Journal of Intelligent & Robotic Systems (2019) 93:85–100
https://doi.org/10.1007/s10846-018-0790-z

Automatic Control for Aerobatic Maneuvering of Agile Fixed-Wing UAVs

Eitan Bulka[1] · **Meyer Nahon**[1]

Received: 22 September 2017 / Accepted: 2 February 2018 / Published online: 13 February 2018
© Springer Science+Business Media B.V., part of Springer Nature 2018

Abstract
The use of unmanned aerial vehicles (UAVs) has become ubiquitous in a broadening range of applications, including many civilian uses. UAVs are typically categorized into two categories: conventional fixed-wing aircraft, which are associated with efficient flight over long distances, and rotor-craft, which are associated with short flights requiring maneuverability. An emerging class of UAVs, agile fixed-wing UAVs, are bridging the gap between fixed-wing and rotor-craft with efficient *and* maneuverable flight capabilities. This article presents a single physics-based controller capable of aerobatic maneuvering of agile fixed-wing UAVs. We first demonstrate autonomous flight in a conventional high-fidelity in-house simulation with this controller, and then implement the algorithm on a Pixhawk micro-controller. A hardware-in-the-loop (HIL) environment is developed and used to further develop autonomous aerobatic flight, followed by outdoor flight tests in windy conditions. Our control system successfully tracks position and orientation times histories to achieve autonomous extreme aerobatic maneuvers including knife-edge, rolling Harrier, hover, and aggressive turnaround.

Keywords Control architectures · Agile Fixed-Wing UAVs · Aerobatics · Unmanned aerial vehicles

1 Introduction

Improving the autonomy of small unmanned aerial vehicles (UAVs) has been a growing topic of research over the past decade due to the many commercial and military applications of these vehicles. A large portion of this research has focused on rotor-craft (helicopter, quadrotor) due to their maneuverability, and conventional fixed-wing aircraft due to their endurance. An emerging class of UAVs, agile fixed-wing UAVs, aims to bridge the gap between rotor-craft and fixed-wing aircraft. Agile aircraft are characterized by high thrust-to-weight ratios (2-3) and large control surfaces capable of large deflections, allowing maneuverability rivaling that of rotor-craft, while maintaining the ability to achieve long endurance in conventional flight.

Pilots have demonstrated the ability of small radio-controlled aircraft to perform extreme aerobatic maneuvers for many years, but it is only recently that researchers have attempted to automate these extreme flight modes. Designing automatic control systems to perform these extreme maneuvers is challenging due to the limited control authority in certain flight regimes (e.g. in low-speed flight) and the under-actuated nature of a conventional fixed-wing aircraft layout.

Some researchers have developed control systems capable of performing aerobatic maneuvers on small RC helicopters. In [1], control strategies have been developed based on mimicking a human pilot's control inputs. This idea is further explored in [2], where the authors apply reinforcement learning techniques to achieve automatic aerobatic helicopter flight. Some research has also focused on developing controllers for aerobatic maneuvering of quadrotors. In [3], aerobatic quadrotor flight is achieved by decomposing maneuvers into discrete phases, where each phase has a local controller. The controllers consist of an outer-loop PID position controller, and an inner-loop PD attitude tracker.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10846-018-0790-z) contains supplementary material, which is available to authorized users.

✉ Eitan Bulka
eitan.bulka@mail.mcgill.ca

Meyer Nahon
meyer.nahon@mcgill.ca

[1] Department of Mechanical Engineering, McGill University, Montreal, QC, H3A OC3, Canada

In contrast to rotor-craft, there has been little research developing control systems for aerobatic maneuvering of agile fixed-wing UAVs. The majority of that effort has been towards developing controllers for level flight and hovering conditions and transitioning between these conditions. A simple approach using a PD control law is presented in [4] while more sophisticated approaches have also been proposed: a nonlinear Lyapunov backstepping controller in [5] and an LQR-Trees controller in [6], including experimental demonstration.

The majority of autonomous aerobatics for fixed-wing UAVs has only been demonstrated in simulation. In [7], a multimodal flight control scheme is presented which is capable of performing many aerobatic maneuvers in simulation. Each maneuver is comprised of a number of flight modes, where each mode is locally controlled by a dynamic sliding mode control law. In [8], a control system is developed where a PD control law is used to track a time varying pitch and roll trajectory. This technique is used to perform a few aerobatic maneuvers in a simulation environment and one maneuver in experiment.

In [9], a control system is developed which can perform many aerobatic maneuvers along a specified flight path outdoors. The control system uses a non-linear path following guidance law in an outer-loop to create an acceleration command. The elevator and rudder are used to track the acceleration command using a PI control law. The roll can be selected independently of the flight path as this component is decoupled from tracking the specified path, which enables for knife-edge and rolling harrier flight. In [10], an agile fixed-wing aircraft automatically flies between objects narrower than its wing span using the knife-edge maneuver in an indoor motion capture environment. An open-loop trajectory is formed using a direct collocation method, which is tracked using a time-varying linear quadratic regulator.

The control system designs in previous works are either tailored to a single maneuver, or contain a library of maneuver specific controllers and control gains. One exception would be in [9], however this control system is limited to tracking a flight path, which excludes the ability to hover or aggressively turnaround. The novel control system developed in this work is capable of tracking virtually any feasible trajectory with a single controller and a single set of gains. This approach is advantageous as it simplifies the tuning process, and enables robust transitions between maneuvers, as the control laws are not changing discontinuously. Using this control system, we demonstrate successful autonomous aerobatics in a conventional simulation, in a hard-ware-in-the-loop (HIL) simulation, and in outdoor flight tests using our agile fixed-wing aircraft shown in Fig. 1.

Fig. 1 McFoamy aircraft

2 Controller Design

An aircraft control system must adjust the aircraft's pose using its available control inputs (aileron, elevator, rudder, and thrust). This is a challenging control problem because the system is under-actuated, and the actuators' effectiveness varies with the aircraft speed. The purpose of this work is to develop a *single* controller capable of tracking any feasible trajectory while maintaining the ability to recover from large deviations from the reference trajectory. These objectives are achieved by developing a nonlinear control system that is based on the physics of the aircraft, allowing simple control laws to achieve precise tracking of highly non-linear dynamics. We avoid any plant linearization, to ensure that the controller will remain effective throughout the entire flight envelope of the agile aircraft.

The control surfaces of a fixed-wing aircraft can control orientation but cannot directly control position. However, the orientation of an aircraft *can* control position. This leads to a modular control architecture where the control surfaces are used to track orientation and orientation is used to track position. The controller structure assumes the position, $\mathbf{p}$, the orientation expressed as a unit quaternion, $\mathbf{q}$, and the velocity, $\mathbf{v}$, can be obtained from on-board sensors and a state estimator.

In the present context, a maneuver may consist of equilibrium or non-equilibrium flight, some examples of which are level flight, hover, aggressive turnaround, knife-edge, and rolling Harrier. The control system is provided with a time history of reference pose and longitudinal speed for each maneuver, by a 'maneuver generator'. These reference trajectories should ideally be kinematically and dynamically feasible, but this is not essential, as the controller has been found to work well, even when this is not the case.

Figure 2 shows the high-level control logic. The reference position, $\mathbf{p}_{\mathbf{ref}}$, and reference orientation quaternion, $\mathbf{q}_{\mathbf{ref}}$, are generated by the maneuver generator, and fed to the

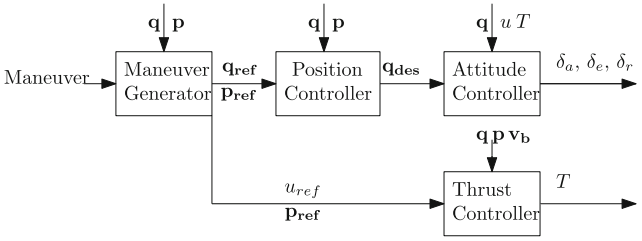

Fig. 2 Control architecture

outer-loop position controller. The position controller compares the reference position to the state estimate of position, $\mathbf{p}$, to obtain position errors, $\Delta\mathbf{p}$. The reference attitude is then modified accordingly to reduce errors in position, thus generating the desired attitude quaternion, $\mathbf{q_{des}}$. In turn, the attitude controller then determines appropriate control surface deflections to track the desired attitude. Thrust control is performed separately, and its goal is to track altitude, h, and longitudinal speed, u. The modularity of the controller is advantageous as it allows the potential to port the control logic to other unmanned aerial systems, and also simplifies the debugging and tuning processes.

2.1 Attitude Controller

The aim of the quaternion-based attitude controller is to track the desired orientation generated by the position controller. Tracking is achieved in the following three step process:

1. determine the angular error about the body axes
2. generate desired moments to exert on the aircraft
3. generate control surfaces deflections to produce the desired moments

2.1.1 Angular Errors

Obtaining angular errors in body frame coordinates simplifies the control logic because each control surface exerts a moment principally about one body axis. This approach is based on [4] which addresses attitude control of a hovering agile aircraft. It originates from the notion that an error quaternion, $\Delta\mathbf{q}$, can be used to describe the rotation from the aircraft attitude, $\mathbf{q}$, to the desired attitude, $\mathbf{q_{des}}$, and so this error quaternion must satisfy:

$$\mathbf{q} \otimes \Delta\mathbf{q} = \mathbf{q_{des}} \tag{1}$$

By left multiplying by the quaternion inverse, which is just the conjugate for a unit quaternion, $\mathbf{q}^*$, we can solve for the error quaternion:

$$\Delta\mathbf{q} = \mathbf{q}^* \otimes \mathbf{q_{des}} \tag{2}$$

to ensure the angular errors remain less than 180°, in Eq. 2 we replace $\mathbf{q_{des}}$ with $-\mathbf{q_{des}}$ if $||\mathbf{q}+\mathbf{q_{des}}|| < ||\mathbf{q}-\mathbf{q_{des}}||$. The equations remain correct since a quaternion and its negative describe the same attitude.

From the definition of a quaternion, the error quaternion can be represented as a rotation of magnitude, Θ, about a unit vector, $\mathbf{u}$. Hence:

$$\begin{aligned}\Delta\mathbf{q} &= [\Delta q_0\ \Delta q_1\ \Delta q_2\ \Delta q_3] \\ &= [\cos\frac{\Theta}{2}\quad u_x \sin\frac{\Theta}{2}\quad u_y \sin\frac{\Theta}{2}\quad u_z \sin\frac{\Theta}{2}]\end{aligned} \tag{3}$$

Now we aim to find the angular error about each of the body axes (E_x, E_y & E_z), which would be the magnitude of rotation along the component of each of these axes:

$$E_x = \Theta u_x,\ E_y = \Theta u_y,\ E_z = \Theta u_z \tag{4}$$

By equating the first component on either side of Eq. 3 we can obtain the magnitude of rotation from:

$$\Theta = 2\cos^{-1}(\Delta q_0) \tag{5}$$

We can also equate the other components on the two sides of Eq. 3 to solve for the axis of rotation:

$$u_x = \frac{\Delta q_1}{\sin\frac{\Theta}{2}},\ u_y = \frac{\Delta q_2}{\sin\frac{\Theta}{2}},\ u_z = \frac{\Delta q_3}{\sin\frac{\Theta}{2}} \tag{6}$$

Using a trigonometric identity, Eq. 5, and the knowledge that we are using a unit quaternion, we can solve for $\sin\frac{\Theta}{2}$:

$$\sin\frac{\Theta}{2} = \sqrt{1-\cos^2\frac{\Theta}{2}} = \sqrt{1-\Delta q_0^2} = \sqrt{\Delta q_1^2+\Delta q_2^2+\Delta q_3^2} \tag{7}$$

Finally, we can substitute Eqs. 7 into 6, and substitute Eqs. 5 & 6 into Eq. 4 to obtain the angular errors about the body frame axes. Combining these equations, we find that the angular errors can be computed [4] as:

$$E_x = 2\cos^{-1}(\Delta q_0)\frac{\Delta q_1}{||\Delta\mathbf{q_{1:3}}||} \tag{8}$$

$$E_y = 2\cos^{-1}(\Delta q_0)\frac{\Delta q_2}{||\Delta\mathbf{q_{1:3}}||} \tag{9}$$

$$E_z = 2\cos^{-1}(\Delta q_0)\frac{\Delta q_3}{||\Delta\mathbf{q_{1:3}}||} \tag{10}$$

2.1.2 Desired Moments

The angular errors can be directly mapped to control surface deflections, as done in [4]. A drawback to this approach is that the effect of deflecting a control surface is not only dependent on the magnitude of the deflection, but is also dependent of the speed of the air flowing over the surface. Air flowing over a deflected surface changes direction, causing a change in momentum of the air, which ultimately exerts a force on the aircraft. For the same magnitude of deflection, faster flowing air undergoes a greater change in momentum, producing a larger force. Forces generated

by control surfaces that are far from the aircraft center of mass will ultimately lead to large moments on the aircraft. To account for this phenomenon, an intermediate step is introduced here. The angular errors calculated in the previous section are mapped to desired moments, and these are combined with the airspeed over the control surfaces to calculate the appropriate deflections.

Figure 3 depicts the desired moments which the aircraft exerts in order to correct errors in orientation. Each moment corresponds to the angular error about its body axis. Hence body frame x, y, z errors are respectively mapped to rolling (L), pitching (M), and yawing (N) moments.

We use one PD control law to map angular errors to desired angular accelerations, and then scale this by the moment of inertia to obtain the desired correctional moment about each axis. This allows us to use the same two gains for all three axes: a proportional, K_{a_p}, and derivative, K_{a_d}, gain. The control law is shown in the following equations:

$$L = (K_{a_p} E_x + K_{a_d} \dot{E}_x) I_x \tag{11}$$

$$M = (K_{a_p} E_y + K_{a_d} \dot{E}_y) I_y \tag{12}$$

$$N = (K_{a_p} E_z + K_{a_d} \dot{E}_z) I_z \tag{13}$$

where the time derivatives of E_x, E_y, E_z are denoted by $\dot{E}_x$, $\dot{E}_y$, $\dot{E}_z$ respectively. The moments of inertia about each body frame axis with respect to the center of mass are denoted by I_x, I_y and I_z.

2.1.3 Control Surface Deflections

The moment exerted on an aircraft due to its control surfaces is a function of the magnitudes of the surface deflections and the speed of the air flowing over the surfaces. This relationship is well understood in the aircraft community and can be expressed by the following equations [11]:

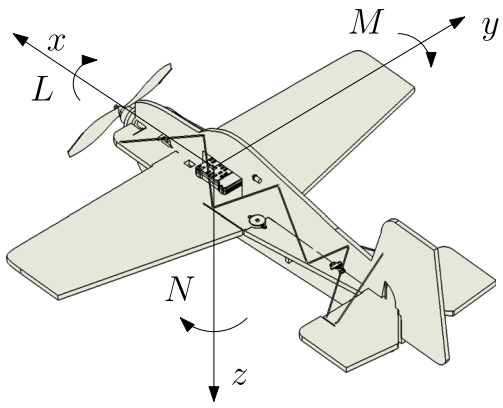

Fig. 3 Aircraft moments

$$L = \frac{1}{2} \rho v^2 S b C_{l_{\delta_a}} \delta_a \tag{14}$$

$$M = \frac{1}{2} \rho v^2 S \bar{c} C_{m_{\delta_e}} \delta_e \tag{15}$$

$$N = \frac{1}{2} \rho v^2 S b C_{n_{\delta_r}} \delta_r \tag{16}$$

where we denote the air density with ρ, the airspeed with v, the wing area with S, the wing span with b and the mean aerodynamic chord with $\bar{c}$. We use δ_a, δ_e and δ_r to denote the aileron, elevator, and rudder deflections and their respective control derivative coefficients are $C_{l_{\delta_a}}$, $C_{m_{\delta_e}}$ and $C_{n_{\delta_r}}$. The wing area, wing span, and mean aerodynamic chord are determined from a CAD model, while the control derivative coefficients are determined experimentally [12] and are a non-dimensional representation of the moment generated per unit control deflection.

The relationships in Eqs. 14–16 were derived for conventional aircraft. In these aircraft, the speed of the airflow over the control surfaces is equivalent to the speed of the aircraft. However, for small agile aircraft the propeller slipstream effects must also be considered. For example, a hovering agile aircraft is stationary but yet generates all of its control authority from the propeller slipstream. Thus we modify Eqs. 14–16 by replacing the aircraft speed, v, with slipstream speed, v_s. We can rearrange the equations to solve for the control surface deflection needed to generate the desired moment:

$$\delta_a = \frac{L}{\frac{1}{2} \rho v_s^2 S b C_{l_{\delta_a}}} \tag{17}$$

$$\delta_e = \frac{M}{\frac{1}{2} \rho v_s^2 S \bar{c} C_{m_{\delta_e}}} \tag{18}$$

$$\delta_r = \frac{N}{\frac{1}{2} \rho v_s^2 S b C_{n_{\delta_r}}} \tag{19}$$

where L, M and N are the desired moments from Eqs. 11–13.

We estimate the slipstream speed, v_s, using momentum theory [13]:

$$v_s = \sqrt{u^2 + \frac{2T}{\rho A_{prop}}} \tag{20}$$

where the propeller disk area is denoted by A_{prop}, and the thrust, T, is approximated by using the thrust command. We then smooth this signal using a second-order low-pass filter with a 2 Hz natural frequency and .707 damping ratio. The low-pass filter introduces some delay, but considering it takes some time for the flow created by the propeller to reach the control surfaces downstream, adding this delay is consistent with the slipstream we are modeling. In

addition, this value is bounded to always be greater than the slipstream in a hover (calculated using Eq. 20, and setting $u = 0$ and $T = mg$), to avoid excessive control action at low slipstream values.

A high-level overview of the attitude controller shown in the block diagram in Fig. 4. Using this logic, the aircraft is capable of tracking the desired orientation, which is obtained from the position controller.

2.2 Position Controller

The position controller modifies the reference orientation in order to reduce errors in position. This error reduction is achieved by commanding a desired attitude which redirects the aircraft towards the reference position. Although this approach is generally not used in traditional aircraft, similar approaches are used for quadrotors and for a hovering agile fixed-wing aircraft in [5]. The position control algorithm begins by evaluating the inertial position error and its derivative:

$$\Delta \mathbf{p_i} = \mathbf{p_{ref}} - \mathbf{p}, \ \Delta \dot{\mathbf{p}}_\mathbf{i} = \frac{d}{dt}(\mathbf{p_{ref}} - \mathbf{p}) \tag{21}$$

To correct the position errors, we will need to rotate the reference orientation of the aircraft, and we therefore need to express the position errors in the frame corresponding to that orientation. This is done as follows:

$$\begin{bmatrix} \Delta x_p \\ \Delta y_p \\ \Delta z_p \end{bmatrix} = C_{ri} \Delta \mathbf{p_i}, \begin{bmatrix} \Delta x_d \\ \Delta y_d \\ \Delta z_d \end{bmatrix} = C_{ri} \Delta \dot{\mathbf{p}}_\mathbf{i} \tag{22}$$

where the rotation matrix from inertial to reference frame is denoted by C_{ri}, which can be formed from $\mathbf{q_{ref}}$:

$$C_{ri} = \begin{bmatrix} q_{r_0}^2 + q_{r_1}^2 - q_{r_2}^2 - q_{r_3}^2 & 2(q_{r_1}q_{r_2} + q_{r_0}q_{r_3}) & 2(q_{r_1}q_{r_3} - q_{r_0}q_{r_2}) \\ 2(q_{r_1}q_{r_2} - q_{r_0}q_{r_3}) & q_{r_0}^2 - q_{r_1}^2 + q_{r_2}^2 - q_{r_3}^2 & 2(q_{r_2}q_{r_3} + q_{r_0}q_{r_1}) \\ 2(q_{r_1}q_{r_3} + q_{r_0}q_{r_2}) & 2(q_{r_2}q_{r_3} - q_{r_0}q_{r_1}) & q_{r_0}^2 - q_{r_1}^2 - q_{r_2}^2 + q_{r_3}^2 \end{bmatrix} \tag{23}$$

it should be noted in Eq. 23 q_r is used to denote q_{ref}.

The aircraft thruster is directed along the reference x-axis. Say the aircraft contains position error in the reference y-axis, the reference orientation should rotate about the reference $+z$-axis to redirect the thruster towards the reference position. Larger positional errors should lead to larger rotations of the reference orientation. This rotation of the reference orientation can be mathematically represented as a quaternion:

$$\mathbf{q_z} = [\cos\frac{\Theta_z}{2}, 0, 0, \sin\frac{\Theta_z}{2}] \tag{24}$$

$$\Theta_z = K_{p_p}\Delta y_p + K_{p_d}\Delta y_d \tag{25}$$

where $-45° \leq \Theta_z \leq 45°$. Limiting the rotation to 45° guarantees that the aircraft will still partially fly in the direction of its reference heading, even when the aircraft is very far from the reference position. The proportional and derivative gains from the PD control law in Eq. 25 are denoted by K_{p_p} and K_{p_d}.

Similarly, say an aircraft has a position error in the positive z direction of the reference frame. Using the the same logic, the reference orientation should rotate negatively about the reference frame $+y$ axis, which is represented by the following quaternion:

$$\mathbf{q_y} = [\cos\frac{\Theta_y}{2}, 0, \sin\frac{\Theta_y}{2}, 0] \tag{26}$$

$$\Theta_y = -(K_{p_p}\Delta z_p + K_{p_d}\Delta z_d) \tag{27}$$

where $-45° \leq \Theta_y \leq 45°$.

The complete rotation can be expressed as a sequence of quaternion multiplications which is used to obtain the desired orientation that the attitude controller aims to track:

$$\mathbf{q_{des}} = \mathbf{q_{ref}} \otimes \mathbf{q_z} \otimes \mathbf{q_y} \tag{28}$$

which can be interpreted as rotating the reference orientation about the reference z axis by Θ_z and then about the reference y axis by Θ_y. Ultimately, these rotations redirect the aircraft toward the reference position, which reduces the error in position. The rotations can be visualized in Fig. 5.

Our approach to controlling position, rotating the reference orientation based positional errors expressed in the same frame, allows the methodology to be robust to variations in aircraft orientation. The aircraft can be in *any* orientation (i.e. level flight, hover, or knife-edge), and the position controller will remain effective. In addition, the

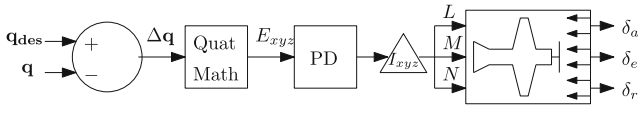

Fig. 4 Attitude control summary

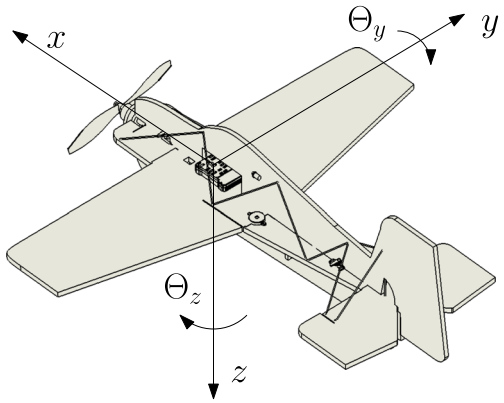

Fig. 5 Reference orientation rotations

position control can handle large errors in position and orientation. The position control design can be summarized by the block diagram in Fig. 6.

2.3 Thrust Controller

As shown in Fig. 2, the thrust control is separate from the control surfaces. The main purpose of the thruster is to control the aircraft height, h, and longitudinal speed, u. Alternatively, we can also use the thruster to increase control surface effectiveness by generating a faster slipstream. In this section we discuss both purposes.

2.3.1 Height and Speed Control

The thrust axis is aligned with the body x-axis, and thus produces force in that direction. We can use this force to generate acceleration along the body x-axis, $\dot{u}$. We adjust the acceleration to track the desired speed and height. We proportionally adjust the acceleration to the amount of speed error, Δu. On the contrary, errors in height, Δh, are scaled by the sine of the pitch angle, $\sin\theta$, to adjust acceleration. The reason is best demonstrated through an example: consider an aircraft with positive altitude error, i.e. the aircraft should increase its altitude. The further the aircraft is pitched down, the more the throttle should be reduced; the more the aircraft is pitched up, the more the throttle should be increased; and if the aircraft has zero pitch the throttle has no effect on altitude. All of these traits are represented by scaling the height error by sine of the pitch angle. We use P and PI control laws for longitudinal speed and height control respectively. The acceleration command is shown as follows:

$$\dot{u} = K_{u_p}\Delta u + (K_{h_p}\Delta h + K_{h_i}\int \Delta h\, dt)\sin\theta \tag{29}$$

where the proportional gain for longitudinal speed is denoted by K_{u_p}, and altitude proportional and integral gains are respectively represented by K_{h_p} and K_{h_i}.

We can approximate all the forces acting on the aircraft to obtain the required thrust which results in the longitudinal acceleration obtained in Eq. 29. Solving for this thrust is easiest using a free body diagram of the aircraft in the vertical plane, as shown in Fig. 7.

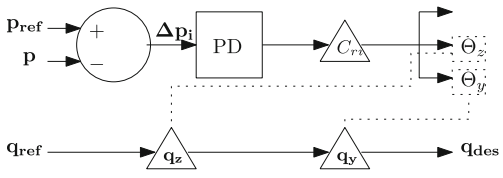

Fig. 6 Position control summary

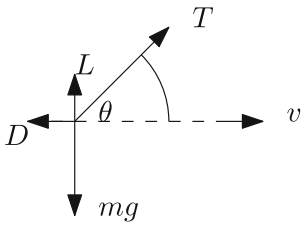

Fig. 7 Free body diagram

We denote the magnitude of thrust, lift, and drag by T, L and D respectively. The mass of the aircraft and the acceleration due to gravity are respectively denoted by m and g. By assuming the aircraft is flying at constant altitude, with constant velocity, v, in the horizontal plane, we can use Newton's 2^{nd} law and sum the forces along the thrust line to obtain:

$$T = mg\sin\theta + m\dot{u} + D\cos\theta - L\sin\theta \tag{30}$$

We approximate the aerodynamic forces using our simulation environment. We record the aerodynamic force acting in the body x-axis for various aircraft speeds in steady level flight, as well as in a hover, and apply a second-order curve fit to obtain the aerodynamic force as a function of speed, which is shown in Fig. 8. In order to account for some discrepancy between our model and the actual aircraft drag properties, we scale this by a gain, K_{aero}, which is tuned in flight:

$$D\cos\theta - L\sin\theta \approx K_{aero}F_{aero}(v) \tag{31}$$

Using the thruster to control height and longitudinal speed can be summarized by the block diagram in Fig. 9.

2.3.2 Attitude Control Augmentation

Generally, changing the control surface deflection is sufficient to generate a desired moment. However, if the control surface is saturated (i.e. at its maximum deflection), we can use the thruster to increase the slipstream speed

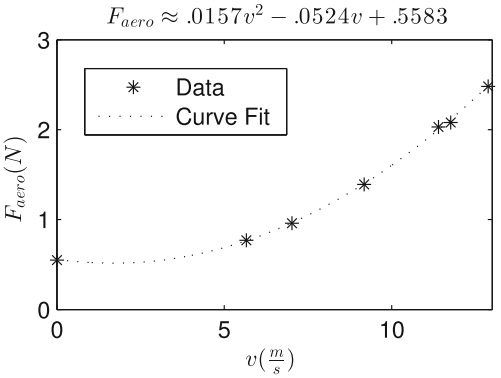

Fig. 8 Aerodynamic force approximation

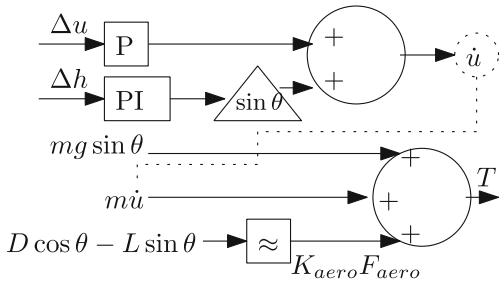

Fig. 9 Height and speed control summary

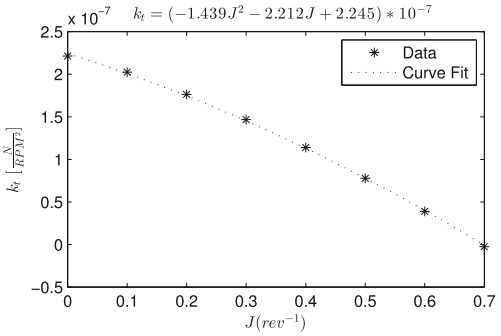

Fig. 10 Thruster coefficient vs advance ratio for electrify PowerFlow 10 × 4.5 propeller

to increase the moment generated by the control surfaces. We can compute a desired slipstream speed based on the desired moment and control surface characteristics, and then use this desired slipstream coupled with momentum theory to calculate a desired thrust force. We outline these calculations for a saturated elevator, although similar calculations could be made for a saturated aileron or rudder:

$$v_{s_{des}} = \sqrt{\frac{M}{\frac{1}{2}\rho S\bar{c}C_{m_{\delta_e}}\delta_{e_{max}}}} \tag{32}$$

$$T = \frac{\rho A_{prop}}{2}(v_{s_{des}}^2 - u^2) \tag{33}$$

The total thrust force is obtained by summing the component to correct height and speed errors, as given by Eq. 30, and the component to increase the control authority when necessary (i.e. a saturated control surface), given by Eq. 33.

2.3.3 Propeller Force to Rotation Conversion

Our thrust controller outputs a thrust force, however in reality we command the thruster with a pulse width modulation (PWM) signal to the electronic speed controller (ESC), which directly maps to a propeller rotational speed, ω_t. Here we discuss the mapping from propeller force to a rotational speed, while the mapping from propeller rotational speed to PWM signal will be discussed in Section 5.

The thrust force quadratically varies with propeller rotational speed, which can be represented by the following equation:

$$\omega_t = \sqrt{\frac{T}{k_t(J)}} \tag{34}$$

Simple thruster models contain a constant thrust coefficient, k_t. However, as the aircraft flies faster the same propeller rotational speed will produce less force, as the difference in airflow velocity entering and leaving the propeller lessens. We account for this phenomenon by modeling the thrust coefficient as a function of advance ratio, J, using the model presented in [14], shown in Fig. 10, where the advance ratio is defined as

$$J = \frac{v}{2R\frac{\omega_{t_{prev}}}{60}} \tag{35}$$

$\omega_{t_{prev}}$ is the thrust command at the previous time step, in RPM. The advance ratio is bounded such that $0 \leq J \leq 0.5$, which ensures the propeller is in its normal working state [12].

3 Maneuver Generator

In the preceding section, we developed a control algorithm capable of tracking a given reference trajectory. That trajectory is generated by a maneuver generator, and consists of a time history of reference motion variables: reference orientation, reference position, and reference longitudinal speed. One approach to do this is to generate reference motion variables for each maneuver using optimization with kinematic and dynamic constraints, such as in [15]. Here, instead, we use a heuristic approach that generates paths that are not necessarily feasible. While it is not possible for the aircraft to perfectly track the output of the maneuver generator, we have found that this approach does allow the maneuvers to be accomplished efficiently.

3.1 Reference Position

Before discussing the generation of specific maneuvers, we address the generation of the reference position, as this applies to multiple maneuvers. Generating the reference position can be done using two approaches, depending on whether the controller aims to track a specified *position*, or if it aims to track a specified *path*. In the first approach, the user simply specifies that reference position. The second approach requires a more detailed explanation, best illustrated by an example.

Consider an aircraft at position **p**, aiming to fly along a line, as shown in the top view of Fig. 11.

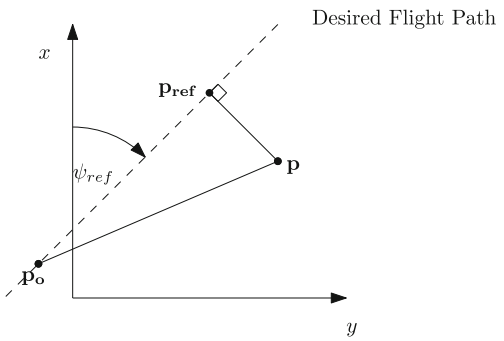

Fig. 11 Line following example

We define the desired flight path as the line extending from the aircraft position at the initiation of the maneuver, $\mathbf{p_0}$, extending in the direction of the reference heading, ψ_{ref}, represented by the dashed line in Fig. 11. The reference position sent to the position controller is the point on the desired flight path closest to the aircraft, which can be calculated as follows:

$$\mathbf{p_{ref_{xy}}} = \mathbf{p}_{\parallel} = ((\mathbf{p} - \mathbf{p_0})_{\mathbf{xy}} \bullet \begin{bmatrix} \cos\psi_{ref} \\ \sin\psi_{ref} \end{bmatrix}) \begin{bmatrix} \cos\psi_{ref} \\ \sin\psi_{ref} \end{bmatrix} + \mathbf{p_{o_{xy}}} \tag{36}$$

where the x and y components of the variable are denoted by $()_{xy}$. In Eq. 36, the term in parentheses represents the length of $(\mathbf{p} - \mathbf{p_0})_{\mathbf{xy}}$ projected onto the desired flight path. This is then multiplied by the unit vector in the direction of the desired flight path to obtain the vector from the initial position to the reference position. It should be noted that $\mathbf{p_{ref_z}}$ is left free to be defined according to the maneuver, as will be discussed later in this section.

3.2 Straight and Level

While level flight is not considered an aerobatic maneuver, we nevertheless discuss how it is specified because it is often used to transition between other aerobatic maneuvers. When flying straight and level, the aim of the aircraft is to fly level while tracking a straight line at constant altitude. This desired motion implies a zero reference roll angle. We allow the reference longitudinal speed and heading to be arbitrarily chosen. The reference pitch angle is chosen to be consistent with the specified reference longitudinal speed [16]. Thus, the maneuver generator for level flight is as follows:

$\theta_{ref} = \theta(u_{ref})$
$\mathbf{q_{ref}} = \text{EulToQuat}(\phi = 0°, \theta = \theta_{ref}, \psi = \psi_{ref})$
$\mathbf{p_{ref}} = [\mathbf{p}_{\parallel},\ p_{0_z}]$
where euler angles are converted to their equivalent quaternion representation.

3.3 Knife-Edge

Knife-Edge maneuvers are useful for flying between obstacles when the passage is narrower than the aircraft's wingspan. The goal of a knife-edge is to maintain 90° roll while tracking a straight line at constant altitude. Similarly to level flight, we allow the reference longitudinal speed and heading to be user-specified, and determine the reference pitch based on the reference speed [16]. The maneuver generator for knife-edge flight is as follows:

$\theta_{ref} = \theta(u_{ref})$
$\mathbf{q_{ref}} = \text{EulToQuat}(\phi = 90°, \theta = \theta_{ref}, \psi = \psi_{ref})$
$\mathbf{p_{ref}} = [\mathbf{p}_{\parallel},\ p_{0_z}]$

3.4 Rolling Harrier

In a rolling Harrier maneuver, the aircraft flies along a constant altitude line, while maintaining a constant roll rate. While this maneuver has little practical utility, it does demonstrate the aircraft's extreme flight capability and allows us to demonstrate the versatility of our control system. As in the two previous maneuvers, the reference longitudinal speed and heading are user-specified, and the reference pitch angle is chosen as a function of the reference longitudinal speed [16]. The maneuver generator for a rolling Harrier is as follows:

$\theta_{ref} = \theta(u_{ref})$
$\mathbf{q_{ref}} = \text{EulToQuat}(\phi = \Omega t, \theta = \theta_{ref}, \psi = \psi_{ref})$
$\mathbf{p_{ref}} = [\mathbf{p}_{\parallel},\ p_{0_z}]$
where the desired roll rate is denoted by Ω, and the time in seconds since the initiation of the maneuver is denoted by t.

3.5 Hover

Hovering can be useful for surveillance tasks, as the aircraft remains stationary in a vertical orientation. Given the nature of the maneuver, the reference pitch is 90° and the reference longitudinal speed is zero. To ensure a smooth transition into the hover, the reference heading is unchanged from the preceding maneuver (most likely level flight) and the reference roll angle is set to zero. This allows the transition to only occur along the pitch axis.

Ideally, we would like the aircraft to stop instantaneously when commanding a hover; thus the reference position is initially set to the initial position of the maneuver. In order to avoid the aircraft backtracking to that position, we change the reference position to the aircraft's current position once the vertical orientation has been achieved. This new reference position, $\mathbf{p_f}$, is kept for the remaining of

the hover. The maneuver generator for the hover maneuver is as follows:

$u_{ref} = 0$

$\mathbf{q_{ref}} = \text{EulToQuat}(\phi = 0, \theta = 90°, \psi = \psi_{ref})$

$$\mathbf{p_{ref}} = \begin{cases} \mathbf{p_0} & \text{prior to vertical orientation} \\ \mathbf{p_f} & \text{otherwise} \end{cases}$$

3.6 Aggressive Turnaround

Some situations may require the aircraft to reverse course quickly in a small space. We design the maneuver assuming the aircraft is initially in level flight, and must end the maneuver in level flight with opposite heading. We first command the aircraft into a hover. Once the pitch angle exceeds 45°, we command inverted flight with a heading opposite that at the start of the maneuver. Finally, once the aircraft pitches down to less than the reference pitch angle, the aircraft is commanded to roll back to level flight. Transitions between stages are unidirectional – i.e. once the maneuver proceeds to the next stage, it cannot go back to a previous stage, regardless of its orientation. The maneuver generator for an aggressive turnaround is defined as follows:

All Stages

$\theta_{ref} = \theta(u_{ref})$

Stage 1 $\theta : \theta_0 \rightarrow 45°$

$\mathbf{q_{ref}} = \text{EulToQuat}(\phi = 0, \theta = 90°, \psi = \psi_{ref})$

$\mathbf{p_{ref}} = \mathbf{p_0}$

Stage 2 $\theta : 45° \rightarrow 90° \rightarrow \theta_{ref}$

$\psi_{ref} \leftarrow \psi_{ref} + 180°$

$\mathbf{q_{ref}} = \text{EulToQuat}(\phi = 180°, \theta = \theta_{ref}, \psi = \psi_{ref})$

$\mathbf{p_{ref}} = \mathbf{p}_{\parallel}$

Stage 3 otherwise

$\mathbf{q_{ref}} = \text{EulToQuat}(\phi = 0, \theta = \theta_{ref}, \psi = \psi_{ref})$

$\mathbf{p_{ref}} = \mathbf{p}_{\parallel}$

4 Simulation

Before implementing the control system on the aircraft, we demonstrate successful autonomous aerobatics in our high-fidelity in-house simulator [12, 17]. We then implement the control system on the flight hardware, and validate the controller in a hardware-in-the-loop (HIL) simulator as an intermediate step between conventional simulation and flight testing. Both conventional and HIL simulators require an accurate agile aircraft dynamics model, which is briefly discussed here.

4.1 Modeling

The model used in the present work represents the six degree of freedom dynamics of an aircraft, based on standard rigid body dynamics equations coupled to accurate models of the aerodynamics [18], thruster dynamics [19] [14], and propeller slipstream [20].

4.1.1 Dynamics and Kinematics

The following differential equations describe the motion of a rigid body in three-dimensional space:

$$\begin{aligned} \dot{\mathbf{p}}_\mathbf{i} &= C_{bi}^T \mathbf{v_b} \\ \dot{\mathbf{v}}_\mathbf{b} &= \frac{1}{m}\mathbf{F_b} - \omega_\mathbf{b} \times \mathbf{v_b} \\ \dot{\mathbf{q}} &= \frac{1}{2}\mathbf{q} \otimes \omega_\mathbf{b} \\ \dot{\omega}_\mathbf{b} &= I_b^{-1}[\mathbf{M_b} - \omega_\mathbf{b} \times I_b \omega_\mathbf{b}] \end{aligned} \tag{37}$$

where $\mathbf{p_i}$ is the aircraft position and $\mathbf{q}$ is the aircraft orientation, expressed as a unit quaternion. The vectors $\mathbf{v_b}$ and $\omega_\mathbf{b}$ denote the aircraft translational and rotational velocity expressed in the body frame; while C_{bi} denotes the rotation matrix from the inertial to the body frame. The aircraft mass is denoted by m, while its inertia matrix resolved in the body frame is I_b. Vectors $\mathbf{F_b}$ and $\mathbf{M_b}$ denote the forces and moments exerted on the aircraft, expressed in the body frame, and are due to aerodynamics, thruster, and gravity. The gravity force is trivial to obtain, but the aerodynamics and thruster forces are predicted using more detailed models. A propeller slipstream model is also needed to accurately estimate the aerodynamic forces.

4.1.2 Aerodynamics

The aerodynamic forces are calculated based on a component breakdown approach that partitions each component of the aircraft (wing, tail, rudder etc.) into small segments that produce lift, drag, and moment about the aircraft's center of mass. Khan's aerodynamics model [18] includes modeling of:

- the full flight envelope i.e ± 180 deg. angle of attack and sideslip range,
- partial flow conditions over the aerodynamic surfaces,
- low aspect ratio surfaces, and
- large control surfaces with large deflections.

4.1.3 Thruster Dynamics

Khan's thruster model [19] is based on the propeller and motor characteristics. It uses blade element momentum theory, coupled with an inflow model to predict the thruster generalized forces. The model also includes thruster gyroscopic effects. The inputs to the model are the propeller rotational speed and the incoming flow speed and direction. The model is able to capture the following flow conditions that can occur in aerobatic flight:

– stationary conditions (i.e. hover),
– pure axial flow conditions (flow aligned with the propeller rotational axis) (i.e. level flight),
– oblique flow conditions (flow at an angle to its rotation axis), due to aerobatic flight or wind gusts, and
– in reverse flow conditions (i.e. aerobatics).

4.1.4 Propeller Slipstream

The propeller slipstream has an important impact on the ability of the control surfaces to generate generalized forces. The slipstream provides additional flow over the control surfaces, enabling the aircraft to maintain control authority at low speeds. Khan's novel slipstream model [20] includes two components: the axial velocity and the swirl velocity. The axial component models both the acceleration and diffusion phenomena that occur in the slipstream. The model has been shown to accurately predict the axial slipstream velocity up to 6 propeller diameters downstream of the propeller. On the other hand, the swirl velocity is known to cause a rolling moment on the aircraft that reduces the thruster reaction moment. To model this, the rolling reaction moment caused by the thruster [17] is reduced accordingly.

4.2 Real-time Simulator

The individual components of the simulator, including aerodynamics, thruster and slipstream, have been validated through wind-tunnel tests and static bench tests. The components are assembled to create a real-time simulation environment using Matlab/Simulink, with visualization using X-Plane. Further validations were performed through pilot-in-loop simulations, in which an experienced professional RC pilot found the simulator to be accurate.

The real-time simulation environment depicted in Fig. 12a is used for initial evaluation of the control algorithm and to adjust control gains. It has been found easiest to tune the gains from the inside out: the thrust can be temporarily set to a constant value, and the position control gains can be set to zero, while the attitude controller is tuned first. The position control gains can then be increased until satisfactory position tracking is achieved. Finally, the thrust control can be tuned to track longitudinal speed and height. Using this approach, we are able to converge to a set of control gains which lead to successful autonomous aerobatic flight in this conventional simulation environment, as shown in [16].

4.3 Hardware-In-The-Loop Simulation

The conventional simulation provides an environment which tests the control system on the simulated plant, but does not consider realistic aspects such as sensor noise,

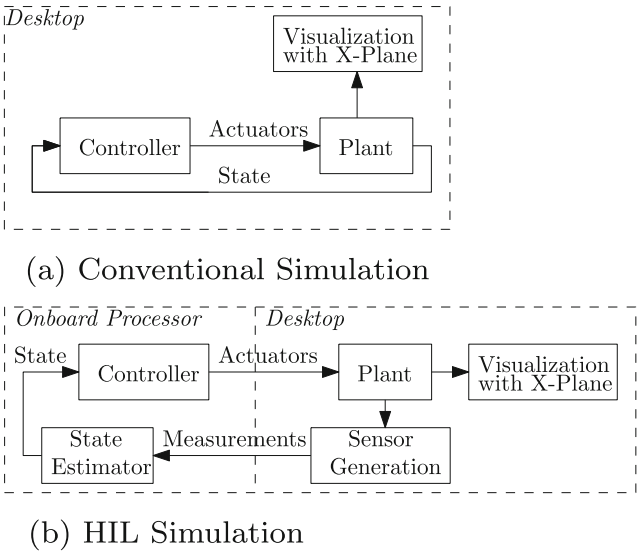

Fig. 12 Simulation environments

state estimation errors, controller discretization, memory overflow, timing delays, etc., all of which can have significant effects on the control system performance. Prior to flight testing, it is beneficial to evaluate these effects using a hardware-in-the-loop (HIL) environment.

In this scenario, depicted in Fig. 12b, the control algorithm is programmed in the flight computing hardware (that will later be mounted on the aircraft). The on-board processor feeds artificial actuator commands to a desktop computer on which the real-time aircraft simulator is running. That simulator computes the system dynamics and generates artificial sensor measurements, which include realistic sensor noise. These measurements are sent back to the on-board processor, which uses its on-board state estimator to provide state inputs to the control algorithm. This is a more realistic environment in which to test out the control system, and more specifically, the aspects related to the hardware implementation.

The Pixhawk flight hardware [21], which will be used in our flight testing, comes with a built-in HIL environment which uses the X-Plane physics engine as the plant. However, as we believe our dynamics model to be more realistic than X-Plane's, we modify this setup to replace the X-Plane physics engine with our in-house Simulink dynamics model. This modification is discussed in detail in [22].

The HIL simulation demonstrated that the control algorithm implemented in conventional simulation needed to be modified with signal smoothing techniques and a reduction of control gains to achieve successful flight in a more realistic HIL simulation environment [22].

Differentiated signals were found to contain unrealistic values that did not occur in the conventional simulation, and these outliers could easily destabilize the control. To reject

these outliers, if the derivative of a position error is greater than 50 $\frac{m}{s}$, or if the derivative of an angular error is greater than 20 $\frac{rad}{s}$, the previous value is used instead.

The noise in the derivative approximations can also destabilize the system and leads to noisy control inputs, which can damage the actuators. To reduce the noise in the position and angular error derivative approximations, second-order low-pass filters are applied to those signals. We use a damping ratio of .707, and natural frequencies of 3 Hz for position, and 10 Hz for angles.

5 Experimental Implementation

The experimental platform is an off-the-shelf RC aircraft, the WM Parkflyers McFoamy, which is retrofitted with additional carbon fiber reinforcements and a custom 3D printed motor mount. The aircraft is equipped with the Electrify SS-25 Brushless ESC, the Great Planes Rimfire 400 brushless motor and the Electrify PowerFlow 10 × 4.5 propeller. A 3DR Pixhawk Mini flight controller [21] runs the open source PX4 flight stack to perform state estimation and control. The default extended Kalman filter, the EKF2, fuses the Pixhawk's embedded IMU (MPU9250, ICM20608), barometer (MS5611), as well as a GPS module (ublox Neo-M8N) with compass (HMC5983) to provide an estimate of the 12 state system (position, attitude, velocity, and body rates). The flight controller contains a 32 bit ARM Cortex M4 Processor (STM32F427 Rev 3) which executes the control loop at 200 Hz. The McFoamy aircraft is displayed in Fig. 1 and the physical properties are shown in the Appendix.

The controller generates three control surface deflections, and one propeller rotational speed. The control surfaces are attached to control rods driven by servo motors while the propeller is driven by a brushless DC motor controlled by an electronic speed controller (ESC). The servos and the ESC are all commanded using pulse-width-modulated (PWM) signals. We therefore had to experimentally characterize the servo linkages and the ESC in order to ensure that the appropriate PWM signals were generated and that the desired control surface deflections and propeller speed were obtained. These characteristics were implemented as curve-fits to translate the controller outputs to PWM signals during operation of the system.

6 Results

The outdoor flight tests took place at the West Island Model Aeronautics Club, Montreal, Canada, which has a flight field of dimension $100m \times 100m$. During the tests, the average wind gusts recorded at a nearby airport were 10-12 knots [23]. All the autonomous flight modes tested were airborne maneuvers, i.e. takeoff and landing were performed manually by a professional RC pilot.

To demonstrate the effectiveness of the controller in flight tests, and to clearly demonstrate differences between the simulations and experiments, each maneuver is performed in the conventional simulation, the HIL simulation, and actual flight, shown in Figs. 13 and 14. For each environment, the same set of control gains is used, shown in the Appendix. We evaluate each maneuver at two reference longitudinal speeds, $u_{ref} = 5\ \frac{m}{s}$ & $u_{ref} = 9\ \frac{m}{s}$. For consistency in the initial conditions for the aerobatic maneuver in each environment, we command level flight for $20m$ prior to the initiation of each maneuver. While only one set of results is shown here for each maneuver, the flight tests were repeated multiple times, each time achieving similar results. An exception to this is for the hover maneuver, where the wind has the largest effect on the aircraft, because the wings present a large surface area to the wind. Although the hover always remained stable, at times there were larger drifts in position.

For each maneuver, the raw data has been altered such that the maneuver starts at the origin, and the reference heading is along the x axis. This allows motion in the y-direction to be viewed as cross-track error, and motion in the z-direction to be viewed as altitude error. For easier comparisons between experiments and simulations, the horizontal axis in the figures corresponds to the x axis in flight, not time. In order to see the transition into the maneuver, two meters of level flight are shown prior to the initiation of the maneuver (i.e. the plots starts at $x = -2\ m$). To evaluate the performance of the attitude tracker, the orientation angles (roll, pitch, and yaw) are displayed along with desired orientation (dotted lines).

In each environment, the aircraft is commanded to perform the same maneuver, for the same distance covered. The desired orientation for each environment differs because the position controller uses errors in position to determine the desired orientation, and each environment has different position errors.

6.1 Hover

As shown in Figs. 13 & 14, we successfully transition from level flight to hover at both $5\frac{m}{s}$ and $9\frac{m}{s}$, and hold the hover for $10s$ in the conventional simulator, in the HIL simulator, and in the outdoor flights. The position control is noticeably worse in experiment, likely because the wind has a large effect on a hovering aircraft, as the wings generate large drag, and there is no actuator to directly cancel the wind.

The aircraft traveling at $5\frac{m}{s}$ pitches up into a hover, and as the wings become more exposed to the wind gusts, the

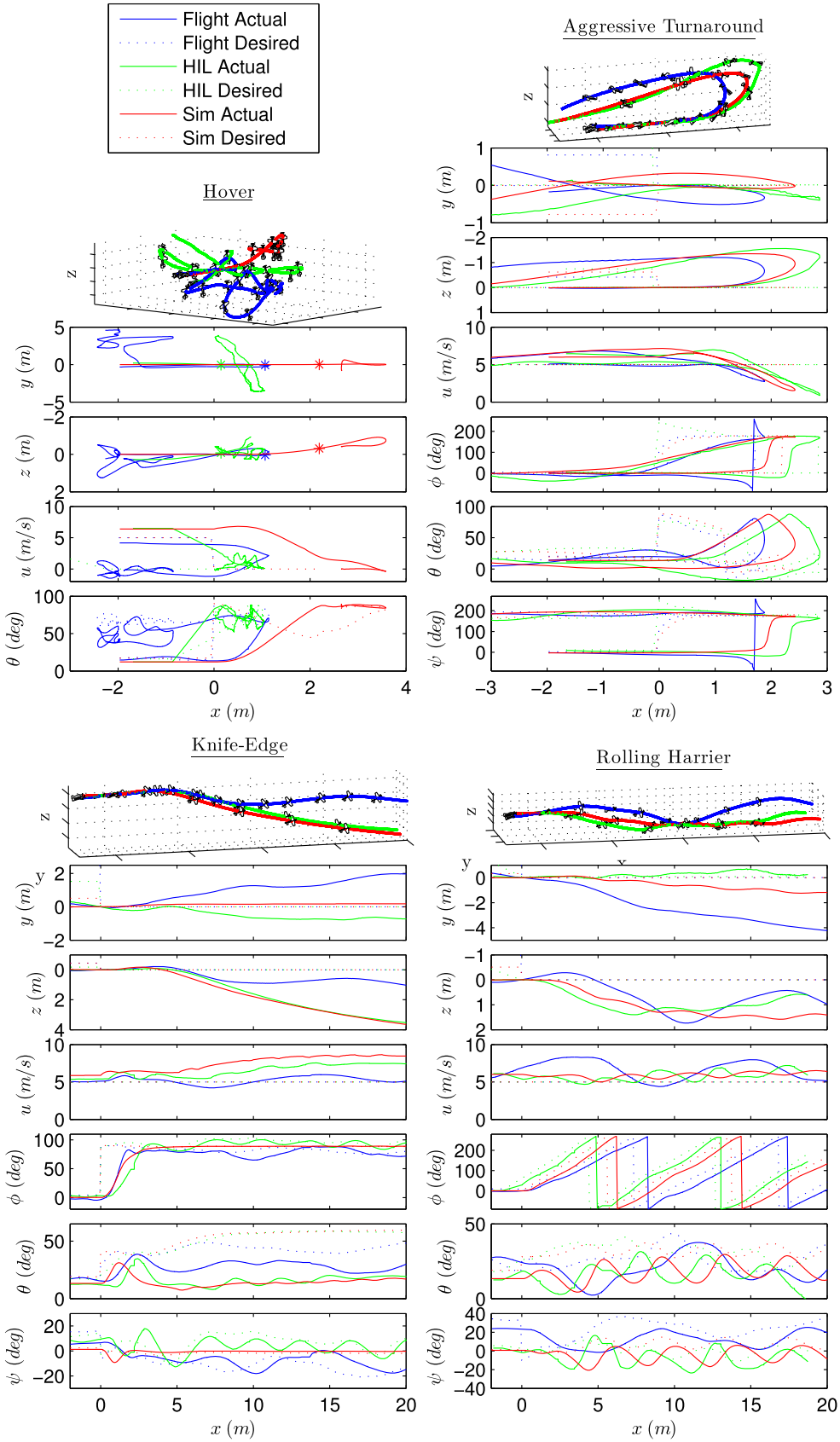

Fig. 13 Results at $5\frac{m}{s}$

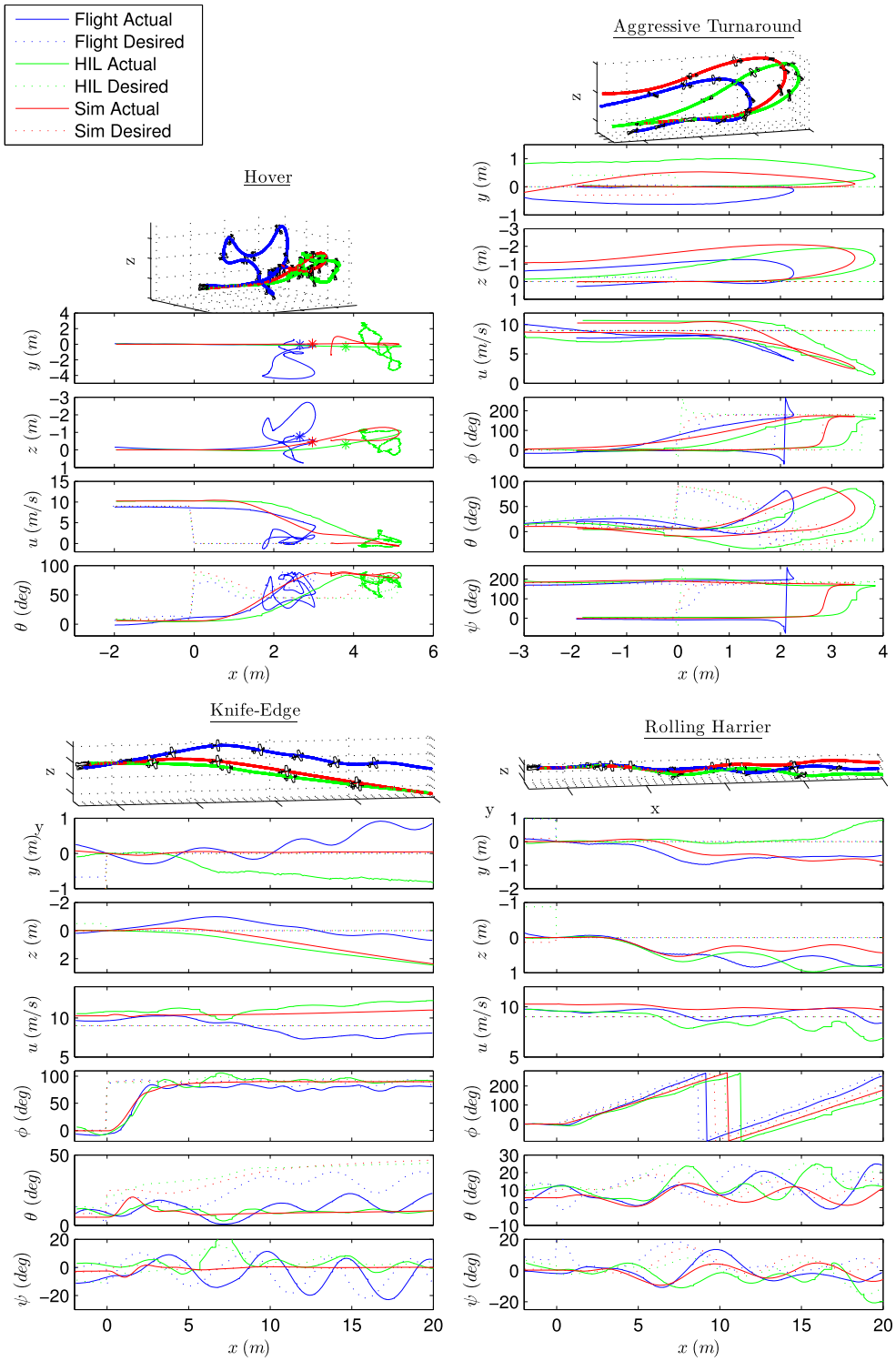

Fig. 14 Results at $9\frac{m}{s}$

aircraft gets pushed away from the reference position, with about $5m$ cross-track error, $3m$ in the direction of flight but with less deviation in altitude. The aircraft eventually finds an orientation which points its nose into the wind, and remains stationary. As seen in Fig. 13, the pitch angle reduces to $40° - 50°$ to counteract the wind, whereas in both simulations without wind, the pitch remains closer to 90°.

For the transition to hover at $9\frac{m}{s}$, it is likely the winds were calmer in the x-direction as the aircraft does not get pushed backwards, and the steady-state pitch angle

is higher. However, there is $4m$ of cross-track error, and the aircraft climbs $3m$. The aircraft climbs higher when transitioning from a faster speed, as there is more kinetic energy to dissipate.

The position control is substantially better in both simulations because the environment is wind-free, and is better in the conventional simulation than in the HIL simulation. Even without wind disturbance, sensor noise, as well as the thruster's gyroscopic moment will create small errors in position, and the aircraft sways side to side correcting these errors. It is possible to eliminate this swaying in simulation by reducing the position controller gains, however, these gains lead to unstable hovering in windy conditions outdoors, as the aircraft is not quick enough to react to wind gusts with reduced position gains.

6.2 Aggressive Turnaround

Once again, Figs. 13 & 14 show that the control system performs the aggressive turnaround effectively in all three environments and at both speeds. In the experiment, at both speeds, the aircraft is able to reverse the heading of the aircraft in about $1s$, covering $\approx 2m$ in the direction flight, with $\approx .5m$ of cross-track error, and a $1m$ gain in altitude during the maneuver. It is somewhat surprising that the aircraft turns around in the same distance when traveling at a higher speed, as one would expect the aircraft to travel further and have a larger increase in altitude, as is the case in the conventional and HIL simulations. A plausible explanation could be that the aircraft was flying against a stronger wind during the $9\frac{m}{s}$ case.

Turning our attention to the orientation results, the pitch angle initially climbs to 90°. At this point, the roll and yaw angle instantaneously jump to 180° due to the Euler angle convention. The pitch angle then decreases, followed by the roll angle going from 180° to 0°. At the end of the maneuver, the aircraft is in level flight with reversed heading. This motion is similar in all three environments at both speeds.

6.3 Knife-Edge

As shown in Fig. 13, when the aircraft is traveling at $5\frac{m}{s}$, the aircraft takes about $1.8m$ ($\frac{1}{3}s$) to roll into the knife-edge, with about a $1m$ drop in altitude and less than $2m$ cross-track error during flight. At $9\frac{m}{s}$, shown in Fig. 14, the aircraft takes about $2.5m$ ($\frac{1}{4}s$) to roll into the knife-edge, with an initial $1m$ climb followed by a $1m$ drop in altitude, and $1m$ cross-track error during flight. These results are consistent with the conventional and HIL simulations, as well as with physical intuition. We expect a faster traveling aircraft to perform the roll in less time, as the aircraft has more control authority, but also expect this roll to occur over a greater distance, as the aircraft is moving faster. A reduction in cross-track error is also expected, as the wind has a smaller effect on a faster moving aircraft.

Once the maneuver is in steady state, one noticeable difference between the flight and the simulations is that the pitch angle remains more elevated in steady state in the real flight. In the simulations, the low steady-state pitch angles seems to result in a dropping altitude. Apart from this, the performance in the simulations is similar to that in flight, and the maneuver is effectively accomplished in all three environments.

6.4 Rolling harrier

We successfully perform a rolling Harrier maneuver with a desired roll rate of $\Omega = 5\frac{rad}{s}$, at both $5\frac{m}{s}$ and $9\frac{m}{s}$ shown in Figs. 13 & 14. At $5\frac{m}{s}$, the flight has about $4m$ of cross-track error and less than $2m$ drop in altitude, while at $9\frac{m}{s}$, the flight has about $1m$ of cross-track error and $1m$ drop in altitude. Again, the reduction in cross-track error at a higher speed is expected due to the wind having a lesser effect. For both speeds, the behavior in the conventional and HIL simulations are similar to experiment, with the exception that the cross-track error at $5\frac{m}{s}$ is less in simulation. In all three environments, the roll angle lags the desired roll, but by less than $0.1\ s$.

7 Conclusions

In this work, we present a physics-based control system capable of autonomously performing a wide-variety of aerobatic maneuvers, while traveling at a wide range of speeds, all using a single control system with a single set of gains. The controller is modular, consisting of an attitude controller, a position controller, a thrust controller; all driven by a maneuver generator. The controllers incorporate nonlinear kinematics valid at all orientations. Prior to flight testing, we evaluate the controller in a conventional simulation environment, followed by a hardware-in-the-loop simulation environment. Successful autonomous aerobatics and hovering are demonstrated in both simulation environments, as well as in outdoor flight testing in moderate winds. We find the aircraft motion to be similar in simulation and experiment, implying our simulation environment is realistic.

In the future, we plan to use these maneuvers in a motion planning environment to perform real-time obstacle detection and avoidance.

Acknowledgments A short version of this paper was presented in ICUAS 2017 [16]. The authors thank Michael Verrecchia for flight testing assistance and Waqas Khan for providing the agile aircraft dynamics model.

Appendix

Table 1 Aircraft properties

Parameter	Symbol	Value	Unit
Mass	m	.45	kg
	I_x	3.922×10^{-3}	kg m^2
Moments of inertia	I_y	1.594×10^{-2}	kg m^2
	I_z	1.934×10^{-2}	kg m^2
Non-zero products of inertia	I_{xz}	3.03×10^{-4}	kg m^2
Wing Area	S	.143	m^2
Wing Span	b	.864	m
Mean aerodynamic chord	$\bar{c}$	.21	m
	$C_{l_{\delta_a}}$	−.0006777	deg^{-1}
Control derivative coefficients	$C_{m_{\delta_e}}$	−.0117747	deg^{-1}
	$C_{n_{\delta_r}}$	−.0035663	deg^{-1}
Maximum aileron deflection	$\delta_{a_{max}}$	52	deg
Maximum elevator deflection	$\delta_{e_{max}}$	59	deg
Maximum rudder deflection	$\delta_{r_{max}}$	49	deg
Propeller disk area	A_{prop}	.0507	m^2

Table 2 Controller gains

Parameter	Symbol	Value	Unit
Position proportional gain	K_{p_p}	.08	rad/m
Position derivative gain	K_{p_d}	.1	$rad/\frac{m}{s}$
Attitude proportional gain	K_{a_p}	160	$\frac{rad}{s^2}/rad$
Attitude derivative gain	K_{a_d}	8	$\frac{rad}{s^2}/\frac{rad}{s}$
Speed proportional gain	K_{u_p}	3	$\frac{m}{s^2}/\frac{m}{s}$
Height proportional gain	K_{h_p}	5	$\frac{m}{s^2}/m$
Height integral gain	K_{h_i}	0.5	$\frac{m}{s^2}/ms$
Aerodynamic force approximation gain	K_{aero}	2	–

References

1. Gavrilets, V., Mettler, B., Feron, E.: Human-inspired control logic for automated maneuvering of miniature helicopter. J. Guid. Control. Dyn. **27**, 752–759 (2004)
2. Abbeel, P., Coates, A., Quigley, M., Ng, A.Y.: An application of reinforcement learning to aerobatic helicopter flight. In: Schölkopf, P.B., Platt, J.C., Hoffman, T. (eds.) Advances in neural information processing systems 19, pp. 1–8. MIT Press, Cambridge (2007)
3. Mellinger, D., Michael, N., Kumar, V.: Trajectory generation and control for precise aggressive maneuvers with quadrotors. Int. J. Robot. Res. **31**, 664–674 (2012)
4. Green, W.E., Oh, P.Y.: A Hybrid MAV for Ingress and Egress of Urban Environments. IEEE Trans. Robot. **25**, 253–263 (2009)
5. Sobolic, F.M.: Agile flight control techniques for a fixed-wing aircraft. Thesis Massachusetts Institute of Technology (2009)
6. Moore, J., Cory, R., Tedrake, R.: Robust post-stall perching with a simple fixed-wing glider using LQR-Trees, vol. 9 (2014)
7. Ure, N., Inalhan, G.: Autonomous control of unmanned combat air vehicles: design of a multimodal control and flight planning framework for agile maneuvering. IEEE Control. Syst. **32**(5), 74–95 (2012)
8. Hall, J.K., McLain, T.W.: Aerobatic maneuvering of miniature air vehicles using attitude trajectories. In: AIAA guidance, navigation and control conference and exhibit American institute of aeronautics and astronautics Inc. (2008)
9. Park, S.: Autonomous aerobatics on commanded path. Aerosp. Sci. Technol. **22**(1), 64–74 (2012)
10. Barry, A.J., Jenks, T., Majumdar, A., Lin, H.T., Ros, I.G., Biewener, A.A., Tedrake, R.: Flying between obstacles with an autonomous knife-edge maneuver. In: 2014 IEEE, international conference on robotics and automation (ICRA), pp. 2559–2559 (2014)
11. Etkin, B.: Dynamics of flight: stability and control. Wiley, Australia, Limited (1982)
12. Khan, W.: Dynamics modeling of agile fixed-wing unmanned aerial vehicles. PhD Thesis, McGill University, Montreal (2016)
13. McCormick, B. Aerodynamics, aeronautics, and flight mechanics, 2nd edn. Wiley, Hoboken (1995)
14. Khan, W., Nahon, M.: A propeller model for general forward flight conditions. International Journal of Intelligent Unmanned Systems **3**, 72–92 (2015)
15. Levin, J.M., Paranjape, A., Nahon, M.: Agile fixed-wing UAV, motion planning with knife-edge maneuvers. In: 2017 international conference on unmanned aircraft systems (ICUAS), pp. 114–123 (2017)
16. Bulka, E., Nahon, M.: Autonomous control of agile fixed-wing UAVs, performing aerobatic maneuvers. In: 2017 international conference on unmanned aircraft systems (ICUAS), pp. 104–113 (2017)
17. Khan, W., Nahon, M.: Modeling dynamics of agile fixed-wing UAVs, for real-time applications. In: 2016 international conference on unmanned aircraft systems (ICUAS), pp. 1303–1312 (2016)
18. Khan, W., Nahon, M.: Real-time modeling of agile fixed-wing UAV, aerodynamics. In: 2015 international conference on unmanned aircraft systems (ICUAS), pp. 1188–1195 (2015)
19. Khan, W., Nahon, M.: Toward an accurate Physics-Based UAV thruster model. IEEE/ASME Trans. Mechatron. **18**, 1269–1279 (2013)
20. Khan, W., Nahon, M.: Development and validation of a propeller slipstream model for unmanned aerial vehicles. J. Aircr. **52**(6), 1985–1994 (2015)
21. 3D Robotics, https://3dr.com/support/articles/quick_start_guide-3/. Accessed 8 Feb 2018
22. Bulka, E., Nahon, M.: Autonomous fixed-wing aerobatics: from theory to flight. In: Accepted to 2018 IEEE international conference of robotics and automation (ICRA) (2018)
23. Environment and Climate Change Canada, Hourly Data Report for September 12, 2017, http://climate.weather.gc.ca/historical_data/search_historic_data_e.html. Accessed 8 Feb 2018

Eitan Bulka received his B.Eng. degree in mechanical engineering from McGill University, Montreal, QC, Canada, in 2016. He is currently working toward the Ph.D. degree at McGill University. His research interests include control and motion planning of robotic systems, with a focus on aerospace applications.

Meyer Nahon received the B.A.Sc. degree in mechanical engineering from Queen's University, Kingston, ON, Canada, the M.A.Sc. degree in aerospace engineering from the University of Toronto, Toronto, ON, Canada, and the Ph.D. degree in mechanical engineering from McGill University, Montreal, QC, Canada. He was a Faculty Member in the Department of Mechanical Engineering at the University of Victoria, Victoria, BC, Canada, from 1991 to 2001. Since then, has been with McGill University, presently as a Professor of mechanical engineering. His research interests include various aspects of dynamics and control of aerial and marine vehicles and systems, with a particular emphasis on tethered systems. Prof.Nahon is an Associate Fellow of the AIAA and a Fellow of the Canadian Aeronautics and Space Institute (CASI). He has received awards from the AIAA and CASI for his work on flight simulator motion systems and on space-based robotics.