Navigation and Control Technologies for Autonomous Micro Vehicles

Randal Beard
beard@byu.edu

Timothy W. McLain
mclain@byu.edu

Follow this and additional works at: https://scholarsarchive.byu.edu/facpub
Part of the Electrical and Computer Engineering Commons

Original Publication Citation
BYU Magic Lab

BYU ScholarsArchive Citation
https://scholarsarchive.byu.edu/facpub/1309

This Report is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_atamangelo@byu.edu.
Final Report

Award No. F08630-03-1-0017

Navigation and Control Technologies for Autonomous Micro Vehicles

August 25, 2005

Principal Investigator/Technical Point of Contact
Randal W. Beard, Associate Professor
Department of Electrical and Computer Engineering
Brigham Young University
Provo, Utah 84604 USA
voice: (801) 422-8392
fax: (801) 422-0201
beard@ee.byu.edu

Co-Principal Investigator
Timothy W. McLain, Associate Professor
Department of Mechanical Engineering
Brigham Young University
Provo, Utah 84604 USA
mclain@byu.edu

AFRL/MN Point of Contact
Carrie Fowler
AFRL/MNAV
101 West Eglin Blvd, Suite 346B
Eglin AFB, FL 32542
(850) 882-8876 x3383
carrie.fowler@eglin.af.mil
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover Page</td>
<td>i</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>iii</td>
</tr>
<tr>
<td>1 Objectives</td>
<td>1</td>
</tr>
<tr>
<td>2 Magnetometer Integration</td>
<td>1</td>
</tr>
<tr>
<td>2.1 Magnetometer Selection</td>
<td>1</td>
</tr>
<tr>
<td>2.2 Motor Noise</td>
<td>2</td>
</tr>
<tr>
<td>2.3 Calibration</td>
<td>2</td>
</tr>
<tr>
<td>2.3.1 Ground Calibration</td>
<td>2</td>
</tr>
<tr>
<td>2.3.2 Airborne Calibration</td>
<td>3</td>
</tr>
<tr>
<td>2.4 Heading Estimation</td>
<td>3</td>
</tr>
<tr>
<td>2.4.1 2-Axis Heading Estimation</td>
<td>3</td>
</tr>
<tr>
<td>2.4.2 3-Axis Heading Estimation</td>
<td>4</td>
</tr>
<tr>
<td>2.5 2-Axis Tilt Compensation</td>
<td>5</td>
</tr>
<tr>
<td>2.6 Magnetometer Wind Estimation</td>
<td>5</td>
</tr>
<tr>
<td>3 Height Above Ground Sensor</td>
<td>9</td>
</tr>
<tr>
<td>3.1 Landing Algorithm</td>
<td>9</td>
</tr>
<tr>
<td>3.1.1 Overview</td>
<td>9</td>
</tr>
<tr>
<td>3.2 Orbit Routine</td>
<td>10</td>
</tr>
<tr>
<td>3.2.1 Heading Field Generation</td>
<td>10</td>
</tr>
<tr>
<td>3.2.2 Correcting Steady State Error</td>
<td>12</td>
</tr>
<tr>
<td>3.2.3 Glide Slope</td>
<td>13</td>
</tr>
<tr>
<td>3.3 Barometric Altimeter Based Landing</td>
<td>14</td>
</tr>
<tr>
<td>3.3.1 Theory</td>
<td>14</td>
</tr>
<tr>
<td>3.3.2 Application</td>
<td>14</td>
</tr>
<tr>
<td>3.3.3 Results</td>
<td>15</td>
</tr>
<tr>
<td>3.4 Ultrasonic Range Finder Based Landing</td>
<td>16</td>
</tr>
<tr>
<td>3.4.1 Theory</td>
<td>16</td>
</tr>
<tr>
<td>3.4.2 Application</td>
<td>16</td>
</tr>
<tr>
<td>3.4.3 Results</td>
<td>18</td>
</tr>
<tr>
<td>3.5 Optic Flow Based Landing</td>
<td>18</td>
</tr>
<tr>
<td>3.5.1 Theory</td>
<td>18</td>
</tr>
<tr>
<td>3.5.2 Application</td>
<td>18</td>
</tr>
<tr>
<td>3.5.3 Results</td>
<td>21</td>
</tr>
<tr>
<td>3.6 Conclusion</td>
<td>22</td>
</tr>
<tr>
<td>4 Automatic Trim and Gain Tuning</td>
<td>23</td>
</tr>
<tr>
<td>4.1 UAV Model</td>
<td>24</td>
</tr>
<tr>
<td>4.2 Scheme A: nonlinear MRAC using backstepping</td>
<td>26</td>
</tr>
<tr>
<td>4.2.1 Pitch Attitude Hold</td>
<td>26</td>
</tr>
<tr>
<td>4.2.2 Roll Attitude Hold</td>
<td>29</td>
</tr>
<tr>
<td>4.3 Scheme B: nonlinear MRAC without backstepping</td>
<td>31</td>
</tr>
<tr>
<td>4.3.1 Pitch Attitude Hold</td>
<td>31</td>
</tr>
<tr>
<td>4.3.2 Roll Attitude Hold</td>
<td>33</td>
</tr>
<tr>
<td>4.4 Scheme C: nonlinear MRAC grouping terms</td>
<td>34</td>
</tr>
<tr>
<td>4.4.1 Pitch Attitude Hold</td>
<td>34</td>
</tr>
<tr>
<td>4.4.2 Roll Attitude Hold</td>
<td>36</td>
</tr>
<tr>
<td>4.5 Scheme D: nonlinear MRAC simple</td>
<td>38</td>
</tr>
<tr>
<td>4.5.1 Pitch Attitude Hold</td>
<td>38</td>
</tr>
<tr>
<td>4.5.2 Roll Attitude Hold</td>
<td>39</td>
</tr>
</tbody>
</table>
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.6 Flight Data</td>
<td>41</td>
</tr>
<tr>
<td>4.6.1 Platform</td>
<td>41</td>
</tr>
<tr>
<td>4.6.2 Implementation</td>
<td>42</td>
</tr>
<tr>
<td>4.6.3 Results</td>
<td>42</td>
</tr>
<tr>
<td>4.7 Conclusion</td>
<td>43</td>
</tr>
<tr>
<td>5 Students Funded Under Project</td>
<td>73</td>
</tr>
<tr>
<td>A Appendix: PNI Micromag Specifications</td>
<td>74</td>
</tr>
</tbody>
</table>
1 Objectives

During this project we focused on four primary objectives which are listed below.

1. **Magnetometer Integration.** Integrate and flight test magnetometers with the current version of the autopilot (Spiral 1).

2. **Height-Above-Ground Sensor Integration.** Integrate and flight test height-above-ground sensors with the current version of the autopilot (Spiral 1).

3. **Automatic Gain Adjustment.** Develop and flight test automatic gain adjustment algorithms that automatically tune the servo loops of the autopilot.

4. **Automatic Trim Seeking.** Develop and flight test automatic trim seeking algorithms to recursively estimate the trim values of the UAV.

The last two objectives were handled within one uniform framework and will therefore be described together.

2 Magnetometer Integration

Adding a magnetic compassing device to the sensing suite of the Kestrel Autopilot is very desirable for several reasons. The BYU autopilot uses GPS for an estimate of ground track. With the use of a magnetometer, true heading ($\psi$) can also be estimated. The use of true heading instead of ground track could significantly improve the accuracy of our gimbaled camera tracking system. Without true heading the gimbaled tracker will be in error by at least the value of the crab angle and the crab angle can be significant in high winds. In addition, magnetic heading allows for the computation of the wind vector which can easily be estimated in real time by using magnetometer heading, GPS velocity, airspeed, and ground track heading.

2.1 Magnetometer Selection

When first selecting a magnetometer to interface with the autopilot several models were considered including the Honeywell HMC 1022 2-axis analog magnetometer and the PNI Micromag 2-axis digital magnetometer. Both sensors have a footprint of less than 0.6 x 0.5 inches. However the PNI Micromag magnetometer was selected because it incorporates built-in temperature compensation and uses a SPI bus protocol that was straightforward to implement on the autopilot. The PNI Micromag also has selectable resolution and sample rate and the technical support at PNI is excellent. PNI developed a 3-axis version of the Micromag within several months which is currently being used on the autopilot. A table of Specifications and pin outs is included in Appendix A.

![Figure 1: PNI Micromag two-axis digital magnetometer](image)
2.2 Motor Noise

It was theorized that the magnetometer heading would be adversely influenced by induced magnetic fields from large brushless motors that are used for propulsion onboard Unmanned Air Vehicles (UAVs). Fortunately, it was found that motor noise could be mitigated by twisting and shielding the high current carrying wiring. The alternating nature of brushless motors causes high frequency low amplitude noise and can also introduce a bias on the magnetometer readings. The high frequency component of the noise can be filtered out digitally. The biasing component of the motor noise is correlated with motor current. Work is currently being done to remove the effects of the electric motor on the magnetometer readings. The magnetometer is mounted on the autopilot which is located about 4 to 6 inches from the motor. If possible this distance should be increased to further reduce the effects of noise. All current carrying wiring should also be placed as far from the magnetometer as possible in addition to being twisted and/or shielded.

2.3 Calibration

In order to use magnetometers effectively a good calibration must be employed. This is easier said than done. First the magnetometer must be placed in the UAV in its completed form. Every time that something metallic is added to the UAV or every time the magnetometer is relocated on the UAV a new calibration should be performed. The magnetometer should be mounted as far as feasibly possible from current carrying wires, ferrous metals, and magnetic components. Then a magnetically clean environment should be located for calibration. A magnetically clean environment is one that is distant from steel and iron structures, power lines, and magnetic items. As a rule of thumb the calibration should be performed at least 5 times the diameter away from the nearest ferrous and magnetic objects. For example, if a 10 foot automobile is nearby, move at least 50 feet away from the automobile before calibrating the magnetometers. Beware that underground power lines and nearby power transformers could significantly affect the calibration.

2.3.1 Ground Calibration

If it is intended to calibrate on the ground, the person performing the calibration should be sure to remove any metal objects and powered devices, such as keys and cell phones, from the area. Next, the ground station operator should set Calibrate Magnetometers to a value of 1. This starts logging data for the X and Y axes and stores the maximum and minimum values \((X_{\text{max}}, X_{\text{min}}, Y_{\text{max}}, Y_{\text{min}})\) for each axis. Then the person calibrating should hold the plane as level as possible while slowly turning in a circle in the horizontal plane through two complete rotations. Next the ground station operator should set Calibrate Magnetometers to a value of 2. This stops logging magnetometer data and calculates the scale factors \((X_{\text{scale}}, Y_{\text{scale}})\) and offset values \((X_{\text{offset}}, Y_{\text{offset}})\) for each axis.

To ensure good calibration it is recommended that the X and Y magnetometer data be plotted on the x-y plane and displayed for the operator. The X and Y magnetometer data should appear to be an ellipsoid. Applying the scale factors and offsets should circularize the ellipsoid and shift its center toward zero on the plot (see Figure 2).

For a 3-axis magnetometer the Z-axis should be calibrated separately. To do this the ground station operator should set Calibrate Magnetometers to 3. This starts searching for maximum and minimum Z-axis values \((Z_{\text{max}}, Z_{\text{min}})\). The airplane must then be rotated about either the X or Y-axis for a full rotation. It is also valid to only turn the aircraft completely upside-down, gathering the data points in both orientations.

Once the rotation is completed Calibrate Magnetometers should be set to 4 to calculate the Z scale factor and offset values \((Z_{\text{scale}} \text{ and } Z_{\text{offset}} \text{ respectively})\). The scale factors and offsets are calculated using the following equations:

\[
\begin{align*}
X_{\text{offset}} &= (X_{\text{max}} + X_{\text{min}}) / 2, \\
Y_{\text{offset}} &= (Y_{\text{max}} + Y_{\text{min}}) / 2, \\
Z_{\text{offset}} &= (Z_{\text{max}} + Z_{\text{min}}) / 2, \\
X_{\text{scale}} &= (X_{\text{max}} - X_{\text{min}}) / 2, \\
Y_{\text{scale}} &= (Y_{\text{max}} - Y_{\text{min}}) / 2, \\
Z_{\text{scale}} &= (Z_{\text{max}} - Z_{\text{min}}) / 2.
\end{align*}
\]

Once the calibration is complete be sure to save the calibration parameters in the autopilot flash memory and to a file.
Figure 2: Y-axis data plotted against X-axis data for a typical two-axis calibration. The uncalibrated data is in red while the data with the applied scaling and offset is blue.

2.3.2 Airborne Calibration

The most effective method for calibrating the magnetometer was found to be an airborne calibration. An airborne calibration places the UAV in an environment that is far from magnetic disturbances on the ground.

To calibrate the magnetometers in the air the first step is to set up an orbit point with a large orbit radius (between 200 and 300 meters). Next, get the UAV up in the air and flying in its commanded orbit. Then start the magnetometer $XY$ calibration by setting Calibrate Magnetometers to 1. This starts logging data for the $X$ and $Y$ axes and stores the maximum and minimum values ($X_{\text{max}}$, $X_{\text{min}}$, $Y_{\text{max}}$, $Y_{\text{min}}$) for each axis. Once the plane has completed at least one complete orbit, set Calibrate Magnetometer to 2 which stops logging magnetometer data and calculates the scale factors ($X_{\text{scale}}$, $Y_{\text{scale}}$) and offset values ($X_{\text{offset}}$, $Y_{\text{offset}}$) for each axis.

To calibrate the $Z$-axis, fly on an east-west flight path. Then set Calibrate Magnetometers to 3. Next, invert the UAV momentarily before returning to normal flight. Barrel rolls could also be performed to achieve the same goal. Finally, set Calibrate Magnetometer to 4. This completes the $Z$-axis calibration and calculates the $Z$ scale factor and offset values ($Z_{\text{scale}}$ and $Z_{\text{offset}}$ respectively).

As an alternative method for calibrating the $Z$-axis, set up a north-south flight path. Then set Calibrate Magnetometers to 3. Next perform a complete loop or split-S maneuver. Then set Calibrate Magnetometers 4. In either case the goal of the calibration routine is to capture the maximum and minimum values of all three axes. Be sure to save the calibration parameters in the autopilot flash memory and to a file.

Once the rotation is completed the scale factors and offsets are calculated using Equations (1).

2.4 Heading Estimation

2.4.1 2-Axis Heading Estimation

To calculate heading, raw magnetometer readings ($B_x$ and $B_y$) are adjusted using the scale factors and offsets calculated during calibration using the following equations:

\[
B_y' = \frac{B_y - Y_0}{Y_g} \\
B_x' = \frac{B_x - X_0}{X_g}
\]
2.4 Heading Estimation

Figure 3: 3-axis heading (red) compared to 2-axis heading (green) along with corresponding roll ($\phi$) and pitch ($\theta$) angles taken during a clockwise circular orbit.

To estimate heading the following formula is used:

$$\psi = \text{atan2}(B'_y, B'_x) + \text{declination}$$

The heading estimate on a 2-axis magnetometer is only valid when the magnetometer is oriented level. A 2-axis magnetometer will be in error by as much as 20 degrees with roll and pitch angles of up to 15 degrees (see Figure 3).

2.4.2 3-Axis Heading Estimation

To calculate heading, raw magnetometer readings ($B_x$, $B_y$, and $B_z$) are adjusted using the scale factors and offsets calculated during calibration, where $B'_x$, $B'_y$, and $B'_z$ are the now calibrated $x$, $y$, and $z$ components of the magnetic field, i.e.,

$$B'_x = \frac{B_x - X_0}{X_g}$$
$$B'_y = \frac{B_y - Y_0}{Y_g}$$
$$B'_z = \frac{B_z - Z_0}{Z_g}.$$

Then the adjusted magnetometer readings are un-rolled and un-pitched using standard rotation matrices as follows:

$$\begin{pmatrix} X_h \\ Y_h \\ Z_h \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{pmatrix} \begin{pmatrix} B'_x \\ B'_y \\ B'_z \end{pmatrix}. \tag{3}$$

The un-pitched and un-rolled horizontal components of the magnetic field then become:

$$X_h = B'_x \cos \theta - B'_y \sin \theta \sin \phi - B'_z \cos \phi \sin \theta$$
$$Y_h = B'_y \cos \phi - B'_z \sin \phi.$$

From these equations heading is computed as:

$$\psi = \text{atan2}(Y_h, X_h) + \text{declination}.$$
2.5 2-Axis Tilt Compensation

A 2-axis magnetometer can be pitch and roll compensated to achieve results comparable to those of a 3-axis magnetometer by estimating the $z$ component of the magnetometer. This is done by using additional information that is known about the Earth’s magnetic field such as inclination and magnitude of the Earth’s field. Inclination is the angle between the Earth’s surface and the total magnetic field vector. This angle, often referred to as the dip angle, mainly varies depending on latitude. Inclination is not to be confused with declination which is the angular difference between true north and magnetic north. The magnitude of the Earth’s magnetic field is relatively constant, however, aperiodic variation of earth’s magnetic field over time and place does occur.

The $Z$-axis component can be estimated using the following expression:

$$
\tilde{B}_z = \frac{\sin \lambda B_{\text{Total}} + \sin \theta B_x' - \cos \theta \sin \phi B_y'}{\cos \theta \cos \phi},
$$

where $\lambda$ is the inclination and $B_{\text{Total}}$ is the magnitude of the Earth’s magnetic field. Both of these values are only valid in the region where the magnetometer is to be used. In the case of the PNI magnetometer which outputs readings in terms of ADC counts rather than in Tesla or Gauss, The value for $B_{\text{Total}}$ must also be input in terms of ADC counts.

After estimating the $z$ component, the adjusted magnetometer readings are un-rolled and un-pitched using Equation (3) with $B_x'$ replaced by $\tilde{B}_z$.

This method has been used successfully to estimate the actual heading of the UAV even while maneuvering at pitch and roll angles of over 35 degrees (see figure 5).

2.6 Magnetometer Wind Estimation

Knowledge of true heading and ground track heading as well as airspeed and ground speed allows for the computation of the wind vector which can easily be estimated in real time. Using what is called the wind triangle, the wind speed and wind direction can both be found using simple trigonometry (see figure 6).

Using the wind triangle to subtract the air velocity vector from the ground velocity vector yields the wind vector. This method has accurately measured the wind speed and wind direction in flight on a UAV (see figure 7).
2.6 Magnetometer Wind Estimation

Figure 5: A simulation comparison of all three methods discussed for estimating psi including using a 3-axis mag (3D Mag), 2-axis mag (2D Mag), and 2-axis tilt compensation by estimating the Z-axis (2D Mag Est 3D Mag). Corresponding roll and pitch angles are given in the lower plot.

Figure 6: Wind triangle showing that the addition of the air velocity vector and wind vectors should equal the ground velocity vector.

Additional magnetometer data is shown in Figure 8 and 9.
Figure 7: Top: Estimated wind direction. Middle: Estimated wind speed. Bottom: GPS Course and Magnetometer heading. The actual wind speed and direction as reported on day of flight were 2 m/s from the north-west (315 degrees).

Figure 8: Top: Magnetometer flight data of $X$, $Y$, and $Z$ axes. Middle: Total magnetic field vector calculated from $X$, $Y$ and $Z$ data. It is relatively constant which indicates that the magnetometer did not enter the presence of any significant magnetic disturbances. Bottom: Roll ($\phi$) and Pitch ($\theta$) data corresponding to the collected magnetometer data in flight.
Figure 9: Heading estimates in a circular orbit using both 2 and 3 axis magnetometers.
3 Height Above Ground Sensor

The usefulness of mini and micro unmanned air vehicles (UAVs) depends largely on the ability to hand launch and recover the UAV in uncontrolled environments. Efforts to implement effective autonomous landing of mini UAVs in uncontrolled environments have been most hindered by lack of small, lightweight, effective sensors to determine height above ground (HAG). Because of the lack of a suitable sensor for determining HAG, small UAVs have traditionally relied solely on the barometric altimeters with which they are already equipped for HAG estimation. This approach works well for controlled environments in which the altitude of the landing environment relative to the altitude of the takeoff environment is known, and the terrain is relatively smooth. However, in uncontrolled environments where this offset is not known, or where the terrain is not acceptably smooth, other methods must be utilized to effectively determine HAG.

Two methods with acceptably small sensor payloads have been proposed as effective means of HAG estimation. These methods are:

1. Estimation of HAG via Ultrasonic Ranging
2. Estimation of HAG via Optic Flow

It is proposed that these sensors can be combined with information from the barometric altimeter to reliably and accurately determine HAG. This accurate estimate of HAG could then be combined with robust control algorithms to effectively and safely land small UAVs in uncontrolled environments. This report presents the experimental results of efforts to estimate HAG and land small fixed-wing UAVs in uncontrolled environments using each of the sensors outlined above. Because each of the sensors tested will couple with the barometric altimeter in determining HAG, the experimental results will be presented as follows:

1. Overview of the landing algorithm,
2. Landing results using only the barometric altimeter,
3. Landing results using an ultrasonic sensor combined with the barometric altimeter,
4. Landing results using an optic flow sensor combined with the barometric altimeter,
5. Comparison of results using each of the sensors.

Each of these sections will address the specific sensor tested and the algorithms used to determine HAG from the data provided by the sensor.

3.1 Landing Algorithm

3.1.1 Overview

The landing algorithm developed and tested for this project consists of two user defined waypoints: the approach point and the landing point (Figure 10).

![Landing Algorithm Diagram](image)

Figure 10: Landing Algorithm

The approach point serves as a point about which the airplane may orbit until it reaches a specified altitude, at which point it will break out of its orbit and follow a glide slope to the landing point. The approach point is fully defined by the following parameters:
3.2 Orbit Routine

1. Airspeed – The desired airspeed which will be maintained throughout the circle-down portion of the landing.

2. Radius – The radius of the orbit which will be maintained throughout the circle-down portion of the landing.

3. Start Altitude – The altitude to which the plane will attempt to climb or descend while en route to the approach point. The true start altitude for the descent will be the altitude to which the plane has successfully climbed or descended by the time it is within 1.5 orbit radii of the landing point.

4. End Altitude - The altitude at which the airplane will break out of its orbit and begin its glide slope in to the landing point.

5. Relative East Coordinate - The distance east of home (in meters) around which the plane will orbit while descending from its start altitude to its end altitude.

6. Relative North Coordinate - The distance north of home (in meters) around which the plane will orbit while descending from it start altitude to its end altitude.

7. Descent Rate - The rate (in meters/second) at which the plane will descend while orbiting the approach point.

The landing point is fully defined by the following parameters:

1. Airspeed - The airspeed which will be maintained throughout the glide slope from approach point to landing point.

2. Flare Height - The height above ground at which the airplane will cut throttle and attempt to hold zero roll while maintaining its glide slope by commanding pitch to control around desired altitude.

3. Relative East Coordinate - The distance east of home (in meters) at which the plane will touch down.

4. Relative North Coordinate - The distance north of home (in meters) at which the plane will touch down.

Altitude control throughout the landing routine is accomplished through activation of various PID loops. For the portion of the landing sequence in which the plane is en route from its last waypoint to the approach point altitude is maintained as follows:

1. Actual altitude is within a preset distance (usually around 15 m) of desired altitude - Pitch from Altitude, Throttle from Airspeed.

2. Actual altitude is greater than the same preset distance from desired altitude and is less than desired altitude - Pitch from Airspeed, Full Throttle.

3. Actual altitude is greater than the same preset distance from desired altitude and is greater than desired altitude - Pitch from Airspeed, Zero Throttle.

For the portion of the landing sequence in which the plane is circling down to its end altitude and then following its glide slope to the landing point altitude is maintained using the pitch from altitude and throttle from airspeed loops. Throughout the entire landing sequence, until the flare at which desired roll is held at zero, heading is controlled using the roll from heading PID loop. Roll rate and pitch rate are also controlled to provide damping. Desired headings are commanded based on the generation of a heading field which forms a limit cycle around the desired orbit radius. A full description of this algorithm is provided in the following section.

3.2 Orbit Routine

3.2.1 Heading Field Generation

The ability to implement the landing routine described in the previous section required the development and implementation of a control algorithm which would allow the airplane to fly accurate orbits, at minimal orbit radii, even in high wind conditions.

The orbit algorithm implemented generates a desired GPS heading based on the position of the airplane such that the corresponding heading field forms a limit cycle around the desired orbit radius (Figure 11). Such a heading field is generated using Equations (4).
3.2 Orbit Routine

Figure 11: Typical Heading Field for a 50 Meter Radius Orbit (Desired Orbit Radius Indicated by Solid Circle).

\[
\chi_d = \begin{cases} 
\chi_{\text{center}} - \tan^{-1} \left( \frac{d}{r_c} \right), & d \geq 2r_c \\
\chi_{\text{center}} - \alpha - \beta r, & r_c \leq d \leq 2r_c \\
\chi_{\text{center}} + \alpha + \pi + \beta r, & d < r_c
\end{cases}
\]

In the above equations \( d \) is the distance from the center of the orbit, \( \chi_d \) is the desired course, \( \chi_{\text{center}} \) is the compass heading to the center of the orbit, \( r_c \) is the desired orbit radius, \( \alpha \) is the included angle between \( \chi_d \) and \( \chi_{\text{center}} \) for \( d = 2r_c \), \( \beta \) is the complimentary angle to \( \alpha \), and \( \gamma \) is a gain corresponding to the field strength of the orbit. Increasing \( \gamma \) makes the field more reactive around the desired orbit radius and decreasing \( \gamma \) makes the field less reactive as shown in Figure 12. \( \tau \) is a parameterization of the distance from the center of the orbit based on \( r_c \). The formulation of \( \tau \) is such that \( 0 \leq \tau \leq 1 \) and \( \tau = 1 \) when \( d = r_c \). The value of \( \tau \) is calculated according to Equations (5).

\[
\tau = \begin{cases} 
1 - \frac{d-r_c}{r_c}, & r_c \leq d \leq 2r_c \\
\frac{d}{r_c}, & d < r_c
\end{cases}
\]

Figure 12: High Gain Heading Field (Left) and Low Gain Heading Field (Right) for a 50 Meter Radius Orbit (Desired Orbit Radius Indicated by Solid Circle).

The desired heading from the heading field described by equations (4) is used to generate a roll command for the airplane according to

\[
\phi_d = k(\chi_d - \chi_a)
\]

11
3.2 Orbit Routine

3.2.2 Correcting Steady State Error

Because the heading control represents a type one system and flying an orbit based on desired heading is essentially tracking a ramp in heading, the actual course of the airplane tracks the desired course with a steady state error. This steady state error in course represents a lag between desired and actual course which translates to actual orbits that are larger than the commanded orbit. The magnitude of the steady state error in orbit radius is a function of the desired radius of the orbit, the groundspeed of the airplane, and the heading field gain $\gamma$. Two different approaches were used to overcome this effect. The first approach involved adapting the radius command to remove the steady state error. The second approach, which was more effective in high winds, involved calculating a nominal roll angle based on the desired heading rate and the coordinated turn equation.

Adaptive Radius Commands  Because the actual orbit radius of the airplane is always exterior to the commanded radius, for every dynamically feasible orbit there exists some smaller radius command that will result in the actual desired orbit. The adaptive radius command progressively increases or decreases an offset between the desired orbit radius entered by the operator and the desired orbit radius passed to the orbit algorithm. This offset is continuously incremented or decremented until the plane is orbiting at the radius requested by the operator. Orbit results with and without the adaptive radius command are shown below in Figure 13. Telemetry from actual flight tests of the adaptive radius method are shown in Figure 14.

![Figure 13: Circular Orbits (From Simulation) with and without an adaptive radius command.](image)

Nominal Roll Angle Based on Desired Heading Rate  The adaptive radius command approach works extremely well for situations where wind speed does not exceed 50% of airspeed. However, high winds cause the groundspeed of the airplane to vary significantly based on the course of the airplane relative to the wind direction. This variation in groundspeed causes a corresponding variation in the radius offset necessary to maintain the desired orbit.

A better approach for high wind conditions was developed which involved calculating a nominal roll angle that resulted in the heading rate necessary to fly the desired orbit. The heading field was then used to give roll commands about this nominal roll value.

Calculating the nominal roll angle first required taking the derivative of the heading field with respect to time. This resulted in the formation of a heading rate field over the same space as the heading field. The equations for the heading rate field are given by

$$\dot{\chi}_d = \frac{V r_c \left( r_c^2 + 2d^2 \right)}{d \left( r_c^2 + d^2 \right)^{\frac{3}{2}}}$$

(7)

$$\dot{\chi}_d = \frac{V}{d} \sin \left( \alpha + \beta \gamma \right) + \left( \frac{\gamma \beta V \cos \left( \alpha + \beta \gamma \right)}{r_c} \right)$$

(8)

$$\dot{\chi}_d = \frac{V}{d} \left( \sin \left( \alpha + \beta \gamma \right) - \gamma \beta \cos \left( \alpha + \beta \gamma \right) \right)$$

(9)

An example of a heading rate field for a given heading field strength and constant groundspeed is shown in Figure 15.

Equations (4) combined with Equations (7)–(9) allow the calculation of both a desired heading and a desired heading rate for any location relative to the center of the orbit. The desired heading rate can then be converted to a
nominal roll angle using the standard coordinated turn equation

\[ \phi_{\text{nom}} = \frac{\dot{x}gV}{g} \]  

and the control law described in Equation (6) can be adapted as

\[ \phi_{\text{nom}} = \frac{\dot{x}gV}{g} \]

\[ \phi_d = \phi_{\text{nom}} + k(\chi_d - \chi_a) \]  

The control law shown in Equation (12) is more effective at tracking orbits in high winds, but somewhat less steady at tracking orbits in low wind conditions than is the adaptive radius command method. A comparison of the two methods flying a 100 meter radius orbit in simulation with a constant wind speed of 8 m/s is shown in Figure 16.

3.2.3 Glide Slope

The glide slope implemented in this landing routine was a simple linear glide slope calculated by decreasing desired altitude as a function of the distance from which the glide slope began. This formulation allows the desired altitude to be driven negative for distances from the initiation of the glide slope which are farther than the desired landing point. This makes the glide slope robust with respect to negative offsets between measured HAG and actual HAG. Expected overshoot or undershoot of the landing point can approximated as

\[ \text{Expected Over / Undershoot} \approx \frac{1}{\tan \theta} \left( \tilde{a} + \tilde{b} \right), \]  

where \( \theta \) is the glide slope, \( \tilde{a} \) is the altitude control error (difference between desired and actual HAG), and \( \tilde{b} \) is the measurement error. The relationship shown in Equation (13) indicates that larger glide slopes will tend to reduce the overall overshoot or undershoot error for the landing provided the plane is capable of matching the commanded glide
3.3 Barometric Altimeter Based Landing

3.3.1 Theory

The Kestrel autopilot comes equipped with a barometric pressure sensor as part of its standard sensor suite. Before a given flight, the barometric pressure sensor is zeroed such that it references the barometric pressure at the takeoff location as the pressure corresponding to zero altitude. Pressure decreases with altitude according to

\[ P = P_{\text{ref}} - \int_{h_{\text{ref}}}^{h} \rho g h \, dh. \]  

(14)

If the density of air (\( \rho \)) is assumed to be sufficiently constant over the range of altitudes being measured, then altitude above the takeoff point can be determined by

\[ h = \frac{P_{\text{ref}} - P_{\text{actual}}}{\rho g} = \alpha (P_{\text{ref}} - P_{\text{actual}}), \]  

(15)

where the constant of proportionality (\( \alpha \)) can be determined by taking readings of the pressure sensor at a number of different known reference altitudes and fitting a straight line to the data set. Once the constant of proportionality has been determined, readings of the pressure sensor from the analog to digital converter can be converted directly into relative altitudes above the takeoff point. This sort of calibration allows the barometric pressure sensor to provide valid and accurate estimates of HAG where the terrain is sufficiently flat. Where this is not true, barometric pressure can still provide a rough estimate of HAG until other sensors are able to collect valid readings of true HAG. After another sensor has reported a valid reading of true HAG variations in barometric pressure from the pressure at which true HAG was determined can be used to further estimate the true HAG.

3.3.2 Application

Temperature Compensation The pressures reported by the pressure sensor on the Kestrel autopilot proved to be somewhat sensitive to the temperature of the sensor. Particularly this would become a problem when the autopilot was allowed to heat up as it sat static during calibration and programming, and then cooled as airflow over the airfoil dissipated heat during flight. This calibration at high temperature and subsequent cooling would cause the autopilot to consistently read approximately 20 meters high. The increase in pressure readings with decreased sensor temperature was empirically determined to be linear over the range of operating temperatures. The constant of proportionality was likewise empirically determined and implemented as

\[ P_{\text{actual}} = P_{\text{measured}} + \beta (T_{\text{measured}} - T_{\text{ref}}) \]  

(16)
3.3 Barometric Altimeter Based Landing

Figure 16: A Comparison of Nominal Roll Angle and Adaptive Radius Based Methods (Simulation).

This temperature compensation makes the pressure readings from the autopilot robust to the variations in temperature that are encountered in standard autopilot operation.

**Compensation for Airflow Over the Airfoil** Another source of error in the readings from the autopilot’s pressure sensor stems from the fact that the MACICC Lab airplanes are not normally equipped with static pressure tubes. While in flight a considerable offset in the absolute pressure measured by the pressure sensor is introduced by airflow over the airfoil in which the autopilot is housed. The increased speed of the airflow over the airfoil results in a Bernoulli effect that causes the pressure measured along the airfoil to be less than the true absolute pressure. This low pressure region in which the actual pressure measurements are taken causes the altitude measurement computed by the autopilot to consistently read high. The relationship between the velocity of the airflow over the airfoil and the induced offset in the pressure reading was determined empirically through wind tunnel testing to be a second order function of airspeed. This functional relationship can be combined with the temperature correction term in Equation (16) to calculate pressure as

\[
P_{\text{actual}} = P_{\text{measured}} + C_1 v_{\text{diffpres}}^2 + C_2 v_{\text{diffpres}} + \beta (T_{\text{measured}} - T_{\text{ref}})
\]  

(17)

where \(v_{\text{diffpres}}\) is the airspeed past the airplane as measured by the airplane’s pitot probe attached to a differential pressure port. Once the constants \((C_1, C_2, \beta)\) have been empirically determined, the offset can be computed each time through the loop, and the barometric altitude reading can be adjusted accordingly. Testing has shown the constants determined for one airplane to be valid for other airplanes of a similar type regardless of slight variations in the airfoil and in autopilot location. This allows the constants for a fleet of similar airplanes to be determined through a rigorous wind tunnel calibration of a single airplane. Further testing is required to determine how robust these constants are with respect to major changes in airframe or autopilot location. This method of compensating for airflow over the airfoil allows a large number of similar airplanes to be flown, using the exact same constants, without each being equipped with static tubes which can add unnecessary weight and complication.

3.3.3 Results

Using the calibration described above and testing over relatively short flights, the MAGICC Lab airplanes consistently read within 1 - 1.5 meters of the actual altitude relative to home. This allows relatively accurate landings where the altitude of the landing point can be safely assumed to be the same as the altitude of home. This works especially well when trying to land the airplane at or near its home location which is most often the case.
If the actual altitude of the airplane matched the desired altitude perfectly, then overshoot or undershoot would be solely a function of error in HAG estimation. For glide slopes of 7 to 12 degrees and a measurement error of 1 - 1.5 meters, a corresponding overshoot or undershoot error of around 4 - 12 meters can be expected (see Equation (13)).

Actual results for 26 consecutive landing runs are shown below in Figure 17. The first 22 autoland runs were each from the main approach point. These were followed by one run apiece from each of the four alternate approach points.

![Figure 17: Results of Twenty-six Consecutive Autolands from Five Different Approach Points.](image)

An average overshoot or undershoot error of 4 - 12 meters was estimated. Actual results from these 26 landing show an average over/undershoot error of 7.6 meters with a standard deviation of 5.4 meters and a median value of 6 meters.

### 3.4 Ultrasonic Range Finder Based Landing

#### 3.4.1 Theory

Ultrasonic sonar transducers determine distance from an object by measuring the time of flight of an ultrasonic signal which is emitted by the transducer and then reflected by the object. The speed of sound in an ideal gas is given by

\[ c = \sqrt{\kappa RT}, \tag{18} \]

where \( \kappa \) is the adiabatic constant (1.402 for air), \( R \) is the universal gas constant (297.05 kJ/(kg K) for air), and \( T \) is the absolute temperature in Kelvin.

Linearizing Equation (18) around 0 degrees C gives a convenient approximation for the speed of sound as

\[ c_{\text{air}} = 331.5 + 0.6T_c. \tag{19} \]

#### 3.4.2 Application

**Sensor** The MUST01 Ultrasonic Sonar Transducer manufactured by Mekatronix was selected and used as the ultrasonic ranging module and transducer for these tests. This sensor provided both the ranging board and transducer in one integrated package as well as allowing a sensing range of up to 12.2 meters with a resolution of 4.8 cm. The entire unit has an overall diameter of 4.3 centimeters and a total mass of 17 grams. The sensor emits a 40 - 50 kHz sound wave and has a nominal beam angle of 15 degrees.

The MUST01 unit emits a signal 10 times a second and returns an analog voltage proportional to each signal’s time of flight. This value is alpha filtered on the ranging board according to

\[ V_{\text{out}} = 0.9V_{\text{new}} + 0.1V_{\text{textlast}}. \tag{20} \]
The voltages from the ultrasonic sensor are read into the autopilot via the autopilot’s analog to digital converter. The ultrasonic sensor was calibrated by holding the sensor at a number of fixed distances from a surface with good acoustical properties and then recording the analog to digital converter values corresponding to that distance. The A/D values were then related directly to their corresponding distances through a constant of proportionality. This constant was determined through a least-squares linear fit of the data.

**Compensation for Airplane Attitude** Because the ultrasonic sensor has a nominal beam angle of 15 degrees and reports the closest item from which it receives a valid return signal, the reading from the ultrasonic sensor may not always reflect HAG depending on the attitude of the airplane. If the airplane is either rolled or pitched sharply the signal emitted by the ultrasonic sensor will reflect off the ground at some distance significantly displaced from the point located directly beneath the airplane. The distance from the airplane to this displaced point will be significantly greater than the distance from the airplane to the point directly beneath it. Thus this reading will be significantly greater than the airplane’s actual HAG. However, if the attitude of the airplane is known, and the earth can be assumed to be sufficiently flat over the span from the point directly below the airplane to the point at which the signal is reflected, then the actual HAG can still be determined by

\[
h_{\text{actual}} = h_{\text{measured}} \cos (|\theta| - \gamma) \cos (|\phi| - \gamma) + d \sin \phi,
\]

where \(\gamma\) is one half the nominal beam angle, \(\theta\) and \(\phi\) refer to pitch angle and roll angle respectively, and \(d\) is the lateral distance from the center of the airplane at which the sensor is mounted. Equation (21) is only valid, and only necessary, when either \(\theta\), \(\phi\), or both are greater than \(\gamma\). It is important to note that when this equation must be used to determine HAG, the maximum distance at which true HAG can be accurately determined becomes a dynamic function of airplane attitude rather than a constant (12.2 meters for the sensor tested). The airplane attitude dependant value of maximum distance at which true HAG can be accurately determined can be found setting \(h_{\text{measured}}\) equal to the maximum distance at which true HAG can be determined for level flight, and then solving Equation (21) for \(h_{\text{actual}}\). Another important consideration when using an ultrasonic sensor to determine HAG is that there will be an angle of incidence associated with every surface at which the ultrasonic signal will not be reflected back toward the sensor. Equation (21) is not valid if the angle of incidence is greater than this angle. Instead there is a discontinuity at which the maximum value of true HAG that can be determined goes immediately to zero because no return signal is received. Flight tests have shown that for all outdoor environments tested this angle of incidence is acceptably small such that this discontinuity is not encountered under normal flight conditions.

**Limitations Due to Propeller Noise** A major limitation of the ultrasonic sensor is its tendency to pick up false readings due to noise from the propeller. For the MAGICC Lab airplanes with the motor running at full throttle to near full throttle any reading over 8 meters would be rejected in favor of noise from the propeller. This makes it impossible to trust readings from the ultrasonic sensor at a range greater than 8 meters. For our testing we set the cutoff at which we would trust readings from the ultrasonic sensor at 7.9 meters. This severely limits the usefulness
of the ultrasonic sensor by severely limiting its maximum attainable range. This also introduces uncertainty as to whether the sensor will be more or less susceptible to noise from a different combination of motor and propeller. The sensor’s maximum range then becomes very platform dependent rather than strictly sensor dependent. The magnitude of the noise which the sensor picks up has been minimized by placing the sensor on the far wingtip of the airplane, as far from the motor and propeller as possible. This, however, introduces a significant payload to the extremity of the airplane causing balancing issues. Also, if the sensor were to be used on an airplane where it could not be mounted a significant distance from the propeller, a significant decrease in maximum range should be expected.

Environmental Factors  In order for the ultrasonic sensor to determine distances it must transmit and then receive back a signal of acceptable magnitude. For the case of the airplane this signal must be of acceptable magnitude such that it will be accepted by the sensor as the significant signal over noise from the propeller. Testing has shown that this is extremely reliable for flights over surfaces that are sufficiently acoustically reflective such as streets or well-trimmed yards. However, for flights over more dense foliage (fields with tall grass for example) the signal can be sufficiently muffled such that the sensor does not receive an acceptably large return signal until it is 1 - 2 meters off the ground. Thus maximum attainable range for the ultrasonic sensor is not only dependant on the particular airframe it is flown in (especially sensor placement on that airframe) and that airplane’s attitude, but it is also highly dependant on the acoustical properties of its flight environment.

3.4.3 Results

In order for the ultrasonic sensor to facilitate autonomous landing in truly uncontrolled environments where landing altitude relative to takeoff altitude is not known, it must allow the airplane to circle down at a steady rate until true HAG is detected, and then break out of its orbit and follow a glide slope to the landing point. The ultrasonic sensor was not able to reliably perform this task. The sensor’s extremely limited range coupled with loss of ultrasonic readings over dense foliage and loss of ultrasonic readings due to airplane attitude were the main contributing factors to the sensor’s unreliability.

3.5 Optic Flow Based Landing

3.5.1 Theory

Optic flow sensors can be used to determine HAG by relating the flow of features across an imaging array to the speed the imaging array is moving past the surface it is imaging and its distance from that surface. The number of pixels a given object moves in the imaging plane can be combined with data from the airplane’s IMU and GPS to determine HAG according to [1]

\[ h = \frac{\delta x}{2 \tan \left( \frac{\gamma \text{fov}}{2p_n} - \frac{\dot{\theta} t_s}{2} \right)} \cos \theta \cos \phi, \]  

(22)

where \( h \) represents HAG, \( \delta x \) represents the displacement of the sensor parallel to the plane being imaged over the sample time \( t_s \), \( \gamma \) represents the average number of pixels of displacement of features across the imaging plane over the same sample time, 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, \( \dot{\theta} \) is the average pitch rate of the sensor over the sample period, \( \theta \) is the average pitch of the sensor over the sample period, and \( \phi \) is the average roll of the sensor over the sample period.

3.5.2 Application

Sensor  Testing of autonomous landing using optic flow to determine HAG was performed using Agilent’s ADNS-2610 optic flow sensor. The ADNS-2610 has a small form factor, measuring only 10 x 12.5 mm and runs at 1500 frames per second. It requires a light intensity of at least 80 mW/m^2 at a wavelength of 639 nm or 100 mW/m^2 at a wavelength of 875 nm. The ADNS-2610 measures the flow of features across an 18 x 18 CMOS imager. It outputs two values, \( dx \) and \( dy \), representing the total optic flow across the sensor’s field of view in both the \( x \) and \( y \) directions respectively. These values are stored in a buffer which can store accumulated 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 flow data in the camera \( y \) direction corresponds to lateral motion of the airplane and for purposes of establishing HAG can be ignored. The flow data in the camera...
3.5 Optic Flow Based Landing

$x$ direction can be combined with data from the IMU, GPS, and the field of view of the sensor to determine HAG according to Equation (22).

**Optics**

A critical consideration in outfitting an airplane with an optic flow sensor is the setup of the optics. Narrow angle lenses increase the distance at which the airplane will be able to pick up appreciable optic flow. This increases the distance at which the sensor will be able to accurately measure HAG for a given groundspeed. This comes at the expense of increasing the longitudinal size 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 $dx$ register or by causing the angular flow rate seen by the sensor to become too "fast" for the sensor to register. 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 airplane in pitch and roll. In addition to focal length, lens diameter is an important consideration. This is especially important with regard to the corresponding f-stop value of the lens. The f-stop is defined as the ratio of a 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 - 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. Because the sensor automatically adjusts shutter speed to keep the light intensity at an ideal level, there is no disadvantage to selecting a lens with a lower f-stop other than 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 longitudinal size of the sensor, but also increase the lateral size 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 of the configurations is shown in Figure 19.

![Figure 19: 1.2, 2.5, and 6.5 degree Field of View Optics Configurations.](image)

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 the slightly greater range while performing better over feature poor surfaces such as asphalt. The one disadvantage of the 1.2 degree field of view setup was its increased susceptibility to noise caused by pitch and roll oscillations.

**Sample Rate**

Another critical consideration when equipping an airplane with an optic flow sensor is the selection of sample rate. The resolution of the sensor (measured in units of distance per sensor count) is given by

$$r = \lim_{\delta h \to 0} \left( \frac{\delta h}{\tan^{-1} \left( \frac{V_s}{2h - \delta h \tan^{-1} \left( \frac{V_s}{2h} \right)} \right)} \right).$$

(23)
For a given sensor with a fixed field of view the sensor’s resolution is a function of sample rate, groundspeed, and HAG. Plots of sensor resolution for a fixed field of view and varying sample rates and groundspeeds are shown in Figure 20.

Figure 20: Plots of Resolution as a function of HAG and Groundspeed and HAG and Sample Rate.

For high values of \( r \) each sensor counts corresponds to a relatively large distance measure. Correspondingly the noise on the sensor measured in sensor counts is amplified by \( r \) into higher magnitude noise in the HAG measurement. This makes large values of \( r \) 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 20. This, however, can be undesirable because it decreases the sensor’s response time and can lead to overflow in the sensor’s \( dz \) register. Furthermore, for a fixed sample rate, maintaining a low value of \( r \) for relatively large values of HAG results in unnecessarily low values of \( r \) for lower values of HAG. This means unnecessarily low sample rates are being maintained where significantly faster sample rates could be used without significant loss in signal to noise ratio. Such a approach also results in premature sensor overflow and corresponding loss of low end range. To solve these problems two approaches were developed using a dynamic rather than static sample rate. The first approach maintains an \( r \) value along a specified resolution curve regardless of groundspeed of the plane. The second approach uses the current velocity estimate and HAG estimate to maintain a constant value of \( r \) regardless of groundspeed and HAG.

**Constant Curve Resolution** In general, flight control systems for UAVs utilize a feedback loop to control around a commanded airspeed rather than groundspeed. Even when the feedback loop is closed around groundspeed, the commanded groundspeed may or may not be realizable due to wind conditions. In either case in the presence of wind it is reasonable to assume that groundspeed may vary significantly depending on the course of the airplane relative to the wind direction. As shown in Figure 20 variations in groundspeed can alter the resolution curve significantly. However, if the sample rate is set according to

\[
t_s = \frac{\tau}{V}, \tag{24}
\]

where \( \tau \) represents the sample rate divisor, the value of \( r \) will follow a fixed resolution curve as a univariate function of HAG. This is useful because it eliminates the variations in resolution due to velocity and provides reliable repeatable resolutions regardless of wind conditions or commanded airspeeds.

**Constant Value Resolution** When an estimate of the current HAG is available, the correct sample rate can be calculated such that the sensor resolution can be maintained constant regardless of HAG or groundspeed. This sample rate is found by solving Equation (23) for \( t_s \). The result is

\[
t_s = \lim_{\delta_h \to 0} \left( \frac{1}{\frac{V^2}{2(h-\delta_h)} - \frac{V}{2h}} \right), \tag{25}
\]

where \( \tau \) now represents a user defined parameter that defines the resolution of the sensor for all combinations of HAG and groundspeed. In practice, the actual value of \( t_s \) is capped at both the high and low end, but allowed to change in order to maintain a fixed resolution inside of some saturation block. In flight tests using a dynamic sample rate to fix sensor resolution cut noise drastically for high values of HAG while significantly increasing the sample rate and thus
improving the response time for low values of HAG. In practice Equation (25) is implemented by dropping the limit expression and fixing \( \delta_h \) to some reasonably small number.

Flight test results using both constant curve resolution and constant value resolution are shown in Figure 21. The spike in the optic flow measurement at sample 400 in the constant value resolution plot and 275 in the constant curve resolution plot corresponds to flight over the asphalt road. In practice such spikes are ignored by ignoring values for which the optic flow sensors reports poor surface quality.

![Figure 21: Optic HAG Results from Glideslope Descents Using Fixed Value Resolution (Left) and Fixed Curve Resolution (Right).](image)

### 3.5.3 Results

**Ranging** The accuracy of the distance measurements calculated from optic flow was tested by comparison with a laser rangefinder. Testing was performed by mounting the optic flow sensor and 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 calculations and the laser rangefinder. The results of this test are shown in Figure 22. The results show very close agreement and validate the readings obtained from the optic flow calculations.

![Figure 22: Range Data from Optic Flow Calculations Compared to Laser Rangefinder Data.](image)

Range data from the optic flow sensor in flight tests showed reliable, repeatable readings for any height above ground less than 40 meters. For the purposes of landing based on HAG calculations from optic flow, values above 40 meters were discarded as noise.
3.6 Conclusion

Using the fixed value resolution sample rate and the 2.65 degree field of view optics, HAG measurements based on optic flow were used to perform twenty-seven consecutive autonomous landings from a single approach point. The results are shown in Figure 19.

For each of these autonomous landings HAG was calculated according to

\[ h = h_{\text{bar}} + h_{\text{offset}} \]  
\[ h_{\text{offset}} = \alpha (h_{\text{bar}} - h_{\text{optic}}) + (1 - \alpha) h_{\text{offset}}. \]

The value of \( h \) was updated every time through the loop and the value of \( h_{\text{offset}} \) was updated every \( t_s \) seconds where \( t_s \) varied with the current HAG estimate and groundspeed.

The readings from the barometric pressure sensor were allowed to drift over the twenty-seven landings in order to simulate an unknown landing point altitude. The barometric altimeter drifted to 15 meters of inaccuracy over the first 22 landings and was then rezeroed to introduce negative offsets between true HAG and barometric altitude. The value of \( h_{\text{offset}} \) (corresponding to the drift in the barometric altimeter) for each of the landings is shown in the third subplot of Figure 19. The second subplot of Figure 23 shows the distance from the landing point for each of the twenty-seven landings.

![Comparison of HAG and Barometric Altitude Over Relatively Flat Terrain](image)

Figure 23: Results of 27 Consecutive Optic Flow Based Autonomous Landings.

The average distance between the desired and actual touchdown points for autonomous landings using optic flow data was 4.3 meters. This represents a 56% improvement over the average distance from the desired touchdown point using only the barometric altimeter. Landings based on HAG from optic flow also had a much narrower spread than those based on barometric altitude alone with a standard deviation of 2.2 for optic flow based landings compared to 5.4 for barometric.

Because the circle down portion of the landing routine is based on barometric altitude until HAG readings fall under a threshold at which they are trusted, it is important to note that the distance from the desired touchdown point does not increase with increasing error in barometric altitude. This demonstrates the ability of the optic flow sensor to land the airplane accurately even when the altitude of the landing point relative to the takeoff point is not known.

3.6 Conclusion

The first step necessary to land a vehicle autonomously is the design of control algorithms that will allow the vehicle to accurately perform a landing maneuver. A method for controlling the landing maneuver based on heading fields has been presented. This method has been tested both in simulation and in hardware and has proven effective, repeatable, and robust to wind. The method has been used effectively to perform hundreds of autonomous landings in a wide variety of wind conditions.
In addition to effectively tracking a landing maneuver, in order to accomplish true autonomous landing in uncon-
trolled environments, the UAV must be able to detect HAG reliably. In environments where the terrain is sufficiently
flat and the altitude of the landing point relative to the takeoff point is known, a well-calibrated barometric altimeter
(with both airflow and temperature compensation) is sufficient to accurately determine HAG and land the UAV.

In environments where this is not the case fusion of data from the barometric altimeter and the optic flow sensor
provides a feasible and robust method for estimating HAG. This estimate of HAG can be combined with robust landing
algorithms to successfully land mini and micro fixed-wing UAVs within meters of their desired landing point. No other
sensor combination to date with a similar payload and computational cost has demonstrated the same capability.

\section{Automatic Trim and Gain Tuning}

Traditional autopilot design involves taking the equations of flight and linearizing them about an operating point. Con-
trollers are then designed with these linearized models. A popular choice for such controllers are PID controllers.
Although such designs are easy to implement and can work well, this method requires that the plane remain within the
design flight conditions and are typically not very failure tolerant. To allow for a larger flight envelope, gain schedul-
ing can be used to bridge different PID controllers designed around different operating points. However such designs
still require tuning at the different operating points, do not address fault tolerance, and must be tuned for different
platforms. Adaptive control was developed to address some of these problems. Examples of different adaptive control
approaches are neural network, least squares estimation, and Lyapunov based methods.

One adaptive approach is to use neural networks. This often entails training a network offline which is used in
the autopilot. However, to compensate for training errors and to adapt to changes in the aircraft (i.e. battle damage,
system failures, etc.), a separate online neural network is also used. This online network is then constantly adapting to
changes on the plane and training error. \cite{2}, \cite{3}, \cite{4}, \cite{5}, and \cite{6} discuss this approach.

Another approach identifies the plane parameters using recursive least squares techniques. These parameters are
then used to adjust the controller. Such controllers have the ability to quickly converge on the plane parameters \cite{7},
and therefore adapt quickly to failures. \cite{8}, \cite{9}, and \cite{5} use least squares estimation.

Also, a Lyapunov based approach can be used. In this approach parameters are adjusted according to a Lyapunov
function to assure stability. These algorithms typically have a small code footprint and require little computing power.
\cite{10}, \cite{11}, and \cite{12} are examples of Lyapunov based algorithms.

In our research, we deal with Miniature Aerial Vehicles (MAVs). We have found that PID values are different for
each MAV, even if they have the same geometry. This is mostly due to the manufacturing process. The desire to have
one autopilot that works on any of our MAVs with only one set of gains has lead us into adaptive control. Because of
the size of our MAVs, we need to use a small autopilot. This means we have code space, memory, and computational
restrictions on our autopilot. Also, because of the differing platforms and an imperfect manufacturing process, we do
not have an accurate model of our MAVs. This means we are not able to use neural networks because they require a
good model of the MAV, require some computational power to implement, and are not well suited for being placed on
MAVs of differing geometries. We also cannot use the least squares algorithms which use inverse matrices that require
memory and computer power to handle. Such resources come at a premium. This has lead us to use the Lyapunov
based approach.

Using model reference design and Lyapunov theory, we will show that a stable, adaptive algorithm can be derived
that is computationally simple. This new autopilot design will automatically adapt itself to whatever MAV it is flying.
Its inputs will be desired pitch and roll angles. We assume that both heading and altitude looped are closed outside of
the current control design and feeding the autopilot with the desired pitch and roll angles.

It should be noted that adaptive control algorithms are primarily used to compensate for actuator and control sur-
face failure. To the authors’ knowledge it is not being used to make a universal autopilot that can be used on a variety
of different platforms. This makes our approach somewhat unique. Further research should be done to see how many
different platforms our autopilot can fly on.
In this paper we will show four different Model Reference Adaptive Control (MRAC) schemes. All four schemes will use the same reference model, but will differ on how much is lumped into the regressor’s bias term. The first scheme will employ backstepping to handle derivatives on angular rates. The second does not use backstepping; instead we assume that the angular rate derivatives change slowly enough that we can assume they are slow-varying parameters. The third and forth schemes further simplify by lumping more slowly changing signals into the bias term. Flight data from three different MAV designs will also be shown.

4.1 UAV Model

If we assume that the inertia matrix has the form

\[
\mathbf{J} = \begin{pmatrix}
J_x & 0 & -J_{xz} \\
0 & J_y & 0 \\
-J_{xz} & 0 & J_z
\end{pmatrix},
\]

and that the aerodynamic moments are linear, then the rotational dynamics of the UAV are given by

\[
\dot{\phi} = p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \\
\dot{\theta} = q \cos \phi - r \sin \phi \\
\dot{\psi} = q \sin \phi \sec \theta + r \cos \phi \sec \theta
\]

\[
\dot{\dot{\phi}} = \Gamma_1 pq - \Gamma_2 qr + \frac{1}{2} \rho V^2 S b \left[ C_{p\alpha} + C_{p\beta} \beta + C_{p\gamma} \frac{bp}{2V} + C_{p\delta} \frac{br}{2V} \right] \\
+ \frac{J_{xz}}{J_y} (r^2 - p^2) + \frac{J_x - J_z}{J_y} pr + \frac{1}{2J_y} \rho V^2 \bar{c} S \left[ C_{m\alpha} + C_{m\beta} \alpha \right] \\
+ C_{m\delta_a} \delta_a + C_{m\delta_r} \delta_r
\]

\[
\dot{\dot{\theta}} = \frac{J_{xz}}{J_x} (r^2 - p^2) + \frac{J_x - J_z}{J_y} pr + \frac{1}{2J_y} \rho V^2 \bar{c} S \left[ C_{m\alpha} + C_{m\beta} \alpha \right] \\
+ C_{m\delta_a} \delta_a + C_{m\delta_r} \delta_r
\]

where

| \phi, \theta, \psi | roll, pitch, yaw angles |
| \phi^d, \theta^d | desired roll and pitch angles |
| \phi_m, \theta_m | reference model roll and pitch angles |
| k_m | reference model gain |
| p, q, r | roll, pitch, yaw rates |
| \alpha, \beta | angle of attack, sideslip angle |
| \chi, \gamma | flight path heading angle, flight path angle |
| \delta_e, \delta_a, \delta_r | elevator, aileron, rudder commands |
| V | airspeed |
| S, b, \bar{c} | wing area, wing span, average cord length |
| J_{ij} | inertial moments about i and j axes |

Table 1: Definition of terms
For all schemes we use the following first order models,

\[ \dot{\theta}_m = k_m (\theta^d - \theta_m) \]  

(34)

for pitch and

\[ \dot{\phi}_m = k_m (\phi^d - \phi_m) \]  

(35)

for roll. We define tacking error as

\[ \tilde{\theta} = \theta - \theta_m \text{ and } \]  

(36)

\[ \tilde{\phi} = \phi - \phi_m \]  

(37)

for pitch and roll, respectively. The derivatives are

\[ \dot{\tilde{\theta}} = \dot{\theta} - \dot{\theta}_m \]
\[ = q \cos \phi - r \sin \phi - \dot{\theta}_m \text{ and } \]  

(38)

\[ \dot{\tilde{\phi}} = \dot{\phi} - \dot{\phi}_m \]
\[ = p + q \sin \phi \tan \theta + r \cos \phi \tan \theta - \dot{\phi}_m. \]  

(39)

We also make the following assumptions:
A1: The following states can be measured: $\theta$, $\phi$, $p$, $r$, $q$.

A2: The pitch angle is limited to $-\bar{\theta} \leq \theta \leq \bar{\theta}$ where $\bar{\theta} < \pi/2$.

A3: The roll angle is limited to $-\bar{\phi} \leq \phi \leq \bar{\phi}$ where $\bar{\phi} < \pi/2$.

A4: The pitch angle $\theta = \gamma + \alpha$ where $\gamma$ is the flight path climb angle.

A5: The pitch angle $\beta = \chi - \psi$ where $\chi$ is the flight path heading angle.

A6: For the pitch controllers $\phi$ is constant.

A7: For the roll controllers $\theta$ is constant.

4.2 Scheme A: nonlinear MRAC using backstepping

4.2.1 Pitch Attitude Hold

For pitch attitude hold, we will work with Equations (29) and (32). Our objective is to design a controller so that the pitch angle follows the reference model (34).

Our desire is to pick $\delta_e$ such that Equation (38) becomes

$$\dot{\theta} = -\gamma_1 \dot{\theta},$$

where $\gamma_1 > 0$.

If we define

$$q^{des} \triangleq \dot{\theta}_m - \gamma_1 \dot{\theta} + r \sin \phi \over \cos \phi,$$

and let $q = q^{des}$ in Equation (38), we get the desired result. Using the actual pitch rate but adding and subtracting $q^{des}$ we obtain

$$\dot{\theta} = q \cos \phi - r \sin \phi - \dot{\theta}_m$$

$$= (q + q^{des} - q^{des}) \cos \phi - r \sin \phi - \dot{\theta}_m$$

$$= q^{des} \cos \phi - r \sin \phi - \dot{\theta}_m + (q - q^{des}) \cos \phi$$

$$= -\gamma_1 \dot{\theta} + \dot{q} \cos \phi,$$

where $\dot{q} \triangleq q - q^{des}$.

Differentiating $\dot{q}$ we get

$$\ddot{q} = \dot{q} - \dot{q}^{des}$$

$$= \frac{J_{xz}}{J_y} (p^2 - r^2) + \frac{J_z - J_x}{J_y} pr + \frac{\rho V^2 S c}{J_y} \left[ C_{m_0} + C_{m_\alpha} (\theta - \gamma) \right]$$

$$+ C_{m_q} \frac{\ddot{q}}{V} + C_{m_{\delta_e}} \delta_e - \dot{q}^{des}.$$

Define the Lyapunov function candidate

$$V = \frac{1}{2} \dot{\theta}^2 + \frac{1}{2} \dot{q}^2.$$
4.2 Scheme A: nonlinear MRAC using backstepping

Differentiating we get

\[ \dot{V} = \dot{\theta} + \dot{\theta} \frac{\theta^2}{2} \cos \phi + \dot{\theta} \frac{\theta^2}{2} + \frac{J_{xz}}{J_y} (p^2 - r^2) + \frac{J_z - J_x}{J_y} pr \]

\[ + \frac{1}{2} \rho V^2 \frac{Sc C_m e}{J_y} \left[ C_{m_0} + C_{m_a} (\theta - \gamma) + C_{m_q} \frac{\theta q}{V} + C_{m_e} \delta_e \right] \].

(41)

If we knew the coefficients, we would like to pick \( \delta_e \) to satisfy

\[ -\gamma_2 \dot{q} = \dot{\theta} - \dot{\theta} \frac{\theta^2}{2} \cos \phi + \dot{\theta} \frac{\theta^2}{2} + \frac{J_{xz}}{J_y} (p^2 - r^2) + \frac{J_z - J_x}{J_y} pr \]

\[ + \frac{1}{2} \rho V^2 \frac{Sc C_m e}{J_y} \left[ C_{m_0} + C_{m_a} (\theta - \gamma) + C_{m_q} \frac{\theta q}{V} + C_{m_e} \delta_e \right] \].

(41)

Solving for \( \delta_e \) gives

\[ \delta_e^{des} = \frac{J_y}{\frac{1}{2} \rho V^2 Sc C_m e} \left[ -\gamma_2 \dot{q} - \dot{\theta} \frac{\theta^2}{2} \cos \phi + \dot{\theta} \frac{\theta^2}{2} + \frac{J_{xz}}{J_y} (p^2 - r^2) + \frac{J_z - J_x}{J_y} pr \right] \]

\[ - \frac{1}{C_{m_e}} \left[ C_{m_0} + C_{m_a} (\theta - \gamma) + C_{m_q} \frac{\theta q}{V} \right] \].

(42)

If we define

\[ K \triangleq \begin{pmatrix} \frac{J_y}{\frac{1}{2} \rho V^2 Sc C_m e} \\ \frac{J_{xz}}{J_y} (p^2 - r^2) + \frac{J_z - J_x}{J_y} pr \end{pmatrix} \quad \text{and} \quad \Psi \triangleq \begin{pmatrix} \frac{-\gamma_2 \dot{q} - \dot{\theta} \frac{\theta^2}{2} \cos \phi + \dot{\theta} \frac{\theta^2}{2} + \frac{J_{xz}}{J_y} (p^2 - r^2) + \frac{J_z - J_x}{J_y} pr} \\ -\gamma_2 \dot{q} - \dot{\theta} \frac{\theta^2}{2} \cos \phi + \dot{\theta} \frac{\theta^2}{2} + \frac{J_{xz}}{J_y} (p^2 - r^2) + \frac{J_z - J_x}{J_y} pr \end{pmatrix}, \]

(43)

then

\[ \delta_e^{des} = \Psi^T K. \]

Let \( \hat{K} \) be the current estimate of \( K \), then we will use the control

\[ \delta_e = \Psi^T \hat{K}. \]

Therefore, plugging into Equation (41) we get

\[ \dot{V} = -\gamma_1 \dot{\theta}^2 + \dot{\theta} \frac{\theta^2}{2} + \frac{J_{xz}}{J_y} (p^2 - r^2) + \frac{J_z - J_x}{J_y} pr \]

\[ + \frac{1}{2} \rho V^2 \frac{Sc C_m e}{J_y} \left[ C_{m_0} + C_{m_a} (\theta - \gamma) + C_{m_q} \frac{\theta q}{V} + C_{m_e} \delta_e + \delta_e^{des} - \delta_e^{des} \right] \]

\[ = -\gamma_1 \dot{\theta}^2 - \gamma_2 \dot{q}^2 + \dot{q} \left( \frac{1}{2} \rho V^2 \frac{Sc C_m e}{J_y} \right) C_{m_e} (\delta_e - \delta_e^{des}) \]

\[ = -\gamma_1 \dot{\theta}^2 - \gamma_2 \dot{q}^2 + \dot{q} \left( \frac{1}{2} \rho V^2 \frac{Sc C_m e}{J_y} \right) V^2 \Psi^T (\hat{K} - K) \]

\[ = -\gamma_1 \dot{\theta}^2 - \gamma_2 \dot{q}^2 - \dot{q} \left( \frac{1}{2} \rho V^2 \frac{Sc C_m e}{J_y} \right) V^2 \Psi^T \hat{K}. \]
4.2 Scheme A: nonlinear MRAC using backstepping

where \( \tilde{K} \triangleq K - \hat{K} \).

To cancel the last term, let the modified Lyapunov function candidate be

\[
V_2 = \frac{1}{2} \tilde{\theta}^2 + \frac{1}{2} \tilde{q}^2 + \left( \frac{\frac{1}{2} \rho V S \tilde{C}_{m,\phi}}{I_y} \right) \frac{1}{2} \tilde{K}^T \Gamma^{-1} \tilde{K},
\]

(44)

where \( \Gamma \) is a diagonal matrix. Then

\[
\dot{V}_2 = -\gamma_1 \tilde{\theta}^2 - \gamma_2 \tilde{q}^2 - \left( \frac{\frac{1}{2} \rho S \tilde{C}_{m,\phi}}{I_y} \right) V^2 \tilde{q} \Psi^T \tilde{K} + \left( \frac{\frac{1}{2} \rho V S \tilde{C}_{m,\phi}}{I_y} \right) \tilde{K}^T \Gamma^{-1} \tilde{K}.
\]

Observing that \( \dot{\tilde{K}} = \tilde{K} - \hat{K} = -\hat{K} \) since \( K \) is constant, gives

\[
\dot{V}_2 = -\gamma_1 \tilde{\theta}^2 - \gamma_2 \tilde{q}^2 - \left( \frac{\frac{1}{2} \rho V S \tilde{C}_{m,\phi}}{I_y} \right) \left[ \Gamma^{-1} \tilde{K} + V \tilde{q} \Psi \right]^T \tilde{K}.
\]

(45)

Therefore letting

\[
\dot{\hat{K}} = -\Gamma V \tilde{q} \Psi,
\]

(46)

gives

\[
\dot{V}_2 = -\gamma_1 \tilde{\theta}^2 - \gamma_2 \tilde{q}^2.
\]

The algorithm can be summarized as follows:

**Algorithm 1** Nonlinear MRAC for pitch attitude hold.

1. Obtain \( \phi, p, r, \alpha, q, \theta \) from sensors.
2. Update the reference model according to \( \dot{\theta}_m = k_m (\theta^d - \theta_m) \).
3. Compute: \( \dot{q}^{\text{des}} = -\hat{q} \cos \phi + \dot{q}^{\text{des}} \)
4. Compute: \( \Psi = \begin{pmatrix} \frac{\gamma_2 \tilde{q} - \hat{q} \cos \phi + \dot{q}^{\text{des}}}{\sqrt{\gamma_2^2 + \gamma_1^2}} & -\gamma_2 \tilde{q}^2 - \frac{pr}{V^2} - 1 - \theta - \frac{\dot{\theta}}{V} \end{pmatrix}^T \)
5. Update the gain estimate according to: \( \dot{\hat{K}} = -\Gamma V \tilde{q} \Psi \)
6. Compute the elevator command: \( \delta_e = \Psi^T K \).
4.2 Scheme A: nonlinear MRAC using backstepping

4.2.2 Roll Attitude Hold

Using Equation (35) as our roll model reference, we want Equation (39) to be \( \dot{\phi} = -\gamma \dot{\phi} \), so we define \( p^{des} \) to be

\[
p^{des} = -\gamma \dot{\phi} - q \sin \phi \tan \phi - r \cos \phi \tan \phi + \dot{\phi}_m.
\]

We define \( \dot{p} \) as

\[
\dot{p} \triangleq p - p^{des}
\]

and take its derivative

\[
\dot{\dot{p}} = \dot{p} - p^{des}.
\]

Substituting equation (31) into equation (48) we get the following.

\[
\dot{p} = \Gamma_1 pq - \Gamma_2 qr + \frac{1}{2} \rho V^2 S \frac{\beta}{2} \left[ C_{p_0} + C_{p_{\alpha}} \beta + \frac{b_p}{2V} + C_{p_r} \frac{br}{2V} + C_{p_{\delta r}} \delta_r \right] - p^{des}
\]

We define our Lyapunov candidate function

\[
V = \frac{1}{2} \dot{\phi}^2 + \frac{1}{2} \dot{p}^2
\]

and take it derivative

\[
\dot{V} = \dot{\phi} \dot{\phi} + \dot{p} \dot{p}
\]

\[
= \ddot{\phi} \left( p + q \sin \phi \tan \theta + r \cos \phi \tan \theta - \dot{\phi}_m \right) + \ddot{p} \dot{p}.
\]

We add and subtract \( p^{des} \) to get

\[
\dot{V} = \ddot{\phi} \left( \ddot{p} + p^{des} + q \sin \phi \tan \theta + r \cos \phi \tan \theta - \dot{\phi}_m \right) + \ddot{p} \dot{p}
\]

\[
= \ddot{\phi} \left( -\gamma_1 \ddot{\phi} + \ddot{p} \right) + \ddot{p} \dot{p}
\]

\[
= -\gamma_1 \ddot{\phi}^2 + \ddot{p} \dot{p} + \ddot{p} \dot{p}
\]

\[
= -\gamma_1 \ddot{\phi}^2 + \ddot{p} \dot{p}.
\]

To arrive at the Lyapunov equation we want, we set \( \dddot{\phi} + \dddot{p} = -\gamma_2 \dddot{p} \). If we now plug this into equation (49) and solve for \( \delta_a \), calling it \( \delta_a^{des} \), gives us

\[
\delta_a^{des} = \frac{4}{\rho V^2 S b C_{p_{\alpha}}^2} \left( -\gamma_2 \dddot{p} + \dddot{p}^{des} \right) = \frac{4 \Gamma_1}{\rho V^2 S b C_{p_{\alpha}}^2} \rho V^2 S b C_{p_{\alpha}} \dddot{p} - \frac{4 \Gamma_2}{\rho V^2 S b C_{p_{\alpha}}^2} \rho V^2 S b C_{p_{\alpha}} \dddot{p} - \frac{4 \Gamma_1}{\rho V^2 S b C_{p_{\alpha}}^2} \rho V^2 S b C_{p_{\alpha}} \gamma_2.
\]

Using the assumption that \( \beta = \psi - \chi \) where \( \chi \) is the flight path heading angle gives us

\[
\delta_a^{des} = \frac{4}{\rho V^2 S b C_{p_{\alpha}}^2} \left( -\gamma_2 \dddot{p} + \dddot{p}^{des} \right) = \frac{4 \Gamma_1}{\rho V^2 S b C_{p_{\alpha}}^2} \rho V^2 S b C_{p_{\alpha}} \dddot{p} - \frac{4 \Gamma_2}{\rho V^2 S b C_{p_{\alpha}}^2} \rho V^2 S b C_{p_{\alpha}} \dddot{p} - \frac{4 \Gamma_1}{\rho V^2 S b C_{p_{\alpha}}^2} \rho V^2 S b C_{p_{\alpha}} \gamma_2.
\]

which can be represented as

\[
\delta_a^{des} = \Psi^T K
\]
4.2 Scheme A: nonlinear MRAC using backstepping

where

\[
\Psi = \begin{pmatrix}
-\gamma_2 \dot{p} - \dot{\phi} + p^{des} \\
-p_a \\
-q_r \\
-1 \\
\chi \\
-\rho \\
-V \\
-\delta_r
\end{pmatrix}, \quad K = \begin{pmatrix}
\frac{4}{b} p V^2 S c_{\psi} p_{\hat{a}} \\
\frac{4}{b} p V^2 S c_{\psi} p_{\hat{a}} \\
\frac{4}{b} p V^2 S c_{\psi} p_{\hat{a}} \\
\frac{4}{b} p V^2 S c_{\psi} p_{\hat{a}} \\
C_{\theta} + C_{\theta z} \beta \\
C_{\theta} + C_{\theta z} \beta \\
C_{\theta} + C_{\theta z} \beta \\
C_{\theta} + C_{\theta z} \beta
\end{pmatrix}.
\]

(56)

However, we do not know the terms in \(K\) and instead only have an estimate \(\hat{K}\) so really we have \(\delta_a = \Psi^T \hat{K}\). Adding and subtracting \(\delta_a^{des}\) to equation (49) we get

\[
\dot{\hat{p}} = \Gamma_1 \rho q + \Gamma_2 qr + \frac{1}{2} \rho V^2 S \frac{b}{2} \left[ C_{\psi} c_{\psi} + C_{\psi d} + C_{\psi d} \frac{b p}{2 V} + C_{\psi d} \frac{b r}{2 V} \right] - \gamma_2 \hat{p} - \hat{\phi}
\]

(57)

Defining a new Lyapunov function candidate

\[
V_2 = \frac{1}{2} \chi^2 + \frac{1}{2} \rho^2 + \frac{\rho V S b C_{\psi} \rho a}{8} \dot{\hat{K}} \Gamma^{-1} \hat{K}
\]

(59)

where \(\Gamma\) is a diagonal matrix. We differentiate the Lyapunov function to get

\[
\dot{V}_2 = \dot{\chi} \dot{\chi} + \rho \dot{\phi} \hat{p} + \frac{\rho V S b C_{\psi} \rho a}{4} \hat{K} \Gamma^{-1} \hat{K} = -\gamma_1 \dot{\phi}^2 + \chi \dot{\phi} + \dot{\phi} \hat{p} + \frac{\rho V S b C_{\psi} \rho a}{4} \hat{K} \Gamma^{-1} \hat{K}
\]

(60)

Substituting the results from (57) we get

\[
\dot{V}_2 = -\gamma_1 \dot{\phi}^2 + \rho \left( \dot{\hat{p}} + \hat{p} \right) + \frac{\rho V S b C_{\psi} \rho a}{4} \hat{K} \Gamma^{-1} \hat{K}
\]

(61)

Setting \(\rho V \Psi^T + \hat{K} \Gamma^{-1} = 0\) we arrive at the update law

\[
\hat{K} = -\Gamma \hat{p} V \Psi.
\]

(62)

The algorithm can be summarized as follows.
4.3 Scheme B: nonlinear MRAC without backstepping

Algorithm 2 Nonlinear MRAC for roll attitude hold.
1: Obtain $\phi$, $p$, $r$, $q$, $\theta$ from sensors.
2: Update the reference model according to $\dot{\phi}_m = k_m (\phi^d - \phi_m)$.
3: Compute: $\dot{p}^{des}$
4: Compute:
   $$\Psi = \left( \begin{array}{c} -\gamma \tilde{p} - \dot{p}^{des} \frac{\dot{p} + p \dot{q}}{\sqrt{p^2}} - \frac{p}{\sqrt{p^2}} - 1 \chi - \frac{p}{\sqrt{p^2}} - \frac{r}{\sqrt{p^2}} - \delta_r \end{array} \right)^T,$$
5: Update the gain estimate according to: $\dot{K} = -\Gamma \rho V \Psi$
6: Compute the aileron command: $\delta_a = \Psi^T K$.

4.3 Scheme B: nonlinear MRAC without backstepping

4.3.1 Pitch Attitude Hold

Using the same definition of tracking error for pitch as above, we design controller without using backstepping. Again, our desire is to pick $\delta_c$ such that Equation (38) becomes

$$\dot{\theta} = -\gamma_1 \tilde{\theta},$$

where $\gamma_1 > 0$. Solving for $q$ in equation (32), and assuming $\dot{q}$ is slow-varying parameters, we get

$$q = \frac{V}{C_{m\alpha} e} \left( \frac{2}{\rho V^2 S c} [J_y \dot{q} - J_{ez} (r^2 - p^2) - (J_z - J_x) p r] - C_{m0} + C_{m\alpha} \gamma - C_{m\alpha} \theta - C_{\delta e} \delta_e \right).$$

If we now plug this into equation (38) we get

$$\dot{\theta} = \frac{V \cos \phi}{C_{m\alpha} e} \left( \frac{2}{\rho V^2 S c} [J_y \dot{q} - J_{ez} (r^2 - p^2) - (J_z - J_x) p r] - C_{m0} + C_{m\alpha} \gamma - C_{m\alpha} \theta - C_{\delta e} \delta_e \right) - r \sin \phi - \dot{\theta}_m$$

If we knew the parameters we could command

$$\delta_e^{des} = \Psi^T K$$

where

$$\Psi^T = \begin{bmatrix} \frac{\gamma_1 \tilde{\theta} - k_m (\phi^d - \theta_m) - r \sin \phi}{V \cos \phi} & \frac{\gamma_0 \tilde{\theta} - k_m (\phi^d - \theta_m) - r \sin \phi}{V \cos \phi} & \frac{\gamma_0 \tilde{\theta} - k_m (\phi^d - \theta_m) - r \sin \phi}{V \cos \phi} \\ \frac{1}{p^2 - r^2} & \frac{1}{p^2 - r^2} & \frac{1}{p^2 - r^2} \\ -pr & -pr & -pr \end{bmatrix} K = \begin{bmatrix} J_y & C_{m\alpha} \gamma & \frac{C_{m\alpha}}{C_{m\alpha} e} + C_{m\alpha} \gamma \\ C_{m\alpha} \gamma & \frac{C_{m\alpha}}{C_{m\alpha} e} + C_{m\alpha} \gamma & \frac{C_{m\alpha}}{C_{m\alpha} e} + C_{m\alpha} \gamma \\ \frac{2}{\rho V^2 S c} [J_y \dot{q} - J_{ez} (r^2 - p^2) - (J_z - J_x) p r] - C_{m0} & \frac{2}{\rho V^2 S c} [J_y \dot{q} - J_{ez} (r^2 - p^2) - (J_z - J_x) p r] - C_{m0} & \frac{2}{\rho V^2 S c} [J_y \dot{q} - J_{ez} (r^2 - p^2) - (J_z - J_x) p r] - C_{m0} \end{bmatrix}$$

Adding and subtracting this in equation (64) we get

$$\dot{\theta} = \frac{V \cos \phi}{C_{m\alpha} e} \left( \frac{2}{\rho V^2 S c} [J_y \dot{q} - J_{ez} (r^2 - p^2) - (J_z - J_x) p r] - C_{m0} + C_{m\alpha} \gamma - C_{m\alpha} \theta - C_{\delta e} (\delta_e^{des} + \delta_e - \delta_e^{des}) \right) - r \sin \phi - \dot{\theta}_m$$

$$= -\gamma_1 \tilde{\theta} + \frac{V \cos \phi}{C_{m\alpha} C_{m\alpha} e} \Psi^T K$$

31
Assuming $\phi$ is constant, we define our Lyapunov function candidate as

$$V = \frac{1}{2} \hat{\theta}^2 + \frac{\cos \phi}{2C_mC_{m_k}} \hat{K}^T \Gamma^{-1} \hat{K}$$

and take its derivative we get

$$\dot{V} = \dot{\theta} \hat{\theta} + \frac{\cos \phi}{C_mC_{m_k}} \hat{K}^T \Gamma^{-1} \hat{K}$$

$$= -\gamma_1 \hat{\theta}^2 + \frac{\hat{\theta} V \cos \phi}{C_mC_{m_k}} \Psi^T \hat{K} + \frac{\cos \phi}{C_mC_{m_k}} \hat{K}^T \Gamma^{-1} \hat{K}$$

$$= -\gamma_1 \hat{\theta}^2 + \frac{\cos \phi}{C_mC_{m_k}} \left( \hat{\theta} V \Psi^T + \hat{K}^T \Gamma^{-1} \right) \hat{K}$$

(67)

We now let $\dot{\theta} V \Psi^T + \hat{K}^T \Gamma^{-1} = 0$ and we get our update law

$$\dot{\hat{K}} = -\Gamma \dot{\theta} V \Psi$$

(68)

We summarize the algorithm.

**Algorithm 3** Nonlinear MRAC for pitch attitude hold.

1: Obtain $\phi, p, r, q, \theta$ from sensors.
2: Update the reference model according to $\dot{\hat{m}} = k_m (\theta^d - \hat{m})$.
3: Compute:
   \[
   \Psi^T = \left( \frac{\gamma_1 \hat{\theta} - k_m (\theta^d - \hat{m}) - r \sin \phi}{\cos \phi} \quad 1 \quad p^2 - r^2 - pr - \theta \right)^T,
   \]
4: Update the gain estimate according to: $\dot{\hat{K}} = -\Gamma \dot{\theta} V \Psi$
5: Compute the elevator command: $\delta_e = \Psi^T \hat{K}$.
4.3 Scheme B: nonlinear MRAC without backstepping

4.3.2 Roll Attitude Hold

Assuming \( \dot{p} \) and \( \dot{r} \) are slow-varying parameters and solving for \( p \) in equation (31) we find that

\[
p = \frac{8}{\rho V S_b^2 C_{p_p}} \dot{p} - \frac{8 \Gamma_1}{\rho V S_b^2 C_{p_p}} p q + \frac{8 \Gamma_2}{\rho V S_b^2 C_{p_p}} q r - \frac{2 V C_{p_0}}{b C_{p_p}} - \frac{2 V C_{p_0} \beta}{b C_{p_p}}
\]

\[
- \frac{C_{p_r}}{C_{p_p}} r - \frac{2 V C_{p_{\delta a}}}{b C_{p_p}} \delta_a - \frac{2 V C_{p_{\delta r}}}{b C_{p_p}} \delta_r.
\]

Substituting this equation in equation (39) give us

\[
\dot{\phi} = \frac{8}{\rho V S_b^2 C_{p_p}} \dot{p} - \frac{8 \Gamma_1}{\rho V S_b^2 C_{p_p}} p q + \frac{8 \Gamma_2}{\rho V S_b^2 C_{p_p}} q r - \frac{2 V C_{p_0}}{b C_{p_p}} - \frac{2 V C_{p_{\delta a}}}{b C_{p_p}} \beta
\]

\[
- \frac{C_{p_r}}{C_{p_p}} r - \frac{2 V C_{p_{\delta a}}}{b C_{p_p}} \delta_a - \frac{2 V C_{p_{\delta r}}}{b C_{p_p}} \delta_r + q \sin \phi \tan \phi + r \cos \phi \tan \phi - \dot{\phi}.
\]

But since we want \( \dot{\theta} = -\gamma \dot{\theta} \) to guarantee that the tracking error goes to 0, we define \( \delta_{a \text{des}} \) to be

\[
\delta_{a \text{des}} = \frac{\beta}{2 V C_{p_{\delta a}}} \left( \gamma \dot{\phi} + q \sin \phi \tan \phi + r \cos \phi \tan \phi - \dot{\phi} \right) - \frac{C_{p_{\delta a}}}{C_{p_{\delta a}}} \beta
\]

\[
- \frac{C_{p_{\delta r}}}{C_{p_{\delta a}}} \delta_r - \frac{C_{p_r}}{2 V C_{p_{\delta a}}} r + \frac{4 \Gamma_1}{\rho V^2 S_b} \dot{p} - \frac{4 \Gamma_2}{\rho V^2 S_b} q r
\]

Using the assumption that \( \beta = \chi - \psi \), then

\[
\delta_{a \text{des}} = \frac{b C_{p_r}}{2 V C_{p_{\delta a}}} \left( \gamma \dot{\phi} + q \sin \phi \tan \phi + r \cos \phi \tan \phi - \dot{\phi} \right) - \frac{C_{p_{\delta a}}}{C_{p_{\delta a}}} \beta
\]

\[
+ \frac{C_{p_{\delta r}}}{C_{p_{\delta a}}} \psi - \frac{C_{p_{\delta a}}}{C_{p_{\delta a}}} \delta_r - \frac{C_{p_r}}{2 V C_{p_{\delta a}}} r + \frac{4 \Gamma_1}{\rho V^2 S_b} \dot{p} + \frac{4 \Gamma_2}{\rho V^2 S_b} q r
\]

Grouping the knowns and unknowns together we get the form \( \delta_{a \text{des}} = \Psi^T K \) gives us

\[
\Psi = \begin{bmatrix}
\gamma \dot{\phi} + q \sin \phi \tan \phi + r \cos \phi \tan \phi - \dot{\phi} \\
\frac{\dot{p}}{\rho V S_b^2 C_{p_p}} \\
-1
\end{bmatrix}
\]

and

\[
K = \begin{bmatrix}
\frac{b C_{p_p}}{2 V C_{p_{\delta a}}} \\
-\frac{4 \Gamma_1}{\rho V^2 S_b} \\
-\frac{2 \Gamma_2}{\rho V^2 S_b}
\end{bmatrix}
\]

Finally, we substitute this equation in equation (33) to get

\[
\delta_{a \text{des}} = \Psi^T K
\]
Adding and subtracting $\delta_a^{des}$ from Equation (70) gives us

$$\phi = \frac{8}{\rho V S b^2 C_{p\phi}} \dot{\phi} - \frac{8 \Gamma_1}{\rho V S b^2 C_{p\phi}} p q + \frac{8 \Gamma_2}{\rho V S b^2 C_{p\phi}} q r - 2 V C_{p\phi} - 2 V C_{p\phi} \beta$$

$$\gamma_1 \phi - \frac{2 V C_{psa}}{b C_{p\phi}} (\delta_a - \delta_a^{des}) - 2 V C_{psa} \delta_r + q \sin \phi \tan \phi$$

From Equation (74) we arrive at our update law

$$\dot{V} = \frac{1}{2} \dot{\phi}^2 + \frac{V C_{psa}}{b C_{p\phi}} \hat{K} T \Gamma^{-1} \hat{K}$$

and take its derivative to get

$$\dot{V} = \dot{\phi}^2 + \frac{2 V C_{psa}}{b C_{p\phi}} \hat{K} T \Gamma^{-1} \hat{K}$$

$$\dot{\phi} = \gamma_1 \dot{\phi} - \frac{2 V C_{psa}}{b C_{p\phi}} \Psi^T \hat{K}$$

If we define our Lyapunov candidate function to be

$$V = \frac{1}{2} \dot{\phi}^2 + \frac{V C_{psa}}{b C_{p\phi}} \hat{K} T \Gamma^{-1} \hat{K}$$

Setting $V \dot{\phi} \Psi^T + \hat{K} T \Gamma^{-1} = 0$ we arrive at our update law

$$\hat{K} = -\Gamma V \dot{\phi} \Psi.$$ 

The algorithm can be summarized as follows.

**Algorithm 4** Nonlinear MRAC for roll attitude hold.
1: Obtain $\phi$, $p$, $r$, $\alpha$, $q$, $\theta$ from sensors.
2: Update the reference model according to $\phi_m = k_m (\phi^d - \phi_m)$.
3: Compute:

$$\Psi = \frac{1}{V} \begin{bmatrix} \gamma_1 \dot{\phi} + \sin \theta \tan \theta + \cos \theta \tan \theta - k_m (\phi^d - \phi_m) \\ V \end{bmatrix} = \frac{1}{V} \begin{bmatrix} -p q & qr & \alpha & -1 & -\chi & -r & -\delta_r \end{bmatrix}^T,$$

4: Update the gain estimate according to: $\hat{K} = -\Gamma V \dot{\phi} \Psi$
5: Compute the aileron command: $\delta_a = \Psi^T \hat{K}$.

### 4.4 Scheme C: nonlinear MRAC grouping terms

#### 4.4.1 Pitch Attitude Hold

We assume that $\dot{q}, \frac{1}{\cos \phi}, p^2 - r^2$, and $pr$ terms are slow-varying parameters. Our desire is to pick $\delta_e$ such that Equation (38) is

$$\dot{\theta} = -\gamma_1 \tilde{\theta},$$

34
where \( \gamma_1 > 0 \). Solving for \( q \) in equation (32) we get
\[
q = \frac{V}{C_m c} \left( \frac{2}{\rho V^2 S c} [J_y \dot{q} - J_x (r^2 - p^2) - (I_z - I_x) pr] - C_m \alpha - C_m \gamma - C_m \theta - C_{\delta \theta} \delta_e \right).
\]

If we now plug this into equation (38) we get
\[
\dot{\theta} = \frac{V \cos \phi}{C_m c} \left( \frac{2}{\rho V^2 S c} [J_y \dot{q} - J_x (r^2 - p^2) - (J_z - J_x) pr] - C_m \alpha - C_m \gamma - C_m \theta - C_{\delta \theta} \delta_e \right) - r \sin \phi \dot{\theta}_m
\]
(78)

If we knew the parameters we could we could command
\[
\delta_e^{des} = \Psi^T K
\]
where
\[
\Psi^T = \begin{pmatrix} \gamma_1 \dot{\theta} - k_m (\theta^e - \theta_m) - r \sin \phi \\ 1 \end{pmatrix}
\]
and
\[
K = \begin{pmatrix} \frac{J_y q}{2 Sc m_k} - \frac{C_m}{C_m c} + \frac{C_m}{C_m c} \gamma + \frac{\dot{\theta}}{2 \rho V^2 Sc m_k} (p^2 - r^2) - \frac{J_z - J_x}{2 \rho V^2 Sc m_k} pr \end{pmatrix}
\]
(79)

Adding and subtracting this in equation (78) we get
\[
\dot{\theta} = \frac{V \cos \phi}{C_m c} \left( \frac{2}{\rho V^2 S c} [J_y \dot{q} - J_x (r^2 - p^2) - (J_z - J_x) pr] - C_m \alpha - C_m \gamma - C_m \theta - C_{\delta \theta} (\delta_e^{des} + \delta_e - \delta_e^{des}) \right) - r \sin \phi \dot{\theta}_m
\]
(80)

Assuming \( \phi \) is constant, we define our Lyapunov function candidate as
\[
V = \frac{1}{2} \dot{\theta}^2 + \frac{\cos \phi}{2 C_m c m_k c} \dot{K}^T \Gamma^{-1} \dot{K}
\]
and take its derivative we get
\[
V = \dot{\theta} \ddot{\theta} + \frac{\cos \phi}{C_m c m_k c} \dot{K}^T \Gamma^{-1} \dot{K} = - \gamma_1 \dot{\theta}^2 + \frac{\dot{\theta} V \cos \phi}{C_m c m_k c} \Psi^T \dot{K} + \frac{\cos \phi}{C_m c m_k c} \dot{K}^T \Gamma^{-1} \dot{K} = - \gamma_1 \dot{\theta}^2 + \frac{\cos \phi}{C_m c m_k c} \left( \dot{\theta} V \Psi^T + \dot{K}^T \Gamma^{-1} \right) \dot{K}
\]
(81)

We now let \( \dot{\theta} V \Psi^T + \dot{K}^T \Gamma^{-1} = 0 \) and we get our update law
\[
\dot{K} = - \Gamma \dot{\theta} V \Psi
\]
(82)

We summarize the algorithm.
4.4 Scheme C: nonlinear MRAC grouping terms

Algorithm 5 Nonlinear MRAC for pitch attitude hold.

1: Obtain $\phi, p, r, \alpha, q, \theta$ from sensors.
2: Update the reference model according to $\dot{\theta}_m = k_m (\theta_d - \theta_m)$.
3: Compute:
$$\Psi = \begin{pmatrix} \frac{\gamma_1 \dot{\phi} - k_m (\theta_d - \theta_m)}{V \cos \phi} & 1 & -\theta \end{pmatrix}^T,$$
4: Update the gain estimate according to: $\dot{K} = -\Gamma V \cos \phi \hat{\theta} \Psi$
5: Compute the elevator command: $\delta_e = \Psi^T K$.

4.4.2 Roll Attitude Hold

Let the derivative of the tracking error be
$$\ddot{\phi} = p + q \sin \phi \tan \phi + r \cos \phi \tan \phi - \phi_m. \quad (83)$$

We make the assumption here that the terms $\dot{p}, \dot{q}, \dot{r}, \sin \theta \tan \phi + r \cos \theta \tan \phi, pq,$ and $\chi$ are slow-varying parameters. Solving for $p$ in equation (31) we find that
$$p = \frac{8}{\rho V S b^2 C_{p_r}} \dot{\phi} - \frac{8 \Gamma_1}{\rho V S b^2 C_{p_p}} pq + \frac{8 \Gamma_2}{\rho V S b^2 C_{p_p}} qr - \frac{2 V C_{p_\phi}}{b C_{p_r}} - \frac{2 V C_{p_\phi} \beta}{b C_{p_p}}$$
$$- \frac{2 V C_{p_{\phi a}}}{b C_{p_r}} r - \frac{2 V C_{p_{\phi a}}}{b C_{p_p}} \delta_a - \frac{2 V C_{p_{\phi c}}}{b C_{p_p}} \delta_c.$$

Substituting this equation in equation (39) give us
$$\ddot{\phi} = \frac{8}{\rho V S b^2 C_{p_r}} \dot{\phi} - \frac{8 \Gamma_1}{\rho V S b^2 C_{p_p}} pq + \frac{8 \Gamma_2}{\rho V S b^2 C_{p_p}} qr - \frac{2 V C_{p_\phi}}{b C_{p_r}} - \frac{2 V C_{p_\phi} \beta}{b C_{p_p}}$$
$$- \frac{2 V C_{p_{\phi a}}}{b C_{p_r}} r - \frac{2 V C_{p_{\phi a}}}{b C_{p_p}} \delta_a + \frac{2 V C_{p_{\phi c}}}{b C_{p_p}} \delta_c + q \sin \phi \tan \phi + r \cos \phi \tan \phi - \phi_m. \quad (84)$$

But since we want $\ddot{\phi} = -\gamma_1 \dot{\phi}$ to guarantee that the tracking error goes to 0, we define $\delta_{\phi, \text{des}}$ to be
$$\delta_{\phi, \text{des}} = \frac{b C_{p_p}}{2 V C_{p_{\phi a}}} \left( \frac{\gamma_1 \dot{\phi} + q \sin \phi \tan \phi + r \cos \phi \tan \phi - \phi_m}{\chi} \right) - \frac{C_{p_{\phi a}}}{C_{p_{\phi a}}} - \frac{C_{p_{\phi c}}}{C_{p_{\phi c}}} \beta$$
$$- \frac{C_{p_{\phi a}}}{C_{p_{\phi a}}} \delta_a - \frac{C_{p_{\phi c}}}{2 V C_{p_{\phi a}}} r + \frac{4 \Gamma_1}{\rho V S b^2} \dot{p} - \frac{4 \Gamma_2}{\rho V S b^2} \dot{q} \ddot{r}.$$

Using the assumption that $\beta = \chi - \psi$, then
$$\delta_{\phi, \text{des}} = \frac{b C_{p_p}}{2 V C_{p_{\phi a}}} \left( \frac{\gamma_1 \dot{\phi} + q \sin \phi \tan \phi + r \cos \phi \tan \phi - \phi_m}{\chi} \right) - \frac{C_{p_{\phi a}}}{C_{p_{\phi a}}} - \frac{C_{p_{\phi c}}}{C_{p_{\phi c}}} \chi$$
$$+ \frac{C_{p_{\phi a}}}{C_{p_{\phi a}}} \psi - \frac{C_{p_{\phi c}}}{C_{p_{\phi c}}} \delta_a - \frac{C_{p_{\phi c}}}{2 V C_{p_{\phi a}}} r + \frac{4 \Gamma_1}{\rho V S b^2} \dot{p} - \frac{4 \Gamma_2}{\rho V S b^2} \dot{q} \ddot{r}.$$

Grouping the knows and unknowns together so we get the form $\delta_{\phi, \text{des}} = \Psi^T K$ gives us
$$\Psi = \begin{pmatrix} \gamma_1 \dot{\phi} - k_m (\phi_d - \phi_m) \\ -1 \\ -\chi \end{pmatrix} \quad \text{and} \quad K = \begin{pmatrix} \frac{b C_{p_{\phi a}}}{2 V C_{p_{\phi a}}} \\ \frac{C_{p_{\phi a}}}{C_{p_{\phi a}}} \\ \frac{C_{p_{\phi c}}}{C_{p_{\phi c}}} \end{pmatrix}.$$
4.4 Scheme C: nonlinear MRAC grouping terms

\[ C_{psa} = \frac{4}{V^2 \rho Sb} \dot{p} + \frac{C_{pa}}{C_{psa}} + \frac{bC_{pa}}{2C_{psa}} (q \sin \theta \tan \theta + r \cos \theta \tan \theta) \]
\[ - 4 \hat{\Gamma}_1 \frac{pq}{\rho Sb V^2} + \frac{4 \hat{\Gamma}_2 \frac{qr}{\rho Sb V^2} - \frac{C_{pa}}{C_{psa}} \dot{\psi} - \frac{C_p}{2C_{psa}} \frac{r}{V} \]

Adding and subtracting \( \delta_a^{des} \) from Equation (89) gives us

\[ \dot{\phi} = - \frac{8}{\rho V Sb^2 C_{pp}} \dot{p} - \frac{8 \hat{\Gamma}_1}{\rho V Sb^2 C_{pp}} pq + \frac{8 \hat{\Gamma}_2}{\rho V Sb^2 C_{pp}} qr - \frac{2 V C_{pa}}{b C_{pp}} - \frac{2 V C_{pa}}{b C_{pp}} \beta \]
\[ - \frac{C_p}{C_{pp}} r - \frac{2 V C_{pa}}{b C_{pp}} (\delta_a - \delta_a^{des}) - \frac{2 V C_{pa}}{b C_{pp}} \delta_r + q \sin \phi \tan \phi \]
\[ + r \cos \phi \tan \phi - \phi^* \]
\[ = - \gamma_1 \dot{\phi} - \frac{2 V C_{pa}}{b C_{pp}} (\delta_a^{des} - \delta_a) \]
\[ = - \gamma_1 \dot{\phi} - \frac{2 V C_{pa}}{b C_{pp}} (\Psi^T \hat{K} - \Psi^T \hat{K}) \]
\[ = - \gamma_1 \dot{\phi} - \frac{2 V C_{pa}}{b C_{pp}} \Psi^T \hat{K}. \] (89)

If we define our Lyapunov candidate function to be

\[ V = \frac{1}{2} \tilde{\phi}^2 + \frac{V C_{pa}}{b C_{pp}} \hat{K}^T \Gamma^{-1} \hat{K} \] (90)

and take its derivative to get

\[ \dot{V} = \dot{\phi} \ddot{\phi} + \frac{2 V C_{pa}}{b C_{pp}} \hat{K}^T \Gamma^{-1} \hat{K} \]
\[ = \dot{\phi} \left( - \gamma_1 \dot{\phi} - \frac{2 V C_{pa}}{b C_{pp}} \Psi^T \hat{K} \right) + \frac{2 V C_{pa}}{b C_{pp}} \hat{K}^T \Gamma^{-1} \hat{K} \]
\[ = - \gamma_1 \dot{\phi}^2 - \frac{2 V \dot{\phi} C_{pa}}{b C_{pp}} \Psi^T \hat{K} + \frac{2 V C_{pa}}{b C_{pp}} \hat{K}^T \Gamma^{-1} \hat{K} \]
\[ = - \gamma_1 \dot{\phi}^2 - \frac{2 V C_{pa}}{b C_{pp}} \left( V \dot{\phi} \Psi^T + \hat{K}^T \Gamma^{-1} \right) \hat{K}. \] (91)

Setting \( V \dot{\phi} \Psi^T + \hat{K}^T \Gamma^{-1} = 0 \) we arrive at our update law

\[ \dot{\hat{K}} = - \Gamma V \dot{\phi} \Psi. \]

The algorithm can be summarized as follows.

**Algorithm 6** Nonlinear MRAC for roll attitude hold.
1: Obtain \( \phi, p, r, \alpha, q, \theta \) from sensors.
2: Update the reference model according to \( \dot{\phi}_m = k_m (\phi^d - \phi_m) \).
3: Compute:
\[ \Psi = \left( \frac{\gamma_1 \dot{\phi} - k_m (\delta_a - \phi)}{2 \gamma_1} 1 - \delta_r - \chi \right)^T. \]
4: Update the gain estimate according to: \( \dot{\hat{K}} = - \Gamma V \dot{\phi} \Psi \)
5: Compute the aileron command: \( \delta_a = \Psi^T \hat{K} \).

37
4.5 Scheme D: nonlinear MRAC simple

4.5.1 Pitch Attitude Hold

We assume that $\dot{q}$, $\frac{1}{\text{comp.}} p^2 - r^2$, and $pr$ terms are slow-varying parameters. We want to pick $\delta_e$ such that

$$\dot{\theta} = -\gamma_1 \dot{\theta},$$

where $\gamma_1 > 0$. Solving for $\theta$ in equation (32) we get

$$q = \frac{V}{C_{m_q}} \left( \frac{2}{\rho V^2 S \bar{c}} [J_y \dot{q} - J_{xz} \left( r^2 - p^2 \right) - (I_z - I_x) pr] - C_{m_0} + C_{m_a} \gamma - C_{m_a} \theta - C_{\delta_e} \delta_e \right).$$

If we now plug this into equation (38) we get

$$\dot{\theta} = \frac{V \cos \phi}{C_{m_a}} \left( \frac{2}{\rho V^2 S \bar{c}} [J_y \dot{q} - J_{xz} \left( r^2 - p^2 \right) - (I_z - I_x) pr] - C_{m_0} + C_{m_a} \gamma - C_{m_a} \theta - C_{\delta_e} \delta_e \right) - r \sin \phi - \dot{\theta}_m$$

If we knew the parameters we could we could command

$$\delta_e^{\text{des}} = \Psi^T K$$

where

$$\Psi^T = \left( \frac{\gamma_1 \dot{\theta} - k_m (\theta_d - \theta_m)}{1} \right)^T$$

and

$$K = \left( \frac{J_y \dot{q}}{\frac{2}{S \bar{c}} C_{m_q} m_e} + \frac{C_{m_0}}{C_{m_a}} \left( \gamma - \dot{\theta} \right) + \frac{\gamma C_{m_p}}{\cos \phi C_{m_a}} \left( r^2 - p^2 \right) = \frac{J_y - J_x}{\frac{2}{S \bar{c}} C_{m_a} m_e} pr \right).$$

Adding and subtracting this in equation (95) we get

$$\dot{\theta} = \frac{V \cos \phi}{C_{m_a}} \left( \frac{2}{\rho V^2 S \bar{c}} [J_y \dot{q} - J_{xz} \left( r^2 - p^2 \right) - (I_z - I_x) pr] - C_{m_0} + C_{m_a} \gamma - C_{m_a} \theta - C_{\delta_e} \delta_e \left( \delta_e^{\text{des}} + \delta_e - \dot{\theta}_m \right) \right) - r \sin \phi - \dot{\theta}_m$$

Assuming $\phi$ is constant, we define our Lyapunov function candidate as

$$V \triangleq \frac{1}{2} \dot{\theta}^2 + \frac{\cos \phi}{2C_{m_q} C_{m_a} \bar{c}} K^T \Gamma^{-1} \dot{K}$$

and take its derivative we get

$$\dot{V} = \frac{\cos \phi}{C_{m_q} C_{m_a} \bar{c}} \dot{\theta}^2 + \frac{\cos \phi}{C_{m_q} C_{m_a} \bar{c}} \dot{K}^T \Gamma^{-1} \dot{K}$$

$$= -\gamma_1 \dot{\theta}^2 + \frac{\dot{\theta} V \cos \phi}{C_{m_q} C_{m_a} \bar{c}} \Psi^T \dot{K} + \frac{\cos \phi}{C_{m_q} C_{m_a} \bar{c}} \dot{K}^T \Gamma^{-1} \dot{K}$$

$$= -\gamma_1 \dot{\theta}^2 + \frac{\cos \phi}{C_{m_q} C_{m_a} \bar{c}} \left( \dot{\theta} V \Psi^T + \dot{K}^T \Gamma^{-1} \right) \dot{K}$$

(96)
4.5 Scheme D: nonlinear MRAC simple

We now let $\hat{\theta} V \hat{\Psi} + \dot{\hat{K}} T \Gamma^{-1} = 0$ and we get our update law

$$
\dot{\hat{K}} = -\Gamma \hat{\theta} V
$$

(97)

We summarize the algorithm.

**Algorithm 7** Nonlinear MRAC for pitch attitude hold.

1. Obtain $\phi, p, r, \alpha, q, \theta$ from sensors.
2. Update the reference model according to $\dot{\theta}_m = -k_m (\theta - \theta_m)$.
3. Compute:

$$
\Psi = \left( \frac{\gamma_1 \dot{\theta} - k_m (\theta - \theta_m)}{\hat{\Psi}} \right)^T
$$

4. Update the gain estimate according to: $\dot{\hat{K}} = -\Gamma \hat{\theta} V \cos \phi \Psi$
5. Compute the elevator command: $\delta_e = \Psi^T K$

4.5.2 Roll Attitude Hold

We make the assumption here that the terms $\dot{\phi}, \dot{r}, \dot{q}, \sin \theta \tan \theta + r \cos \theta \tan \theta, pq, qr,$ and $\chi$ are slow-varying parameters Solving for $\phi$ in equation (31) we find that

$$
p = \frac{8}{\rho V S h^2 C_{pr}} \dot{p} - \frac{8 \Gamma_1}{\rho V S h^2 C_{pr}} pq + \frac{8 \Gamma_2}{\rho V S h^2 C_{pr}} qr - \frac{2 V C_{p0}}{b C_{pp}} - \frac{2 V C_{p1}}{b C_{pp}} \beta
$$

$$
- \frac{C_{pr}}{b C_{pp}} \frac{r}{r - \frac{2 V C_{p1}}{b C_{pp}} \delta_a - \frac{2 V C_{p1}}{b C_{pp}} \delta_r}
$$

(98)

Substituting this equation in equation (39) give us

$$
\ddot{\phi} = 8 \frac{\rho V S h^2 C_{pp} \dot{p}}{C_{pp}} - \frac{8 \Gamma_1}{C_{pp}} pq + \frac{8 \Gamma_2}{\rho V S h^2 C_{pr}} qr - \frac{2 V C_{p0}}{b C_{pp}} - \frac{2 V C_{p1}}{b C_{pp}} \beta
$$

$$
- \frac{C_{pr}}{b C_{pp}} \frac{r}{r - \frac{2 V C_{p1}}{b C_{pp}} \delta_a - \frac{2 V C_{p1}}{b C_{pp}} \delta_r} + q \sin \phi \tan \phi + r \cos \phi \tan \phi - \phi_m.
$$

(99)

But since we want $\dot{\theta} = -\gamma_1 \dot{\theta}$ to guarantee that the tracking error goes to 0, we define $\delta^a_{des}$ to be

$$
\delta^a_{des} = \frac{b C_{p0}}{2 C_{p1s}} \left( \gamma_1 \dot{\phi} + q \sin \phi \tan \phi + r \cos \phi \tan \phi - \phi_m \right) - \frac{C_{p0}}{C_{p1s}} \beta
$$

$$
- \frac{C_{pr}}{C_{p1s}} \delta_a - \frac{4 \Gamma_1}{\rho V S h^2 b C_{pp}} \dot{p} + \frac{4 \Gamma_2}{\rho V S h^2 b C_{pp}} qr
$$

(100)

Using the assumption that $\beta = \chi - \psi$, then

$$
\delta^a_{des} = \frac{b C_{p0}}{2 C_{p1s}} \left( \gamma_1 \dot{\phi} + q \sin \phi \tan \phi + r \cos \phi \tan \phi - \phi_m \right) - \frac{C_{p0}}{C_{p1s}} \chi
$$

$$
+ \frac{C_{pr}}{C_{p1s}} \psi - \frac{C_{pr}}{C_{p1s}} \delta_r + \frac{C_{pr}}{2 C_{p1s}} \delta_r + \frac{4 \Gamma_1}{\rho V S h^2 b C_{pp}} \dot{p} + \frac{4 \Gamma_2}{\rho V S h^2 b C_{pp}} qr
$$

(101)

Grouping the knowns and unknowns together so we get the form $\delta^a_{des} = \Psi^T K$ gives us

$$
\Psi^T = \left( \frac{\gamma_1 \dot{\phi} - k_m (\phi - \phi_m)}{1 - \delta_r} \right)^T \text{ and } K = \left( \frac{b C_{p1s}}{2 C_{p1s}} \frac{b C_{p1s}}{C_{p1s}} \frac{b C_{p1s}}{C_{p1s}} \right)
$$

(102)

39
where
\[
C_{p \Delta a} = \frac{4}{\rho S \beta} \frac{\dot{p}}{\rho S \beta} \frac{\dot{q}}{\rho S \beta} + \frac{b C_{p a}}{2 C_{p \Delta a}} (q \sin \theta \tan \theta + r \cos \theta \tan \theta) - \frac{4 \Gamma_1}{\rho S \beta} \frac{\dot{q}}{\rho S \beta} \frac{\dot{r}}{\rho S \beta} \frac{C_{p b}}{C_{p \Delta a}} (-\dot{\psi} + \chi) - \frac{C_{p b}}{2 C_{p \Delta a}} \frac{r}{V}
\]

Adding and subtracting \( \delta_a^{des} \) from Equation (99) gives us
\[
\dot{\phi} = \frac{8}{\rho S \beta^2 C_{p a}} \frac{\dot{p}}{\rho S \beta^2 C_{p a}} \frac{\dot{q}}{\rho S \beta^2 C_{p a}} \frac{\dot{r}}{\rho S \beta^2 C_{p a}} - \frac{2 C_{p b}}{b C_{p a}} (\delta_a^{des} - \delta_a^{des}) - \frac{2 V C_{p a}}{b C_{p a}} \frac{\dot{q} \sin \phi \tan \phi}{\beta} + r \cos \phi \tan \phi - \phi_m
\]
\[
= - \gamma_1 \dot{\phi} - 2 V C_{p a} \left( \Psi^T \dot{K} - \dot{\Psi}^T K \right)
\]
\[
= - \gamma_1 \dot{\phi} - 2 V C_{p a} \Psi^T \dot{K}.
\]

If we define our Lyapunov candidate function to be
\[
V = \frac{1}{2} \dot{\phi}^2 + \frac{V C_{p a}}{b C_{p a}} \dot{K}^T \Gamma^{-1} \dot{K}
\]
and take its derivative to get
\[
\dot{V} = \dot{\phi} \dot{\phi} + \frac{2 V C_{p a}}{b C_{p a}} \dot{K}^T \Gamma^{-1} \dot{K}
\]
\[
= \dot{\phi} \left( -\gamma_1 \dot{\phi} - \frac{2 V C_{p a}}{b C_{p a}} \Psi^T \dot{K} \right) + \frac{2 V C_{p a}}{b C_{p a}} \dot{K}^T \Gamma^{-1} \dot{K}
\]
\[
= - \gamma_1 \dot{\phi}^2 - \frac{2 V C_{p a}}{b C_{p a}} \Psi^T \dot{K} + \frac{2 V C_{p a}}{b C_{p a}} \dot{K}^T \Gamma^{-1} \dot{K}
\]
\[
= - \gamma_1 \dot{\phi}^2 - \frac{2 V C_{p a}}{b C_{p a}} (V \dot{\phi} \Psi^T + \dot{K}^T \Gamma^{-1}) \dot{K}.
\]

Setting \( V \dot{\phi} \Psi^T + \dot{K}^T \Gamma^{-1} = 0 \) we arrive at our update law
\[
\dot{K} = -V \dot{\phi} \Psi
\]

The algorithm can be summarized as follows.

**Algorithm 8** Nonlinear MRAC for roll attitude hold.
1. Obtain \( \phi, p, r, \alpha, q, \beta \) from sensors.
2. Update the reference model according to \( \dot{\phi}_m = -k_m (\phi - \phi_m) \).
3. Compute:
   \[
   \Psi = \left( \frac{\gamma_1 \dot{\phi} - k_m (\phi - \phi_m)}{V} \right)^T
   \]
4. Update the gain estimate according to: \( \dot{K} = -V \dot{\phi} \Psi \)
5. Compute the aileron command: \( \delta_a = \Psi^T \dot{K} \).
4.6 Flight Data

4.6.1 Platform

The algorithms were flown on MAVs equipped with the Kestrel Autopilot running a 29 MHz Rabbit microcontroller with 512K Flash and 512K RAM. The autopilot is equipped with a variety of sensors, including rate gyros, accelerometers, an absolute pressure sensor for measuring altitude, a differential pressure sensor for measuring airspeed, and a GPS receiver. The autopilot is programmed in C. The adaptive control algorithms run at about 80-90 hertz with this platform.

The airframes flown were a Zagi-style flying wing design (Deliverance), a larger and heavier flying wing design (Ray), and a fixed-wing glider design (Phidipides). In the initial phase of testing, we used Deliverance to tune adaptive control gains and to obtain a comparison between the four algorithms presented in this paper. Afterward, we narrowed our testing to one algorithm (D) and flew with the other airframes.

Deliverance
Weight: 0.734 kg
Wingspan: 1.207 m
Mean geometric chord: 0.254 m

Ray
Weight: 1.321 kg
Wingspan: 1.511 m
Mean geometric chord: 0.305 m

Phidipides
Weight: 0.969 kg
Wingspan: 1.524 m
Mean geometric chord: 0.168 m
Nose to tail: 0.914 m
4.6 Flight Data

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pitch $k_m$</td>
<td>3.0</td>
</tr>
<tr>
<td>Pitch $\gamma_1$</td>
<td>45.0</td>
</tr>
<tr>
<td>Pitch $\Gamma_1$</td>
<td>0.06</td>
</tr>
<tr>
<td>Pitch Leakage gain 1</td>
<td>0.001</td>
</tr>
<tr>
<td>Pitch Leakage gain 2</td>
<td>0.0005</td>
</tr>
<tr>
<td>Roll $k_m$</td>
<td>4.0</td>
</tr>
<tr>
<td>Roll $\gamma_1$</td>
<td>140.0</td>
</tr>
<tr>
<td>Roll $\Gamma_1$</td>
<td>0.005</td>
</tr>
<tr>
<td>Roll Leakage gain 1</td>
<td>0.001</td>
</tr>
<tr>
<td>Roll Leakage gain 2</td>
<td>0.0001</td>
</tr>
</tbody>
</table>

Table 2: Adaptive control gains used in Algorithm D

4.6.2 Implementation

While Lyapunov theory provides for stability, adaptive control engineers have several techniques available to ensure good stability in practice. Many times, flight conditions do not provide sufficient excitation for all of the adaptive parameters. Sensor noise and other disturbances can also cause parameter drift. [13] Some of the parameters may diverge such that when flight conditions change quickly (i.e., we switch from straight, level flight into a climbing turn), the plane can go unstable. We use three techniques in our implementation to improve stability: dead zone, leakage, and normalization.

With dead zone, adaption is stopped once the actual value gets within a certain region around the desired value. Once we get into a flight pattern with little excitation, the $\hat{K}$ parameters will adapt until the plane is close to the desired value. Then adaptation freezes until flight conditions change such that we are no longer within the no-adaption ("dead") zone.

Leakage subtracts off a small percentage of each $\hat{K}$ parameter each time the code is executed. This prevents parameters from ever becoming too large. So as not to affect performance, the leakage gain is kept small, about 0.0001 for most parameters.

Normalization is similar to leakage. Each time through the control loop, all of the $\hat{K}$ parameters are normalized with respect to themselves. This prevents one parameter from become much larger than the others, but preserves the relative weighting between each parameter.

Because all of our MAVs do not have a rudder we have implemented the controller with $\delta_r = 1$.

4.6.3 Results

We wanted to compare the four algorithms against each other and against the usual PD controller used in our lab to control pitch and roll on the MAVs. To obtain this comparison, we flew an hourglass-shaped pattern with legs between 200 and 300 meters long on the same airframe (Deliverance) with each of the four algorithms. The flights incorporated auto-takeoff and auto-land functions, giving the adaptive algorithms full control for the entire flight. We also obtained data for the flight plan flown with more conventional PD control, with gains tuned by hand. For all of the adaptive algorithms, transients die out within about 3 seconds for pitch and about 10 seconds for roll.

Along with the adaptive control plots, we show the values of the $\hat{K}$ parameters adapting with time. In most cases, these parameters settle to a steady value after adaption has been running for a while. Note that while the parameters represent certain combinations of aerodynamic factors and other estimated quantities, they do not, in most cases, converge to these true values. The Lyapunov stability proof used to derive the algorithms does not guarantee that these parameters will converge to the true values. Rather, the proof shows that the parameters stabilize to values that will cause the tracking error to go to zero.

The results show that each adaptive control algorithm’s performance is comparable to PD control. Before each algorithm could be flown, several adaptive control gains (namely $k_m$, $\gamma_1$, and $\Gamma$) had to be tuned. Algorithm D, with the fewest number of gains, was easiest to tune. Algorithm D also is the simplest algorithm to implement and takes up the smallest amount of autopilot code space.
Table 3: Average error in degrees for each algorithm - from last third of hourglass flight tests on Deliverance

We used average error as a metric to compare between the algorithms. We define average error as the average of $\sqrt{(desired - actual)^2}$ over the sample. Each figure included in this paper displays average error over the total sample as well as the average error over each third of the flight time. This shows the gradual improvement in performance characteristic of adaptive control as the parameters become more finely tuned. In most of the figures, the last third of the flight has less average error than the first third. The longer the adaptive algorithms run, the better the performance will be.

The adaptive gains $k_m$, $\gamma_1$, $\gamma_2$, and $\Gamma$ were very difficult to tune for most of the algorithms, especially algorithm C. Algorithms A and B were tuned to give good performance, but did not have very good stability. Based on low average error, good stability, and simplicity of implementation, we chose to focus on algorithm D for further testing. None of the algorithms outperformed PD control, but bear in mind that the control gains were finely tuned to Deliverance’s airframe.

To test the robustness of adaptive control, we attached a flap near the center of Deliverance’s right wing and deployed it mid-flight. By doing this, we simulated what a change in the airframe may have on the controllability of the MAV. With the flap deployed, PD control can still fly the MAV, but with substantial steady-state error on roll. However, algorithm D can be seen to adapt to the change in aerodynamics within a few seconds. When the flap goes back down, the parameters once again adapt back to their previous states.

To further test adaptive control’s robustness, we flew two very different airframes; Ray, a heavier Zagi-style airframe, and Phidipides, a fixed-wing glider design. The plots presented in this paper are flight data from flying Algorithm D with the adaptive gains tuned while flying Deliverance. Both Ray and Phidipides fly well with these same gains. Phidipides in particular was very hard to tune PD gains for, but algorithm D had no problems. This shows the main strength of adaptive control: the ability to fly the same code on different airframes without having to hand-tune gains.

### 4.7 Conclusion

In this paper we have shown the derivation for four adaptive control schemes for pitch and roll. We also present flight test data for each of the schemes. Through comparing performance, stability, and simplicity, Algorithm D was chosen as the best algorithm to pursue for further testing and development on the autopilot. We show that the algorithm performs well on several different airframes without having to tune any gains.
Figure 24: Algorithm A pitch on Deliverance flying an hourglass
4.7 Conclusion

Figure 25: Algorithm A pitch parameters on Deliverance flying an hourglass
Figure 26: Algorithm A roll on Deliverance flying an hourglass
Figure 27: Algorithm A roll parameters on Deliverance flying an hourglass
Figure 28: Algorithm B pitch on Deliverance flying an hourglass
Figure 29: Algorithm B pitch parameters on Deliverance flying an hourglass
Figure 30: Algorithm B roll on Deliverance flying an hourglass
Figure 31: Algorithm B roll parameters on Deliverance flying an hourglass
Figure 32: Algorithm C pitch on Deliverance flying an hourglass
Figure 33: Algorithm C pitch parameters on Deliverance flying an hourglass
Figure 34: Algorithm C roll on Deliverance flying an hourglass
Figure 35: Algorithm C roll parameters on Deliverance flying an hourglass
Figure 36: Algorithm D pitch on Deliverance flying an hourglass
Figure 37: Algorithm D pitch parameters on Deliverance flying an hourglass
4.7 Conclusion

Figure 38: Algorithm D roll on Deliverance flying an hourglass
Figure 39: Algorithm D roll parameters on Deliverance flying an hourglass
Figure 40: PD pitch on Deliverance flying an hourglass
Figure 41: PD roll on Deliverance flying an hourglass
Figure 42: Algorithm D Pitch and Roll with Flap Deployment flying an hourglass. One transit around the hourglass is completed with the flap deployed.
Figure 43: Algorithm D Pitch and Roll parameters with Flap Deployment flying an hourglass. Notice $k_2$ for roll adapting to a new value when the flap is deployed and adapting back to the previous value when the flap is stowed.
Figure 44: PD Control with Flap Deployment flying an hourglass. Notice the steady-state error of about 10 degrees in roll.
4.7 Conclusion

Figure 45: Algorithm D pitch on Ray flying between two waypoints
4.7 Conclusion

Figure 46: Algorithm D pitch parameters on Ray flying between two waypoints
Figure 47: Algorithm D roll on Ray flying between two waypoints
4.7 Conclusion

Figure 48: Algorithm D roll parameters on Ray flying between two waypoints
Figure 49: Algorithm D pitch on Phidipides flying an hourglass
4.7 Conclusion

Figure 50: Algorithm D pitch parameters on Phidipides flying an hourglass
Figure 51: Algorithm D roll on Phidipides flying an hourglass
Figure 52: Algorithm D roll parameters on Phidipides flying an hourglass
5 Students Funded Under Project

<table>
<thead>
<tr>
<th>Student</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steve Griffiths</td>
<td>Magnetometer integration</td>
</tr>
<tr>
<td>Blake Barber</td>
<td>Height above ground sensor</td>
</tr>
<tr>
<td>Andrew Eldridge</td>
<td>Magnetometer integration</td>
</tr>
<tr>
<td>Josh Matthews</td>
<td>Adaptive control</td>
</tr>
<tr>
<td>Nathan Knoebel</td>
<td>Adaptive control</td>
</tr>
<tr>
<td>Steve Osborne</td>
<td>Adaptive control</td>
</tr>
</tbody>
</table>

References


Appendix: PNI Micromag Specifications
### Absolute Maximum Ratings

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Max</th>
<th>Units</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{DD}$</td>
<td>DC Supply Voltage</td>
<td>5.25</td>
<td>VDC</td>
<td></td>
</tr>
<tr>
<td>$V_{IN}$</td>
<td>Input Pin Voltage</td>
<td>VDD+0.3</td>
<td>VDC</td>
<td></td>
</tr>
<tr>
<td>$I_{IN}$</td>
<td>Input Pin Current</td>
<td>10.0</td>
<td>mA DC</td>
<td>25°C</td>
</tr>
<tr>
<td>$T_{STG}$</td>
<td>Storage Temperature</td>
<td>125</td>
<td>°C</td>
<td></td>
</tr>
</tbody>
</table>

Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### Module Characteristics

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Conditions</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Measured at $V_{DD}$</td>
<td>3 VDC, Rbias=43 Ω</td>
<td>0.4</td>
<td>0.5</td>
<td>mA RMS</td>
<td></td>
</tr>
<tr>
<td>Field Measurement Range (1)</td>
<td>3 VDC, Rbias=43 Ω</td>
<td>-1100</td>
<td>1100</td>
<td>μT</td>
<td></td>
</tr>
<tr>
<td>Gain (2) P/N 11594</td>
<td>3 VDC, Rbias=43 Ω</td>
<td>31.24</td>
<td>Count/μT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linearity</td>
<td>Error from best fit straight line at ±300 μT</td>
<td>0.6</td>
<td>1</td>
<td>% of 300 μT</td>
<td></td>
</tr>
<tr>
<td>Resolution</td>
<td>1/Gain</td>
<td>μT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensor Frequency</td>
<td>3 VDC, Rbias=43 Ω, within free Earth’s magnetic field</td>
<td>175</td>
<td>KHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating Temperature Range</td>
<td>-20</td>
<td>70</td>
<td>°C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Storage Temperature Range</td>
<td>-40</td>
<td>125</td>
<td>°C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(1) Field Measurement Range is defined as the monotonic region of the output characteristic curve.

(2) Gain is defined as the change in the number of counts from the ASIC, when the period select is set to 2048, per change in the magnetic field in μT. For situations requiring higher gain and less field measurement range, the gain and resolution can be increased by a factor of 2 by setting the ASIC period select to 4096. When setting higher period selects, beware that the ASIC counter can overflow at the field is strong enough drive the count beyond a signed 16-bit integer. Period select set to 2048 is the highest setting where it is impossible to overflow the counter.

In practical magnetometer applications, a sensor gain calibration is normally performed when the sensor module is in the host system.

Gain will change in response to a 5 VDC supply voltage or with the use of alternate bias resistor values. PNI recommends changing the Rbias value to 75 Ω for 5 VDC operation.
## Pad Descriptions

<table>
<thead>
<tr>
<th>PIN</th>
<th>NAME</th>
<th>FUNCTION</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MOSI</td>
<td>Serial data input (Master Out Slave In)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SSNOT</td>
<td>Active low Chip select for SPI port</td>
<td></td>
</tr>
<tr>
<td>3, 7</td>
<td>AVSS,</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>4, 6</td>
<td>AVDD,</td>
<td>Supply Voltage</td>
<td>Connect to VDD</td>
</tr>
<tr>
<td>5</td>
<td>GND</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>COMP</td>
<td>Comparator output</td>
<td>Used for diagnostics</td>
</tr>
<tr>
<td>9</td>
<td>RESET</td>
<td>Reset input</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>DRDY</td>
<td>Data Ready</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>DHST</td>
<td>High speed oscillator output (output is 1/2 clock speed)</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>VDDBY</td>
<td>Input protection clamp diode common</td>
<td>Connect to VDD</td>
</tr>
<tr>
<td>13</td>
<td>SCLK</td>
<td>Serial Clock input for SPI port</td>
<td>1MHz Max (Relax=100K)</td>
</tr>
<tr>
<td>14</td>
<td>MISO</td>
<td>Serial data output (Master In Slave Out)</td>
<td></td>
</tr>
</tbody>
</table>

## Host Processor Interface

All accesses to and from the PNI-11096 are through a hardware handshaking, synchronous serial interface that adheres to the Motorola SPI protocol. The interface consists of six signals: SCLK, MOSI, MISO, SSNOT, RESET and DRDY. See “PNI-11096, 3-Axis Magnetoresistive Sensor Driver and Controller with SPI Serial Interface” data sheet for detailed information on the interface.

## SPI Port Line Descriptions

- **MOSI** – Master Out, Slave In for the Module SPI port.
- **SSNOT** – Slave Select for the Module SPI port. SSNOT must remain low until the command response is decoded out.
- **SCLK** – Serial Clock input for SPI port.
- **MISO** – Master In, Slave Out for the Module SPI port.

## Hardware Handshaking Line Descriptions

- **RESET** – RESET is usually low. RESET must be toggled from low-high-low.
- **DRDY** – Data Ready. The module returns DRDY. DRDY is low after a RESET; after a command has been received and the sample is ready, DRDY is raised. The table below shows the minimum delay from the end of the command bit being decoded out until DRDY is raised for the various period selections.

<table>
<thead>
<tr>
<th>Period Select</th>
<th>/32</th>
<th>/64</th>
<th>/128</th>
<th>/256</th>
<th>/512</th>
<th>/1024</th>
<th>/2048</th>
<th>/4096</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Delay (msec)</td>
<td>22</td>
<td>22</td>
<td>23</td>
<td>26</td>
<td>26</td>
<td>36</td>
<td>49</td>
<td>77</td>
</tr>
</tbody>
</table>

PNI Corporation