



---

All Faculty Publications

---

2008-8

# Aerobatic Maneuvering of Miniature Air Vehicles Using Attitude Trajectories

James K. Hall

*Brigham Young University - Provo*, hallatjk@gmail.com

Timothy W. McLain

*Brigham Young University - Provo*, mclain@byu.edu

Follow this and additional works at: <https://scholarsarchive.byu.edu/facpub>



Part of the [Mechanical Engineering Commons](#)

## Original Publication Citation

James Hall and Timothy McLain. "Aerobatic Maneuvering of Miniature Air Vehicles Using Attitude Trajectories", AIAA Guidance, Navigation and Control Conference and Exhibit, Guidance, Navigation, and Control and Co-located Conferences, (2008). <http://dx.doi.org/10.2514/6.2008-7257>

---

## BYU ScholarsArchive Citation

Hall, James K. and McLain, Timothy W., "Aerobatic Maneuvering of Miniature Air Vehicles Using Attitude Trajectories" (2008). *All Faculty Publications*. 1508.

<https://scholarsarchive.byu.edu/facpub/1508>

# Aerobatic Maneuvering of Miniature Air Vehicles Using Attitude Trajectories

James K. Hall\* and Timothy W. McLain†

July 24, 2008

## Abstract

We develop aerobatic maneuvering for miniature air vehicles (MAVs) using time-parameterized attitude trajectory generation and an associated attitude tracking control law. We develop two methodologies, polynomial or trigonometric, for creating smooth functions that specify pitch and roll angle trajectories. For both approaches, the functions are constrained by the maneuver boundary conditions for aircraft position and velocity. We develop a feedback control law to regulate aircraft orientation throughout the maneuvers. The performance of our trajectory generation algorithm and our attitude tracking control law is demonstrated through simulated and actual flight tests of aerobatic maneuvers.

## 1 Introduction

The use of miniature air vehicles (MAVs) for aerial surveillance has increased dramatically in recent years as various technologies such as low-cost solid-state sensors and cameras have been developed and integrated. MAVs have been utilized in applications such as tracking forest fire perimeters, searching for lost persons, and urban tactical reconnaissance. Operating in an urban or mountainous environment is particularly challenging for MAVs due to the obstacles that can be encountered. Therefore, much research is being conducted to develop technologies needed to autonomously avoid obstacles while traversing specified regions.

For autonomous aircraft, the problem of avoiding obstacles has been addressed using several different approaches. In [1], a small, unmanned helicopter is used to autonomously explore an unknown urban environment by generating real-time, planar trajectories to navigate around obstacles. The work reported in [2] demonstrated the ability to generate paths for an aircraft flying among obstacles, and [3, 4] demonstrated the ability to plan and control planar trajectories to avoid obstacles. The common element in the autonomous obstacle avoidance literature is the extension of ground robot methods to aircraft, inadvertently imposing an artificial constraint on the maneuver space. By developing the ability to autonomously fly aerobatic maneuvers, we exploit the inherent ability of aircraft to operate in three-dimensions.

Aircraft aerobatic maneuvers have been addressed in the literature under the general category of motion planning. A maneuver is defined as a transition from one steady-state flight condition to another by passing through a sequence of desired locations with an associated desired attitude [5]. Thus, maneuvers can connect segments of steady flight and the concept a library of maneuvers, as presented in [6] for helicopters, would be desirable for autonomous aircraft. In [7], each maneuver in the library was constructed using flight data to build polynomial splines of position and orientation. Thus, developing continuous functions to specify an aerobatic maneuver is subject to constraints on position and velocity at the beginning and ending of the maneuver.

Applying the maneuver boundary conditions, a library of aerobatic maneuvers should include basic maneuvers such as aileron rolls, and loops. Additionally, we include maneuvers specific to autonomous aircraft obstacle avoidance of fixed-wing aircraft, namely: the Immelmann<sup>1</sup> and the Close-Q. The Immelmann, performed by flying a half loop followed immediately by a half roll, would be used when the aircraft needs to reverse directions but lacks sufficient space to turn in the horizontal plane. The Close-Q maneuver, characterized as a loop with a quarter twist, would be used for making a 90 degree turn in minimum horizontal space (see Section 3.2). Each of these maneuvers can be commanded by generating the desired attitude at each instant in time according to functions that meet the boundary conditions of initial and final position and velocity.

---

\*Graduate Research Assistant, Mechanical Engineering

†Professor, Mechanical Engineering, Brigham Young University, Provo, UT 84602

<sup>1</sup>Named for Max Immelmann, German pilot in World War I.

Two different approaches are developed for generating pitch and roll angle trajectories. A simple fifth-order polynomial can be used to generate trajectories in pitch angle or roll angle separately. This simple method can be used to fly a loop maneuver and an aileron roll maneuver or connected in series to fly an Immelmann. For aerobatic maneuvers characterized as the simultaneous variation of the pitch and roll angles, a methodology for obtaining the rotation matrix from the first and second derivatives of desired position [8] will be exploited. Given the desired attitude trajectory produced by the maneuver library, a feedback control law of the Euler angles for pitch and roll and the associated angular rates was developed. We demonstrated the ability to fly aerobatic maneuvers using the orientation tracking feedback control law through flight simulations.

To aid in the development of aircraft control technologies, the Brigham Young University (BYU) Multi-AGent Intelligent Coordination and Control (MAGICC) lab has developed a high-fidelity flight simulation environment for developing autonomous aircraft control methodologies that can be programmed directly onto the MAV autopilot [9, 10] for flight demonstrations. The MAGICC lab development environment was used to develop and validate the attitude tracking control law.

We present a foundational explanation of aircraft attitude representations in Section ???. Two different approaches to time-parameterized attitude trajectories is presented in Section 2, and the trajectory tracking control law is presented in Section 4. The performance of the attitude trajectory generation and tracking algorithms are verified through flight simulations, with the results of various maneuvers shown in Section 5. We present concluding remarks in Section 6.

## 2 Obstacle Avoidance Maneuver Paths

### 3 Attitude Trajectories

The maneuvers in our library were created using both fifth-order polynomials and trigonometric functions. The motivation for applying both approaches is that the aileron roll maneuver cannot be built as a trigonometric function. An idealized aileron roll never deviates from a single heading, altitude, or velocity, meaning that the aircraft is not experiencing any linear accelerations; because there are no linear accelerations, the kinematic constraints allow flight in a straight line at any roll rate. Thus, the polynomial trajectory in roll angle is the correct way to generate the commands for an aileron roll maneuver.

#### 3.1 Polynomial Functions

A time-parameterized trajectory for a loop maneuver and an aileron roll maneuver can be generated using a fifth-order polynomial function [11] in pitch angle or roll angle, respectively. These polynomial functions have the attractive property of being easily generated at each time step in the autopilot code, and the desired angle and its first derivative are continuous, smooth functions. For the aileron roll maneuver, the functions for the desired roll angle and desired roll rate are

$$\phi_d(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (1)$$

$$\dot{\phi}_d(t) = a_1 + 2a_2t + 3a_3t^2 + 4a_4t^3 + 5a_5t^4, \quad (2)$$

where the subscript  $d$  indicates the desired value. For a lateral-axis-only maneuver (aileron roll), the derivative of the roll angle,  $\dot{\phi}$ , is approximately equal to the angular rate about the body-frame  $x$ -axis, denoted as  $p$ . The polynomial coefficients ( $a_n$ ,  $n = 1, \dots, 5$ ) are calculated from the boundary conditions on the roll angle and its first two derivatives, according to the equations:

$$\begin{aligned} a_0 &= \phi_0 \\ a_1 &= \dot{\phi}_0 \\ a_2 &= \frac{\ddot{\phi}_0}{2} \\ a_3 &= \frac{20(\phi_f - \phi_0) - (8\dot{\phi}_f + 12\dot{\phi}_0)t_f + (\ddot{\phi}_f - 3\ddot{\phi}_0)t_f^2}{2t_f^3} \\ a_4 &= \frac{30(\phi_0 - \phi_f) + (14\dot{\phi}_f - 16\dot{\phi}_0)t_f - (2\ddot{\phi}_f - 3\ddot{\phi}_0)t_f^2}{2t_f^4} \\ a_5 &= \frac{12(\phi_f - \phi_0) - (6\dot{\phi}_f + 6\dot{\phi}_0)t_f + (\ddot{\phi}_f - \ddot{\phi}_0)t_f^2}{3t_f^5}, \end{aligned}$$

where the subscript 0 indicates an initial condition ( $t = 0$ ) and subscript  $f$  indicates a final condition ( $t = t_f$ ).

With this formulation, an aileron roll maneuver can be flown by generating the desired roll angle and first derivative as a function of time and specifying that  $\phi_0 = 0$  and  $\phi_f = 2\pi$ . Thus, the trajectory in roll angle completes a full revolution and the aircraft tracking these angles will fly an aileron roll maneuver. The aggressiveness of the maneuver is specified by  $t_f$ , where a smaller final time results in faster turning. For the MAGICC lab MAVs, there is not a method to make an accurate analytical prediction of the maneuver envelope; hence, the limits on  $t_f$  were found through multiple trials of simulated flight maneuvers.

Similarly, a loop maneuver can be flown by following a time-parameterized fifth-order polynomial function for pitch angle and its time derivative, which can be written as

$$\theta_d(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (3)$$

$$\dot{\theta}_d(t) = a_1 + 2a_2t + 3a_3t^2 + 4a_4t^3 + 5a_5t^4, \quad (4)$$

The loop is a purely longitudinal-mode maneuver, implying that  $\dot{\theta}_d$  is equal to the body-frame angular rate about the y-axis, denoted as  $q$ . By tracking the attitude trajectory for desired pitch angle ( $\theta_d = 0 \rightarrow 2\pi$ ), the aircraft will fly a loop maneuver, where the maneuver aggressiveness is again specified by  $t_f$ . As with the aileron roll maneuver, the range of  $t_f$  was found by trial and error using the flight simulator.

The individual aileron roll and loop maneuvers can be combined sequentially to create more interesting maneuvers such as the Immelmann. Fighter pilots developed the Immelmann maneuver to trade airspeed for altitude while reversing directions; it is accomplished by flying a half loop ( $\theta_d = 0 \rightarrow \pi$ ) followed by a half roll ( $\phi_d = -\pi \rightarrow 0$ ).

As mentioned previously, the aggressiveness of these maneuvers is fully specified by the maneuver duration,  $t_f$ , the limits of which are dictated by the turning performance and the thrust characteristics of each individual aircraft. The primary shortcoming of the polynomial attitude trajectories is the inability to specify a maneuver that is a simultaneous combination of pitch and roll angles. Therefore, a novel approach was developed that uses trigonometric functions to describe time-parameterized paths from which a desired attitude can be calculated at each instant in time.

## 3.2 Trigonometric Functions

Building a three-dimensional aerobatic flight maneuver requires simultaneous knowledge of both pitch and roll angles, which can be derived from functions of position and its derivatives in the inertial  $x$ ,  $y$ , and  $z$ -axes. These time-parameterized trigonometric functions can be created using boundary conditions for position and velocity at the start and finish of the maneuver.

### 3.2.1 Time-Parameterized Position Functions

For the purposes of MAV obstacle avoidance, we devised a three-dimensional aerobatic maneuver that we refer to as the Close-Q (see Figure 1), which can be thought of as a loop with a quarter twist. This maneuver would be used for making a 90-degree turn in a minimum amount of horizontal space. The Close-Q boundary conditions are aircraft position at maneuver start—level flight pointing in any compass direction and in the exact same position at the end of the maneuver, pointing in a direction  $\pm 90$  degrees from the initial heading, depending on the desired final direction. The example boundary conditions shown in Figure 1 are aircraft position at maneuver start with wings-level, pointing north and in the exact same position at the end of the maneuver, with wings level, pointing in the negative east direction. The vectors in Figure indicate the aircraft body-frame  $x$  and  $z$ -axes.

Given these boundary conditions, the vector of functions specifying desired position for a Close-Q maneuver can be written as

$$\mathbf{p}_d = \begin{bmatrix} \frac{1}{2}r \sin(\omega t) + r \sin\left(\frac{1}{2}\omega t\right) \\ \frac{1}{2}r \sin(\omega t) - r \sin\left(\frac{1}{2}\omega t\right) \\ -r + r \cos(\omega t) \end{bmatrix}, \quad (5)$$

where  $r$  is the maneuver radius;  $t$  is the maneuver time, which varies from ( $0 \rightarrow 2\pi/\omega$ ); and  $\omega$  is the maneuver rate, which is the ratio of nominal flight velocity and maneuver radius ( $\omega = V/r$ ). Note that the  $z$ -axis function is defined according to the positive-down convention and should not be confused with altitude, which is defined as positive up. The maneuver aggressiveness variable for the trigonometric functions is the maneuver radius,  $r$ . As with the polynomial trajectories, the

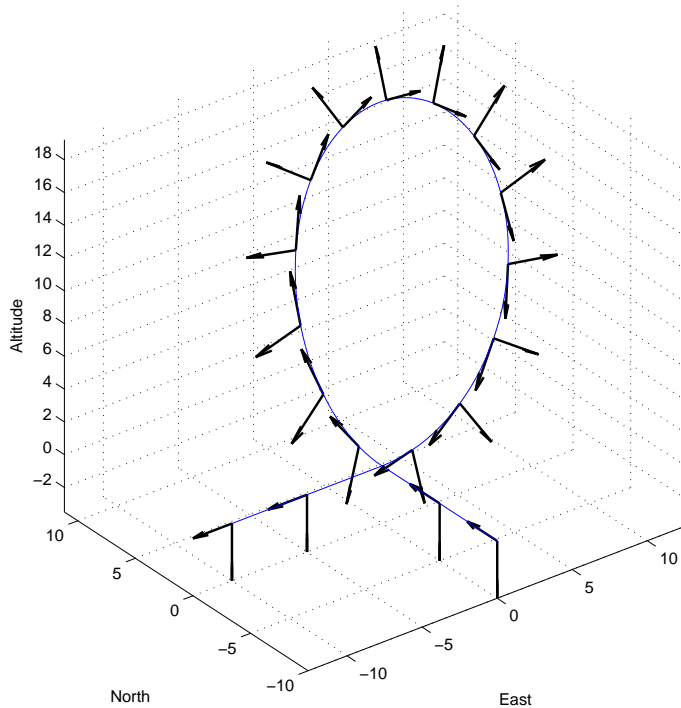


Figure 1: The Close-Q maneuver can be flown to minimize the maneuver footprint when making a 90-degree turn to avoid obstacles in the flight path.

range of  $r$  is limited by the aircraft flight envelope and was computed experimentally. By analyzing the functions in each inertial frame axis individually, as shown in Figure 2, we can gain insight into the construction of the maneuver. Note that by adding trigonometric functions, the  $x_i$  position starts and ends at zero, but the slope at  $t = 0$  is unity, indicating that all of the airspeed is in the  $x_i$  direction at the start of the maneuver. Similarly, by subtracting the same two trigonometric functions for  $y_i$  position, the maneuver again starts and ends at zero, but the slope of the position curve at  $t = 0$  is zero and at  $t = t_f$ , the slope is unity, indicating that all of the airspeed at the end of the maneuver is in the  $y_i$  direction. The plot of  $z_i$  position starts and ends at zero, with zero slope at both instants in time.

Given this framework of position and velocity constraints, one may derive a functional description for many desired maneuvers, such as loops, barrel rolls, etc. Additionally, the Close-Q maneuver can be used to execute a right-angle turn to the left by simply changing the sign on the  $y$ -axis function.

### 3.2.2 Generating Combined Pitch and Roll Trajectories

The process of generating attitude trajectories that simultaneously specify pitch and roll angles involves the following steps: 1) determine a trigonometric function that specifies the desired time-parameterized position in inertial space and meets the boundary conditions for position and velocity; 2) find the first and second derivatives of this function and ensure attitude boundary conditions are satisfied; 3) calculate the desired body-to-inertial reference frame rotation matrix; 4) determine the desired pitch and roll angles; and 5) compute the desired body-frame angular rates. The derivation of this process is based on the assumption that the aircraft velocity vector is aligned with the body-frame  $x$ -axis and aircraft kinematics. These kinematic constraints are specifically smooth transitions to and from steady, level flight, maximum and minimum limits on airspeed, and limits on the curvature of turns.

**Step 1 – Time-Parameterized Position Function** This is accomplished by first specifying the desired, inertial reference frame,  $x$ ,  $y$ , and  $z$ -axis positions as continuously differentiable functions of time subject to the boundary conditions for

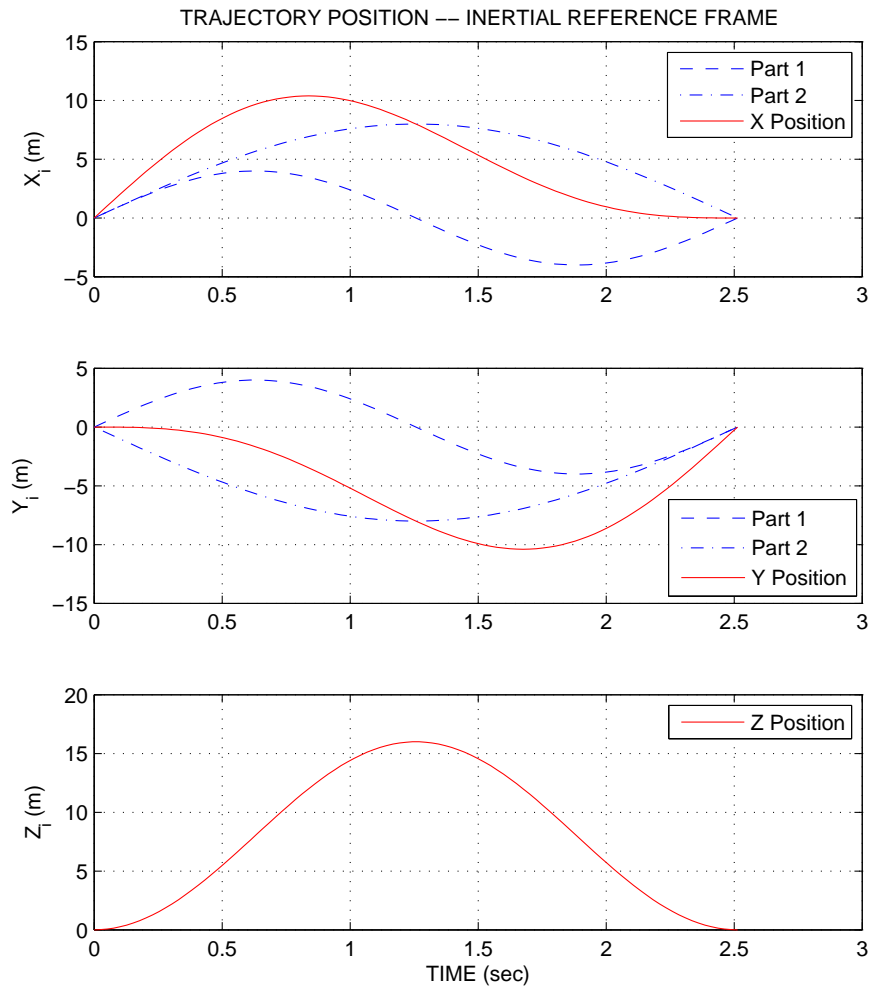


Figure 2: The trigonometric functions for position in each inertial frame axis are constructed by adding various functions to meet the boundary condition constraints. Note that the sign is reversed on the  $z_i$  plot to give a more intuitive view of the third component of the maneuver.

position and velocity. Using the Close-Q maneuver as an example, the position functions are as shown in Equation (5). As noted earlier for the Close-Q maneuver, each axis position function starts and ends at the current aircraft north-east-altitude position, in keeping with the position boundary conditions. The velocity boundary conditions are met in Step 2.

**Step 2 – Position Function Time Derivatives** Taking the first and second derivatives of the Close-Q position functions (5) yields

$$\dot{\mathbf{p}}_d = \begin{bmatrix} \frac{1}{2}V \cos(\omega t) + \frac{1}{2}V \cos\left(\frac{1}{2}\omega t\right) \\ \frac{1}{2}V \cos(\omega t) - \frac{1}{2}V \cos\left(\frac{1}{2}\omega t\right) \\ V \sin(\omega t) \end{bmatrix} \quad (6)$$

and

$$\ddot{\mathbf{p}}_d = \begin{bmatrix} -\frac{1}{2}\omega V \sin(\omega t) - \frac{1}{4}\omega V \sin\left(\frac{1}{2}\omega t\right) \\ -\frac{1}{2}\omega V \sin(\omega t) + \frac{1}{4}\omega V \sin\left(\frac{1}{2}\omega t\right) \\ \omega V \cos(\omega t) \end{bmatrix}, \quad (7)$$

where  $V$  is the nominal airspeed of the aircraft. Examining  $\dot{\mathbf{p}}_d$ , one can see that at time  $t = 0$  the velocity is all in the inertial frame  $x$ -axis and at time  $t = t_f$  the velocity is in the  $y$ -axis, thus meeting the boundary conditions for velocity. Using the expressions for the inertial components of velocity and acceleration allows the derivation of the tangent and normal vector of our position curve. This fact will be exploited in the next step.

**Step 3 - Instantaneous Rotation Matrix** The procedure for finding the rotation matrix ( $\mathbf{R} = [ \mathbf{c}_1 \ \mathbf{c}_2 \ \mathbf{c}_3 ]$ ) is to find the first and third columns of the matrix using kinematic relationships (see Appendix A for the complete derivation) and then use the vector cross product to find the remaining column. The first column of the rotation matrix is found by enforcing the assumption that the velocity vector aligns with the body-frame  $x$ -axis. Thus, the expression for  $\mathbf{c}_1$  (a unit-length vector) can be written as

$$\mathbf{c}_1 = \frac{\dot{\mathbf{p}}_d}{V}, \quad (8)$$

where  $V$  is the magnitude of the aircraft velocity and  $\dot{\mathbf{p}}_d$  is a vector whose components are the velocity vector transformed into the inertial reference frame.

Column three of the rotation matrix,  $\mathbf{c}_3$ , is found by projecting the accelerations onto the unit-vector perpendicular to  $\mathbf{c}_1$ . Defining  $\mathbf{I}$  as a  $3 \times 3$  identity matrix and  $\mathbf{g}$  as the gravity vector expressed in the inertial-frame, the third column of the rotation matrix can be written as

$$\mathbf{c}_3 = \frac{(\mathbf{I} - \mathbf{c}_1 \mathbf{c}_1^T) (\ddot{\mathbf{p}}_d - \mathbf{g})}{-\|(\mathbf{I} - \mathbf{c}_1 \mathbf{c}_1^T) (\ddot{\mathbf{p}}_d - \mathbf{g})\|}, \quad (9)$$

where the negative sign results from defining positive gravitational acceleration in the direction of the inertial reference frame positive  $z$ -axis.

Finally, the second column of the rotation matrix is computed using the cross product

$$\mathbf{c}_2 = \mathbf{c}_3 \times \mathbf{c}_1. \quad (10)$$

Using this procedure, the desired rotation matrix can be computed at each instant of time.

**Step 4 - Desired Euler Angles** Once the rotation matrix is computed, the Euler angle formulation of the rotation matrix, which can be written as

$$\mathbf{R}_{(\phi\theta\psi)} = [ \mathbf{c}_1 \ \mathbf{c}_2 \ \mathbf{c}_3 ], \quad (11)$$

where

$$\begin{aligned} \mathbf{c}_1 &= \begin{bmatrix} \cos \theta \cos \psi \\ -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi \\ \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi \end{bmatrix} \\ \mathbf{c}_2 &= \begin{bmatrix} \cos \theta \sin \psi \\ \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi \\ -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi \end{bmatrix} \\ \mathbf{c}_3 &= \begin{bmatrix} -\sin \theta \\ \sin \phi \cos \theta \\ \cos \phi \cos \theta \end{bmatrix} \end{aligned}$$

can be used to calculate the desired pitch and roll angles ( $\theta_d$  and  $\phi_d$ ) using the following equations:

$$\theta_d = -\sin^{-1}[\mathbf{c}_3(1)] \quad (12)$$

$$\phi_d = \tan^{-1}[\mathbf{c}_3(2), \mathbf{c}_3(3)], \quad (13)$$

which were extracted from the components of Equation 11.

**Step 5 - Desired Angular Rates** The complete derivations for the desired angular rates are shown in Appendix A. We define the variable  $g_{bz}$  as the component of gravity in the body-frame  $z$ -axis and  $a_{bz}$  as the component of the aircraft acceleration vector also in the body-frame  $z$ -axis. Thus, the desired angular rate about the body-frame  $y$ -axis can be written as

$$q_d = \frac{-(g_{bz} + a_{bz})}{V}. \quad (14)$$

Similarly, we define  $a_{bx}$  as the component of the aircraft acceleration vector also in the body-frame  $x$ -axis. The variable  $r$  is defined as the measured angular rate about the  $z$ -axis (yaw rate). Lastly, the variable  $p_{by}^{(3)}$  is the component of jerk in the body-frame  $y$ -axis. Given these terms, the desired angular rate about the body-frame  $x$ -axis can be calculated using the expression

$$p_d = \frac{1}{a_{bz}} \left( r a_{bx} + p_{by}^{(3)} \right). \quad (15)$$

Examination of the equations for the trigonometric approach to trajectory generation makes it evident why this approach cannot be used for the aileron roll maneuver; the roll angle is varied without changing the direction of the velocity vector as needed in Equation (20) and without any linear body-frame accelerations as needed in Equation (25). This is because in the differentially flat aircraft flight model, roll rate is independent of the body-frame accelerations as long as the velocity vector alignment assumption holds. A complete library of maneuvers should incorporate the ability to command aileron rolls as well as three-dimensional maneuvers, thus both the polynomial and trigonometric approaches are important.

The function shown in Equation (5) is the trajectory for a Close-Q maneuver. Similarly, the function for a loop maneuver can be written as

$$\mathbf{p}_d = \begin{bmatrix} r \sin(\omega t) \\ 0 \\ -r + r \cos(\omega t) \end{bmatrix} \quad (16)$$

where the aggressiveness of the maneuver is determined by the radius,  $r$ , and  $\omega = V/r$ . The duration of the maneuver is  $2\pi/\omega$ . The first portion of an Immelmann maneuver uses the exact same trajectory function as the loop, however, the duration is cut in half,  $\pi/\omega$ . Additionally, subsequent to the half-loop portion of the maneuver, the controller must command a half-roll using a polynomial trajectory in roll angle.

Given the desired pitch and roll angles and the body-frame angular rates, we can build a feedback control law to track the attitude trajectories to autonomously fly aerobatic maneuvers.



## 4 Attitude Trajectory Tracking Control Law

Defining a maneuver as the transition between two steady-state flight conditions leads naturally to a simple and effective proportional-derivative (PD) maneuver control law that can be tuned for a specific aircraft and maneuver aggressiveness. The control law can be written as

$$\delta_{el} = k_{\theta} (\theta_d - \theta) + k_q (q_d - q) \quad (17)$$

$$\delta_{ail} = k_{\phi} (\phi_d - \phi) + k_p (p_d - p) \quad (18)$$

where the actual values of  $\theta$ ,  $\phi$ ,  $p$  and  $q$  are aircraft state variables, and  $k_{\theta}$ ,  $k_q$ ,  $k_{\phi}$  and  $k_p$  are the feedback gains. These four gains must be tuned for the specified maneuver aggressiveness and airframe turn rate and thrust characteristics. For loop, aileron roll, and Immelmann maneuvers, the  $\delta_{el}$  and  $\delta_{ail}$  are entirely independent of each other so only two gains are tuned for each of these maneuvers. However, for the Close-Q maneuver, all four gains need to be adjusted simultaneously to achieve good tracking of the desired aircraft orientation. In addition to tuning the controller gains, the maneuver aggressiveness variables (maneuver duration or maneuver radius) need to be adjusted to ensure the aircraft has sufficient thrust and control authority to complete the maneuver. The tuning of the maneuver variables was accomplished using an iterative process in the flight simulation environment. For example, the MAV used in this research was unable to turn a loop faster than 2 seconds for lack of control authority and had insufficient thrust available avoid stall speed during a 5-second loop.

## 5 Results

The trajectory generation algorithms and maneuver tracking controller were developed in simulation where the gains were tuned prior to being flight tested.

### 5.1 Simulation Results

Flight simulations of various maneuvers commanded by the attitude trajectory generation algorithms and tracking controller were conducted using the Aviones flight simulator. The aileron roll maneuver was executed using the polynomial functions to generate the attitude trajectories, and the loop, Immelmann and Close-Q maneuvers were performed using the combined attitude trajectory based on trigonometric functions. The maneuver aggressiveness variables are annotated with each figure. For each maneuver the starting altitude was 100 meters, the flight airspeed was 13 meters per second, and the initial heading was north.

The simulation results for the aileron roll maneuver show that this idealized model is not completely accurate; because the velocity vector is not perfectly aligned with the body-frame  $x$ -axis, it produces a loss of altitude when the aircraft is inverted. Additionally, there is a loss of lift when the roll angle approaches 90 and 270 degrees, contributing to the loss of altitude shown in Figure 3.

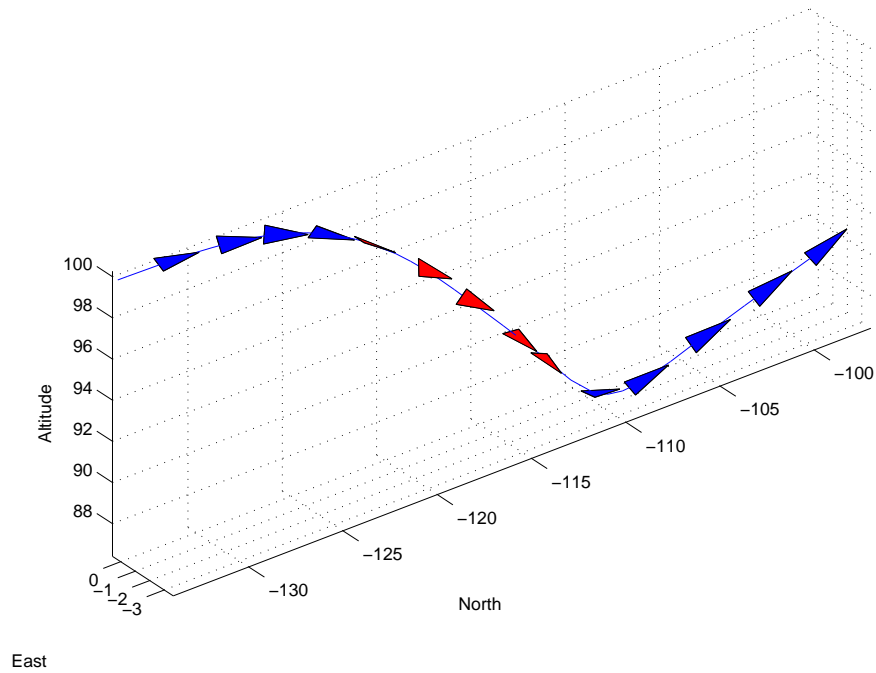


Figure 3: An aileron roll maneuver ( $t_f = 2$ ) was flown autonomously in the flight simulator. The bi-colored triangles indicate the aircraft orientation at various positions.

As mentioned, the aileron roll maneuver was accomplished using the fifth-order polynomial trajectories, however, the loop, Immelmann and Close-Q maneuvers shown in Figures 4- 6 were generated using the trigonometric functions for desired attitude and angular rates. The results for a loop maneuver are shown in Figure 4. Note that although a loop maneuver was flown, the shape of the path followed by the aircraft is not a perfect circle and the final altitude is below the initial level. This reflects the fact that the controller has worked to match the attitude specified by the trajectory generation algorithm, not necessarily to follow a circular path.

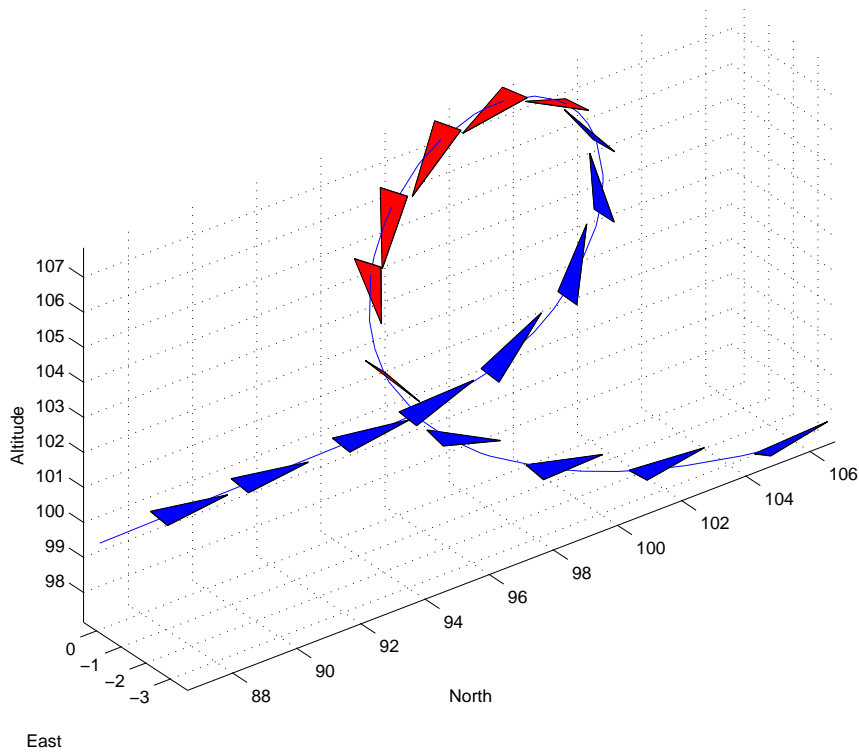


Figure 4: An autonomous loop maneuver ( $t_f = 2$ ) is shown using bi-colored triangles to indicate the aircraft orientation at various locations.

By combining the half loop with the half aileron roll in series, two new maneuvers can be performed, the Split-S and the Immelmann. Flying half of a loop followed by an half roll results in an Immelmann maneuver as shown in Figure 5. Notice that the aircraft was in an inverted, slightly nose-down attitude when the aileron roll portion of the maneuver was executed, causing a rapid decrease in altitude back to the nominal flight level.

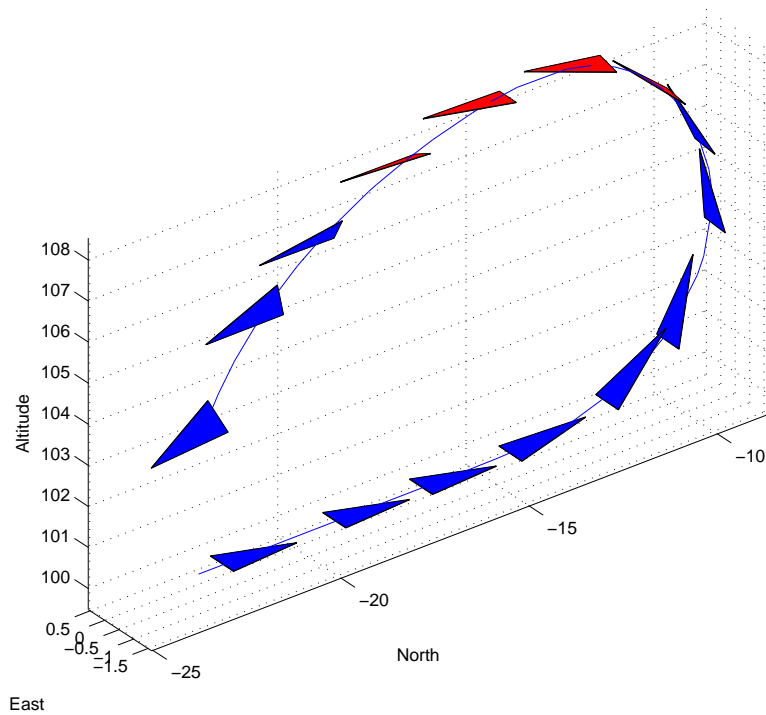


Figure 5: An Immelmann maneuver with combined half loop maneuver duration ( $t_f = 2$ ) and half roll maneuver duration ( $t_f = 2$ ).

The tuning of the gains to achieve accurate attitude tracking of both roll and pitch angles was extremely sensitive to errors in roll angle; a small error in roll angle resulted in greatly degraded maneuver performance. Therefore, the controller gains were tuned iteratively to ensure accurate tracking of the desired aircraft attitude during all four maneuvers with a single set of gains.

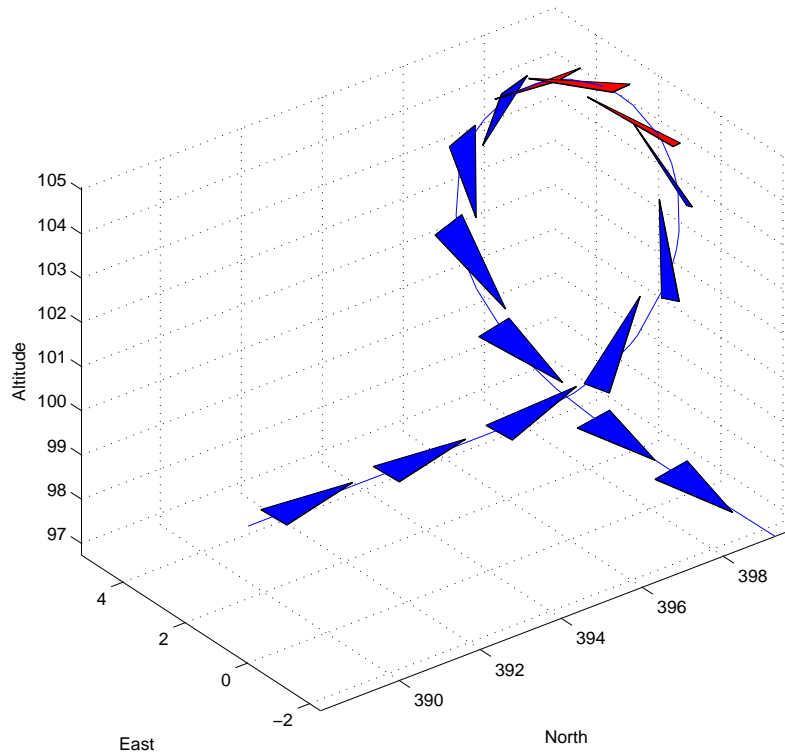


Figure 6: The autonomous Close-Q maneuver ( $r = 5$ ) is demonstrated making a right-hand turn. The bi-colored triangles shown the aircraft orientation throughout the maneuver.

## 5.2 Flight Results

The polynomial trajectory generation algorithm was used to generate roll angle and rate commands to fly an aileron roll maneuver. The aircraft was flying a constant altitude, waypoint navigation, flight plan when the aileron roll maneuver was commanded. Referring to Figure 7, the pitch and roll angles ( $\theta$  and  $\phi$ ) are plotted as a function of time. The roll maneuver can be seen in the angle  $\phi$  rapidly increasing to 180 degrees, wrapping around to -180 degrees and proceeding back to 0. It is important to note that the same gains found through simulations were used for the flight test. The greater control authority of the actual MAV as compared to the simulator model resulted in turning the maneuver in less than two seconds and the large overshoot, as can be seen in Figure 7.

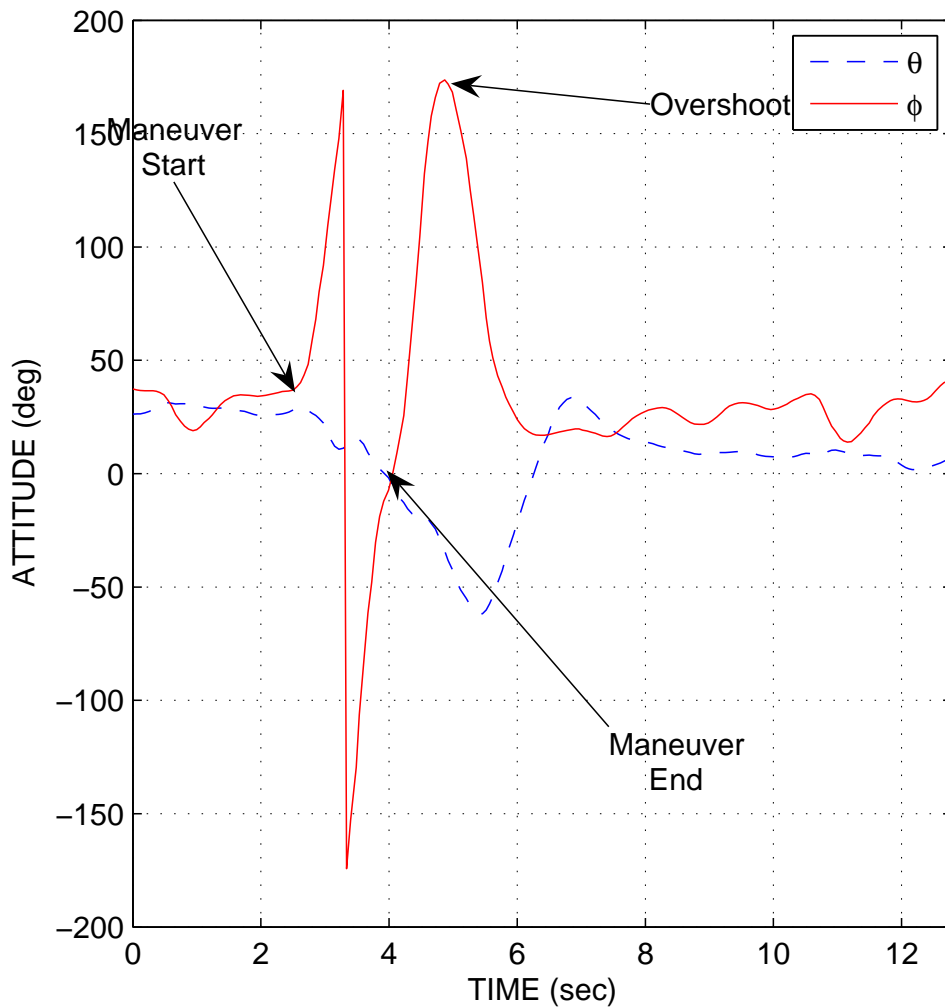


Figure 7: An autonomous aileron roll maneuver ( $t_f = 2$ ) was flown by the MAV. The plot shows  $\phi$  increasing up to 180 degrees, wrapping to -180 degrees and continuing back to 0, indicating a complete revolution about the x-axis.

The MAV experienced large control surface deflections immediately after the maneuver, see Figure 8, to compensate for the large overshoot in roll angle and the loss of altitude. By aggressively actuating the ailerons and elevator, the MAV was able to quickly recover to the desired altitude and course after completing the maneuver.

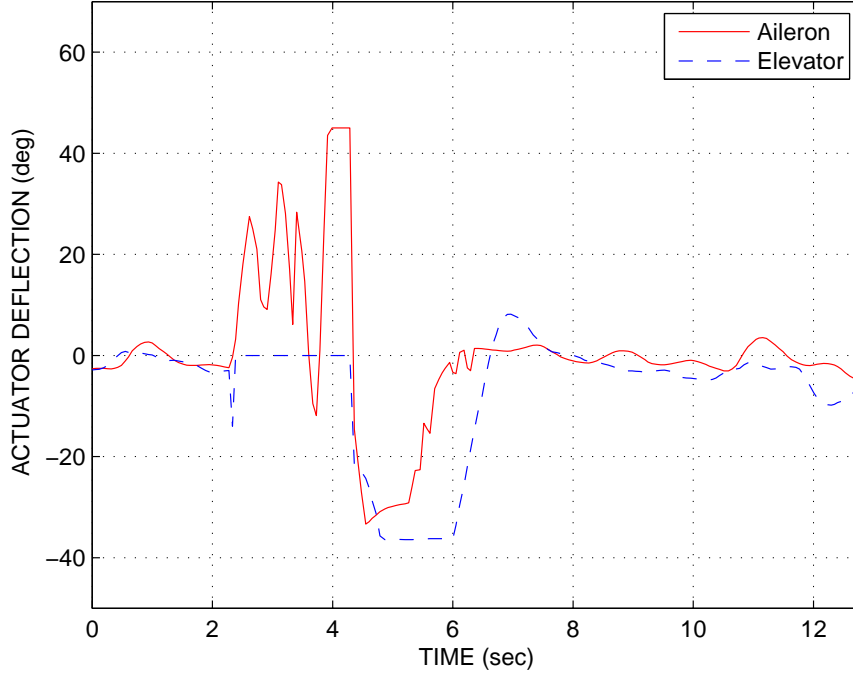


Figure 8: The aileron and elevator deflections for the aileron roll maneuver show the large deflection necessary to track the desired roll angle. The control surface deflections immediately after maneuver completion were caused by the aircraft rejoining the waypoint path and altitude.

The simulation and flight results for the various maneuvers demonstrated the ability of the two trajectory generation algorithms, polynomial and trigonometric, to command maneuvers using time-parameterized trajectories of attitude. The results also validate the performance of the attitude tracking control law in commanding the aircraft through the maneuvers.

## 6 Conclusion

We have presented a methodology for flying aerobatic maneuvers using time-parameterized attitude trajectories. We developed a systematic approach to creating polynomial and trigonometric functions of position and the derivatives from which the attitude trajectories were generated. The key feature used when defining the position functions was meeting the desired maneuver boundary conditions.

We were able to define and execute a variety of maneuvers in the flight simulation environment. The trajectory generation and tracking was used to successfully complete flight tests of an aileron roll maneuver, however, hardware issues prevented other maneuvers from being flown prior to paper publication.

## Appendix A - Derivation of the Rotation Matrix and Angular Rates

**Rotation Matrix, Column 1 ( $c_1$ )** The first column,  $c_1$ , of the rotation matrix can be derived from the expression for the velocity vector alignment assumption. Given an airspeed,  $V$ , and an inertial-frame unit vector  $\hat{e}_x = [1, 0, 0]^T$ , the assumption that the velocity vector aligns with the x-axis can be written as

$$\dot{\mathbf{p}} = VR\hat{e}_x, \quad (19)$$

which can be expanded to yield

$$\dot{\mathbf{p}} = V \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \mathbf{c}_3 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

Performing the multiplication and solving for  $\mathbf{c}_1$  yields

$$\mathbf{c}_1 = \frac{\dot{\mathbf{p}}}{V}, \quad (20)$$

where  $V = \|\dot{\mathbf{p}}\|$  is the airspeed of the MAV.

**Rotation Matrix, Column 3 ( $\mathbf{c}_3$ )** To find the third column,  $\mathbf{c}_3$ , we use the dynamic expression for the second derivative of the inertial reference frame position vector,  $\ddot{\mathbf{p}}$ , which can be written as,

$$\ddot{\mathbf{p}} = \mathbf{g} + \mathbf{R}\mathbf{a}_b, \quad (21)$$

where  $\mathbf{g}$  is the inertial frame gravity vector ( $\begin{bmatrix} 0 & 0 & g \end{bmatrix}^T$ ),  $\mathbf{R}$  is the body-to-inertial rotation matrix, and  $\mathbf{a}_b$  is the vector of body-frame accelerations. Differentiating the rotation matrix orthogonality constraint ( $\mathbf{I} = \mathbf{R}\mathbf{R}^T$ ) yields a kinematic relationship for the time-rate-of-change of the rotation matrix,

$$\dot{\mathbf{R}} = \mathbf{R}\hat{\boldsymbol{\omega}}, \quad (22)$$

where  $\hat{\boldsymbol{\omega}}$  is a skew-symmetric matrix of body-frame angular rates,

$$\hat{\boldsymbol{\omega}} = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}. \quad (23)$$

where  $p$ ,  $q$ , and  $r$  are defined as the angular rates about the  $\mathbf{x}$ ,  $\mathbf{y}$ , and  $\mathbf{z}$ -axis, respectively. The third column of the rotation matrix is found from Equation (21), which can be written as

$$\begin{aligned} \ddot{\mathbf{p}} - \mathbf{g} &= \mathbf{R}\mathbf{a}_b \\ &= \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \mathbf{c}_3 \end{bmatrix} \begin{bmatrix} a_{bx} \\ 0 \\ a_{bz} \end{bmatrix} \end{aligned} \quad (24)$$

Recalling that the columns of the rotation matrix are orthogonal unit vectors, the following two properties for inner products are true:

$$\begin{aligned} \mathbf{c}_i^T \mathbf{c}_j &= 0, \text{ if } i \neq j \\ \mathbf{c}_i^T \mathbf{c}_j &= 1, \text{ if } i = j. \end{aligned}$$

Thus, we can solve Equation (24) by multiplying both sides by the vector  $\mathbf{c}_1^T$  calculated in Equation (20) to yield

$$\mathbf{c}_1^T (\ddot{\mathbf{p}} - \mathbf{g}) = a_{bx} + 0,$$

which can be substituted back into Equation (24) to yield

$$\begin{aligned} \ddot{\mathbf{p}} - \mathbf{g} &= \mathbf{c}_1^T (\ddot{\mathbf{p}} - \mathbf{g}) \mathbf{c}_1 + a_{bz} \mathbf{c}_3 \\ a_{bz} \mathbf{c}_3 &= \ddot{\mathbf{p}} - \mathbf{g} - \mathbf{c}_1 \mathbf{c}_1^T (\ddot{\mathbf{p}} - \mathbf{g}) \\ a_{bz} \mathbf{c}_3 &= (\mathbf{I} - \mathbf{c}_1 \mathbf{c}_1^T) (\ddot{\mathbf{p}} - \mathbf{g}). \end{aligned}$$

Physically, it can be seen that the matrix  $(\mathbf{I} - \mathbf{c}_1 \mathbf{c}_1^T)$  projects the acceleration vector onto the unit vector perpendicular to  $\mathbf{c}_1$ . The magnitude of the acceleration is found from the norm of  $(\mathbf{I} - \mathbf{c}_1 \mathbf{c}_1^T) (\ddot{\mathbf{p}} - \mathbf{g})$  and its direction is negated to account for the aircraft definition of positive  $\mathbf{z}$ -axis pointing out the aircraft belly. Thus, the third column of the rotation matrix can be written

$$\mathbf{c}_3 = \frac{(\mathbf{I} - \mathbf{c}_1 \mathbf{c}_1^T) (\ddot{\mathbf{p}} - \mathbf{g})}{-\|(\mathbf{I} - \mathbf{c}_1 \mathbf{c}_1^T) (\ddot{\mathbf{p}} - \mathbf{g})\|}. \quad (25)$$



**Body-Frame Angular Rates** We derive expressions for the desired body-frame angular rates,  $p_d$  and  $q_d$ , starting with the derivative of the coordinated flight condition shown in Equation (19), which can be written as

$$\dot{\mathbf{p}} = \dot{V}\mathbf{R}\hat{\mathbf{e}}_x + V\dot{\mathbf{R}}\hat{\mathbf{e}}_x. \quad (26)$$

Setting this relationship equal to the right side of Equation (21) yields

$$\mathbf{g} + \mathbf{R}\mathbf{a}_b = \dot{V}\mathbf{R}\hat{\mathbf{e}}_x + V\dot{\mathbf{R}}\hat{\mathbf{e}}_x,$$

which can be expanded according to

$$\begin{aligned} \mathbf{g} + \mathbf{R}\mathbf{a}_b &= \dot{V}\mathbf{R}\hat{\mathbf{e}}_x + V\dot{\mathbf{R}}\hat{\mathbf{e}}_x \\ \mathbf{g} + \mathbf{R}(a_{bx}\hat{\mathbf{e}}_x + a_{bz}\hat{\mathbf{e}}_z) &= \dot{V}\mathbf{R}\hat{\mathbf{e}}_x + V\mathbf{R}\boldsymbol{\omega} \times \hat{\mathbf{e}}_x \\ \mathbf{g} + \mathbf{R}(a_{bx}\hat{\mathbf{e}}_x + a_{bz}\hat{\mathbf{e}}_z) &= \dot{V}\mathbf{R}\hat{\mathbf{e}}_x + V\mathbf{R}(r\hat{\mathbf{e}}_y - q\hat{\mathbf{e}}_z) \\ \mathbf{R}^T\mathbf{g} + a_{bx}\hat{\mathbf{e}}_x + a_{bz}\hat{\mathbf{e}}_z &= \dot{V}\hat{\mathbf{e}}_x + Vr\hat{\mathbf{e}}_y - Vq\hat{\mathbf{e}}_z. \end{aligned}$$

This expression can be isolated into  $x$ ,  $y$ , and  $z$  components, where the  $z$ -axis component yields the desired angular rate about the body-frame  $y$ -axis. This desired angular rate can be written as

$$q_d = \frac{-(g_{bz} + a_{bz})}{V}, \quad (27)$$

where  $g_{bz}$  is the component of gravity in the body-frame  $z$ -axis.

The body-frame  $x$ -axis desired angular rate,  $p_d$ , is derived by differentiating Equation (21), yielding the third derivative of position,

$$\begin{aligned} \mathbf{p}^{(3)} &= \dot{\mathbf{g}} + \dot{\mathbf{R}}\mathbf{a}_b + \mathbf{R}\dot{\mathbf{a}}_b \\ &= 0 + \mathbf{R}(\boldsymbol{\omega} \times \mathbf{a}_b + \dot{\mathbf{a}}_b) \\ &= \mathbf{R} \left( \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} a_{bx} \\ 0 \\ a_{bz} \end{bmatrix} + \begin{bmatrix} \dot{a}_{bx} \\ 0 \\ \dot{a}_{bz} \end{bmatrix} \right) \\ &= \mathbf{R} \left( \begin{bmatrix} qa_{bz} \\ ra_{bx} - pa_{bz} \\ -qa_{bx} \end{bmatrix} + \begin{bmatrix} \dot{a}_{bx} \\ 0 \\ \dot{a}_{bz} \end{bmatrix} \right). \end{aligned} \quad (28)$$

Equation (28) can be rearranged and reduced to vector components (using the unit vector  $\hat{\mathbf{e}}_y = [0, 1, 0]^T$ ) to yield

$$\begin{aligned} p_d &= \frac{1}{a_{bz}} \left( ra_{bx} + \hat{\mathbf{e}}_y^T \mathbf{R}^T \mathbf{p}^{(3)} \right) \\ &= \frac{1}{a_{bz}} \left( ra_{bx} + \mathbf{p}_{by}^{(3)} \right), \end{aligned} \quad (29)$$

the desired body-frame angular rate about the  $x$ -axis.

## 7 Acknowledgments and Disclaimer

The authors would like to express gratitude to Dr. John Hauser for insight into autonomous maneuvering flight.

The views expressed in this document are those of the author and do not necessarily reflect the official policy or position of the Air Force, the Department of Defense, or the U.S. Government.

## References

- [1] D. Shim, H. Chung, H. Kim, and S. Sastry, "Autonomous Exploration in Unknown Urban Environments for Unmanned Aerial Vehicles," *2005 AIAA Guidance, Navigation, and Control Conference and Exhibit*, pp. 1–8, 2005.

- [2] L. Singh and J. Fuller, "Trajectory Generation for a UAV in Urban Terrain, using Nonlinear MPC," *Proceedings of the American Control Conference*, vol. 3, 2001.
- [3] S. Griffiths, J. Saunders, A. Curtis, T. McLain, and R. Beard, "Obstacle and Terrain Avoidance for Miniature Aerial Vehicles."
- [4] S. Park, J. Deyst, and J. How, "Performance and Lyapunov Stability of a Nonlinear Path-Following Guidance Method," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 6, pp. 1718–1728, 2007.
- [5] S. Lindemann and S. LaValle, "Current Issues in Sampling-Based Motion Planning," *Proceedings of the International Symposium of Robotics Research*, 2003.
- [6] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-Time Motion Planning for Agile Autonomous Vehicles," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 116–129, 2002.
- [7] M. Mockli, "Guidance and Control for Aerobatic Maneuvers of an Unmanned Airplane," dissertation, Swiss Federal Institute of Technology, Zurich, ETH Zentrum, ML; CH-8092 Zurich; Switzerland, 2006.
- [8] J. Hauser and R. Hindman, "Aggressive Flight Maneuvers," *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 5, 1997.
- [9] R. S. Christiansen, "Design of an Autopilot for Small Unmanned Aerial Vehicles," Master's thesis, Brigham Young University, 2004.
- [10] D. Kingston, R. Beard, T. McLain, M. Larsen, and W. Ren, "Autonomous Vehicle Technologies for Small Fixed Wing UAVs," *AIAA 2nd Unmanned Unlimited Systems, Technologies, and Operations—Aerospace, Land, and Sea Conference and Workshop & Exhibit*, pp. 2003–6559, 2003.
- [11] J. Craig, *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.