\delta$ centers the desired path between the detected obstacles as shown in Figure 3.3. This also shifts the vector field in the same manner while maintaining the same path course as depicted in Figure 3.4.

If at each time step along the path the SUAV computes and shifts its desired path by $\delta$, then the desired path will resemble a curve that is centered between the terrain walls and whose waveform approximates the average of the wave forms of the left and right walls as shown in Figure 3.5. In each subfigure the bold lines represent the canyon walls, the blue line represents the planned path, and the dashed green line represents the offset path.

### 3.3 Path Biasing Implementation

Once a waypoint path has been selected, the UAV will pursue the first waypoint in the preplanned path using the vector field course controller. At every time step along the path the UAV will follow the steps outlined in Table 3.1.
Table 3.1: Steps of Path Offsetting Method.

1. Compute $D_{\text{left}}$ and $D_{\text{right}}$ from optic flow readings
2. Compute path offset $\delta$ according to equation 3.1
3. Compute the tracking error $y$
4. Shift the desired path segment by setting $y^+ = y^- - \delta$
5. Compute desired course based on new offset lateral-tracking error $y^+$

One disadvantage of the path offsetting method is that it only permits the planned path to be offset by the maximum lateral range of the sensors divided by two. For the optic flow sensors used in these experiments $D_{\text{max}} = 50$ m. This results in a maximum path offset $\delta_{\text{max}}$ of 25 m. Therefore, this method will only work when the preplanned path is in error by less than 25 m. As a result, if the the planned path is in error by more than $\delta_{\text{max}}$ the SUAV won’t be able to bias itself far enough to avoid the terrain hazards. The chosen solution to this problem is to select a minimum safe distance, $D_{\text{safe}}$, such that if either $D_{\text{left}}$ or $D_{\text{right}}$ are less than $D_{\text{safe}}$ then the SUAV no longer tries to follow the offset path. Instead the SUAV tries to balance $D_{\text{left}}$ and $D_{\text{right}}$ by changing its desired course according to

$$
\chi_c = \chi + k_p(D_{\text{right}} - D_{\text{left}}),
$$

(3.2)
where $k_p$ is positive proportional gain, $\chi$ is the actual course of the UAV, and $\chi_c$ is the commanded course to take in order to balance left and right distances. Once both $D_{\text{left}}$ and $D_{\text{right}}$ are greater than $D_{\text{safe}}$ the UAV resumes its normal path following.

### 3.4 Simulation Results

In simulation a terrain map of Goshen Canyon was loaded. A waypoint path was chosen through the simulated canyon so that the planned path was intentionally placed inside the east canyon wall. This was done to test how well the simulated SUAV was able to offset its path to avoid the terrain walls. Simulation results show that the SUAV biased itself 10 to 25 meters to the right of its preplanned path as it flew in a south-easterly direction through the canyon. In so doing, the SUAV was able to successfully avoid the canyon walls and fly a path closer to center of the canyon as shown in Figure 3.7 where the planned path is shown in orange, the offset path in green, and the actual path in blue.

Figure 3.6: Screen shot from a terrain navigation simulation using path offsetting.

biased itself 10 to 25 meters to the right of its preplanned path as it flew in a south-easterly direction through the canyon. In so doing, the SUAV was able to successfully avoid the canyon walls and fly a path closer to center of the canyon as shown in Figure 3.7 where the planned path is shown in orange, the offset path in green, and the actual path in blue.
3.5 Flight Test Results

3.5.1 Tree Avoidance

In preparation for testing the path offsetting method in a canyon environment, it was first tested in a wide open area where the planned path could be placed along side a
long row of trees as shown in Figure 3.9(a). The planned flight path was placed 13 meters to the left of the row of trees and the SUAV was commanded to fly at 9 meters above the ground. The SUAV used for these tests is shown in 3.8. Also shown are two optic flow sensors pointing to the left and right at a 30 \text{ deg} forward angle.

![Figure 3.8: SUAV used for tree avoidance tests.](image)

During a series of low passes, the SUAV flew along its path until the optic flow ranging sensor on the right detected the trees. As the SUAV sensed the obstacles and computed their lateral distances, it biased its preplanned path between 5 and 15 meters to the left as depicted in Figure 3.9(b). By tracking the offset path the SUAV was able to distance itself by as much as 15 meters from a potential hazardous encounter with some of the closer trees as shown in Figure 3.10. The planned path is shown in red while the actual path flown by the SUAV using path offsetting is shown in blue. All trees are represented by green circles.

### 3.5.2 Canyon Navigation

Goshen Canyon, in central Utah, was the subject of simulation and flight tests. This canyon was selected for its steep winding canyon walls that reach over 75 meters high. This
Figure 3.9: Photos of tree avoidance site and SUAV flight tests.

Figure 3.10: Results from tree avoidance flight tests.
canyon was also ideal because of its proximity to BYU campus and low automobile traffic. Canyon navigation simulation results were experimentally verified with flight tests through Goshen Canyon using a fixed-wing SUAV. Photos of the SUAV navigating Goshen Canyon are shown in Figure 3.11. The left figure shows the SUAV exiting the canyon having successfully offset its path to navigate through the terrain using the optic flow ranging sensors. The top right of both images show photos taken from the SUAVs onboard camera. In the first flight through the canyon, the planned path was selected to follow the road. The SUAV navigated the canyon with only minor adjustments to its path. For the second and third flights the planned path was intentionally biased toward the east canyon wall to simulate path planning error and to verify that the SUAV would bias its planned path to navigate the canyon and center itself through the terrain.

In Figure 3.13 data collected from the second flight shows that the SUAV biased its planned path to the right between 3 and 10 meters for a duration of approximately 25 seconds to better center itself through the terrain. By tracking the offset path, the SUAV more closely followed the road which was more or less centered between the canyon walls as depicted in Figure 3.12 where the planned path is shown in green and the actual path...
is shown in blue. If the SUAV had not biased its path it would have crashed into the east canyon wall. For this terrain, flight test results support the simulation results in showing that this simple path biasing method can be effective at safely guiding UAVs through hazardous canyon terrain.
Chapter 4

Curved Path Following

4.1 Introduction

4.1.1 Overview

For most applications of unmanned aerial vehicles (UAVs), the ability to follow paths composed of straight lines and arcs is sufficient to accomplish mission objectives. However, certain applications require accurate following of curved paths including tactical maneuvers, formation flying, convoy support, and low altitude missions through narrow mountainous canyons.

There are several advantages associated with using SUAVs to accomplish many of these missions. In addition to being less costly than larger UAVs, they are also more easily transported and can be launched even in rough terrain. In general, they are also less complex and therefore require fewer ground operators. Their simplicity also allows SUAVs to be deployed more quickly than larger UAV systems. Possibly, the greatest advantage inherent in SUAVs is their heightened potential to navigate narrow passages and confined urban environments. For fixed-wing SUAVs, the minimum turn radius is typically in the range of 10 to 50 m compared to the range of 250 to 2000 m for large non-aerobatic UAVs like Predator and Global Hawk. This translates into a significant increase in the spatial frequency of paths that can be tracked by SUAVs. Path tracking algorithms must be able utilize the full turning capability of the SUAV in order exploit this key advantage.
4.1.2 Literature Review

One path planning strategy that requires the UAV to follow curved paths for convoy support is presented in [12] calls for a UAV to fly a sinusoidal path that was precisely computed to keep the faster-flying UAV at the same forward progress along the road as the slower-moving ground vehicles.

A new methodology for the design of UAV guidance, navigation, and control systems is established in [9]. Their approach attempts to design the guidance and control simultaneously. This integrated approach achieves zero steady-state error at any trimming trajectory by using gain scheduling theory. Their approach also addresses the stability of the combined guidance and control system.

One approach for curved path trajectory tracking is presented in [17]. In this approach an optimal path is divided into curved segments. Each segment is then approximated using a 7th-order polynomial. These curved segments are then followed using a controller derived from a kinematic model with the addition of an anticipatory control element that improves the tracking capability for curved paths. This method also accounts for wind disturbances by the addition of an adaptive kinematic factor. Their paper reports experimental tracking errors of 1.6 meters RMS.

A vector pursuit method of path following for UGVs presented in [18] generates a desired orientation based on the current position and orientation of the vehicle relative to the position of a look-ahead point on the planned path and the desired orientation along the path at the look-ahead point. Experimental results show that this technique is less sensitive to the look-ahead distance and vehicle speed while still providing an anticipatory element to improve tracking on paths with high curvature.

Accurate curved path following was achieved in simulation for autonomous underwater vehicles (AUVs) by Lapierre, et al in [11]. While the dynamics of AUVs are notably different than those of UAVs, the Lyapunov approach taken by the authors to derive and prove a nonlinear path following control law is insightful for similar path following approaches for UAVs.

An important limitation of SUAVs is that they are much more adversely affected by windy conditions than large UAVs. This limitation is mostly caused by their light weight
and the low range of airspeeds they are capable of. Since wind speeds can often surpass 40 percent of the SUAV airspeed, effective path tracking strategies must be developed to overcome the effects of these environmental disturbances.

One successful and robust method that allows SUAVs to follow straight lines and arcs in the presence of wind has been developed by Nelson, et al. in [16]. This path following approach uses a vector field of course commands to guide the SUAV on the waypoint path with minimal overshoot. Experimental results have yielded cross-tracking errors of 0.7 m on average with wind speeds up to 50 percent of the SUAVs airspeed.

The work presented in this paper builds on the work presented by Nelson, et al. by adapting their vector field approach for straight line paths and arcs to extend to arbitrary curved paths. One advantage of this method is that the vector field approach lends itself well to reactive collision avoidance methods that alter the path in real time as obstacles are detected. In addition, this vector field approach takes into account the instantaneous speed of the UAV which allows for close following with minimal overshoot over a wide range of groundspeeds. Experimental results validate the feasibility of this approach for SUAVs and demonstrate accurate curved path following even in moderately windy conditions.

### 4.2 Problem Description

The objective of this paper is to present a new method for UAVs to accurately follow curved paths in the presence of wind. This method is designed to work for all curved paths including piecewise continuous functions composed of straight lines, arcs, and curves without requiring the addition of any special cases for each path type. Additionally, a series of polynomial path segments can be followed using an approach similar to one described in [17] where optimally planned paths are segmented based on angular and distance constraints then linear regression is used to fit each segment with a polynomial.

Unlike trajectory tracking where the objective is to track a moving point in time, the objective of path following is to spatially track a path without regard for temporal constraints as remarked in [15]. The path following approach in this paper uses a vector field around the path to be followed to guide the SUAV onto the path asymptotically. Figure 4.1 shows an example of a vector field for a curved path segment.
4.3 Technical Approach

For this method, it is assumed that altitude \( h \) is accurately controlled by pitch and altitude control loops. It is also assumed that the airspeed of the fixed-wing SUAV \( V_a \) is maintained at a safe level above stall speed \( V_{\text{stall}} \). A model of the navigational dynamics used for this path following approach are given according to

\[
\begin{align*}
\dot{x} &= V_g \cos \chi, \\
\dot{y} &= V_g \sin \chi,
\end{align*}
\]

where \( V_g \) is the groundspeed and \( \chi \) is the course of the UAV. In using these navigational dynamics, constant wind disturbances are already accounted for since the motion of the UAV is represented in terms of groundspeed and course which are independent of the wind speed and direction. Another key assumption is that the UAV follows first-order course
dynamics given by
\[ \dot{\chi} = \alpha (\chi^c - \chi), \]  
\hspace{2cm} (4.3)
where \( \chi^c \) is the commanded course, and \( \alpha \) is a positive constant that physically represents the reciprocal of the time constant. Provisions for when \( \alpha \) is not exactly known are given in Section 4.3.3.

4.3.1 Control Approach

In the development of the this curved path following approach, let \( d \) be the lateral distance from the the curved path to the SUA V as shown in Figure 4.2. Another important variable is the path course \( \chi^p \). For the curve, \( y = f(x) \), \( \chi^p \) can be found according to
\[ \chi^p = \tan^{-1}\left( \frac{dy}{dx} \right). \]  
\hspace{2cm} (4.4)

The rate of change of the lateral-tracking error \( \dot{d} \) is defined as a function of the UAVs

Figure 4.2: Vector field for curved path following.
groundspeed $V_g$ and relative orientation by

$$
\dot{d} = V_g \sin(\chi - \chi_p).
$$

(4.5)

To direct the UAV onto the path, an asymptotically stable vector field of course commands is constructed around the curved path segment according to

$$
\chi^d(d) = \chi^p - \frac{2\chi^\infty}{\pi} \tan^{-1}(kd)
$$

(4.6)

where $\chi^d(d)$ is the desired course expressed as a function of the lateral-tracking error $d$ and $\chi^p$ is the course of the path at the point from which $d$ was computed. Note that as $d$ gets smaller, the desired course transitions from $\chi^\infty$ at $d = \infty$ to $\chi^p$ at $d = 0$. The positive constant $k$ controls the rate of this transition and can be thought of as the strength of the vector field. Ultimately, for the SUAV to get on the path this approach must drive $d$ to zero by driving $\chi$ toward $\chi^d$. The angle $\chi^\infty$ is limited to the range $(0, \frac{\pi}{2}]$ so that

$$
-\frac{\pi}{2} < \frac{2\chi^\infty}{\pi} \tan^{-1}(kd) < \frac{\pi}{2}
$$

over all values of $d$.

To solve for the appropriate control command $\chi^c$ that will force the tracking error to converge, first consider the Lyapunov function $W_1 = \frac{1}{2}d^2$ with derivative

$$
\dot{W}_1 = dd = dV_g \sin(\chi - \chi^p)
$$

(4.7)

so that if $\chi = \chi^d$, then $d \to 0$ asymptotically. In this paper a sliding mode approach will be used to ensure that $\chi = \chi^d$. Under the assumption that $\chi = \chi^d$ and substituting (4.6) into (4.7), we find that

$$
\dot{W}_1 = -dV_g \sin \left( \frac{2\chi^\infty}{\pi} \tan^{-1}(kd) \right)
$$

(4.8)
which is negative definite thus meeting Lyapunov stability criteria and ensuring \( d \to 0 \) asymptotically. Next we must derive a controller to ensure \( \chi \to \chi^d \). Let \( \hat{\chi} = \chi - \chi^d(d) \) and differentiate to give

\[
\dot{\hat{\chi}} = \dot{\chi} - \dot{\chi}^d(d) = \alpha(\chi^c - \chi) - \chi^p + \frac{2\chi^\infty}{\pi} \frac{k}{1 + (kd)^2} V_g \sin(\chi - \chi^p).
\] (4.9)

Next, let \( W_2 = \frac{1}{2} \tilde{\chi}^2 \). Differentiating and substituting from (4.9) gives

\[
\dot{W}_2 = \tilde{\chi} \dot{\tilde{\chi}} = \tilde{\chi} \left( \alpha(\chi^c - \chi) - \dot{\chi}^p + \frac{2\chi^\infty}{\pi} \frac{k}{1 + (kd)^2} V_g \sin(\chi - \chi^p) \right).
\] (4.10)

To ensure that (4.10) is negative definite we will choose the control command as

\[
\chi^c = \chi + \frac{\dot{\chi}^p}{\alpha} - \frac{1}{\alpha} \frac{2\chi^\infty}{\pi} \frac{k}{1 + (kd)^2} V_g \sin(\chi - \chi^p) - \frac{\beta}{\alpha} \text{sign}(\tilde{\chi}).
\] (4.11)

where \( \beta \) is a positive constant and

\[
\text{sign}(\tilde{\chi}) = \begin{cases} 
1 & \text{if } \tilde{\chi} > 0 \\
0 & \text{if } \tilde{\chi} = 0 \\
-1 & \text{if } \tilde{\chi} < 0.
\end{cases}
\]

As a result, \( \dot{W}_2 \leq -\beta |\tilde{\chi}| \) and we can conclude that \( \tilde{\chi} \to 0 \) in finite time. Unfortunately, using \( \text{sign}(\tilde{\chi}) \) causes undesirable control chatter in implementation. One solution to this problem is to replace \( \text{sign}(\tilde{\chi}) \) with \( \text{sat}(\frac{\tilde{\chi}}{\epsilon}) \) where

\[
\text{sat}(\frac{\tilde{\chi}}{\epsilon}) = \begin{cases} 
\frac{\tilde{\chi}}{\epsilon} & \text{if } |\frac{\tilde{\chi}}{\epsilon}| \leq 1 \\
\text{sign}(\frac{\tilde{\chi}}{\epsilon}) & \text{otherwise}
\end{cases}
\]

The width of the sliding mode boundary region is represented by the positive constant \( \epsilon \). Implementing this change results in proportional feedback around the course error inside
the boundary region and yields the final control command

\[
\chi^c = \chi + \frac{\dot\chi}{\alpha} - \frac{1}{\alpha} \frac{2\chi^\infty}{\pi} \frac{k}{1 + (kd)^2} V_g \sin(\chi - \chi^p) - \frac{\beta}{\alpha} \alpha(\chi^c).
\]  

(4.12)

It is important to note that large values of \( k \) will cause large changes in \( \chi^d \) as the UAV gets close to the curved path that are difficult to track. Equally important is the \( \dot\chi^p \) term which represents the rate of change of the path over time, which will be discussed in some detail in Section 4.3.2.

So far we have shown that the control command in (4.12) will drive the UAV inside the sliding mode boundary region in finite time, however, it must still be shown that the system converges to the origin \((d, \bar{\chi}) = (0, 0)\). To accomplish this, we define a candidate Lyapunov function as

\[
W = \frac{1}{2}d^2 + \frac{1}{2}\bar{\chi}^2.
\]  

(4.13)

For the system to converge \( \dot{W} \) must be negative definite. Differentiating (4.13) we obtain

\[
\dot{W} = d\dot{d} + \bar{\chi}\dot{\bar{\chi}}
\]

\[
= dV_g \sin(\chi - \chi^p) + \bar{\chi} \alpha \left( \chi^c - \chi - \frac{\dot\chi}{\alpha} + \frac{2\chi^\infty}{\alpha\pi} \frac{k}{1 + (kd)^2} V_g \sin(\chi - \chi^p) \right).
\]

Inside the boundary region \( \chi^c \) can be written as

\[
\chi^c = \chi + \frac{\dot\chi}{\alpha} - \frac{1}{\alpha} \frac{2\chi^\infty}{\pi} \frac{k}{1 + (kd)^2} V_g \sin(\chi - \chi^p) - \frac{\beta}{\alpha} \alpha \bar{\chi}.
\]  

(4.14)

To simplify the development, we will let

\[
\bar{\chi}^d(d) \triangleq \chi^d(d) - \chi^p = \frac{2\chi^\infty}{\pi} \tan^{-1}(kd).
\]  

(4.15)
Substituting (4.14) and (4.15) into $\dot{W}$ gives

$$\dot{W} = V_g d \sin (\tilde{\chi}^d (d) + \tilde{\chi}) - \frac{\beta}{\epsilon} \tilde{\chi}^2$$

$$= - \frac{\beta}{\epsilon} \tilde{\chi}^2 + V_g d \sin (\tilde{\chi}^d (d)) + V_g d \left( \sin (\tilde{\chi}^d (d) + \tilde{\chi}) - \sin (\tilde{\chi}^d (d)) \right)$$

$$\leq - \frac{\beta}{\epsilon} \tilde{\chi}^2 + V_g d \sin (\tilde{\chi}^d (d)) + V_g |d| \sin (\tilde{\chi}^d (d) + \tilde{\chi}) - \sin (\tilde{\chi}^d (d)) \right).$$

This expression can be simplified by noting that

$$\sin (\tilde{\chi}^d (d) + \tilde{\chi}) - \sin (\tilde{\chi}^d (d)) = \left| \sin (\tilde{\chi}^d (d)) \cos (\tilde{\chi}) + \cos (\tilde{\chi}^d (d)) \sin \tilde{\chi} - \sin (\tilde{\chi}^d (d)) \right|$$

$$= \left| \sin (\tilde{\chi}^d (d)) (\cos (\tilde{\chi}) - 1) + \cos (\tilde{\chi}^d (d)) \sin (\tilde{\chi}) \right|$$

$$\leq |\cos (\tilde{\chi}) - 1| + |\sin (\tilde{\chi})|$$

$$\leq 2 |\tilde{\chi}|,$$

which yields

$$\dot{W} \leq - \frac{\beta}{\epsilon} \tilde{\chi}^2 + 2V_g |d| |\tilde{\chi}| + V_g d \sin (\tilde{\chi}^d (d))$$

$$\leq - \frac{\beta}{\epsilon} \tilde{\chi}^2 + 2V_g |d| |\tilde{\chi}| - V_g d \sin \left( \frac{2\chi^\infty}{\pi} \tan^{-1}(kd) \right).$$

Next, let

$$\phi(d) \triangleq d \sin \left( \frac{2\chi^\infty}{\pi} \tan^{-1}(kd) \right),$$

Taking the derivative of $\phi(d)$ we get

$$\left| \frac{\partial \phi}{\partial d} \right| = \left| \sin \left( \frac{2\chi^\infty}{\pi} \tan^{-1}(kd) \right) + d \cos \left( \frac{2\chi^\infty}{\pi} \tan^{-1}(kd) \right) \frac{2\chi^\infty}{\pi} \frac{k}{1 + (kd)^2} \right|$$

$$\leq \left| \frac{2\chi^\infty}{\pi} \tan^{-1}(kd) \right| + \left| \frac{2\chi^\infty}{\pi} \frac{kd}{1 + (kd)^2} \right|$$

$$\leq 2 |d| \frac{2\chi^\infty}{\pi} k.$$

Recalling that for two continuous functions, $f$ and $g$, where $f(0) = g(0)$ and $|f'(x)| \leq
\[ |g'(x)| \text{ then } |f(x)| \leq |g(x)|, \text{ we get} \]

\[ \dot{W} \leq -\frac{\beta}{\epsilon} \chi^2 + 2V_g |d| |\bar{\chi}| - 2V_g \frac{2\chi^\infty}{\pi} k d^2. \]

Rewriting this quadratic equation in matrix form gives

\[ \dot{W} \leq -V_g \begin{pmatrix} |\bar{\chi}| & |d| \end{pmatrix} \begin{pmatrix} \frac{\beta}{4V_g} & -1 \\ -1 & \frac{4\chi^\infty}{\pi} \end{pmatrix} \begin{pmatrix} |\bar{\chi}| \\ |d| \end{pmatrix}. \tag{4.16} \]

The quadratic nature of \( \dot{W} \) guarantees exponential stability if it is negative definite. It can be shown that \( \dot{W} \) is negative definite if

\[ \frac{\beta}{\epsilon V_g} \left( \frac{4\chi^\infty k}{\pi} \right) > 1. \tag{4.17} \]

If the constants are selected to satisfy (4.17) then the system is globally exponentially stable. However, it will be difficult to satisfy (4.17) for large values of \( V_g \). Intuitively this makes sense. Higher groundspeeds result in a larger turning radius which will make a curved path more difficult or impossible to track. Small values of \( \beta \) and \( k \) will also make it difficult to satisfy this constraint. However, large values of \( \beta \) could cause unrealistic control effort. In summary, the control command in (4.12) will lead to finite time convergence to the course field described by (4.6). In turn, this will lead to asymptotic decay of the lateral-tracking error all as long as (4.17) is satisfied.

\subsection*{4.3.2 Implementation Issues}

The physical heading rate constraints associated with fixed-wing UAVs place a limit on the path curvature that can accurately be followed. The path course rate \( \dot{\chi}^p \), represented in the control command in (4.12), is an important factor in determining whether a curved path is flyable. It also has physical meaning, representing the functional relationship between the groundspeed and relative orientation of the UAV and the spatial rate of change of the path. If the curved path to be followed is \( C^2 \) continuous, then \( \dot{\chi}^p \) can be solved for
explicitly. Consider the sinusoidal path

\[ y = A \sin(\omega x), \quad (4.18) \]

where \( A \) is the amplitude, \( \omega \) is the spatial frequency measured in radians per meter, and \( x \) is the distance along the path in meters. The path course rate \( \dot{\chi}^p \) can be found by first expressing it as

\[ \dot{\chi}^p = \frac{d\chi^p}{dx} \frac{dx}{dt}, \quad (4.19) \]

which allows the solution to be divided into two parts. The first part, \( \frac{d\chi^p}{dx} \), is found by taking the expression for \( \chi^p \) given by

\[ \chi^p = \tan^{-1} \left( \frac{dy}{dx} \right) = \tan^{-1}(A \omega \cos(\omega x)), \quad (4.20) \]

and differentiating to yield

\[ \frac{d\chi^p}{dx} = \frac{-A \omega^2 \sin(\omega x)}{1 + A^2 \omega^2 \cos^2(\omega x)}. \quad (4.21) \]

The second part, \( \frac{dx}{dt} \), is related to the groundspeed of the UAV according to

\[ \frac{dx}{dt} = V_g \cos(\chi^\ell - \chi), \quad (4.22) \]

where \( V_g \) is groundspeed and \( \chi^\ell \) is the directional sense of the path segment as shown in Figure 4.2. Finally, the solution is obtained by substituting (4.21) and (4.22) into (4.19) to yield

\[ \dot{\chi}^p = \frac{-A \omega^2 \sin(\omega x) V_g \cos(\chi^\ell - \chi)}{1 + A^2 \omega^2 \cos^2(\omega x)}. \quad (4.23) \]
The maximum value for $\dot{\chi}^p$ occurs at the peaks of the sine wave or when $x = \frac{n\pi}{2}$, where $n$ is an odd integer. At its maximum value, (4.23) can be reduced to

$$\dot{\chi}_{\text{max}}^p = |A\omega^2 V_g|.$$  

In Figure 4.3, a contour plot shows how $\dot{\chi}_{\text{max}}^p$ varies with spatial frequency and amplitude for sinusoidal paths while holding a constant groundspeed. For straight line segments,

$$\frac{d\chi^p}{dx} = 0$$

and therefore $\dot{\chi}^p$ will have a value of zero, thus simplifying the course command equation. When following circular paths, $\dot{\chi}^p$ will vary mainly with groundspeed. For curved paths, if $\dot{\chi}^p$ is ever greater than the maximum achievable turn rate of the SUA V, the SUAV will not be able to closely follow the curve until $\dot{\chi}^p$ drops to a manageable level. One way to reduce $\dot{\chi}_{\text{max}}^p$ without changing the path is to reduce the SUAV groundspeed by commanding a lower airspeed. This is only effective to a certain degree and caution must

![Contour plot of $\dot{\chi}_{\text{max}}^p$ in deg/s for $V_g = 13$ m/s.](image)

Figure 4.3: Contour plot of $\dot{\chi}_{\text{max}}^p$ in deg/s for $V_g = 13$ m/s.
be taken to keep the SUAV airspeed above $V_{\text{stall}}$. For path segments represented by polynomials, $\dot{\chi}^p$ can also be found in general terms following the same steps outlined in (4.19) - (4.23).

In practice, $\dot{\chi}^p$ can be approximated using a simple backward difference numerical derivative according to

$$\dot{\chi}^p \approx \frac{\chi^p - \chi^p_{\text{old}}}{dt}, \quad (4.25)$$

where $\chi^p_{\text{old}}$ is the path course computed in the previous time step. In the limit, as $dt$ approaches zero, (4.25) approaches the true value of $\dot{\chi}^p$. This numerical method has the advantage of being both computationally cheap and effective at estimating $\dot{\chi}^p$ for any path type, not exclusively paths represented by known differentiable functions.

### 4.3.3 Effects of Model Error

Up to this point it has been assumed that the constant $\alpha$ in the course rate model in (4.3) was accurately known. However, $\alpha$ is difficult to determine experimentally and inevitably there will be some error in its estimated value, which will result in less precise path following. In this section, limits will be placed on parameters as a function of the percent error in $\alpha$. First, let $\alpha = \hat{\alpha} + \tilde{\alpha}$ where $\hat{\alpha}$ is the estimate of $\alpha$ and $\tilde{\alpha}$ is the error in $\hat{\alpha}$. The course rate model can then be expressed according to

$$\dot{\chi} = \hat{\alpha}(\chi^c - \chi) + \tilde{\alpha}(\chi^c - \chi), \quad (4.26)$$

Recalling that

$$\tilde{\chi} = \chi - \chi^d$$

$$= \chi - \chi^p + \frac{2\chi^\infty}{\pi} \tan^{-1}(kd),$$

47
its derivative becomes

\[
\dot{\chi} = \dot{\chi} - \dot{\chi}^p + \frac{2\chi^\infty}{\pi} k\dot{d} \frac{kd}{1 + (kd)^2}
\]

\[
= -\hat{\alpha} \chi + \hat{\alpha} \chi^c + \hat{\alpha} (\chi^c - \chi) - \dot{\chi}^p + \frac{2\chi^\infty}{\pi} k\dot{d} \frac{kd}{1 + (kd)^2}. \tag{4.27}
\]

Then taking the control command found in (4.12) and substituting it into (4.27) yields

\[
\dot{\chi} = -\beta \text{sign}(\chi) + \hat{\alpha} \left( \frac{\dot{\chi}^p}{\hat{\alpha}} \frac{2\chi^\infty}{\pi} k\dot{d} \frac{kd}{1 + (kd)^2} - \frac{1}{\hat{\alpha}} \beta \text{sign}(\chi) \right)
\]

\[
= -\beta \text{sign}(\chi) + \left( \frac{\alpha}{\hat{\alpha}} \dot{\chi} - \frac{\hat{\alpha} 2\chi^\infty}{\hat{\alpha}} k\dot{d} \frac{kd}{1 + (kd)^2} - \frac{\hat{\alpha}}{\hat{\alpha}} \beta \text{sign}(\chi) \right). \tag{4.28}
\]

Assuming that \(\hat{\alpha}\) is in error by less than 100 percent or

\[
\left| \frac{\hat{\alpha}}{\alpha} \right| < k_o < 1, \tag{4.29}
\]

then taking the Lyapunov function \(W_1 = \frac{1}{2} \dot{\chi}^2\) and differentiating gives

\[
\dot{W}_1 = \ddot{\chi} \dot{\chi}
\]

\[
= \dot{\chi} \left( -\beta \text{sign}(\chi) + \frac{\hat{\alpha}}{\alpha} |\dot{\chi}^p| - \frac{\hat{\alpha} 2\chi^\infty}{\hat{\alpha}} kV_g - \frac{\hat{\alpha}}{\alpha} \beta |\dot{\chi}| \right)
\]

\[
< -\beta |\dot{\chi}| + \frac{\alpha}{\hat{\alpha}} |\dot{\chi}^p| |\dot{\chi}| + \frac{\hat{\alpha} 2\chi^\infty}{\hat{\alpha}} kV_g |\dot{\chi}| - \frac{\hat{\alpha}}{\alpha} \beta |\dot{\chi}|
\]

\[
< -|\dot{\chi}| \left( -k_o |\dot{\chi}^p| - k_o \frac{2\chi^\infty}{\pi} kV_g + \beta (1 + k_o) \right). \tag{4.30}
\]

To satisfy the Lyapunov criteria and give finite-time convergence to the sliding surface, \(\dot{W}_1\) must be less than zero. This will be true if

\[
\beta \geq \frac{k_o \left( \frac{2\chi^\infty}{\pi} kV_g + |\dot{\chi}^p| \right)}{1 + k_o}. \tag{4.31}
\]

When \(\hat{\alpha}\) is small, \(k_o\) will be small and (4.31) can be satisfied more easily. By observation, since \(|\dot{\chi}^p|\) is a function of groundspeed (4.31) will be difficult to satisfy for paths with sharp curvature at high groundspeeds.
4.3.4 Bounding Following Error

Everything above has ignored error caused by course-rate saturation, unmodeled dynamics, and imperfect course measurements, all of which will degrade path following performance. Much of this error will emerge in $\tilde{\chi}$ preventing it from decaying to zero. In this section, a method to show input-to-state stability (ISS) can be used to bound the lateral following error as a function of $\tilde{\chi}$ as described in [10]. In the process of showing that (4.5) is ISS, we will find an equation $\rho(|\tilde{\chi}|)$ that will bound lateral-tracking error so that

$$\dot{V} < -W(d) \quad \forall \quad d > \rho(|\tilde{\chi}|),$$

(4.32)

where $W(d)$ is continuous and positive definite, and $\rho(|\tilde{\chi}|)$ is a class $\mathcal{K}$ function. Letting $V = \frac{1}{2}d^2$ then

$$\dot{V} = dd$$

$$= dV_g \sin(\chi - \chi^p)$$

$$= dV_g \sin(\chi^d(d) + \tilde{\chi} - \chi^p)$$

$$= dV_g \sin(\tilde{\chi}^d(d) + \tilde{\chi}).$$

(4.33)

To continue we will define a constant $\theta \in (0, 1)$ and rewrite (4.33) as

$$\dot{V} = dV_g \sin \left((1 - \theta)\tilde{\chi}^d(d) + \theta\tilde{\chi}^d(d) + \tilde{\chi}\right).$$

(4.34)

Noticing that $dV_g \sin \left((1 - \theta)\tilde{\chi}^d(d)\right) > 0$ we will let $W(d) = -dV_g \sin \left((1 - \theta)\tilde{\chi}^d(d)\right)$. Next, conditions on $|d|$ must be found so that

$$dV_g \sin \left((1 - \theta)\tilde{\chi}^d(d) + \theta\tilde{\chi}^d(d) + \tilde{\chi}\right) < dV_g \sin \left((1 - \theta)\tilde{\chi}^d(d)\right).$$

(4.35)

It is important to note that the sine function is monotonic over the interval $[-\frac{\pi}{2}, \frac{\pi}{2}]$. There are two cases that will be examined to show satisfaction of (4.35).
Case I:  \((d > 0) \Leftrightarrow \ddot{x}(d) < 0\)

\[
\ddot{x} + \theta \dot{x}^d(d) < 0 \\
\ddot{x} < -\theta \dot{x}^d(d) \\
\ddot{x} < \theta \chi \frac{2}{\pi} \tan^{-1}(kd) \quad (4.36)
\]

Case II:  \((d < 0) \Leftrightarrow \ddot{x}(d) > 0\)

\[
\ddot{x} + \theta \dot{x}^d(d) > 0 \\
\ddot{x} > -\theta \dot{x}^d(d) \\
\ddot{x} > \theta \chi \frac{2}{\pi} \tan^{-1}(kd) \quad (4.37)
\]

Combining (4.36) and (4.37) gives

\[
|\ddot{x}| < \theta \chi \frac{2}{\pi} \tan^{-1}(k|d|). \quad (4.38)
\]

Then solving for \(|d|\) yields

\[
|d| > \frac{1}{k} \tan \left( \frac{|\ddot{x}| \pi}{2 \theta \chi} \right), \quad (4.39)
\]

which function is strictly increasing and is equal to zero when the input \(\ddot{x} = 0\), thus meeting the requirements of a class \(\mathcal{K}\) function in [10]. Letting

\[
\rho(|\ddot{x}|) = \frac{1}{k} \tan \left( \frac{|\ddot{x}| \pi}{2 \theta \chi} \right) \quad (4.40)
\]

satisfies (4.32) which can now be written as

\[
\dot{V} < -dV_{\rho} \sin((1 - \theta)\dot{x}^d(d)) \quad \forall \ |d| > \frac{1}{k} \tan \left( \frac{|\ddot{x}| \pi}{2 \theta \chi} \right). \quad (4.41)
\]

This represents a conservative bound on the lateral-tracking error. The significance of this is that the magnitude of lateral-tracking error \(|d|\) is guaranteed to decrease asymptotically until it is less than \(\rho(|\ddot{x}|)\) where it will remain thereafter. However, it cannot be shown
that $|d|$ will converge to zero. For small values of $\tilde{\chi}$, $\rho(|\tilde{\chi}|)$ will be small and therefore the bound on $|d|$ will be tight, resulting in accurate curved path following as long as $\dot{\chi}^p$ does not exceed the maximum turn rate of the SUAV, as discussed in Section 4.3.2. In the following example, we will select typical values for the parameters in (4.39). For $W(d)$ to be negative definite, $\theta$ must be less than 1. However, to get the tightest bound possible, $\theta$ must be approximately 1, thus we will let $\theta = 0.99$. Letting $k = 0.02$, and $\chi^\infty = \frac{\pi}{2}$, when $\tilde{\chi} = 4$ degrees, the bound on $|d|$ can then be found as

$$|d| > \frac{1}{0.02} \tan \left( \frac{4 \pi}{180} \frac{\pi}{2(0.99)(\frac{\pi}{2})} \right)$$

$$> 3.5 \text{ m},$$

thus the lateral-tracking error will decay asymptotically until it is less than 3.5 m.

4.4 Flight Test Results

This path following algorithm was first tested in an open source UAV simulation environment before being experimentally tested. Simulation results showed effective, predictable, and safe navigation over a broad range of curved path types. Simulation results from an irregularly shaped circular path are shown in Figure 4.4. The planned path is shown in green while the actual path is shown as white dots. The figure on the right shows the associated vector field for the curved path on the left.

In flight tests, an SUAV was commanded to fly a variety of sinusoidal path segments with varying amplitudes and spatial periods. During the flight tests the average wind speed measured 2.4 m/s (18 percent of the commanded airspeed) from the northeast. Wind gusts of 3.2 m/s (26 percent of the commanded airspeed) were also recorded. Flight test results with accompanying RMS tracking error (RMSE) are shown in Figure 4.5. The desired path is shown in dotted red and the actual path is shown in solid blue. The mean following error on the 350 m period path was 3.7 m with a standard deviation of 3.75 m and a $\dot{\chi}^p_{\text{max}}$ of 35 deg/sec. As expected, the SUAV was able to accurately follow the large period sinusoids with an average error of less than 3 m for the three paths where the period was 500 m. The
Figure 4.4: Simulation results of SUAV following an irregular circular path.

The mean error was lowest for the path shown in Figure 4.5(e) at only 2.5 m with a standard deviation of 3.4 m. Unsurprisingly, this path also had the lowest $\dot{\chi}_p$ at 5.5 deg/sec. The low value of $\dot{\chi}_p$ probably contributed to its low mean error. These results also show that the SUAV had difficulty precisely following the path with the shortest period, $T = 125$ m, and corresponding $\dot{\chi}_p = 19$ deg/sec. The mean lateral-tracking error for the 125 m period path was 5.1 m, while the standard deviation of the path error was 4.3 m.

During these tests, the SUAV used the numerical computation of $\dot{\chi}_p$, given in (4.25), for path following. However, for comparison purposes $\dot{\chi}_p$ was also computed using the analytical approach in (4.23). Data comparing these approaches are shown in Figure 4.6. The top figure shows flight data comparing analytical results (red) and numerical results (black) for $\dot{\chi}_p$, in degrees per second, for an SUAV following the sinusoidal path shown in the lower figure. In the lower figure the desired path is indicated by the dotted red line and the actual path is indicated by solid blue line. While the numerical method is noisier than the analytical method, it is easier to implement, requires less computation time, and can be employed for all smooth arbitrary curved paths. If computation time is not an issue, then $\dot{\chi}_p$ should be computed using the explicit solution whenever possible because it is more precise than using numerical derivatives and is less susceptible to noise.
4.5 Conclusions

A new method for SUAVs to follow curved paths has been introduced. It has been shown using Lyapunov stability criteria that this vector field approach provides asymptotic path following for curved path segments even in the presence of constant wind disturbances. The effectiveness of the vector field method has been verified experimentally using a fixed-wing SUAV. Lateral-tracking errors for this method averaged 3.4 m for amplitudes ranging between 10 and 100 m and spatial periods between 125 and 500 m.
(a) $A = 10\,\text{m},\ T = 125\,\text{m},\ \text{RMSE} = 5.1\,\text{m}$

(b) $A = 20\,\text{m},\ T = 167\,\text{m},\ \text{RMSE} = 3.3\,\text{m}$

(c) $A = 25\,\text{m},\ T = 250\,\text{m},\ \text{RMSE} = 3.2\,\text{m}$

(d) $A = 35\,\text{m},\ T = 400\,\text{m},\ \text{RMSE} = 2.9\,\text{m}$

(e) $A = 50\,\text{m},\ T = 500\,\text{m},\ \text{RMSE} = 2.5\,\text{m}$

(f) $A = 60\,\text{m},\ T = 500\,\text{m},\ \text{RMSE} = 3.0\,\text{m}$

(g) $A = 100\,\text{m},\ T = 350\,\text{m},\ \text{RMSE} = 3.7\,\text{m}$

(h) $A = 100\,\text{m},\ T = 500\,\text{m},\ \text{RMSE} = 3.4\,\text{m}$

Figure 4.5: Flight test results for curved path following.
Figure 4.6: Flight results comparing the numerical and analytical results for $\dot{\chi}^p$. 
Chapter 5

Conclusions and Future Work

This chapter includes a summary of conclusions and important results from each previous chapter and makes recommendations for future enhancements and additional testing that should be performed for the optic flow sensors, the reactive terrain navigational methods, and the path following schemes discussed in this thesis.

5.1 Optic Flow Ranging

Passive optic flow sensors have proven to be useful at accurately measuring distances and are a viable alternative to laser rangefinders because of their low cost, light weight, low power consumption, and accurate ranging capabilities. Using the ADNS-2610 optic flow sensor, accurate ranging has been achieved up to 50 m. However, laser rangefinders should be used when measuring long distances or when accurate ranging is required along the direction of travel.

Encouraging ranging results could be dramatically improved by using the Agilent ADNS-3080 optic flow sensor. In contrast to ADNS-2610 with a 18×18 pixel CMOS sensor array that computes optic flow at 1500 fps, the ADNS-3080 has a higher resolution 30×30 pixel CMOS sensor that can process optic flow at a blazing 6400 fps, more than four times faster than the ADNS-2610. This new sensor promises increased range with less noise. This could significantly improve the performance for HAG estimation and the effectiveness of the path biasing techniques used in reactive terrain navigation for SUAVs.
5.2 Reactive Terrain Navigation

In reactive terrain navigation tests, the SUAV biased its path away from the canyon walls, following the road which was approximately centered between the canyon walls. If the SUAV had not biased its path it would have crashed into the east canyon wall. Flight test results support simulation results in showing that this simple path biasing method can be effective at safely guiding UAVs through hazardous canyon terrain. However, since the scenarios for which this algorithm was tested were not as complex as most real-world scenarios would be, additional flight tests should be performed to expand the operational performance envelope of the biasing method.

One area for future efforts will be to combine a forward sensing laser rangefinder or computer vision system with the side-mounted optic flow sensors for a more complete collision avoidance system for SUAVs. The enhanced range of the optic flow sensor would improve centering capabilities of the path biasing strategy by increasing the look ahead time.

5.3 Curved Path Following

A new method for curved path following has been introduced. It has been shown using Lyapunov stability criteria that this vector field approach provides asymptotic path following for curved path segments even in the presence of constant wind disturbances. Practical limitations on parameters have also been established. The effectiveness of the vector field method has been verified experimentally using a fixed-wing SUAV. Following errors for this method averaged 3.4 m for amplitudes ranging between 10 and 100 m and spatial periods between 125 and 500 m.

Future work to improve this method would be to test the method while following a wider variety of curved paths, not just sinusoids to further explore the set of flyable paths for similar small UAVs. Additionally, a set of tests should be run on a calm day to gather "best-case-scenario" data to compare with data from windy days. This data would serve to better characterize the wind disturbance rejection of the path following algorithm.
Another matter for future work would be to test path biasing through mountainous terrain while flying a curved path. It is predicted that this would allow for better performance than the path biasing results collected in Goshen Canyon which were performed using the straight line following algorithms.
Bibliography


