Hover, Transition, and Level Flight Control Design for a Single-Propeller Indoor Airplane

Technical Report

Aerospace Controls Laboratory

Department of Aeronautics and Astronautics

Massachusetts Institute of Technology

Adrian Frank, James McGrew, Mario Valenti, Daniel Levine, Jonathan P. How

a_frank@mit.edu, jsmcgrew@mit.edu, valenti@mit.edu, denu@mit.edu, jhow@mit.edu

May 14, 2007

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
14 MAY 2007		**00-00-2007 to 00-00-2007**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Hover, Transition, and Level Flight Control Design for a Single-Propeller Indoor Airplane	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Massachusetts Institute of Technology,Department of Aeronautics and Astronautics,Aerospace Controls Laboratory,Cambridge,MA,02139**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
U.S. Government or Federal Rights License

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**43**	

Abstract

This paper presents vehicle models and test flight results for an autonomous fixed-wing airplane that is designed to take-off, hover, transition to and from level-flight modes, and perch on a vertical landing platform in a highly space constrained environment. By enabling a fixed-wing UAV to achieve these feats, the speed and range of a fixed-wing aircraft in level flight are complimented by hover capabilities that were typically limited to rotorcraft. Flight and perch landing results are presented. This capability significantly eases support and maintenance of the vehicle. All of the flights presented in this paper are performed using the MIT Real-time Autonomous Vehicle indoor test ENvironment (RAVEN).

1 Introduction

Unmanned aerial vehicles (UAVs) are being used to perform a wide variety of missions in different environments. The ability to perform transitions between hover and flight (and vice-versa) in a confined environment enables fixed-wing UAVs to perform a multitude of missions previously limited to rotary-wing UAVs. Reconnaissance, search and rescue, surveillance, and other missions scenarios benefit from having a sensory unit that can maintain a stationary position. In an open environment, a fixed-wing UAV would fly a loiter pattern to remain on station for extended time periods. However, with added spatial constraints imposed by most urban environments, a loiter pattern may not be feasible.

In addition, using fixed-wing UAVs for sensory missions in enclosed spaces can provide additional operational advantages. For example, a fixed-wing UAV can perform quick dashes from one position to another, allowing for quick relocation. Additionally, the capability to hover enables vertical take-offs and landings, which do not require runways or facilities with large spaces. These combined factors make fixed-wing UAVs with transition capabilities ideal for missions where spatial constraints are dominant.

The usefulness of transition capabilities in fixed-wing UAVs has been acknowledged by numerous research teams. The first successful manually controlled transitions were performed in 1954 with the Convair XFY-1 "Pogo" [14]. Currently, there are several research groups working to develop vehicles that possess the capability to fly in both a level-flight and hover configuration. A custom designed, radio-controlled (R/C) airplane was developed at Drexel University. The airplane is manually-controlled in level-flight operations and transitions to a computer-controlled hover configuration upon user input. The airplane has onboard processing and sensing using a commercially available internal measurement unit (IMU) [13]. At Brigham Young University (BYU) a trajectory generator has been used to simulate the actions of a commercially available R/C model of the Convair XFY-1 to study the autonomous hover, flight, and transitions problem [8].

In addition, Researchers at University of Sydney in Australia have designed and built a T-wing tail-sitter currently capable of autonomous hover [9]. An online video presents the vehicle's capability to autonomously take-off from a vertical orientation, hover, transition to conventional

flight, fly a path according to predefined waypoints, transition back to hover, and land. The vehicle uses onboard processing and sensing including GPS measurements for position sensing [11]. Successful autonomous transitions from flight to hover (and vice-versa) have also been performed by researchers at Georgia Tech. Their experiments used a large, commercially available, R/C airplane equipped with appropriate electronics [3].

Although transitions from a traditional aircraft level-flight mode to the hover position (and vise-versa) have been discussed as a means of enabling missions to be performed in enclosed areas, the projects aforementioned have all been undertaken in areas where space has not been a limiting factor. In previous research, autonomous take-offs and landings are generally performed either from a flat, open surface, or by hand-launch. This effectively limits the possibility to launch and land in an environment where space is limited or access is difficult. In addition, to the best of our knowledge the first reported instance of a fixed-wing autonomous UAV perching on vertical landing platform is shown in [12]. However, flight test data for this landing experiment was not provided in this video submission.

This paper presents both vehicle models and test flight results for an autonomous fixed-wing airplane that is designed to take-off to a hover, transition to and from traditional fixed-wing level flight modes, and perch on a vertical landing platform in a space-constrained environment. These test flights were performed indoors, thus introducing critical spatial constraints comparable to elements of an urban environment. A model of the fixed-wing airplane in both hover and level flight is presented, followed by descriptions of the control systems and guidance logic. In addition, this paper presents flight and perching results for a fixed-wing aircraft in hover configuration. Flight test results for the vehicle flying in the transition configuration and level flight mode are also provided. All of the flights provided in this paper were performed using the MIT Real-time Autonomous Vehicle indoor test ENvironment (RAVEN).

2 Hardware and Control Structure

The first objective of the project was to find a vehicle platform that could be used to study and implement technology which allows a fixed-wing airplane to hover, land/perch, transition and fly in

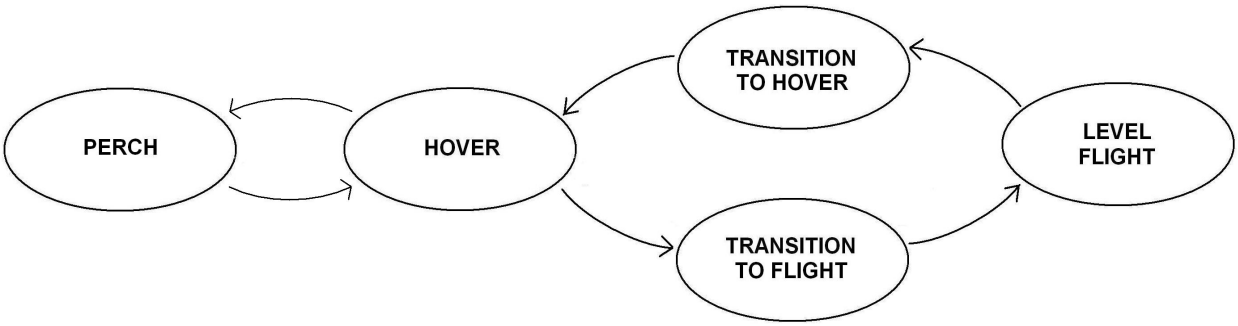


Figure 1: Automaton structure of control logic

an indoor environment and other heavily constrained spaces. Two different commercially-available off-the-shelf (COTS) R/C fixed-wing aircraft were used in these tests. First, the Blade 3D foam airplane was used for early flight testing due to its lightweight, low-cost and durable airframe. Later, the Ikarus Yak 54 Shock Flyer foam airplane was selected for the final flight testing due to its better aerodynamic behavior and higher structural rigidity. Both airplanes have the capability to make the tight turns necessary to enable level flight and transitions in RAVEN's constrained flight space.

Using the control architecture described in [7], each air vehicle can be controlled by a ground-based computer via the trainer port on an R/C transmitter. In addition, the attitude and position of the vehicle can be sensed by a motion capture system by placing lightweight reflective markers on the vehicle. Therefore, all computations are done off-board, allowing the vehicle to maintain its off-the-shelf configuration.

The control structure used for the airplane is that of an automaton with five states: land/perch, hover, transition from hover to flight, flight, and transition from flight to hover. The automaton structure can be seen in Figure 1. Similar structures have been used in the past to control rotary-wing vehicles performing complex maneuvers with success [4]. Hover is defined as a state in which the airplane maintains a steady or semi-steady position with the aircraft's nose perpendicular to the ground. The airplane can be given movement commands in hover. Next, the transition states are used to safely rotate the vehicle from the hover state to the level flight condition and vice-versa. Level flight is a state in which the airplane's wings generate the main portion of the lift and the nose of the airplane is parallel to the ground.

In each of our experiments, the airplane was able to take off from either the ground or vertically from a perching stand before entering the hover state. Both capabilities have been demonstrated and are available on videos located at http://aerobatics.mit.edu. While in hover, the airplane can be moved and rolled to desirable positions and headings. In addition, the vehicle can transition to and from level flight mode, allowing the vehicle to move quickly between desired locations. The level flight state is maintained for as long as waypoints are available. As the airplane approaches the last defined level-flight waypoint, a transition to hover is initiated, causing the airplane to stay over the last waypoint until commanded otherwise. The five parts of the controller are discussed in detail in Sections 4, 5, and 6.

3 Quaternion Attitude Description

The true challenge of performing flights involving transitions is to have a global attitude reference that allows the vehicle's attitude to be described in a non-singular manner regardless of the attitude of the vehicle. There are several methods of describing the attitude of an airplane, the most common being the Euler angle representation. A primary benefit of using Euler angles is that the angles intuitively correspond to roll, pitch, and yaw. However, a deficiency inherent in the Euler angle description becomes clear as a vehicle approaches a pitch angle of $\pm 90^\circ$, when the Euler angle description fails to distinguish between roll and yaw [1].

This singularity does not pose a problem as long as the vehicle's flight envelope is limited in pitch. Given the nature of lift generation in rotary-wing aerial vehicles, it is apparent that large pitch angles are undesirable during nominal operation. The assumption of a limited range of pitch angles also applies to most airplanes, where flight is the only condition of interest. For this project, both hover and flight modes of an airplane are of interest, where hover is defined as a nose-up attitude. Therefore, a pitch angle-limited flight envelope is unfeasible, and a singularity-free description of an arbitrary attitude is necessary.

In order to circumvent the singularity exhibited by the Euler angle attitude description, a quaternion attitude description is adopted as a global attitude reference. A local Euler angle attitude description is then defined based on the global attitude for control. The introduction of

local attitude references allow for control to be done in Euler angles, while relying on a quaternion attitude description to provide a global nonsingular reference. The controller can then switch between different Euler angle attitude descriptions and select one that is optimal for the control task at hand.

A quaternion may be regarded as a 4-tuple of real numbers and of unit length, limiting it to three degrees of freedom. Furthermore, a quaternion can be viewed as a rotation by an angle $\zeta/2$ about a three-dimensional unit-vector $\bar{E}$, such that

$$q = \begin{bmatrix} \cos(\zeta/2) \\ \bar{E}\sin(\zeta/2) \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}.$$

In addition, we define the *zero-quaternion* $q^{(0)} = [1, 0, 0, 0]^T$ to be a reference attitude corresponding to an Euler angle description of $\phi = \theta = \psi = 0°$. Throughout this paper, the zero-quaternion refers to the attitude in which the nose of the airplane is pointed straight North, with the airplane in level flight.

In order to appreciate the quaternion math behind the control logic elucidated in this paper, two concepts merit presentation: quaternion multiplication and quaternion splitting. A quaternion can be viewed as a rotation from the zero-quaternion to the current attitude, which can be represented as a matrix rotation about a vector. Quaternion multiplications are successive rotations applied to a zero-quaternion that are non-commutative, and possess laws similar to those of matrix multiplications. Quaternion multiplication is defined as [5, 6]

$$p \star q = M(p) \cdot q = M(p) \cdot M(q) \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

where p and q are quaternions and $M(p)$ and $M(q)$ are *quaternion matrices* defined as

$$M(\eta) = \begin{bmatrix} \eta_0 & -\eta_1 & -\eta_2 & -\eta_3 \\ \eta_1 & \eta_0 & -\eta_3 & \eta_2 \\ \eta_2 & \eta_3 & \eta_0 & -\eta_1 \\ \eta_3 & -\eta_2 & \eta_1 & \eta_0 \end{bmatrix}.$$

The inverse of the quaternion matrix also follows the rules of matrix multiplication. Since quaternions have a unit length, no scaling occurs and the relationship $M^{-1} = M^T$ is valid.

In controlling a vehicle, the current attitude q^{current} is not necessarily of interest, but rather the deviation, or error, q^{dev} from a desired reference attitude q^{ref}. A deviation quaternion can be defined using quaternion multiplication as

$$q^{\text{current}} = q^{\text{dev}} \star q^{\text{ref}}. \tag{1}$$

Given this definition of a deviation from the reference state, a quaternion split is specified as

$$q^{\text{dev}} \quad = \quad q^x \star q^y \star q^z. \tag{2}$$

The three quaternions resulting from (2) are defined as separate rotations about the x_B-, y_B- and z_B-axis, respectively, applied in reverse order. Consequently, these quaternions represent the rotations performed in an Euler angle description, where components of quaternions $q^x = [q_0^x, q_1^x, 0, 0]^T$, $q^y = [q_0^y, 0, q_2^y, 0]^T$, and $q^z = [q_0^z, 0, 0, q_3^z]^T$ are defined as

$$q_0^x = \cos(\phi) = \frac{1-2\left(q_1^2+q_2^2\right)}{\sqrt{1-4\left(q_0q_2-q_1q_3\right)^2}}, \qquad q_1^x = \sin(\phi) = \frac{2\left(q_0q_1+q_2q_3\right)}{\sqrt{1-4\left(q_0q_2-q_1q_3\right)^2}},$$

$$q_0^y = \cos(\theta) = \sqrt{1-4\left(q_0q_2-q_1q_3\right)^2}, \qquad q_2^y = \sin(\theta) = 2\left(q_0q_2-q_1q_3\right),$$

$$q_0^z = \cos(\psi) = \frac{1-2\left(q_2^2+q_3^2\right)}{\sqrt{1-4\left(q_0q_2-q_1q_3\right)^2}}, \qquad q_3^z = \sin(\psi) = \frac{2\left(q_1q_2+q_0q_3\right)}{\sqrt{1-4\left(q_0q_2-q_1q_3\right)^2}}.$$

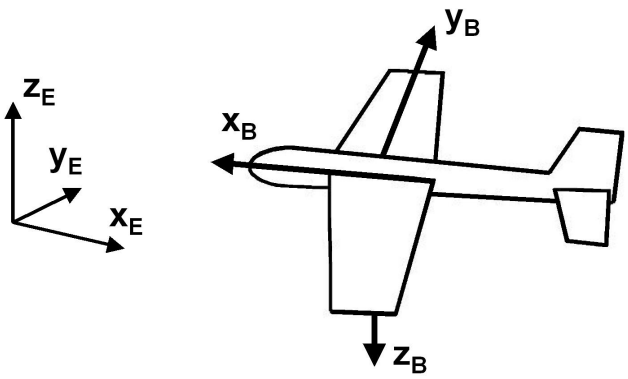


Figure 2: Reference frame and body frame

Even though split quaternions present an intuitive way of working with quaternions, it should be noted that the split defined by (2) is not singularity free. Since the split quaternions in (2) correspond to Euler angles, the singularity at a pitch of $\pm 90^\circ$ is reintroduced. In the control laws suggested in this paper, the quaternion split is only used for control error, or more specifically, the quaternion deviation as defined by (1). To ensure that the magnitude of the pitch error will always be less than 90°during a transition between hover and flight, the controller is equipped with logic that switches the reference frames well before the pitch error reaches $\pm 90^\circ$during the transition. Therefore, the potentially singular behavior does not pose a threat to control performance.

4 Airplane in Hover

4.1 Modeling

In this paper, it is assumed that the Earth reference frame, (x_E, y_E, z_E), is inertial. In addition, there exists a set of conversions between the Earth frame and the body frame, (x_B, y_B, z_B), affixed to the airplane, assumed to be rigid. An illustration of the reference system used can be found in Figure 2.

Since the airplane used in this research has only one propeller, a throttle input will produce both thrust and a moment due to rotation of the propeller. The propeller moment comes from two main contributors. The first contributor is the moment created by a change in propeller speed.

This propeller acceleration moment is calculated as

$$M_{\text{acc}} = \frac{I_{\text{prop}}\,\dot{\omega}}{I_x}, \tag{3}$$

where I_{prop} and I_x are the moments of inertia of the propeller and of the airframe, respectively, both defined in the x_B-direction. The moment M_{acc} about the airframe x_B-axis is created by the angular acceleration of the propeller, $\dot{\omega}$.

Propeller drag is the second contributor to the total propeller moment and can be estimated using the NACA-standardized thrust and power coefficients C_T and C_P [2].

$$C_T = \frac{T}{\rho n^2 d^4} \tag{4}$$

$$C_P = \frac{P}{\rho n^3 d^5} \tag{5}$$

In equations (4) and (5) the thrust $T = mg$ is determined using Newton's second law, where the thrust is the force necessary to keep the airplane in a sustained hover, that is, the mass of the airplane m times the gravitational acceleration g. In this derivation, it is assumed the thrust generated by the propeller is the only force counteracting gravity. The flow over the propeller is assumed to be essentially incompressible, and hence, the density of the air ρ is constant. The remaining parameters are the diameter of the propeller d, and the rotational speed of the propeller in revolutions per second, n. With a known thrust coefficient, the rotational speed of the propeller can be estimated using (4). The moment created by propeller drag can then be estimated using the relationship for power, $P = M_{\text{drag}}\, 2\pi n$. The moment can be expressed as

$$M_{\text{drag}} = \frac{T d C_P}{2\pi C_T}. \tag{6}$$

During hover, the control surfaces are only partially covered by the airflow from the propeller. Utilizing actuator disk theory, the velocity of the propeller airflow at the control surfaces can be estimated. With the additional assumptions of steady flow, discontinuous pressure, and continuous velocity, the velocity of the propeller airflow at the control surfaces in hover can be estimated using

Bernoulli's equation as

$$u_{\text{prop}} = \sqrt{\frac{2T}{\rho A_{\text{disk}}}}, \tag{7}$$

where u_{prop} is the axial velocity of the propeller airflow, and $A_{\text{disk}} = \pi d^2/4$ is the area of the actuator disk representing the spinning propeller.

The lift coefficient derivatives for the ailerons, elevator, and rudder, $C_{L_{\delta_{\text{ail}}}}$, $C_{L_{\delta_{\text{ele}}}}$, and $C_{L_{\delta_{\text{rud}}}}$, are determined using flat plate approximations of the control surfaces. These approximations are predicated on airplane geometry. Lift produced by the wings can be calculated as

$$L_{\text{wing}} = \frac{1}{2}\rho u_\infty^2 A_{\text{wing}} C_{L_\alpha} \alpha. \tag{8}$$

Again, a flat plate approximation of the wing lift coefficient derivative C_{L_α} is valid because of airfoil geometry. The angle of attack can, using small angle approximation, be described as $\alpha = \dot{z}_B/u_\infty$.

While in a hover, aileron deflection does not greatly affect roll. Although the deflection of the ailerons necessary to maintain a desired heading is considerable, a significant amount of drag is produced. This drag contribution can be modeled as

$$D_{\text{ail}} = \left(C_{D,0} + \frac{C_{L_{\delta_{\text{ail}}}}^2 \delta_{\text{ail}}^2}{\pi e \mathcal{AR}}\right) \frac{1}{2}\rho u_{\text{prop}}^2 A_{\text{aileron,prop}}. \tag{9}$$

As a pitching or yawing motion is induced to the airframe, the propeller airflow creates an apparent angle of attack of the control surfaces. This apparent angle of attack in turn causes damping in the aircraft pitch and yaw dynamics. The damping due to propeller airflow lag can be modeled as

$$
\begin{aligned}
q_{\text{damp}} &= \frac{L_{\text{elevator}}\ \alpha_{\text{elevator}}\ A_{\text{elevator}}\ \frac{1}{2}\rho u_{\text{prop}}^2}{I_y} && (10)\\
r_{\text{damp}} &= \frac{L_{\text{rudder}}\ \alpha_{\text{rudder}}\ A_{\text{rudder}}\ \frac{1}{2}\rho u_{\text{prop}}^2}{I_z} && (11)\\
\alpha_{\text{elevator}} &= \frac{L_{\text{prop2elevator}} q}{u_{\text{prop}}} && (12)\\
\alpha_{\text{rudder}} &= \frac{L_{\text{prop2rudder}} r}{u_{\text{prop}}} && (13)\\
C_{\text{ele}_{\text{d}}} &= \frac{q_{\text{damp}}}{q u_{\text{prop}}} && (14)\\
C_{\text{rud}_{\text{d}}} &= \frac{r_{\text{damp}}}{r u_{\text{prop}}} && (15)
\end{aligned}
$$

where q_{d} and r_{d} are the pitch and roll damping moments, $L_{\text{prop2elevator}}$ and $L_{\text{prop2rudder}}$ are the distances from the propeller to the elevator and rudder, and α_{elevator} and α_{rudder} are the apparent angles of attack, respectively. Separating variables from constants, the dimensional elevator and rudder drag constants $C_{\text{ele}_{\text{d}}}$ and $C_{\text{rud}_{\text{d}}}$ can be defined as above.

Based on the model developed so far, the general equations of motion for an airplane can be expressed using a quaternion attitude description as

$$
\begin{aligned}
x_E &= 2\big(q_1q_2 + q_0q_3\big)x_B + \Big[1 - 2\big(q_1^2 + q_3^2\big)\Big]y_B + 2\big(q_2q_3 - q_0q_1\big)z_B\\
y_E &= \Big[1 - 2\big(q_2^2 + q_3^2\big)\Big]x_B + 2\big(q_1q_2 - q_0q_3\big)y_B + 2\big(q_0q_2 + q_1q_3\big)z_B \qquad (16)\\
z_E &= -2\big(q_1q_3 - q_0q_2\big)x_B - 2\big(q_0q_1 + q_2q_3\big)y_B - \Big[1 - 2\big(q_1^2 + q_2^2\big)\Big]z_B
\end{aligned}
$$

$$
\begin{aligned}
\ddot{x}_B &= 2\big(q_1q_3 - q_0q_2\big)g + \frac{1}{m}\delta_{\text{throttle}} - \Big(C_{D,0} + \frac{C_L^2}{\pi e \mathcal{AR}}\Big)\ \frac{1}{2}\rho u_\infty^2 A_{\text{wing}} - \frac{D_{\text{ail}}}{m}\\
\ddot{y}_B &= 2\big(q_0q_1 + q_2q_3\big)g + C_{L_{\delta_{\text{rud}}}}\frac{1}{2}\rho\Big(u_{\text{prop}}^2 A_{\text{rudder,prop}} + u_\infty^2 A_{\text{rudder}}\Big)\delta_{\text{rudder}} \qquad (17)\\
\ddot{z}_B &= \Big[1 - 2\big(q_1^2 + q_2^2\big)\Big]g + C_{L_{\delta_{\text{ele}}}}\frac{1}{2}\rho\Big(u_{\text{prop}}^2 A_{\text{elevator,prop}} + u_\infty^2 A_{\text{elevator}}\Big)\delta_{\text{elevator}} - L_{\text{wing}}
\end{aligned}
$$

$$
\begin{aligned}
\dot{p} &= q\,r\left(\frac{I_y - I_z}{I_x}\right) + C_{L_{\delta_{\text{ail}}}}\frac{1}{2}\rho\Big(u_{\text{prop}}^2 A_{\text{aileron,prop}} L_{\text{aileron,prop}} + u_\infty^2 A_{\text{aileron}} L_{\text{aileron}}\Big)\frac{1}{I_x}\delta_{\text{aileron}} \\
&\quad - \frac{M_{\text{acc}} + M_{\text{drag}}}{I_x} \\
\dot{q} &= p\,r\left(\frac{I_z - I_x}{I_y}\right) + C_{L_{\delta_{\text{ele}}}}\frac{1}{2}\rho\Big(u_{\text{prop}}^2 A_{\text{elevator,prop}} + u_\infty^2 A_{\text{elevator}}\Big)\frac{L_{\text{elevator}}}{I_y}\delta_{\text{elevator}} \\
&\quad - C_{\text{ele}_\text{d}} u_{\text{prop}} q \\
\dot{r} &= p\,q\left(\frac{I_x - I_y}{I_z}\right) + C_{L_{\delta_{\text{rud}}}}\frac{1}{2}\rho\Big(u_{\text{prop}}^2 A_{\text{rudder,prop}} + u_\infty^2 A_{\text{rudder}}\Big)\frac{L_{\text{rudder}}}{I_z}\delta_{\text{rudder}} \\
&\quad - C_{\text{rud}_\text{d}} u_{\text{prop}} r
\end{aligned}
\tag{18}
$$

$$
\begin{aligned}
\dot{q}_0 &= \frac{1}{2}\big(-pq_1 - qq_2 - rq_3\big) \\
\dot{q}_1 &= \frac{1}{2}\big(\;\; pq_0 - qq_3 + rq_2\big) \\
\dot{q}_2 &= \frac{1}{2}\big(\;\; pq_3 + qq_0 - rq_1\big) \\
\dot{q}_3 &= \frac{1}{2}\big(-pq_2 + qq_1 + rq_0\big)
\end{aligned}
\tag{19}
$$

where the vector $[q_0, q_1, q_2, q_3]^T$ is the attitude quaternion. The variables $A_{\text{aileron,prop}}$, $A_{\text{elevator,prop}}$, $A_{\text{rudder,prop}}$, A_{aileron}, A_{elevator}, and A_{rudder} are the areas of the control surfaces covered by the propeller airflow and the total areas of the control surfaces affected by freestream airflow, respectively. The freestream airflow velocity is conventionally denoted as u_∞. The lengths $L_{\text{aileron,prop}}$, L_{aileron}, L_{elevator}, and L_{rudder} are the moment arms of the various control surfaces.

Drag is calculated using the induced drag coefficient $C_{Di} = \frac{C_L^2}{\pi e \mathit{A\!R}}$ and parasitic drag coefficient $C_{D,0}$, where $C_L = C_{L_\alpha}\alpha$ is the lift coefficient of the wing, e is the wing efficiency, and $\mathit{A\!R}$ is the aspect ratio of the wing. The drag acting on the airplane while in a hover condition is negligible. This assumption can be made since the freestream velocity is very low. The force terms in the y_B- and z_B-axis due to control surface deflection are small in hover, again due to very low freestream velocity, and can hence be disregarded.

Due to symmetry of the airframe, the cross-coupled inertia terms $I_{xy} = I_{yz} = 0$. Additionally, the term I_{xz} is considerably smaller than diagonal terms of the inertia tensor (I_x, I_y, and I_z) and is hence disregarded. Since the moment of inertia of the propeller I_{prop} is considerably smaller

than I_x, the propeller torque due to change in rotational speed of the propeller is negligible. All rotational speeds p, q, and r are expected to be small enough such that products thereof can be disregarded without impacting the model accuracy.

The speed of the axial propeller airflow is considered constant while the airplane is in a hover state. It is also known that $u_\infty = 0$ when the vehicle is stationary, and therefore

$$\begin{aligned} \delta_{\text{aileron}} &= \frac{2}{C_{L_{\delta_{\text{ail}}}} \rho u_{\text{prop}}^2 A_{\text{aileron,prop}} L_{\text{aileron,prop}}} M_{\text{drag}} + \hat{\delta}_{\text{aileron}} \\ \delta_{\text{throttle}} &= mg + D_{\text{ail}} + \hat{\delta}_{\text{throttle}} \end{aligned}$$

where $\hat{\delta}_{\text{throttle}}$ is a small deviation from the expected throttle input necessary to maintain hover. The contribution of aileron drag to throttle setting will always be considerably less than that of gravity. Consequently, the aileron drag can be neglected in the horizontal movement terms.

A deviation quaternion can be calculated by projecting the current quaternion onto the product of a heading quaternion $q^{hdg} = [\cos(hdg/2), \sin(hdg/2), 0, 0]^T = [\bar{c}, \bar{s}, 0, 0]$ and a vertical quaternion $q^v = [\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2}, 0]^T$, where hdg is the heading reference. The deviation quaternion represents the attitude error from the reference quaternion (in this case the vertical quaternion) to the attitude in which the airplane is currently hovering.

$$\begin{aligned} q^{\text{current}} &= q^{\text{dev}} \star q^{hdg} \star q^v \\ &= \frac{\sqrt{2}}{2} \begin{bmatrix} \bar{c}\, q_0^{\text{dev}} - \bar{s}\, q_1^{\text{dev}} - \bar{c}\, q_2^{\text{dev}} + \bar{s}\, q_3^{\text{dev}} \\ \bar{s}\, q_0^{\text{dev}} + \bar{c}\, q_1^{\text{dev}} + \bar{s}\, q_2^{\text{dev}} + \bar{c}\, q_3^{\text{dev}} \\ \bar{c}\, q_0^{\text{dev}} - \bar{s}\, q_1^{\text{dev}} + \bar{c}\, q_2^{\text{dev}} - \bar{s}\, q_3^{\text{dev}} \\ -\bar{s}\, q_0^{\text{dev}} - \bar{c}\, q_1^{\text{dev}} + \bar{s}\, q_2^{\text{dev}} + \bar{c}\, q_3^{\text{dev}} \end{bmatrix} \end{aligned}$$

The deviation from the desired state is expected to be small. Consequently, all products of deviated quaternion components other than q_0^{dev} can be disregarded (as q_0^{dev} is near unity when deviation is small).

Using deviated quaternions, the simplified quaternion model can be written as

$$\begin{aligned} \ddot{x}_B &= \frac{1}{m}\hat{\delta}_{\text{throttle}} \\ \ddot{y}_B &= g q_3^{\text{dev}} \\ \ddot{z}_B &= -g q_2^{\text{dev}} \end{aligned} \tag{20}$$

$$\begin{aligned} \dot{p} &= C_{L_{\delta_{\text{ail}}}} \frac{1}{2}\rho u_{\text{prop}}^2 A_{\text{aileron,prop}} L_{\text{aileron,prop}} \frac{1}{I_x}\hat{\delta}_{\text{aileron}} \\ \dot{q} &= C_{L_{\delta_{\text{ele}}}} \frac{1}{2}\rho u_{\text{prop}}^2 A_{\text{elevator,prop}} \frac{L_{\text{elevator}}}{I_y}\delta_{\text{elevator}} - C_{\text{ele}_{\text{d}}} u_{\text{prop}} q \\ \dot{r} &= C_{L_{\delta_{\text{rud}}}} \frac{1}{2}\rho u_{\text{prop}}^2 A_{\text{rudder,prop}} \frac{L_{\text{rudder}}}{I_z}\delta_{\text{rudder}} - C_{\text{rud}_{\text{d}}} u_{\text{prop}} r \end{aligned} \tag{21}$$

$$\dot{q}^{\text{dev}} = \begin{bmatrix} 0 \\ \frac{p}{2} \\ \frac{q}{2} \\ \frac{r}{2} \end{bmatrix} \tag{22}$$

The simplified model can be separated into two fourth-order and two second-order decoupled linear time-invariant (LTI) systems. The fourth-order systems describe the dynamic behavior of pitch and z_B-position, and yaw and y_B-position, respectively. The pitch-z_B loop is affected by the elevator input, whereas the yaw-y_B loop is affected by rudder input. The second order systems describe the x_B loop and roll loop, respectively. Here, throttle controls altitude and the ailerons control roll. It can be noted that when the airplane is near hover, and has a heading reference of 0°, the system maps approximately to $x_E = y_B$, $y_E = z_B$, and $z_E = x_B$.

4.2 Hover Control

The hover controller uses full state feedback from the sensing system, which is made possible by accurate measurement data being available at 100 Hz [7]. In order to improve position-keeping, integrators are incorporated into all four control schemes. The controller gains are optimized using linear quadratic (LQ) control techniques. Large penalties are assigned to pitch, roll, and their

respective derivatives to ensure that the vehicle maintains a hover attitude. Small penalties on the position deviations prevent the vehicle from making sudden movements should the position change. The penalty on altitude deviation is large to ensure that the airplane accurately maintains an altitude, facilitating "perching."

Due to the low structural rigidity of foam airplanes, quick control surface actuation will cause the airframe to twist. This twisting moment can be perceived by the motion capture system as a movement of the airplane, adding noise to the position and attitude measurements. This undesirable effect is partially mitigated by the fact that the servos cannot instantly deflect the control surfaces. Since the servo time constants are small enough to not completely resolve the issue, the LQ design is focused on minimizing rapid changes in control surface position.

5 Level Flight

5.1 Modeling

For an airplane in flight, it is more convenient to use Euler angles rather than quaternions. For this reason, we first introduce the full equations of motion of the airplane in Euler angles.

$$\begin{aligned} x_E &= \cos\theta\ \sin\psi\ x_B + \big(\cos\phi\ \cos\psi + \sin\phi\ \sin\theta\ \sin\psi\big) y_B \\ &\quad + \big(-\sin\phi\ \cos\psi + \cos\phi\ \sin\theta\ \sin\psi\big) z_B \\ y_E &= \big(\cos\theta\ \cos\psi\big) x_B + \big(-\cos\phi\ \sin\psi + \sin\phi\ \sin\theta\ \cos\psi\big) y_B \\ &\quad + \big(\sin\phi\ \sin\psi + \cos\phi\ \sin\theta\ \cos\psi\big) z_B \\ z_E &= \sin\theta\, x_B - \sin\phi\ \cos\theta\, y_B - \cos\phi\ \cos\theta\, z_B \end{aligned} \tag{23}$$

$$\begin{aligned} \ddot{x}_B &= -\sin\theta\, g + \frac{1}{m}\delta_{\text{throttle}} - \left(C_{D,0} + \frac{C_L^2}{\pi e \cancel{AR}}\right)\frac{1}{2}\rho u_\infty^2 A_{\text{wing}} - \frac{D_{\text{ail}}}{m} \\ \ddot{y}_B &= \sin\phi\ \cos\theta\, g + C_{L_{\delta_{\text{rud}}}}\frac{1}{2}\rho\Big(u_{\text{prop}}^2 A_{\text{rudder,prop}} + u_\infty^2 A_{\text{rudder}}\Big)\delta_{\text{rudder}} \\ \ddot{z}_B &= \cos\phi\ \cos\theta\, g + C_{L_{\delta_{\text{ele}}}}\frac{1}{2}\rho\Big(u_{\text{prop}}^2 A_{\text{elevator,prop}} + u_\infty^2 A_{\text{elevator}}\Big)\delta_{\text{elevator}} - L_{\text{wing}} \end{aligned} \tag{24}$$

$$
\begin{aligned}
\dot{p} &= q\,r\left(\frac{I_y - I_z}{I_x}\right) + C_{L_{\delta_{\text{ail}}}}\frac{1}{2}\rho\Big(u_{\text{prop}}^2 A_{\text{aileron,prop}} L_{\text{aileron,prop}} + u_\infty^2 A_{\text{aileron}} L_{\text{aileron}}\Big)\frac{1}{I_x}\delta_{\text{aileron}} \\
&\quad - \frac{M_{\text{acc}} + M_{\text{drag}}}{I_x} \\
\dot{q} &= p\,r\left(\frac{I_z - I_x}{I_y}\right) + C_{L_{\delta_{\text{ele}}}}\frac{1}{2}\rho\Big(u_{\text{prop}}^2 A_{\text{elevator,prop}} + u_\infty^2 A_{\text{elevator}}\Big)\frac{L_{\text{elevator}}}{I_y}\delta_{\text{elevator}} \\
&\quad - C_{\text{ele}_\text{d}} u_{\text{prop}} q \\
\dot{r} &= p\,q\left(\frac{I_x - I_y}{I_z}\right) + C_{L_{\delta_{\text{rud}}}}\frac{1}{2}\rho\Big(u_{\text{prop}}^2 A_{\text{rudder,prop}} + u_\infty^2 A_{\text{rudder}}\Big)\frac{L_{\text{rudder}}}{I_z}\delta_{\text{rudder}} \\
&\quad - C_{\text{rud}_\text{d}} u_{\text{prop}} r
\end{aligned}
\tag{25}
$$

$$
\begin{aligned}
\dot{\phi} &= p + \big(q\sin\phi + r\cos\phi\big)\tan\theta \\
\dot{\theta} &= q\cos\phi - r\sin\phi \\
\dot{\psi} &= \big(q\sin\phi r\cos\phi\big)\frac{1}{\cos\theta}
\end{aligned}
\tag{26}
$$

Unlike an airplane in hover, the airplane in level flight mover follows a predefined trajectory. Hence, a description of position derivatives ($\ddot{x}_B$ and $\ddot{y}_B$) does not necessarily clarify the airplane's dynamic behavior in the level flight condition. Therefore, a modification of the hover model of the airplane (16 - 19) is required for level flight control. The trajectory follower used during flight provides the controller with a lateral acceleration necessary to keep the vehicle on the desired trajectory. Due to the provided reference, a more intuitive way of considering the airplane dynamics is by looking at forward velocity v_{x_B}, altitude z_E, and lateral acceleration a_{lat} rather than a position in the earth frame of reference.

For level flight, the pitch angle can be approximated using small angle approximations for sufficiently large velocities, i.e. not a stalled configuration. The pitch angle necessary to maintain an altitude in level flight, that is, to create sufficient lift to counteract the gravitational force on the airplane, is

$$
\theta_{\text{flight}} = \frac{2mg}{\rho u_\infty^2 C_{L_{\text{wing}}} A_{\text{wing}} + \delta_{\text{throttle}}}. \tag{27}
$$

Based on airplane geometry, the wing lift coefficient $C_{L_{\text{wing}}}$ can be estimated using flat plate ap-

proximation. It can furthermore be assumed that the aileron deflection will be small during flight, due to the increased airflow over the ailerons and the increase in wetted aileron area. The reduction in aileron deflection compared to hover aileron deflection renders the aileron drag component D_{ail} unnecessary.

Now we define

$$\begin{aligned}
\delta_{\text{aileron}} &= 2\frac{M_{acc} + M_{drag} - qr\big(I_y - I_z\big)}{C_{L_{\delta_{\text{ail}}}}\rho\big(u_{\text{prop}}^2 A_{\text{aileron,prop}} L_{\text{aileron,prop}} + u_\infty^2 A_{\text{aileron}} L_{\text{aileron}}\big)} + \tilde{\delta}_{\text{aileron}} \\
\delta_{\text{elevator}} &= 2\frac{-pr\big(I_z - I_x\big)}{C_{L_{\delta_{\text{ele}}}}\rho\big(u_{\text{prop}}^2 A_{\text{elevator,prop}} + u_\infty^2 A_{\text{elevator}}\big) L_{\text{elevator}}} + \tilde{\delta}_{\text{elevator}} \\
\delta_{\text{rudder}} &= 2\frac{-pq\big(I_x - I_y\big)}{C_{L_{\delta_{\text{rud}}}}\rho\big(u_{\text{prop}}^2 A_{\text{rudder,prop}} + u_\infty^2 A_{\text{rudder}}\big) L_{\text{rudder}}} + \tilde{\delta}_{\text{rudder}} \\
\delta_{\text{throttle}} &= m\Big(C_{D,0} + \frac{C_L^2}{\pi e \mathcal{AR}}\Big)\,\frac{1}{2}\rho u_\infty^2 A_{\text{wing}} + 2mg\sin\theta + \tilde{\delta}_{\text{throttle}}
\end{aligned}$$

Altitude changes of the airplane are made mainly by varying the pitch angle of the airplane and thereby pointing the velocity vector slightly upward or downward. Due to this behavior, the altitude is a function of forward velocity and pitch angle as

$$\dot{z}_E = v_{x_B}\sin\big(\theta - \theta_{\text{flight}}\big).$$

The forward velocity during flight can be considered a semi-steady state, where different velocities are possible, but with few changes of reference velocity. Hence, we can define $v_{x_B} = u_\infty$. Note that this relationship is used only for altitude. The assumption of a semi-steady velocity also promotes the description of u_{prop} as a set speed.

In the same manner as the forward velocity, the roll angle can be considered as a semi-steady state. Hence, a desired roll angle ϕ_{ref} and desired pitch angle θ_{ref} can be defined and used to determine the equations of motion for set roll and pitch angles.

The simplified equations of motion for the airplane in flight can be written as

$$\begin{aligned} \dot{v}_{x_B} &= \frac{1}{m}\tilde{\delta}_{\text{throttle}} \\ a_{\text{lat}} &= \big(q\sin\phi_{ref} + r\cos\phi_{ref}\big)u_\infty \\ \dot{z}_E &= \big(\theta - \theta_{\text{flight}}\big)u_\infty \end{aligned} \tag{28}$$

$$\begin{aligned} \dot{p} &= C_{L_{\delta_{\text{ail}}}}\frac{1}{2}\rho\Big(u_{\text{prop}}^2 A_{\text{aileron,prop}} L_{\text{aileron,prop}} + u_\infty^2 A_{\text{aileron}} L_{\text{aileron}}\Big)\frac{1}{I_x}\tilde{\delta}_{\text{aileron}} \\ \dot{q} &= C_{L_{\delta_{\text{ele}}}}\frac{1}{2}\rho\Big(u_{\text{prop}}^2 A_{\text{elevator,prop}} + u_\infty^2 A_{\text{elevator}}\Big)\frac{L_{\text{elevator}}}{I_y}\tilde{\delta}_{\text{elevator}} - C_{\text{ele}_{\text{d}}} u_{\text{prop}}\, q \\ \dot{r} &= C_{L_{\delta_{\text{rud}}}}\frac{1}{2}\rho\Big(u_{\text{prop}}^2 A_{\text{rudder,prop}} + u_\infty^2 A_{\text{rudder}}\Big)\frac{L_{\text{rudder}}}{I_z}\tilde{\delta}_{\text{rudder}} - C_{\text{rud}_{\text{d}}} u_{\text{prop}}\, r \end{aligned} \tag{29}$$

$$\begin{aligned} \dot{\phi} &= p + \big(q\sin\phi_{ref} + r\cos\phi_{ref}\big)\theta_{ref} \\ \dot{\theta} &= q\cos\phi_{ref} - r\sin\phi_{ref} \end{aligned} \tag{30}$$

The equation for the heading or yaw ψ is ignored, since the heading compensator is designed into the guidance logic.

5.2 Flight Control

The model of the airplane in flight can again be divided into four LTI systems. However, the systems are not fully decoupled. Depending on the roll angle, the rudder and elevator will influence pitch and lateral acceleration to different extents respectively. Forward velocity of the aircraft is influenced solely by the throttle. Similarly, roll is controlled by the ailerons.

Full state feedback with an integrator is used to control the forward velocity of the airplane v_{x_B} using throttle. The z_E-control loop uses full state feedback with an integrator term to produce a desired pitch angle θ_{ref}. Depending on the roll angle of the airplane at a specific instant, a mixing of elevator and rudder is used to get the airplane to the desired pitch attitude. Lateral acceleration is commanded by the trajectory follower, which is detailed in Section 7. The lateral acceleration

is produced by a mixing of elevator and rudder in the same manner as the pitch angle. The two contributions to elevator and rudder deflection are then aggregated to produce the true commands. The lateral acceleration reference a_{cmd} is also used to produce a desired roll angle as

$$\phi_{ref} = \arctan\left(\frac{a_{cmd}}{g}\right)$$

With the assumption that the airplane is not ascending or descending rapidly, this reference will put the airplane in an attitude where only the elevator will be used to achieve the desired pitch angle and lateral acceleration, that is, in an attitude corresponding to what would be considered normal flight, where the acceleration along the y_B-axis is zero. All loops are controlled using a PID controller structure, with the exception of the lateral acceleration loop. This loop is controlled with a proportional gain only, since the trajectory follower itself emulates a PD controller.

6 Transition Controller

Transitioning the aircraft between flight regimes allows the vehicle to take advantage of its level flight capabilities and benefits (range, speed, etc.) while enabling the vehicle to operate and land in relatively small spaces using its hovering capabilities. However, enacting this transition can be difficult in a small space. For example, in order to transition from a hover to level flight, the controller must adjust the vehicle's throttle so as to allow the gravitational force acting on the airplane to pull the vehicle's nose down in a "controlled descent," while trimming out its control surfaces to ensure that the vehicle has the heading, pitch, and roll angles needed to complete the transition without concern. Although executing a transition between flight modes may be easy for a well-trained pilot in large spaces, transitions must be performed efficiently in small spaces to allow the vehicle to recover both altitude and position before following its level flight trajectory. In this section, we describe these controller states and how these transitions are performed automatically.

6.1 Transition to Flight

In order to transition to flight in a confined space, the airplane needs to quickly gain speed while losing a minimal amount of altitude. This is achieved by using the hover controller. With an instant movement of the reference waypoint to a distant location, in the direction of the desired transition, these goals can be met satisfactorily. As the airplane tries to move quickly to the new location, the pitch angle is reduced. By careful selection of the distance, the loss of altitude is small, while the airplane gains enough speed to meet the requirements of horizontal flight. It was also found during flight tests that a 5% reduction in throttle helped the airplane pitch forward slightly quicker. This improvement can be attributed to the propeller airflow lag that causes the airplane to resist the desired forward pitching moment. A larger reduction in throttle causes an even quicker drop in pitch. However, if the pitch of the airplane is brought down faster, the airplane loses more altitude. Hence, a balance between altitude loss and pitch rate has to be achieved.

6.2 Transition to Hover

The challenge of transitioning to hover is obviously the opposite of the challenges of transitioning to flight. As the airplane transitions to hover, altitude needs to be maintained, and the forward velocity needs to quickly go from flight velocity to zero. In order to achieve the desired transition characteristics, the elevator and rudder are used in combination to minimize the time used to pitch the airplane to a nose up attitude. This rapid procedure puts the vehicle in a high angle of attack situation throughout the maneuver. Due to the dynamics of the airplane, a major portion of the forward velocity is lost to drag. To prevent the airplane from entering a complete stall, which would cause the aircraft's nose to pitch forward and comprimise the desired transition, the throttle is set slightly lower than what is necessary to keep the airplane in a hover. During transition to hover, the ailerons work in the same manner as during regular flight, that is, the ailerons are used to keep the airplane along the predefined trajectory leading up to the hover reference point.

The transition logic used to transition the airplane from flight to hover is characterized by

$$\delta_{\text{elevator}} = \cos(\phi)\ \delta_{\text{elevator}_{\max}} \frac{C_{L_{\delta_{\text{rud}}}} A_{\text{rudder}} \frac{L_{\text{rudder}}}{I_z}}{C_{L_{\delta_{\text{ele}}}} A_{\text{elevator}} \frac{L_{\text{elevator}}}{I_y}} \tag{31}$$

$$\delta_{\text{rudder}} = \sin(\phi)\ \delta_{\text{rudder}_{\max}} \tag{32}$$

$$\delta_{\text{throttle}} = k\ mg, \tag{33}$$

where k is a number slightly lower than unity. As can be seen, transition to hover is done in an open-loop manner for all controls except for ailerons. The scaling on elevator is based on the assumption that the airplane will have a larger elevator control authority than rudder control authority, which was true for both airplane models used in our flight testing. The scaling factor ensures that the combination of elevator and rudder creates a pure pitching moment regardless of the roll angle of the airplane during transition to hover.

7 Vehicle Guidance Logic

In order to move the airplane in a coordinated manner, a waypoint follower guides the airplane while in hover. During flight, the airplane is guided by a trajectory follower to ensure that the airplane follows a plausible trajectory to get to a desired point. The waypoint follower was originally designed for use with quad rotors, but works for any hovering flight vehicle. The trajectory following logic is based on [10].

7.1 Waypoint Following

While in hover, the airplane is guided by a reference containing an (x, y, z)-position and a heading. In order to move the airplane, this reference point is moved. The waypoint follower implements an open loop solution that performs this movement in a manner which ensures that no steps in the reference input occur. By moving the waypoint at a constant speed towards the final destination of the vehicle, the waypoint follower allows for a slow build up of the proportional error of the controller. As the proportional error increases, a build up of the integral error starts. The integral error eventually gets the vehicle to a point where it has fully caught up with the moving waypoint.

As the waypoint follower reaches the final destination of the vehicle, the movement of the waypoint abruptly stops, causing a slight positional overshoot as the integrator is integrated back down to a hover state.

While in hover, the airplane receives a point at which it should hover. In order to move the airplane, this reference point is moved. This movement of the waypoint is performed by the waypoint follower. The waypoint follower implements an open loop structure and can be described as a rabbit running away from its pursuer. The waypoint changes position in three dimensions, moving at a constant or variable speed towards the final position. As it moves, the proportional part of the controller gets the vehicle moving, and the integral part brings it up to speed. Due to the integral action, some overshoot is expected once the waypoint reaches its final position, where the constant movement of the waypoint stops abruptly. At that point, the integral is built up to maintain constant velocity with the moving waypoint, which will cause the vehicle to overshoot once this excess integral value is no longer needed. The waypoint follower is used for mapping the (x,y)-position, altitude, and heading.

7.2 Trajectory Following

The trajectory follower selects a reference point on the desired trajectory, at a predefined distance L_1 from the vehicle. Based on the angle η between the vehicle velocity vector and a line from the vehicle to the reference point, a lateral acceleration command is calculated as illustrated in Figure 3. The lateral acceleration command a_{cmd} is calculated as

$$a_{cmd} = 2\frac{V^2}{L_1}\sin\eta. \tag{34}$$

It can be noted that the trajectory follower will always try to move the vehicle in the direction of the L_1 line segment, hence pulling the vehicle towards the desired trajectory. It is also readily shown that the trajectory follower produces a lateral acceleration command equal to the centripetal

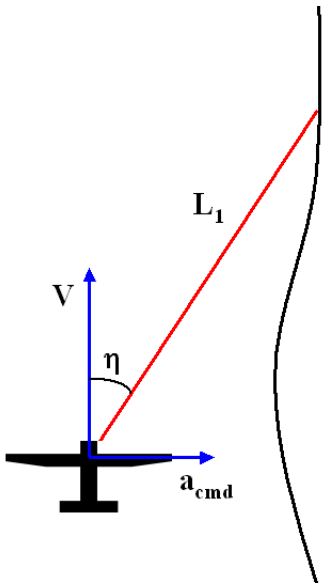


Figure 3: Diagram for trajectory follower

acceleration necessary to maintain a flight path on a circular segment by noting that

$$L_1 = 2R \sin \eta. \tag{35}$$

This also shows an inherent limit in the choices of L_1, since the minimum radius of a circle that can be followed by the trajectory follower is $R_{min} = L_1/2$. This is also made apparent by realizing that with the assumption that the vehicle is on the desired trajectory, no reference point on a circle can be selected if the reference distance L_1 is larger than the diameter of the circle that the vehicle is trying to follow.

Assuming that the above limitation on L_1 is not exceeded, the choice of L_1 is arbitrary. However, due to the manner in which the trajectory follower works, a small L_1 effectively makes the controller more aggressive, whereas a larger L_1 gives a smoother vehicle response.

7.3 Simple Trajectory Generation

It is apparent that a trajectory follower needs a predefined trajectory to follow. In order to provide a simple way of generating trajectories, a simple trajectory generator which connects straight lines and circle arcs was designed.

To better understand the workings of the trajectory generator, let us first review the equations of circles and lines in two dimensions. The equation of a circle can be written using three parameters on the form

$$\big(x-a\big)^2+\big(y-b\big)^2=r^2, \tag{36}$$

where (a,b) is the location of the center of the circle and r is the radius of the circle. Likewise, the equation for a line can be written as

$$x\cos\theta+y\sin\theta-p \;\;=\;\; 0, \tag{37}$$

where p is the shortest distance from the line to the origin, and θ is the angle of the line, where $\theta=0$ indicates a vertical line. Since a distance is never negative, $\theta=0$ also indicates that $x\geq 0$, where the vertical position is determined by the value of p.

The trajectory generator takes an even number of waypoints as inputs. If there are only two waypoints given, a straight line is formed between the two points. If four or more waypoints are available, the first and second waypoints are connected by a straight line, second and third by a circle arc, third and fourth with a straight line, and so on. Due to this trajectory structure, the trajectory generation process can be performed for the first four points, and then performed in the same manner for points three through six, five through eight, etc. A set of four waypoints is used to demonstrate the generation process.

The generation process is initiated by the generation of three straight lines, connecting points one and two, two and three, and three and four. A set of example waypoints can be seen in Figure 4. The lines are generated as

$$x\cos\theta+y\sin\theta-p \;\;=\;\; 0,$$

where

$$\begin{aligned} \theta_k &= \arctan \frac{x_k - x_{k+1}}{y_{k+1} - y_k}, \\ p_k &= x_k \cos\theta + y_k \sin\theta, \end{aligned}$$

where the index k indicates waypoint k or parameter for a segment starting from waypoint k. If $p < 0$ the equation is negated to give a positive p.

If the angles between segments one and two and segments two and three are not equal, either point two or point three are moved along lines one or three to make both angles equal. This movement is always an extension of the trajectory to ensure that all initial waypoints are still on the trajectory.

By rotating and moving the frame of reference, such that point one is at the origin, and point two is on the positive y-axis, the centerpoint of the circle arc that will connect points two and three will be located at point A in Figure 4. The circle parameters are found as (refer to figure for an explanation of the variables)

$$f = \sqrt{\left(x_{2'} - x_3\right)^2 + \left(y_{2'} - y_3\right)^2}, \tag{38}$$

$$\alpha = \arctan \frac{y_3 - y_{2'}}{x_{2'} - x_3}, \tag{39}$$

$$a_{circle} = \frac{f}{2\cos\alpha}, \tag{40}$$

$$b_{circle} = y_{2'}, \tag{41}$$

$$r_{circle} = a_{circle}. \tag{42}$$

The true circle parameters are found by retrieving the original frame of reference. The new waypoints, along with the parameters describing the lines and circles, are now passed on to the trajectory follower.

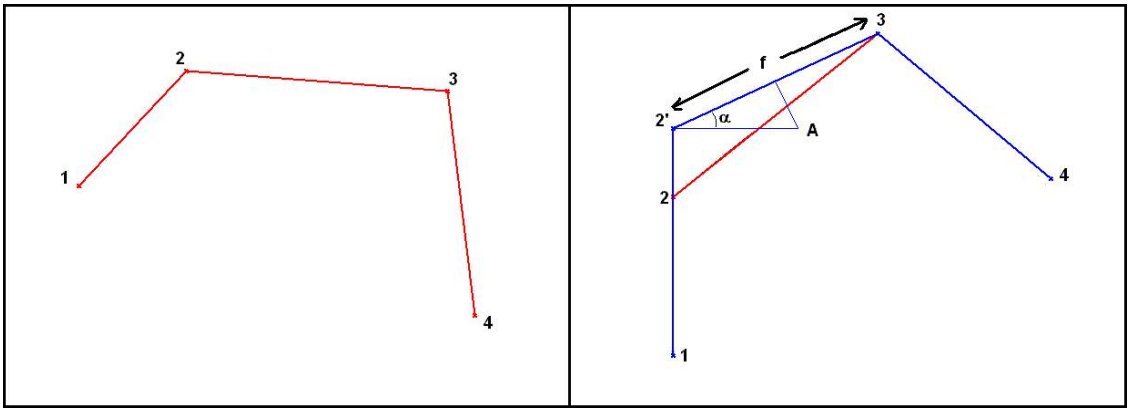


Figure 4: Left: Original waypoints connected by lines. Right: Adjusted waypoints overlapped on original waypoints. Frame of reference has been moved such that waypoint 1 is at the origin and rotated to make the first segment vertical.

7.4 Determination of η

The trajectory follower has five separate ways of determining trajectory intercepts: startup waypoint following, close line following, distant line following, close arc following, and distant arc following.

As the code starts up, the vehicle heads along a straight line running from the center of the vehicle to the initial starting point. The initial straight line following to the first waypoint ensures that the vehicle will head straight for the trajectory if it is initially offset from the trajectory. The initial line is modified at every time sample to always run from the center of the vehicle. Once the vehicle reaches within the predefined distance L_1 of the first waypoint, it assumes following of the first segment. The update of waypoints is always performed as soon as the vehicle reaches within L_1 of the current segment's end waypoint.

While following lines, it is first determined whether the vehicle is within the distance L_1 of the trajectory line or not. If this is not the case, the vehicle heads along a course perpendicular to the line in order to reach the line in as swift a manner as possible. If the line is close enough to be followed, an intercept is determined.

The reference frame is initially moved and rotated such that the vehicle is located at the origin and the trajectory line the vehicle is trying to follow is vertical, with North being the direction of

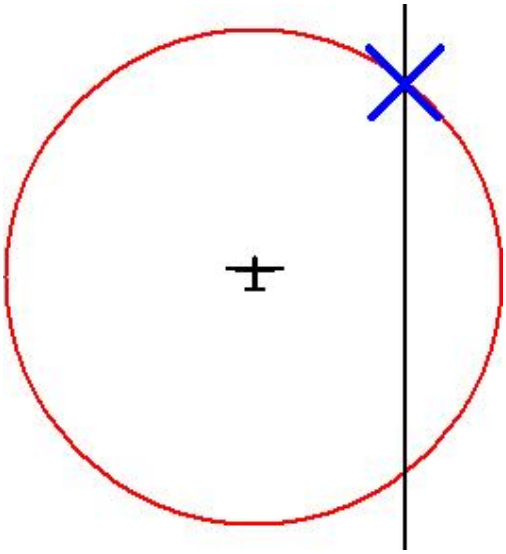


Figure 5: Line interception

the next waypoint. The intercept can now be determined as the interception point between a circle of radius L_1 extending from the vehicle and the line. The intercept will yield two points, of which the one of interest is the one with a positive y-value. See Figure 5 for details.

$$x^2 + y^2 = L_1^2 \qquad \text{Circle equation} \tag{43}$$

$$x - p = 0 \qquad \text{Line equation with } \theta = 0 \tag{44}$$

$$y = \sqrt{L_1^2 - p^2} \tag{45}$$

$$(x, y) = (p, \sqrt{L_1^2 - p^2}) \tag{46}$$

The value of η is found as the difference between the vehicle heading and the direction of the line extending from the center of the vehicle to the intersection point.

While following circle arcs, the trajectory follower does the same check for distance to the arc as was previously done for distance to lines. If the vehicle is too far away to follow the arc, it heads straight for it. The distance is checked as

$$r + L_1 > \sqrt{\left(x_{vehicle} - a\right)^2 + \left(y_{vehicle} - b\right)^2} > r - L_1. \tag{47}$$

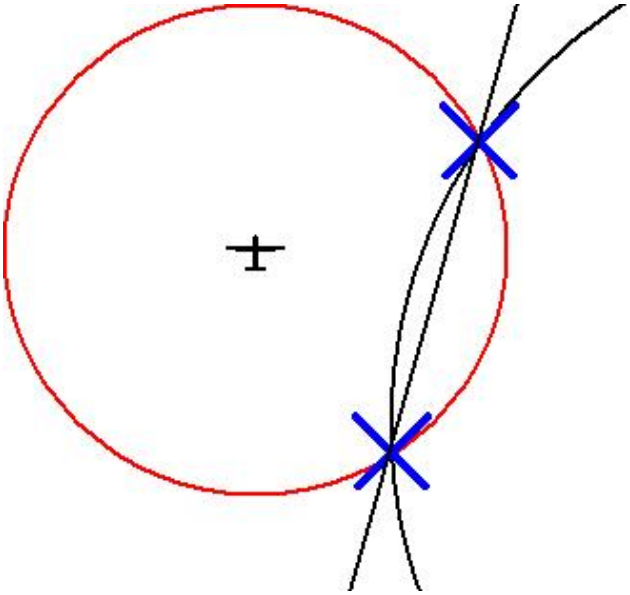


Figure 6: Circle interception

If the inequality is fulfilled, the vehicle is close enough and an intercept is determined.

Initially, the reference frame is moved such that the vehicle is sitting at the origin. The trajectory circle and a circle with radius L_1 centered on the vehicle are compared.

$$\left(x - a'\right)^2 + \left(y - b'\right)^2 = r^2 \tag{48}$$

$$x^2 + y^2 = L_1^2 \tag{49}$$

The variables a' and b' represent the rotated and moved center point of the trajectory circle arc. When these two equations are combined, a line passing through the two intersection points is found.

$$L_1^2 + (a')^2 + (b')^2 - r^2 = 2\left(a'x + b'y\right) \tag{50}$$

The problem is now reduced to a line intersection with a circle. This is again solved in the same manner as the line following problem. See Figure 6 for an illustration.

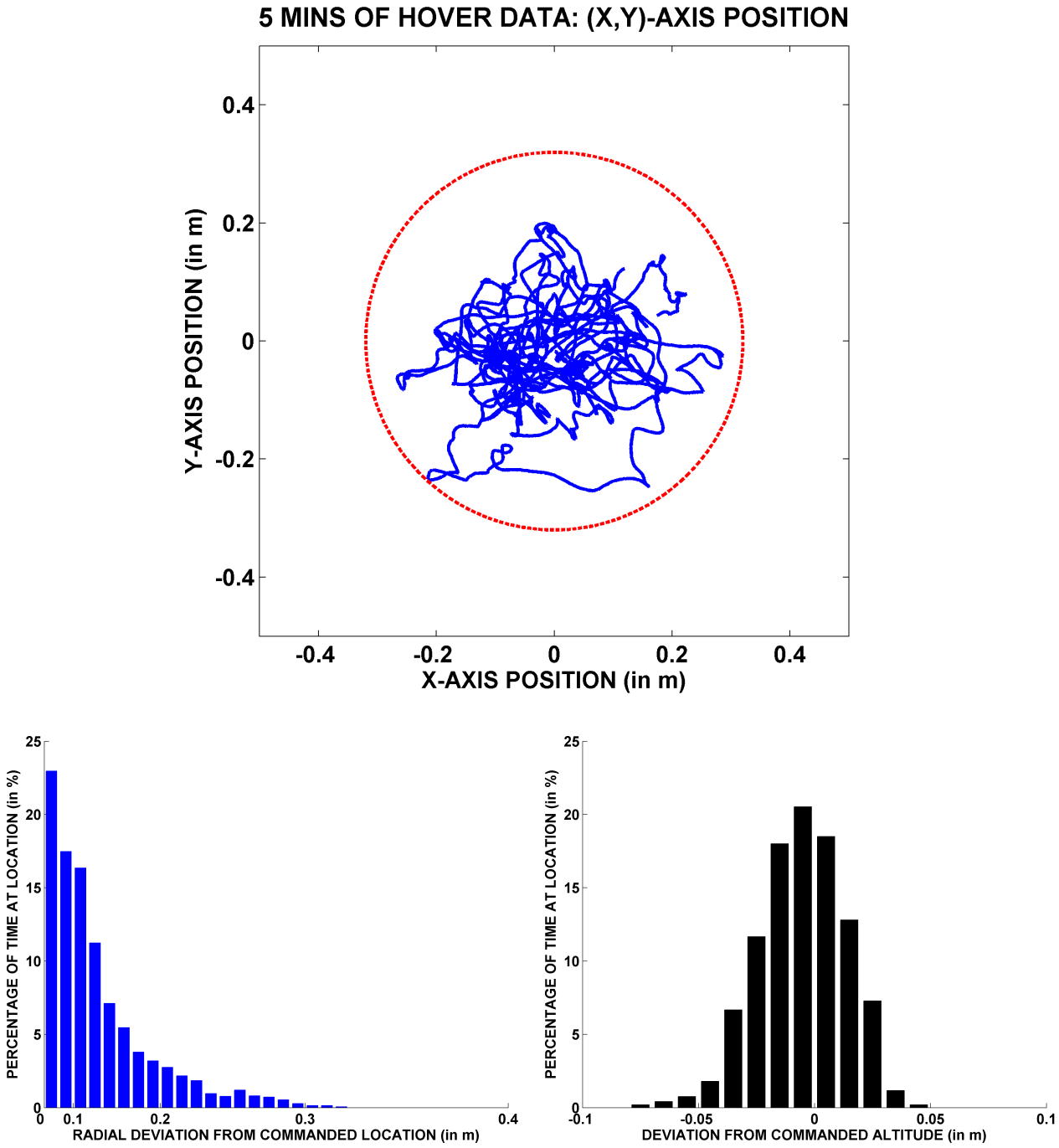


Figure 7: Airplane hover experiment – Airplane commanded to hover at $(x, y, z) = (0, 0, 1.5)$ m for 5 min: x–y plot of vehicle position (top), histograms showing the radial distance error for the vehicle from (x,y) = (0,0) m and altitude error from $z = 1.5$ m (bottom)

Figure 8: Autonomous landing in hover mode – perching on the portable vertical landing platform apparatus

8 Results

Numerous hover and flight tests were performed using the Blade 3D and Ikarus YAK 54 Shock Flyer foam aircraft in the RAVEN at MIT [7]. First, typical results are shown in Figure 7 in which the vehicle was commanded to hold its position at $(x_E, y_E, z_E) = (0, 0, 1.5)$ m for five minutes. Figure 7 shows three plots, including a plot of the vehicle x–y location while it maintained its position and attitude. The dashed red circle in the picture has a radius of 0.32 m. This circle would correspond to the allowable movement in a 1.50 m wide corridor or equivalent, based on the vehicle's wingspan. As shown in Figure 7, the vehicle maintained its position inside this circle during the five-minute test period. The remaining plots give histograms of the vehicle's x-y and z positions. The first histogram shows the radial distance error for the vehicle from $(x, y) = (0, 0)$ m, while the second histogram shows the vehicle's altitude error from $z = 1.5$ m. These plots confirm that the vehicle was within a 20 cm circle for over 87% of the 5 min flight. These plots also show that the vehicle precisely maintained its altitude (staying between 1.4 to 1.6 m) during the entire hover test.

Next, since the airplanes are autonomous from take-off to landing, the aircraft takes off and lands in a perching configuration. Figures 8 and 9 show photographed sequences of the airplanes "perching" on a portable vertical landing platform (in Figure 8) and landing on the fixed cement pole located in the middle of the RAVEN flight space (in Figure 9) from hover mode.

Test flight results for five consecutive take-off and landing sequences where the airplane was commanded to land in the hover configuration onto the fixed landing platform on the cement pole are shown in Figures 10 and 11. In this test, the landing pad on the cement pole is located at (x,y,z) = (0,2.67,1.03) m in the room and the vehicle was commanded to take off, hover for about

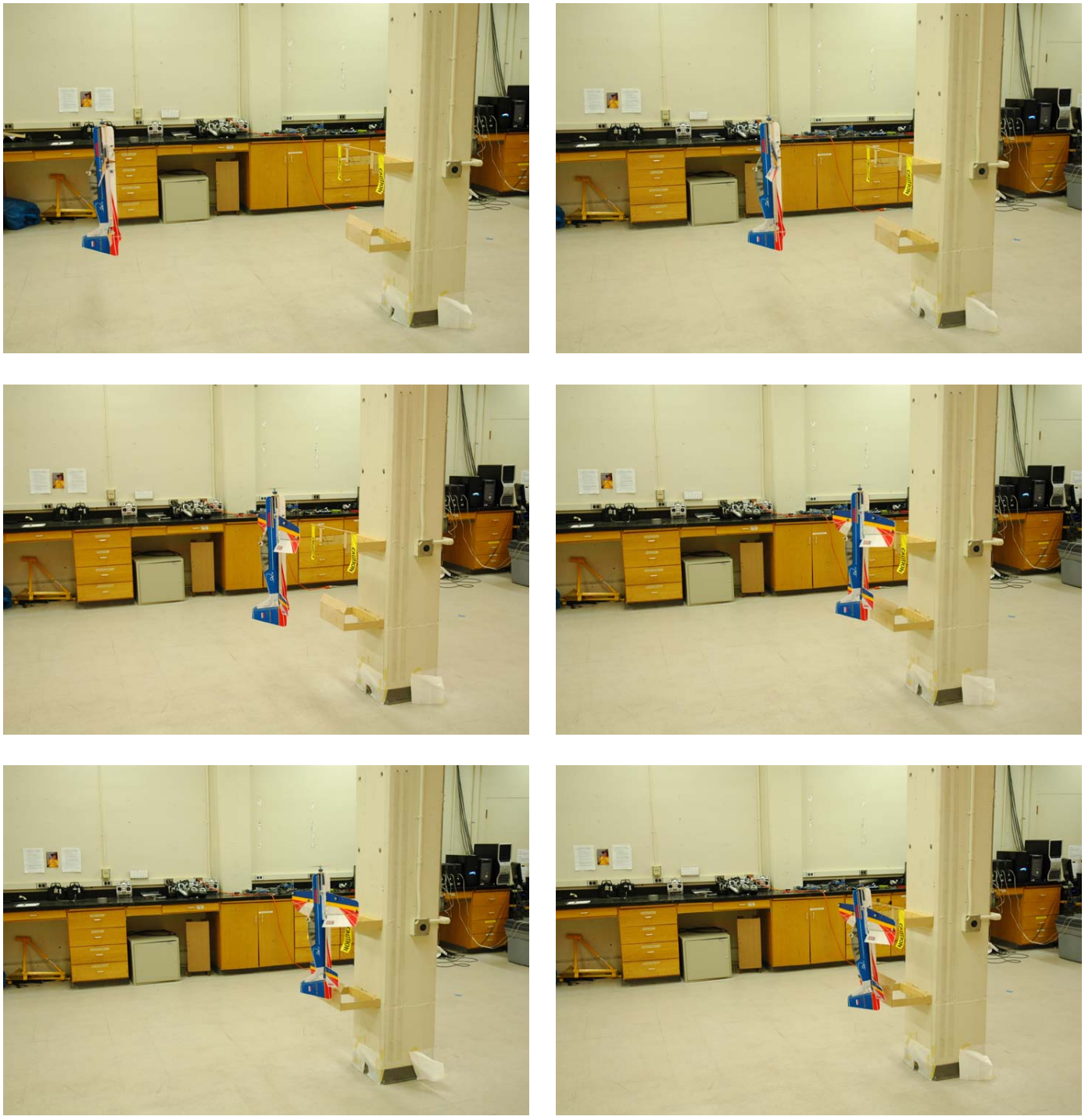


Figure 9: Autonomous landing in hover mode – perching on the fixed landing platform on a cement pole

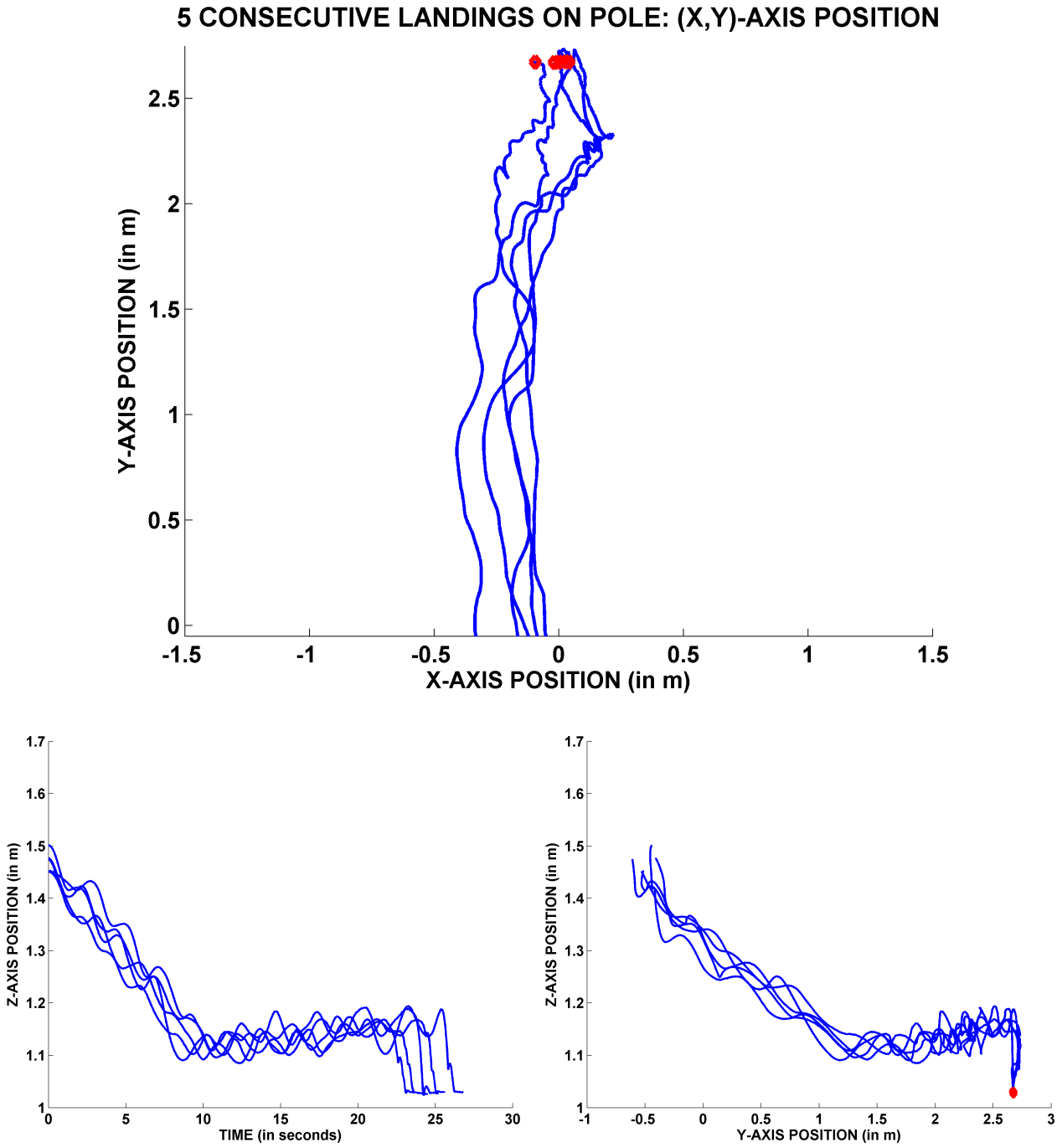


Figure 10: Five consecutive landings – Airplane commanded to perch on the cement pole landing apparatus at $(x, y, z) = (0, 2.67, 1.03)$ m: x–y plot of vehicle position (top), z-axis position vs time plot and y–z plot of vehicle position (bottom)

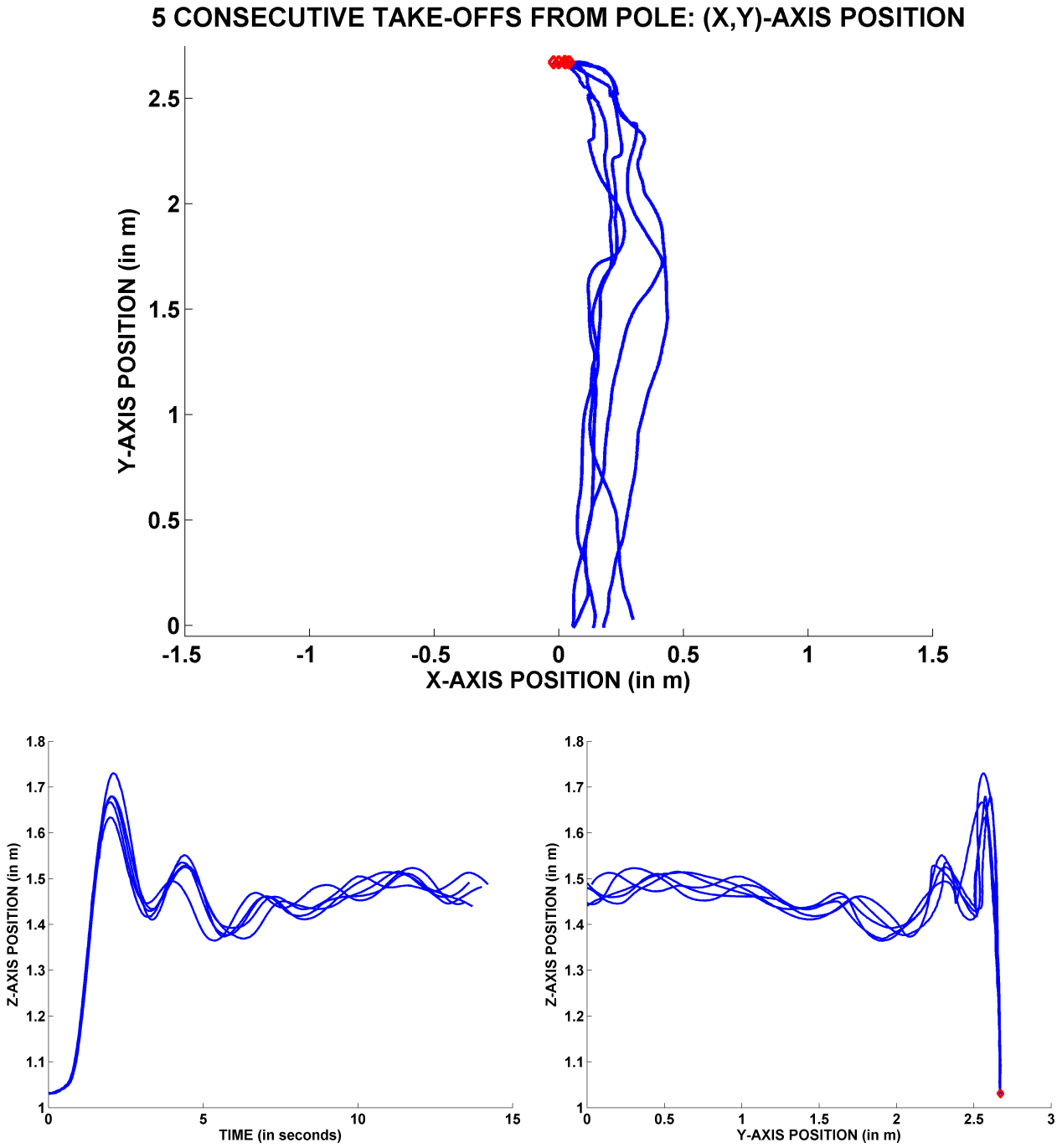
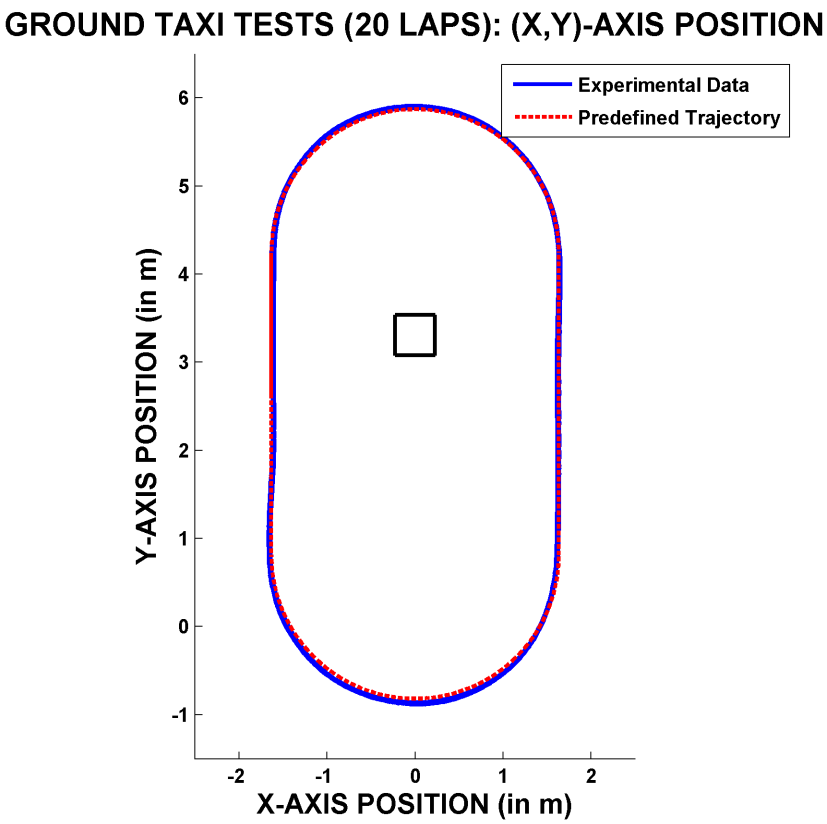


Figure 11: Five consecutive take-offs – Airplane commanded to perch on the cement pole landing apparatus at $(x, y, z) = (0, 2.67, 1.03)$ m: x–y plot of vehicle position (top), z-axis position vs time plot and y–z plot of vehicle position (bottom)

15 s and then attempt another landing. In these plots, the vehicle begins its approach near (x,y) = (0,0,1.5) m and moves toward the landing platform in the hover mode. As it approaches the landing location, the vehicle lowers itself to move just above the landing site. Then, once the vehicle detects that it is in the proper location with a correct heading, the vehicle descends into the landing location to await the next take-off command. Once the take-off command is provided, the throttle is accelerated to pull the vehicle directly above the landing platform before moving the vehicle away from the pole.

In both Figures 10 and 11 the red diamonds at the end of each trajectory show the landing/take-off location of the vehicle. Throughout the entire sequence, the fully-autonomous vehicle was commanded to take off from the location where it had landed. Note that all of these take-off locations are clustered within 0.07 m of each other. Likewise all of the landing locations are within 0.15 m of each other, mainly because the fifth and final landing of the sequence landed the vehicle on the left-hand part of the landing apparatus. Thus, each of the first four landings are clustered within 0.07 m of each other. In addition, note that that the vehicle's descent into the platform over each landing and take-off is consistent. During take-off, there is an overshoot as the vehicle pulls itself out of the platform. This overshoot is the result of a safety feature designed into the vehicle's take-off command sequence. As the vehicle is commanded to take-off, the throttle command is quickly increased to the throttle setting used in the previous landing. Since the integrator gain on the vehicle's vertical take-off altitude controller is small and the commanded hover location after take-off is located just above the landing platform, it takes the vehicle a few seconds to clear the landing platform. During this time, the controller runs a risk of ramping up the elevator integrator while it tries to move the airplane over to the commanded hover position. To prevent this command from turning the aircraft into the landing platform, the throttle command during take-off is increased slightly above the necessary take-off command value needed to quickly move the vehicle into the air and clear of the landing platform. In the case where a previous landing reference throttle setting is not available, a predetermined throttle command value (based on empirical data) is used. A video of two consecutive take-off-hover-landing sequences is available online at http://aerobatics.mit.edu.



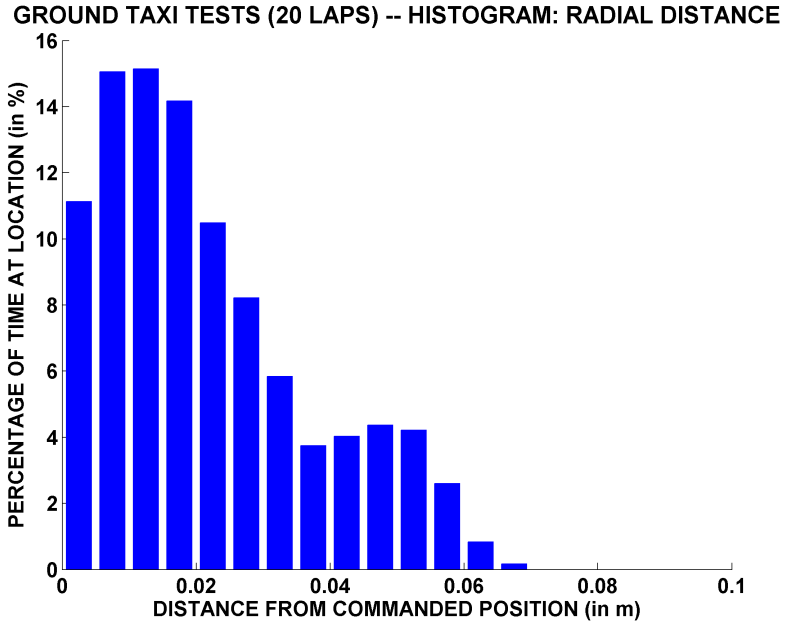


Figure 12: Ground taxi test for 20 consecutive laps – Airplane commanded to follow an oval trajectory in the RAVEN around the cement pole: x–y plot of vehicle position (top), histogram showing the radial distance error from the vehicle's trajectory (bottom)

Next, before attempting level flight, a ground taxi test was used to ensure the integrity of the vehicle's capabilities in following trajectory commands as calculated using the vehicle's trajectory follower. As shown in Figure 12, the vehicle was commanded to follow an oval trajectory (shown as a dashed red line) around the floor for 20 consecutive laps. The histogram showing the radial distance error from the vehicle's commanded trajectory demonstrates that the vehicle's trajectory follower was able to accurately command the vehicle to hold the trajectory and was ready for level flight testing.

Before attempting level flight tests with the aircraft, the vehicle's transition logic was also demonstrated. Figure 13 shows data from one of these transition tests. In this test, the vehicle autonomously moved from the hover take-off location near the pole to the northeastern area of the flight space, and the test began from the red diamond and flew south as shown in the x–y vehicle position plot in Figure 13. First, note that this test occurred in under 6 s. In fact, each vehicle transition takes approximately 1.5 s to complete. As the vehicle transitions to level flight mode the vehicle loses altitude while picking up speed. In this test after the vehicle achieves the level flight mode (i.e., the vehicle's calculated pitch angle is less than 30 degrees), the vehicle is immediately commanded back to hover mode. Notice that as the vehicle begins to pull out of the level flight state, the vehicle gains speed until the system reduces the throttle control as the vehicle gains altitude. In addition, after the vehicle exceeds the desired hover altitude of 1.5 m, the vehicle's velocity reduces quickly and the vehicle's calculated pitch increases dramatically. Note also that the pitch angle calculation was made using the quaternion measurement. Here, the pitch angle measure was unwrapped at 90 degrees to show that the vehicle actually pitches past the 90 degree point as the vehicle tries to reduce speed and altitude quickly to regain the hover control.

Finally, a hover to level flight (3 laps) to hover test was demonstrated with the aircraft in the RAVEN as shown in Figure 14. Figure 15 shows data from two of the three lap tests. Once again, the vehicles autonomously moved from the hover take-off location near the pole to the northeastern area of the flight space shown in Figure 15. From this location, the vehicle transitioned to level flight mode, flew three laps around the pole (as shown in the x-y location plot in Figure 15) and then transitioned back to the hover state before attempting a landing. During the level flight

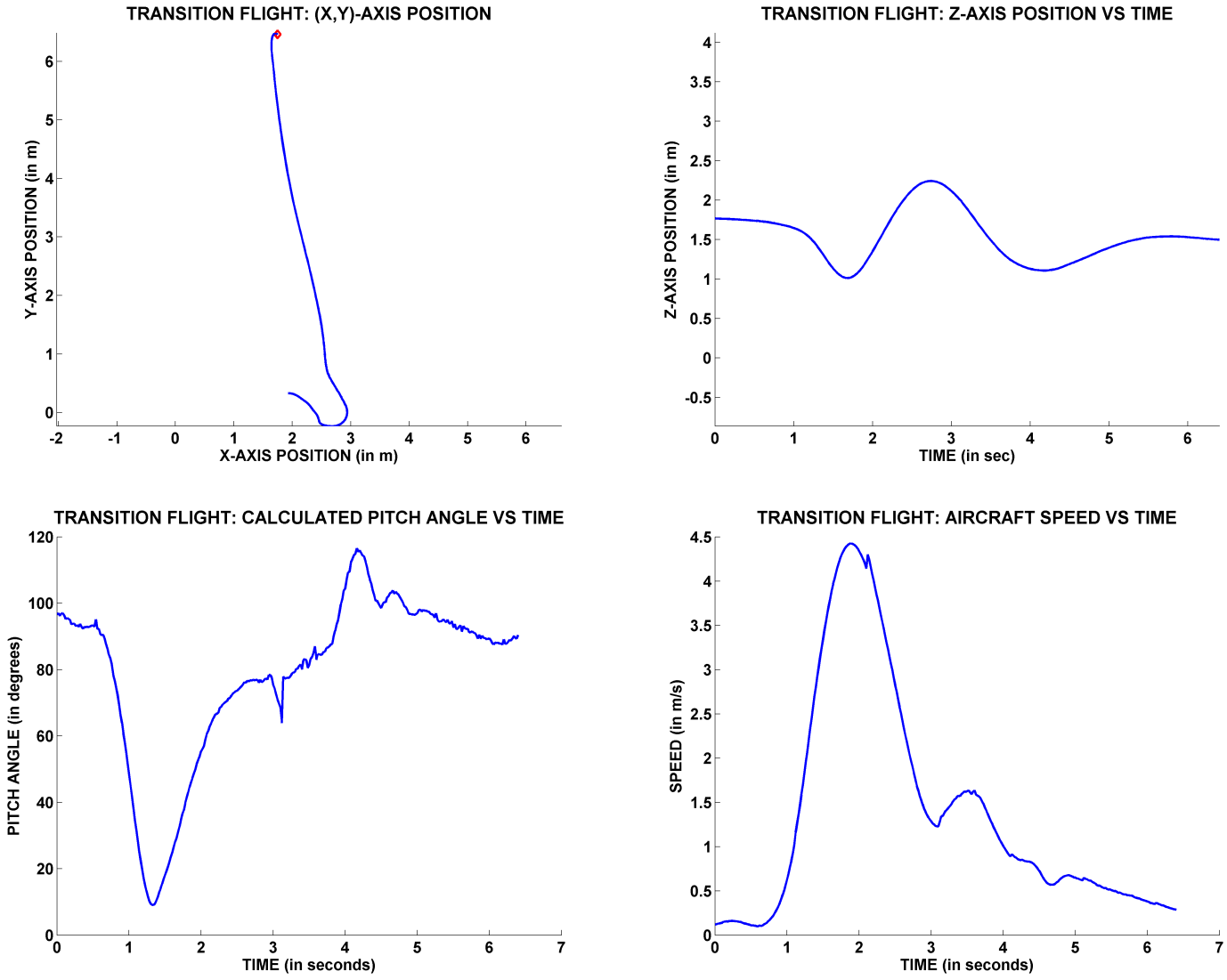


Figure 13: Transition test – Starting from a hover, transition to and from level flight mode: x–y plot of vehicle position (top left), z-axis position vs time plot (top right), calculated pitch angle (via the quaternion measurement) vs time plot (bottom left) and vehicle speed vs time plot (bottom right)

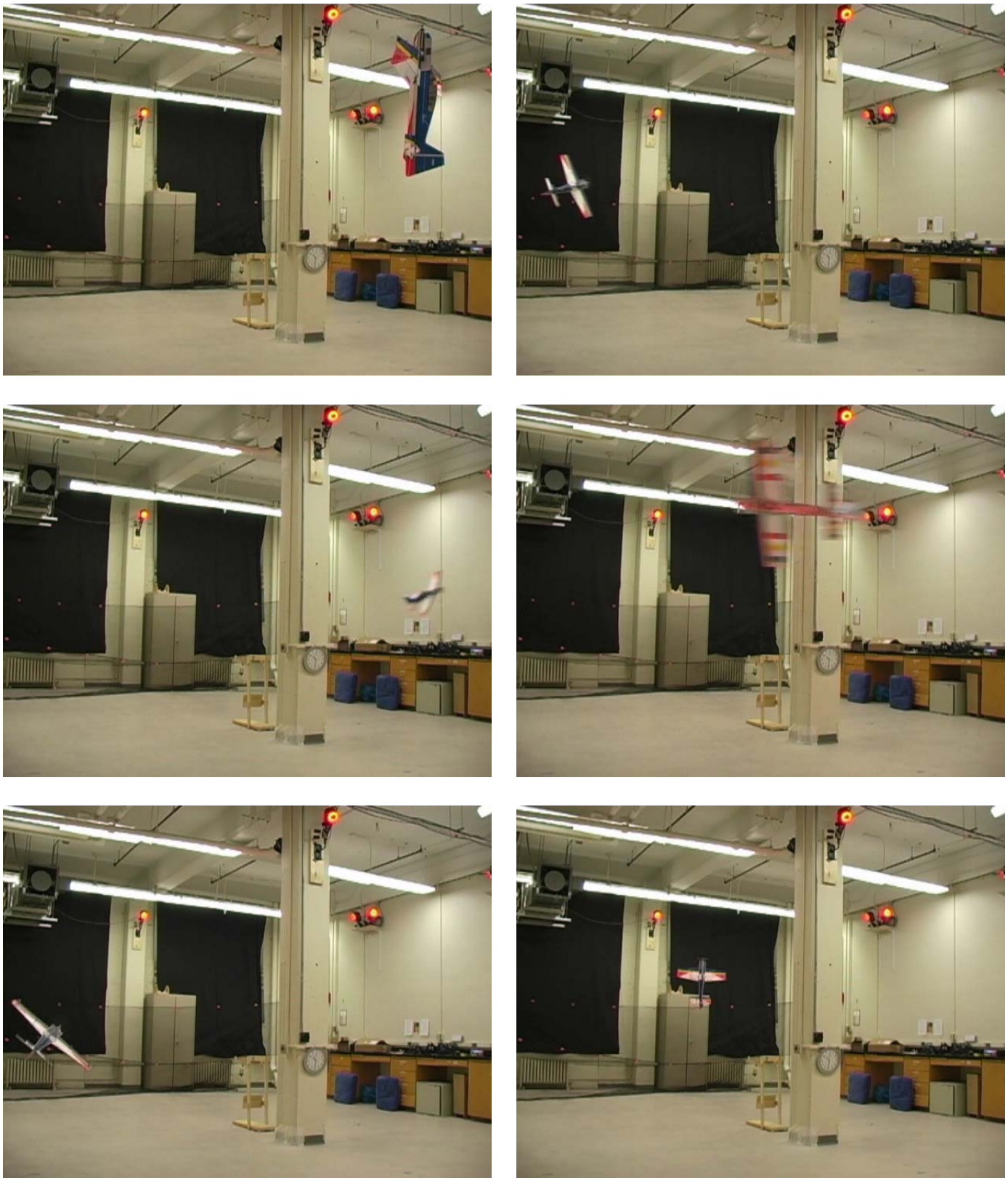


Figure 14: Pictures from an autonomous hover to level flight (3 lap) to hover mode test

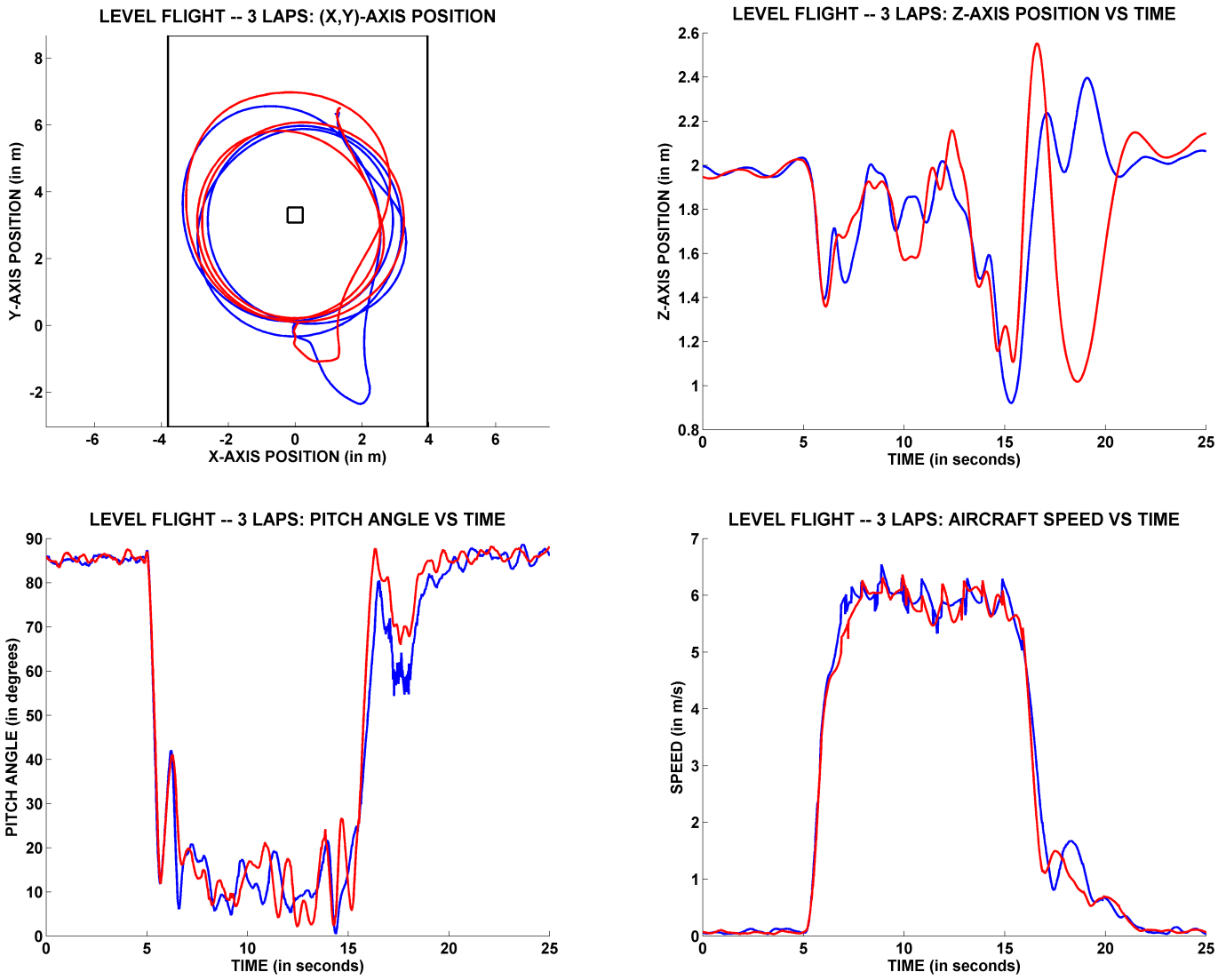


Figure 15: Level flight test in constrained environment – Starting from a hover, the vehicle transitions to level flight mode, flies three laps around the room and then transitions back to level flight mode: x–y plot of vehicle position (top left), z-axis position vs time plot (top right), measured pitch angle vs time plot (bottom left) and vehicle speed vs time plot (bottom right)

in the RAVEN, the vehicle maintains a speed of approximately 6 m/s during each of the three laps. In addition, the vehicle is flying a circular route with a radius of about 2 m. Therefore, the three laps take about 10 s from the time the vehicle enters level flight mode to the time the vehicle transitions back to the hover state. A flight from one of these tests can be found at http://aerobatics.mit.edu.

9 Conclusion

This paper presents both vehicle models and test flight results for an autonomous fixed-wing airplane that is designed to take-off to a hover, transition to and from traditional fixed-wing level flight modes, and perch on a vertical landing platform in a highly space constrained environment. These tests are the first tests shown in the literature of autonomous aggressive indoor flight using a fixed-wing R/C aircraft. Note that results for this COTS fixed-wing model airplane are shown for both hover and level flight modes as demonstrated in the MIT Real-time Autonomous Vehicle indoor test ENvironment (RAVEN). Future work will expand on these test to investigate landing on moving objects in a hover and performing other aerobatic maneuvers in the indoor test environment.

Acknowledgements

The authors would like to thank Spencer Ahrens, Brett Bethke, Luc Brunet, Eli Cohen, Daniel Dale, Brandon Luders, and Adam Woodworth for their assistance in the project. This research has been supported by the Boeing Company, Phantom Works, Seattle and AFOSR grant FA9550-04-1-0458.

References

[1] B. L. Stevens and F. L. Lewis. *Aircraft Control and Simulation, Second Edition.* J. W. Wiley and Sons, 2003.

[2] D. Biermann and R. N. Conway. Propeller Charts for the Determination of the Rotational Speed for the Maximum Ratio of the Propeller Efficiency to the Specific Fuel Consumption. Technical report, National Advisory Committee on Aeronautics, 1942.

[3] E. N. Johnson, M. A. Turbe, A. D. Wu, S. K. Kannan, and J. C. Neidhoefer. Flight Test Results of Autonomous Fixed-Wing UAV Transitions to and from Stationary Hover. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference Exhibit*, Monterey, CO, August 2006.

[4] E. Frazzoli. *Robust Hybrid Control for Autonomous Vehicle Motion Planning*. Department of aeronautics and astronautics, Massachusetts Institute of Technology, Cambridge, MA, June 2001.

[5] A. J. Hanson. *Visualizing Quaternions*. Morgan Kaufman Publishers, San Francisco, CA, 2006.

[6] J. B. Kuipers. *Quaternions and Rotation Sequences*. Princeton University Press, Princeton, NJ, 1999.

[7] M. Valenti, B. Bethke, G. Fiore, J. How, and E. Feron. Indoor multi-vehicle flight testbed for fault detection, isolation, and recovery. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, August 2006.

[8] N. Knoebel, S. Osborne, D. Snyder, T. Mclain, R. Beard, and A. Eldredge. Preliminary modeling, control, and trajectory design for miniature autonomous tailsitters. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, August 2006.

[9] R. H. Stone. Control Architecture for a Tail-Sitter Unmanned Air Vehicle. In *Proceedings of the 5th Asian Control Conference*, Melbourne, Australia, July 2004.

[10] S. Park, J. Deyst, and J. P. How. A New Nonlinear Guidance Logic for Trajectory Tracking. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Providence, RI, August 2004.

[11] University of Sydney. T-Wing Aircraft Homepage. Avaiable at http://www.aeromech.usyd.edu.au/uav/twing/, October 2006.

[12] M. Valenti, B. Bethke, D. Dale, A. Frank, J. McGrew, S. Ahrens, J. P. How, and J. Vian. The MIT Indoor Multi-Vehicle Flight Testbed. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA '07)*, Rome, Italy, April 2007. Video Submission.

[13] W. E. Green and P. Y. Oh. A MAV That Flies Like an Airplane and Hovers Like a Helicopter. In *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Monterey, CA, July 2005.

[14] W. F. Chana and J. F. Coleman. World's First VTOL Airplane Convair/Navy XFY-1 Pogo. In *Proceedings of the International Powered Lift Conference*, pages 99–104, Jupiter, FL, November 1996.