



ORIGINAL RESEARCH

A unified control strategy for autonomous aerial vehicles

Eitan Bulka¹ · Meyer Nahon¹

Received: 18 March 2019 / Accepted: 25 August 2021 / Published online: 9 September 2021
 © Crown 2021

Abstract

Unmanned aerial vehicles (UAVs) have become popular in a wide range of applications, including many military and civilian uses. State-of-the-art control strategies for these vehicles are typically tailored to a specific platform and are often limited to a portion of the vehicle’s flight envelope. This article presents a single physics-based controller capable of aggressive maneuvering for the majority of UAVs. The controller is applicable to UAVs with the ability to apply a force along a body-fixed direction, and a moment about an arbitrary axis, which includes UAVs such as multi-copters, conventional fixed-wing, agile fixed-wing, most flying-wings, most tailsitters, some tilt-rotor/wing platforms, and some flapping-wing vehicles. We describe the implementation of this controller on numerous platforms, and demonstrate autonomous flight in outdoor flight tests for a quadrotor and an agile fixed-wing aircraft. To specifically demonstrate the extreme maneuvering capability of the control logic, we perform a rolling flip with the quadrotor and a rolling Harrier and an aggressive turnaround with the fixed-wing aircraft, all using a single controller.

Keywords Aerial robotics · Control · UAV · Aerobatics

List of symbols

A	Closed-loop position error state transition matrix
a^{des}	Desired acceleration
C_{bi}	Direction cosine matrix from \mathcal{F}_i to \mathcal{F}_b
C_{ri}	Direction cosine matrix from \mathcal{F}_i to \mathcal{F}_r
c_j	Control surface constant for the j th actuator
$\hat{\mathbf{d}}_j$	Direction of force for the j th actuator
e_b	Angular error about the body frame axes
f^{aero}	Aerodynamic force
f^c	Control force
f^c	Magnitude of control force ($\ \mathbf{f}^c\ $)
f^{nc}	Non-control force
$\hat{\mathbf{f}}$	Direction of control force ($\frac{\mathbf{f}^c}{f^c}$)
$\hat{\mathbf{f}}^{ref}$	Direction of control force of the reference aircraft
\mathcal{F}_b	Body frame
\mathcal{F}_i	Inertial frame
\mathcal{F}_r	Reference body frame
g	Acceleration due to gravity
g(u^s_j)	Flapping thrust model

Δh	Height error
I	Moment of inertia with respect to center of mass
J_j	Propeller advance ratio for the j th actuator
k_t	Propeller thrust coefficient
k_q	Propeller torque coefficient
K_{ad}	Derivative attitude control gain
K_{ap}	Proportional attitude control gain
K_{hi}	Integral height control gain
K_{hp}	Proportional height control gain
K_{pd}	Derivative position control gain
K_{pp}	Proportional position control gain
K_v	Proportional speed control gain
m	Mass
m^c	Control moment
m^{nc}	Non-control moment
p	Position
p^{ref}	Reference position
Δp	Position error
q	Orientation quaternion
q^{ref}	Reference orientation quaternion
$\bar{\mathbf{q}}^{ref}$	Augmented reference orientation quaternion
q^x	Quaternion rotation of θ_x
q^y	Quaternion rotation of θ_y
q^z	Quaternion rotation of θ_z
Δq	Error quaternion

✉ Eitan Bulka
 eitan.bulka@mail.mcgill.ca
 Meyer Nahon
 meyer.nahon@mcgill.ca

¹ Department of Mechanical Engineering, McGill University, Montreal, QC, Canada

UAV	Unmanned Aerial Vehicle
\mathbf{r}_j	Position vector from the center of mass to the j th actuator
R	Propeller radius
t	Time
\mathbf{u}_j^f	Force generated by the j th actuator
\mathbf{u}^s	Column matrix of actuator signals
u_j^s	Actuator signal for the j th actuator
u_j	Normalized actuator signal for the j th actuator
\mathbf{u}_j^τ	Torque generated by the j th actuator
V	Lyapunov function
\mathbf{v}	Velocity
\mathbf{v}^{ref}	Reference velocity
$\Delta \mathbf{v}$	Velocity error
$v_{s,j}$	Slipstream speed over the j th actuator
VTOL	Vertical Takeoff and Landing Aircraft
x	x -position in \mathcal{F}_i
y	y -position in \mathcal{F}_i
z	z -position in \mathcal{F}_i
μ	Torque cancellation constant
$\boldsymbol{\omega}$	Angular velocity
$\boldsymbol{\omega}^{ref}$	Reference angular velocity
$\boldsymbol{\theta}$	Triad of rotations for position control ($[\theta_x \ \theta_y \ \theta_z]$)
ρ	Air density
ϕ	Roll
θ	Pitch
ψ	Yaw
γ	Wing tilt angle
\square_b	Resolved in the body frame
\square_i	Resolved in the inertial frame
\square_r	Resolved in the reference body frame
\square_x	The x component of a vector
\square_y	The y component of a vector
\square_z	The z component of a vector
\square_{prev}	Value at the previous time step
\square_0	Scalar part of quaternion
$\square_{1:3}$	Vector part of quaternion
\odot	Hamilton quaternion product
\square^T	Transpose
\square^*	Conjugate
$\dot{\square}$	Time derivative

1 Introduction

Furthering the autonomous capabilities of unmanned aerial vehicles (UAVs) has become a popular research topic due to their many civilian and military applications. In practice, the ideal UAV platform is dependent on the mission, as UAV designs typically trade off maneuverability and flight efficiency, as shown in Fig. 1. For missions in confined cluttered spaces, rotor-craft are the ideal platform. For missions requir-

ing long range flight in uncluttered environments, fixed-wing aircraft are the ideal platform. In missions requiring both long range flight and flight in confined cluttered spaces, agile fixed-wing aircraft or tailsitters may be the most suitable platform.

Over the past decade, researchers have developed control algorithms to automatically track agile trajectories, using both rotor-craft and fixed-wing aircraft, as well as more recently developed hybrid-type platforms such as tailsitter aircraft. Autonomous aggressive quadrotor flight is achieved in [17], where the quadrotor autonomously flies through vertical windows and perches. The flight controller is decomposed into discrete phases, where each phase has a local controller consisting of an outer-loop PID position controller, and an inner-loop PD attitude tracker. In [5], a quaternion-based PD attitude control algorithm is presented for a quadrotor which can recover from collisions with walls.

Autonomous agile flight with a tailsitter is achieved in [21], by demonstrating a level flight to hover transition using a physics-based cascaded control algorithm. The level flight to hover transition has also been demonstrated autonomously with agile fixed-wing aircraft by using a PD control law in [9], a nonlinear Lyapunov backstepping controller in [23] and an LQR-Trees controller in [18]. In [19], a control system is developed that can perform aerobatic maneuvers along a path. The elevator and rudder are used to track an acceleration command using a PI control law, while the roll can be selected independently as it is decoupled from path tracking. This decoupling allows for rolling harrier (constant non-zero roll-rate) or knife-edge (90° roll) flight.

These state-of-the-art control strategies are all tailored to a specific platform, and many of them are tailored to a specific maneuver. Use of these strategies would require a different algorithm for each platform; and even on the same platform, many approaches require separate controllers and tuning for each maneuver. While modern autopilot hardware/software systems, such as the Pixhawk with PX4 are able to control a multitude of vehicles in conventional flight regimes, they do so using different control modules for multicopter, fixed-wing and VTOL vehicles; and these modules use a different control structure for each type of vehicle.

In this work, we propose a *single* control strategy which is capable of performing a wide range of maneuvers and can be applied to a wide range of platforms. A universal controller has many advantages; including portability between different platforms. Using a single controller for many maneuvers is also advantageous for agile maneuvering; as the transition between maneuvers is easier to achieve when the same controller is used in all flight modes—controller switching is often difficult to achieve smoothly.

The goal of a UAV control system, irrespective of the platform, is to track position and orientation (six degrees of freedom). The control strategy is strongly dependent on the

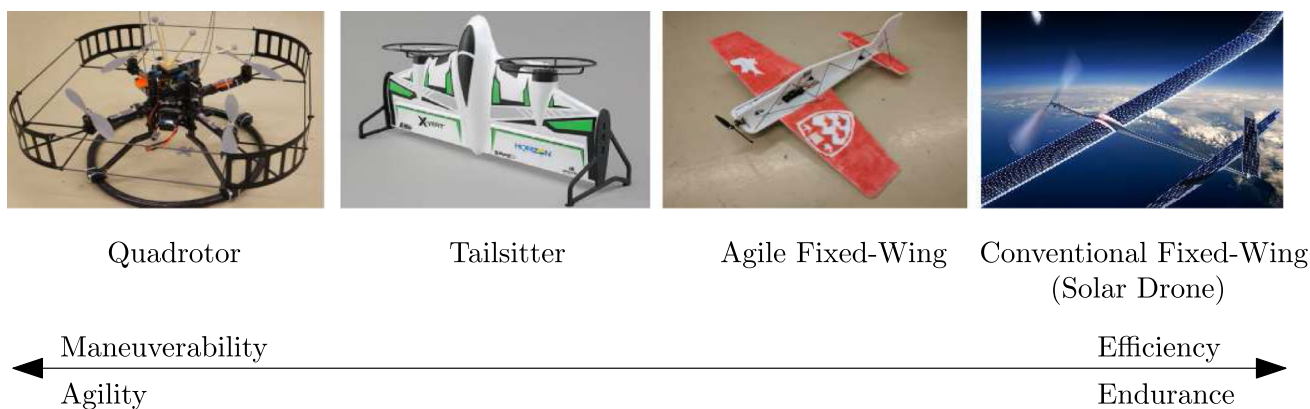


Fig. 1 Types of unmanned aerial vehicles

UAV's actuators, which vary with each platform. However, for the majority of UAVs, the final effect of the actuators is the *same*: they produce a force along a body-fixed axis, and moments about three linearly independent axes, which is equivalent to a moment about an arbitrary axis. The reason for this similarity is most applications require complete control of the UAV orientation, thus requiring the ability to exert a moment about an arbitrary axis. The UAV requires the ability to generate a force in order to counteract gravity, and accelerate the vehicle in a desired direction. Although the control problem would be simplified if this force could be directed in an arbitrary manner, achieving this would require additional actuators, which would in turn add weight and cost to the platform. In order to save this weight, most UAV platforms are built with one body-fixed direction of force, and the under-actuated system controls position by re-orienting itself to re-direct this force. It is for this class of UAVs for which our controller is applicable, which includes multi-copters, conventional fixed-wing, agile fixed-wing, most flying-wings, most tailsitters, some tilt-rotor/wing platforms, and some flapping-wing vehicles.

In [10], a generalized control strategy is developed for the class of vehicles with the ability to generate a body-fixed force, and three linearly independent moments. However, this strategy specifically assumes the vehicle generates small lift and drag forces. This assumption causes the strategy to be applicable for most wingless VTOL vehicles such as multi-rotors, but *not* applicable for winged vehicles such as fixed-wing aircraft or tailsitters.

A continuation of this approach is presented in [20], in which the controller design does not assume small lift and drag forces. The authors present a controller that aims to exert a desired force by using angular velocity as a control input to align the body-fixed thrust with the direction of the desired force. This desired force is calculated based on a reference acceleration, position and velocity feedback, gravity, and aerodynamic forces. A limitation to this approach is that

it is designed for axisymmetric aerial vehicles, which exclude vehicles with asymmetric body shapes such as a conventional fixed-wing aircraft.

This approach is continued further in [11], which allows the control logic to be applied to fixed-wing aircraft, and is successfully implemented in a simulation environment. While this work presents a methodology towards a unified control approach in theory, it is unclear whether this method would work as well in practice. The controller feedback is based on the angle of attack; which is difficult to sense and estimate. In addition, the simulation control inputs are an applied torque and thrust. Although aircraft control surfaces effectively apply torque, it is difficult to know the magnitude of this torque, and thus in a hardware implementation, it's likely that the torque to control surface mapping is not accurate.

In [4], we presented a control architecture that can be applied to any aerial system capable of producing a moment about an arbitrary axis and a force along a body-fixed axis. We specifically demonstrated how it would be applied to a quadrotor and an agile fixed-wing aircraft, and validated the approach in simulation. In this work, in addition to a quadrotor and agile fixed-wing aircraft, we demonstrate how this approach could be applied to a flapping-wing UAV, a tailsitter aircraft, and a tilt-wing aircraft. We also verify this unified control approach with experiments; specifically on a quadrotor (see Fig. 3) and an agile fixed-wing platform (see Fig. 4). We perform both simple and aggressive maneuvers with both platforms using a single control strategy. This work presents a novel unified control framework that can be applied to a wide range of unmanned aerial vehicles, and to the best of our knowledge, this is the only control architecture that can be applied to many types of UAVs that has been verified experimentally.

2 Modeling

Three coordinate systems are used in this paper to describe the controller and the dynamics of a UAV. \mathcal{F}_i is the ground-fixed inertial frame with north-east-down basis vectors; \mathcal{F}_b is the body-fixed frame; and a third frame, \mathcal{F}_r , is the coordinate system fixed to the *reference* orientation of the UAV, called the reference body frame, and will be used in the controller methodology presented in Sect. 3.

The translational and rotational dynamics of the UAVs of interest can be derived from the Newton-Euler equations and Poisson’s equation for a single rigid body, which can be succinctly stated in the following first-order form:

$$\dot{\mathbf{p}}_i = \mathbf{C}_{bi}^T \mathbf{v}_b \tag{1}$$

$$\dot{\mathbf{v}}_b = \frac{1}{m} (\mathbf{f}_b^{nc} + \mathbf{f}_b^c) - \boldsymbol{\omega}_b \times \mathbf{v}_b \tag{2}$$

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{q} \odot \boldsymbol{\omega}_b \tag{3}$$

$$\dot{\boldsymbol{\omega}}_b = \mathbf{I}_b^{-1} ((\mathbf{m}_b^{nc} + \mathbf{m}_b^c) - \boldsymbol{\omega}_b \times \mathbf{I}_b \boldsymbol{\omega}_b) \tag{4}$$

where \odot is the Hamilton quaternion product, \mathbf{p}_i is the absolute position of the UAV centre of mass expressed as components in \mathcal{F}_i (designated with subscript i) and \mathbf{q} is the aircraft orientation, expressed as a unit quaternion, $\mathbf{q} = [q_0, \mathbf{q}_{1:3}^T]^T$. Analogously at the velocity level, \mathbf{v}_b is the translational velocity of the centre of mass and $\boldsymbol{\omega}_b$ is the rotational velocity of the UAV, both expressed in the body-fixed frame (designated with subscript b). The rotation matrix \mathbf{C}_{bi} describes the orientation of the body frame relative to the inertial frame; m is the mass of the UAV, and \mathbf{I}_b is the second moment of mass of the UAV relative to the center of mass, in the body frame. The cumulative forces and moments acting on the UAV are separated into a control and non-control force, \mathbf{f}_b^c and \mathbf{f}_b^{nc} , and a control and non-control moment, \mathbf{m}_b^c and \mathbf{m}_b^{nc} . The control force is caused by a thrust from either a propeller and or flapping-wing, whereas the non-control force stems from gravity, lift, and drag. The control moment can be generated by thrusts being applied far from the center of mass, propeller drag torque, and control surfaces, whereas the non-control moments can be caused by propeller gyroscopic effects, lift, and drag. Given we are specifically designing our controller for unmanned aerial vehicles capable of exerting a *body-fixed* force, we can further simplify the control force equation:

$$\mathbf{f}_b^c = f^c \hat{\mathbf{f}}_b \tag{5}$$

where $\hat{\mathbf{f}}_b$ is a unit vector in the direction of the body-fixed force, resolved in the body frame, and f^c is the magnitude of the control force.

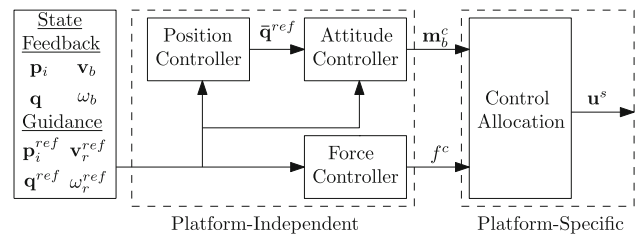


Fig. 2 Control architecture

3 Controller

We design a non-linear controller based on the underlying physics of the UAV while avoiding any plant linearization approximations. This allows simple control laws to achieve good performance throughout the entire flight envelope of the UAV. The control logic is comprised of two parts. The first stage determines the control moment (along an arbitrary axis) and force (along a body-fixed axis) needed to track the reference trajectory of the UAV, and the second stage allocates the control moment and force to UAV-specific actuator signals. The architecture is modular, containing a position controller, a force controller, an attitude controller, and a control allocation block. This modular structure is shown by the block diagram in Fig. 2.

We assume a state estimator provides pose and twist estimates of the aircraft, and a trajectory generation algorithm specifies a reference pose and twist (\mathbf{p}_i^{ref} , \mathbf{q}^{ref} , \mathbf{v}_r^{ref} , and $\boldsymbol{\omega}_r^{ref}$) to be tracked. First, the position controller uses the linear position and velocity errors to modify the reference orientation. This new augmented orientation, $\bar{\mathbf{q}}^{ref}$, is similar to the reference orientation \mathbf{q}^{ref} , but modified to correct translational motion errors. The attitude controller then generates control moments that track this augmented orientation. The force control is decoupled from above and its goal is to counteract gravity and the aerodynamic forces, as well as track the height and velocity of the UAV. The control allocation is achieved by determining the forces and moments that are exerted on the UAV given a set of actuator commands (\mathbf{u}^s), and then this relationship is inverted to obtain a set of actuator commands that apply the commanded control moment and force.

3.1 Position controller

The position controller augments the reference attitude of the UAV in order to redirect the body-fixed force to a direction that reduces translational errors. While there are many ways of achieving this, we augment our reference attitude by performing three successive rotations. This triad of rotations is computed using the following equation:

$$\boldsymbol{\Theta} = \hat{\mathbf{f}}_r^{ref} \times (K_{pp} \mathbf{C}_{ri} (\mathbf{p}_i^{ref} - \mathbf{p}_i) + K_{pd} (\mathbf{v}_r^{ref} - \mathbf{C}_{ri} \mathbf{C}_{bi}^T \mathbf{v}_b)) \tag{6}$$

where the direction of body-fixed force associated with the reference orientation is denoted by $\hat{\mathbf{f}}_r^{ref}$, and the PD control law on translational motion are both resolved in the body frame associated with the reference attitude, \mathcal{F}_r . Note that $\hat{\mathbf{f}}_r^{ref}$ and $\hat{\mathbf{f}}_b$ are component-wise equal. The proportional and derivative gains are denoted by K_{pp} and K_{pd} respectively. Equation (6) uses the direction cosine matrices from inertial to body, \mathbf{C}_{bi} , and inertial to reference, \mathbf{C}_{ri} , which are formed from the attitude, \mathbf{q} , and reference attitude, \mathbf{q}^{ref} , respectively.

The gains K_{pp} and K_{pd} are chosen to be small enough that typical errors in position and velocity lead to the components of $\frac{1}{2}\boldsymbol{\Theta}$, $(\frac{\Theta_x}{2}, \frac{\Theta_y}{2}, \frac{\Theta_z}{2})$, being small angles. In the chance that any component of $\frac{1}{2}\boldsymbol{\Theta}$ becomes large, we limit each component of $\boldsymbol{\Theta}$ to 45° to ensure the half of each component can be considered small. Without this limitation, very large position errors could cause the components of $\boldsymbol{\Theta}$ to become so large that the augmented orientation no longer points the thruster in a direction that reduces errors in position.

We form three quaternion rotations from Θ_x , Θ_y and Θ_z :

$$\mathbf{q}^x = \left[\cos \frac{\Theta_x}{2}, \sin \frac{\Theta_x}{2}, 0, 0 \right]^T \tag{7a}$$

$$\mathbf{q}^y = \left[\cos \frac{\Theta_y}{2}, 0, \sin \frac{\Theta_y}{2}, 0 \right]^T \tag{7b}$$

$$\mathbf{q}^z = \left[\cos \frac{\Theta_z}{2}, 0, 0, \sin \frac{\Theta_z}{2} \right]^T \tag{7c}$$

We can then form our augmented reference orientation by performing three successive rotations of the original reference orientation:

$$\bar{\mathbf{q}}^{ref} = \mathbf{q}^{ref} \odot \mathbf{q}^z \odot \mathbf{q}^y \odot \mathbf{q}^x \tag{8}$$

which can be interpreted as taking the reference orientation, and then subsequently rotating it about the z axis of \mathcal{F}_r , and then rotating by intermediary y axis, and then an intermediary x axis. The order of rotations affect the outcome, and thus this order is chosen based on the body frame definitions discussed in Sect. 4.

While this augmented reference orientation could be computed in various other ways, such as treating $\boldsymbol{\Theta}$ as an axis-angle rotation, we elect to use three successive rotations in order to keep the augmented reference orientation close to the original reference orientation. For any UAV for which this controller is applicable, there is part of the attitude that does not affect the thrust direction, such as the roll of a fixed-wing aircraft, or the yaw of quadrotor. Ideally, the position controller wouldn't affect this portion of the augmented reference attitude. Performing these three successive rotations

has a smaller effect on that portion of the augmented reference attitude then say, treating $\boldsymbol{\Theta}$ as an axis-angle rotation.

Ultimately, these rotations redirect the thrust in order to reduce linear position and velocity error that are not along the force axis; the errors along the force axis are corrected by the force controller discussed in Sect. 3.2. This approach is generalized to an arbitrary thrust axis, making it suitable for quadrotors, tailsitters, agile aircraft, or even tilt rotors undergoing transitions. In addition, this methodology has no limitations on the actual or reference orientation, and remains valid for *any* orientation of the UAV.

Another advantage to this approach is the modularity of the architecture. The modularity makes it very easy to turn on and off position control (by setting the gains to zero). This is advantageous for gain tuning, as it is easier to first focus on tuning the attitude controller without use of the position controller, and once the attitude tracker is satisfactory the position controller can then be tuned. Another case where the ability to easily turn off the position controller is advantageous is in extreme maneuvering. Say the higher level goal of a maneuver is to perform a flip, and the position of the UAV is irrelevant, one could simply turn off the position controller during the flip to achieve better attitude tracking. The last scenario where the ability to turn off the position controller is advantageous is in a manual/pilot assist mode, where a pilot is the 'position controller' and specifies the augmented reference attitude with a joystick, and then the inner-loop attitude controller can still track this manually generated augmented reference attitude.

3.2 Force controller

The control force is chosen such that it cancels the aerodynamic and gravity forces, and accelerates the vehicle according to

$$\mathbf{a}_b^{des} = K_v(\mathbf{C}_{bi}\mathbf{C}_{ri}^T\mathbf{v}_r^{ref} - \mathbf{v}_b) + K_{hp}\Delta\mathbf{h}_b + K_{hi} \int \Delta\mathbf{h}_b dt \tag{9}$$

where

$$\Delta\mathbf{h}_b = \mathbf{C}_{bi}[0, 0, (p_{i,z}^{ref} - p_{i,z})]^T \tag{10}$$

The desired acceleration is based on a proportional control law on the velocity error, using proportional velocity gain K_v , and a proportional-integral control law on the height error, with the proportional and integral height gains denoted by K_{hp} and K_{hi} respectively.

We compute the control force such that it will counteract gravity, counteract the aerodynamic forces (lift and drag), as well accelerate the vehicle according to Eq. (9). The magnitude of the control force, f^c , which is applied along the

body-fixed force axis, is calculated as follows:

$$f^c = (-m\mathbf{C}_{bi}\mathbf{g}_i - \mathbf{f}_b^{aero} + m\mathbf{a}_b^{des})^T \hat{\mathbf{f}}_b \tag{11}$$

where the UAV mass is denoted by m , the acceleration due to gravity, expressed in the inertial frame is denoted by \mathbf{g}_i , and the aerodynamic force is denoted by \mathbf{f}_b^{aero} , which is a function of the UAV’s geometry and current velocity. A curve fit of this force as a function of velocity can be approximated from a simulation environment for winged vehicles, and can be approximated as zero or a pure drag force for wingless vehicles. Ultimately, the desired total force in the parentheses in Eq. (11) would be the desired control force if force could be generated in any direction, however, since it can only be generated along the body-fixed force axis, $\hat{\mathbf{f}}_b$, the commanded control force, f^c , is the projection of the desired total force onto this axis.

3.3 Attitude controller

The goal of the attitude controller is to compute a control moment that will track the augmented reference orientation output from the position controller. The attitude controller computes an error quaternion which is used to obtain angular errors about the body axes, which are in turn, mapped to desired moments.

The error quaternion, $\Delta\mathbf{q}$, is calculated by premultiplying the augmented reference quaternion, $\bar{\mathbf{q}}^{ref}$, by the attitude quaternion conjugate, \mathbf{q}^* :

$$\Delta\mathbf{q} = \mathbf{q}^* \odot \bar{\mathbf{q}}^{ref} \tag{12}$$

The error quaternion is then converted to angular errors about the body frame axes, \mathbf{e}_b , using:

$$\mathbf{e}_b = \begin{cases} 2 \cos^{-1}(\Delta q_0) \frac{\Delta\mathbf{q}_{1:3}}{\|\Delta\mathbf{q}_{1:3}\|}, & \|\Delta\mathbf{q}_{1:3}\| \neq 0 \\ \mathbf{0}, & \|\Delta\mathbf{q}_{1:3}\| = 0 \end{cases} \tag{13}$$

where $\Delta\mathbf{q}_{1:3}$ refers to the vector part of the error quaternion, and Δq_0 refers to the scalar component. To ensure the angular errors remain less than 180° , if $\Delta q_0 < 0$, then $-\Delta\mathbf{q}$ is used instead of $\Delta\mathbf{q}$ in Eq. (13).

A PD controller is used to calculate the moment needed to correct the angular error. Two gain matrices, \mathbf{K}_{ap} and \mathbf{K}_{ad} , are used to map an angular error to a desired angular acceleration, which is multiplied by the inertia matrix to obtain a desired control moment about each axis. This multiplication by the inertia matrix could be removed and factored into the control gains, but it allows for easier transitioning to different platforms while using a similar set of control gains. The

control moment is calculated by:

$$\mathbf{m}_b^c = \mathbf{I}_b(\mathbf{K}_{ap}\mathbf{e}_b + \mathbf{K}_{ad}(\mathbf{C}_{bi}\mathbf{C}_{ri}^T\boldsymbol{\omega}_r^{ref} - \boldsymbol{\omega}_b)) + \mu(\boldsymbol{\omega}_b \times \mathbf{I}_b\boldsymbol{\omega}_b) \tag{14}$$

where \mathbf{m}_b^c is the control moment to be applied by the aircraft’s actuators. The second part of Eq. (14) can precisely cancel the gyroscopic coupling torque by setting $\mu = 1$. However in practice, whether $\mu = 0$ or $\mu = 1$ there is little difference due to the relatively small inertia tensor and angular velocity values.

A stability analysis of the attitude controller is shown in Sect. 3.4.1, and a similar attitude controller presented in [25], is shown to be globally stable.

3.4 Stability analysis

Since a UAV has four control inputs, it is only possible to achieve asymptotic tracking for at most four output states. This leads to most researchers developing controllers for either position tracking flight modes (and one attitude state), or attitude tracking flight modes (and one position state) [14]. However, there are many circumstances where it is beneficial to forego the asymptotic tracking of four states in order to have some control over all six states (the entire position and attitude). Consider a conventional fixed wing aircraft, the ultimate goal is to control position, however, there are critical aspects of the attitude that must be controlled, such as avoiding stall, or flying with an inefficient orientation. In this scenario, foregoing asymptotic tracking of position, for the ability to avoid stalling, is worthwhile if the controller can still keep the tracking errors within certain bounds. By contrast, asymptotic tracking of attitude while having no position control does not make sense for a conventional fixed-wing aircraft. Aerobatic flight is another example where controlling all six states is necessary, since the desired maneuver is a trajectory in both attitude and position. For these reasons, we develop a controller that tries to track all six states, where the controller constantly trades off between position and attitude tracking. While it is not possible to achieve asymptotic tracking of all six states, we are able to demonstrate the attitude controller is globally asymptotically stable with regards to regulation of the *augmented* reference orientation, which is the original reference orientation, but modified to control position. For the position error stability analysis we assume the aircraft attitude is the augmented reference attitude. This assumption is based on the fast rotational and slow translational dynamics of aircraft [22]. Under this assumption we show the position errors are Lyapunov or asymptotically stable, depending on the reference orientation and platform.

3.4.1 Attitude

We recall the attitude kinematics as

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_0 \\ \dot{\mathbf{q}}_{1:3} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_0 \\ \mathbf{q}_{1:3} \end{bmatrix} \odot \begin{bmatrix} 0 \\ \boldsymbol{\omega}_b \end{bmatrix}. \tag{15}$$

and the error quaternion defined as

$$\Delta \mathbf{q} = \mathbf{q}^* \odot \bar{\mathbf{q}}^{ref}. \tag{16}$$

We can obtain our error quaternion kinematics by taking the time derivative of both sides of the equation

$$\Delta \dot{\mathbf{q}} = \dot{\mathbf{q}}^* \odot \bar{\mathbf{q}}^{ref} + \mathbf{q}^* \odot \dot{\bar{\mathbf{q}}}^{ref}. \tag{17}$$

We show the stability analysis for regulation, so $\dot{\bar{\mathbf{q}}}^{ref} = \mathbf{0}$. By simplifying and substituting Eq. (15) into Eq. (17) we obtain

$$\Delta \dot{\mathbf{q}} = \frac{1}{2} \left(\begin{bmatrix} q_0 \\ \mathbf{q}_{1:3} \end{bmatrix} \odot \begin{bmatrix} 0 \\ \boldsymbol{\omega}_b \end{bmatrix} \right)^* \odot \bar{\mathbf{q}}^{ref}. \tag{18}$$

Using the property for two quaternions \mathbf{a} and \mathbf{b} , $(\mathbf{a} \odot \mathbf{b})^{-1} = \mathbf{b}^{-1} \odot \mathbf{a}^{-1}$, and for unit quaternions, $(\mathbf{a} \odot \mathbf{b})^* = \mathbf{b}^* \odot \mathbf{a}^*$, Eq. (18) becomes

$$\Delta \dot{\mathbf{q}} = \frac{1}{2} \begin{bmatrix} 0 \\ -\boldsymbol{\omega}_b \end{bmatrix} \odot \begin{bmatrix} q_0 \\ \mathbf{q}_{1:3} \end{bmatrix}^* \odot \bar{\mathbf{q}}^{ref}. \tag{19}$$

Recalling our error quaternion definition from Eq. (16), Eq. (19) becomes

$$\Delta \dot{\mathbf{q}} = -\frac{1}{2} \begin{bmatrix} 0 \\ \boldsymbol{\omega}_b \end{bmatrix} \odot \Delta \mathbf{q}, \tag{20}$$

and by multiplying through becomes

$$\Delta \dot{\mathbf{q}} = \begin{bmatrix} \Delta \dot{q}_0 \\ \Delta \dot{\mathbf{q}}_{1:3} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \boldsymbol{\omega}_b^T \Delta \mathbf{q}_{1:3} \\ -\Delta q_0 \boldsymbol{\omega}_b - \boldsymbol{\omega}_b \times \Delta \mathbf{q}_{1:3} \end{bmatrix}. \tag{21}$$

Turning our attention to the attitude dynamics, we recall the rotational equations of motion, subject to no disturbances, as

$$\mathbf{I}_b \dot{\boldsymbol{\omega}}_b = \mathbf{m}_b^c - \boldsymbol{\omega}_b \times \mathbf{I}_b \boldsymbol{\omega}_b \tag{22}$$

where our control torque \mathbf{m}_b^c comes from Eqs. (13 and 14). For the regulation task, $\boldsymbol{\omega}_r^{ref} = \mathbf{0}$, and the control torque becomes

$$\mathbf{m}_b^c = \mathbf{I}_b (\mathbf{K}_{ap} 2 \cos^{-1}(\Delta q_0) \frac{\Delta \mathbf{q}_{1:3}}{\|\Delta \mathbf{q}_{1:3}\|} - \mathbf{K}_{ad} \boldsymbol{\omega}_b) + \mu (\boldsymbol{\omega}_b \times \mathbf{I}_b \boldsymbol{\omega}_b). \tag{23}$$

We can replace $\|\Delta \mathbf{q}_{1:3}\|$ with $\sqrt{1 - \Delta q_0^2}$ in Eq. (23) since $\Delta \mathbf{q}$ is a unit quaternion, which can be substituted into Eq. (22) to obtain the closed-loop attitude dynamics

$$\mathbf{I}_b \dot{\boldsymbol{\omega}}_b = \mathbf{I}_b (\mathbf{K}_{ap} 2 \cos^{-1}(\Delta q_0) \frac{\Delta \mathbf{q}_{1:3}}{\sqrt{1 - \Delta q_0^2}} - \mathbf{K}_{ad} \boldsymbol{\omega}_b) + (\mu - 1) (\boldsymbol{\omega}_b \times \mathbf{I}_b \boldsymbol{\omega}_b). \tag{24}$$

We consider a Lyapunov function candidate

$$V = \frac{1}{2} \boldsymbol{\omega}_b^T (\mathbf{I}_b \mathbf{K}_{ap})^{-1} \mathbf{I}_b \boldsymbol{\omega}_b + 2(\cos^{-1}(\Delta q_0))^2 \tag{25}$$

At the equilibrium point, $\Delta \mathbf{q} = [1 \ 0 \ 0 \ 0]^T$ and $\boldsymbol{\omega}_b = \mathbf{0}$, $V = 0$. We require $\mathbf{K}_{ap} > 0$, which makes $V > 0$ for all error quaternion and angular velocity combinations besides the equilibrium point. We can then take the time derivative of the Lyapunov function candidate and obtain

$$\dot{V} = \boldsymbol{\omega}_b^T (\mathbf{I}_b \mathbf{K}_{ap})^{-1} \mathbf{I}_b \dot{\boldsymbol{\omega}}_b - \frac{4 \cos^{-1}(\Delta q_0)}{\sqrt{1 - \Delta q_0^2}} \Delta \dot{q}_0. \tag{26}$$

We can substitute Eqs. (21 and 24) into Eq. (26) to obtain

$$\begin{aligned} \dot{V} = & \boldsymbol{\omega}_b^T (\mathbf{I}_b \mathbf{K}_{ap})^{-1} \left(\mathbf{I}_b (\mathbf{K}_{ap} 2 \cos^{-1}(\Delta q_0) \frac{\Delta \mathbf{q}_{1:3}}{\sqrt{1 - \Delta q_0^2}} \right. \\ & \left. - \mathbf{K}_{ad} \boldsymbol{\omega}_b \right) + (\mu - 1) (\boldsymbol{\omega}_b \times \mathbf{I}_b \boldsymbol{\omega}_b) - \frac{4 \cos^{-1}(\Delta q_0)}{\sqrt{1 - \Delta q_0^2}} \\ & \times \left(\frac{1}{2} \boldsymbol{\omega}_b^T \Delta \mathbf{q}_{1:3} \right), \end{aligned} \tag{27}$$

which simplifies to

$$\begin{aligned} \dot{V} = & -\boldsymbol{\omega}_b^T (\mathbf{I}_b \mathbf{K}_{ap})^{-1} \mathbf{I}_b \mathbf{K}_{ad} \boldsymbol{\omega}_b + \boldsymbol{\omega}_b^T (\mathbf{I}_b \mathbf{K}_{ap})^{-1} (\mu - 1) \\ & \times (\boldsymbol{\omega}_b \times \mathbf{I}_b \boldsymbol{\omega}_b) + \boldsymbol{\omega}_b^T (\mathbf{I}_b \mathbf{K}_{ap})^{-1} (\mathbf{I}_b \mathbf{K}_{ap}) 2 \cos^{-1}(\Delta q_0) \\ & \times \frac{\Delta \mathbf{q}_{1:3}}{\sqrt{1 - \Delta q_0^2}} - \frac{4 \cos^{-1}(\Delta q_0)}{\sqrt{1 - \Delta q_0^2}} \left(\frac{1}{2} \boldsymbol{\omega}_b^T \Delta \mathbf{q}_{1:3} \right), \end{aligned} \tag{28}$$

which simplifies to

$$\begin{aligned} \dot{V} = & -\omega_b^T \mathbf{K}_{ap}^{-1} \mathbf{I}_b^{-1} \mathbf{I}_b \mathbf{K}_{ad} \omega_b + \omega_b^T (\mathbf{I}_b \mathbf{K}_{ap})^{-1} (\mu - 1) \\ & \times (\omega_b \times \mathbf{I}_b \omega_b) + \frac{2 \cos^{-1}(\Delta q_0)}{\sqrt{1 - \Delta q_0^2}} (\omega_b^T \Delta \mathbf{q}_{1:3} - \omega_b^T \Delta \mathbf{q}_{1:3}). \end{aligned} \tag{29}$$

If we set $\mu = 1$ we get a precise cancellation of the gyroscopic coupling torque. Alternatively, this term could go to zero by removing the multiplication of \mathbf{I}_b in Eq. (23), and forcing \mathbf{K}_{ap}^{-1} to be a linear combination of \mathbf{I}_b and the identity matrix, which is shown in [25]. We can further simplify our Lyapunov function derivative to

$$\dot{V} = -\omega_b^T \mathbf{K}_{ap}^{-1} \mathbf{K}_{ad} \omega_b. \tag{30}$$

As we can see in Eq. (30), if we require $\mathbf{K}_{ap}^{-1} \mathbf{K}_{ad} > 0$ our Lyapunov function derivative is negative semi-definite, showing the attitude errors are globally stable. We can show these errors are globally asymptotically stable using Lasalle’s invariant set theorem.

We solve for the error states when $\dot{V} = 0$:

$$\dot{V} = -\omega_b^T \mathbf{K}_{ap}^{-1} \mathbf{K}_{ad} \omega_b = 0 \Rightarrow \omega_b = \mathbf{0} \Rightarrow \dot{\omega}_b = \mathbf{0}$$

Since $\dot{V} = 0$ implies $\omega_b = \mathbf{0}$ and $\dot{\omega}_b = \mathbf{0}$, then when $\dot{V} = 0$ the closed-loop attitude dynamics, represented by Eq. (24), simplifies to

$$\mathbf{0} = \mathbf{I}_b \mathbf{K}_{ap} 2 \cos^{-1}(\Delta q_0) \frac{\Delta \mathbf{q}_{1:3}}{\sqrt{1 - \Delta q_0^2}}. \tag{31}$$

Since $\mathbf{I}_b > 0$ and $\mathbf{K}_{ap} > 0$, for this equation to hold $\Delta \mathbf{q} = [1 \ 0 \ 0 \ 0]^T$.

Applying Lasalle’s invariance principle, since $\dot{V} \leq 0$ for all $\Delta \mathbf{q}$, ω_b , and the only solution to $\dot{V} = 0$ exists when $\Delta \mathbf{q} = [1 \ 0 \ 0 \ 0]^T$ and $\omega_b = \mathbf{0}$, then we can conclude $\Delta \mathbf{q} = [1 \ 0 \ 0 \ 0]^T$ and $\omega_b = \mathbf{0}$ is an asymptotically stable equilibrium point. We can combine this with our Lyapunov analysis to conclude the attitude errors are globally asymptotically stable.

3.4.2 Position

We now turn over attention to the stability analysis of position and velocity errors. First we define our position error, resolved in the inertial frame, as

$$\Delta \mathbf{p}_i = \mathbf{p}_i^{ref} - \mathbf{p}_i \tag{32}$$

and our velocity error, also resolved in the inertial frame, as

$$\Delta \mathbf{v}_i = \Delta \dot{\mathbf{p}}_i = \dot{\mathbf{p}}_i^{ref} - \dot{\mathbf{p}}_i = \mathbf{v}_i^{ref} - \mathbf{v}_i. \tag{33}$$

For our stability analysis we consider constant velocity trajectories, causing the velocity error to propagate according to

$$\Delta \dot{\mathbf{v}}_i = -\dot{\mathbf{v}}_i = -\mathbf{g}_i - \frac{\mathbf{f}_i^{aero}}{m} - \frac{f^c}{m} \hat{\mathbf{f}}_i. \tag{34}$$

The direction, resolved in the inertial frame, in which the aircraft can exert thrust, $\hat{\mathbf{f}}_i$, will depend on the aircraft’s orientation as shown in

$$\hat{\mathbf{f}}_i = \mathbf{C}_{bi}^T \hat{\mathbf{f}}_b. \tag{35}$$

Since aircraft have fast rotational and slow translational dynamics [22], in the position stability analysis we assume the aircraft attitude, represented by both \mathbf{q} and \mathbf{C}_{bi} , is the attitude output from the position controller, the augmented reference attitude, which is represented by both $\bar{\mathbf{q}}^{ref}$ and $\mathbf{C}_{\bar{r}i}$. This assumption causes Eq. (35) to become

$$\hat{\mathbf{f}}_i = \mathbf{C}_{\bar{r}i}^T \hat{\mathbf{f}}_b. \tag{36}$$

We utilize Eqs. (6–8) to compute $\bar{\mathbf{q}}^{ref}$. We can substitute our position and velocity error definitions in Eqs. (32 and 33) into Eq. (6) to obtain

$$\Theta = \hat{\mathbf{f}}_r^{ref} \times \mathbf{C}_{ri} (K_{pp} \Delta \mathbf{p}_i + K_{pd} \Delta \mathbf{v}_i). \tag{37}$$

As mentioned in Sect. 3.1, the gains K_{pp} and K_{pd} are chosen to be small enough that typical errors in position and velocity lead to $\frac{\Theta_x}{2}$, $\frac{\Theta_y}{2}$, and $\frac{\Theta_z}{2}$ being small. This small angle assumption allows us to make the approximations: $\sin \frac{\theta_x}{2} \approx \frac{\theta_x}{2}$, $\sin \frac{\theta_y}{2} \approx \frac{\theta_y}{2}$, $\sin \frac{\theta_z}{2} \approx \frac{\theta_z}{2}$, $\cos \frac{\theta_x}{2} \approx 1$, $\cos \frac{\theta_y}{2} \approx 1$, and $\cos \frac{\theta_z}{2} \approx 1$. These approximations are substituted into Eqs. (7a–7c), which are then substituted into Eq. (8) to obtain $\bar{\mathbf{q}}^{ref}$. The direction cosine matrix $\mathbf{C}_{\bar{r}i}$ can be obtained from $\bar{\mathbf{q}}^{ref}$.

For the remainder of the proof, we must specify the type of UAV and a reference orientation. We consider an agile fixed-wing aircraft in both a level flight and hover reference orientation, followed by a quadrotor in a hover reference orientation. For brevity, we only present analysis using these three examples but we have also analyzed an agile fixed-wing aircraft performing a knife-edge, and more platforms/reference orientations could be analyzed.

3.4.3 Agile fixed-wing level flight

By using a reference orientation of 0° roll, 0° pitch, and 0° yaw, and neglecting higher order terms because of the small K_{pp} and K_{pd} assumption, we can obtain the direction of the

thrust in the inertial frame as

$$\hat{\mathbf{f}}_i = \begin{bmatrix} 1 \\ K_{p_p} \Delta p_y + K_{p_d} \Delta v_y \\ K_{p_p} \Delta p_z + K_{p_d} \Delta v_z \end{bmatrix}, \tag{38}$$

which can be substituted into Eq. (34) to obtain the velocity error dynamics

$$\Delta \dot{\mathbf{v}}_i = \begin{bmatrix} -\frac{f^c + f_x^{aero}}{m} \\ -\frac{f^c}{m} (K_{p_p} \Delta p_y + K_{p_d} \Delta v_y) \\ -g - \frac{f_z^{aero}}{m} - \frac{f^c}{m} (K_{p_p} \Delta p_z + K_{p_d} \Delta v_z) \end{bmatrix}. \tag{39}$$

As we can see, aerodynamic force only acts in the x (drag) and z (lift) directions, because of the 0° roll, 0° pitch, and 0° yaw reference orientation. We assume the reference orientations are chosen such that the lift and weight forces cancel out. The control force, f^c , can be obtained from Eqs. (9 and 11), which simplifies to $f^c = K_v \Delta v_x - f_x^{aero}$ with this reference orientation and neglecting higher-order terms because of the small K_{p_p} and K_{p_d} assumption. We also must realize the control force is the magnitude of an actuator output (usually a propeller) which has a physical limitation, denoted by $f^{c,max}$. This bounds f^c : $0 \leq f^c \leq f^{c,max}$. We can further simplify our velocity error dynamics to

$$\Delta \dot{\mathbf{v}}_i = \begin{bmatrix} -\frac{K_v \Delta v_x}{m} \\ -\frac{f^c}{m} (K_{p_p} \Delta p_y + K_{p_d} \Delta v_y) \\ -\frac{f^c}{m} (K_{p_p} \Delta p_z + K_{p_d} \Delta v_z) \end{bmatrix}. \tag{40}$$

By not substituting the control law for f^c into the y and z components, and allowing it to be any value such that $0 \leq f^c \leq f^{c,max}$, the closed-loop translational dynamics become linear, and can be written in the form

$$\begin{bmatrix} \Delta \dot{\mathbf{p}}_i \\ \Delta \dot{\mathbf{v}}_i \end{bmatrix} = \mathbf{A} \begin{bmatrix} \Delta \mathbf{p}_i \\ \Delta \mathbf{v}_i \end{bmatrix}, \tag{41}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{K_v}{m} & 0 & 0 \\ 0 & -\frac{K_{p_p} f^c}{m} & 0 & 0 & -\frac{K_{p_d} f^c}{m} & 0 \\ 0 & 0 & -\frac{K_{p_p} f^c}{m} & 0 & 0 & -\frac{K_{p_d} f^c}{m} \end{bmatrix}. \tag{42}$$

We can compute the closed-loop eigenvalues of \mathbf{A} and obtain

$$\lambda_1 = 0 \tag{43}$$

$$\lambda_2 = \frac{-K_v}{m} \tag{44}$$

$$\lambda_{3,4} = \frac{-K_{p_d} f^c \pm \sqrt{-f^c(4K_{p_p} m - K_{p_d}^2 f^c)}}{2m} \tag{45}$$

$$\lambda_{5,6} = \frac{-K_{p_d} f^c \pm \sqrt{-f^c(4K_{p_p} m - K_{p_d}^2 f^c)}}{2m} \tag{46}$$

If $-f^c(4K_{p_p} m - K_{p_d}^2 f^c) \leq 0$, then the real part of all eigenvalues of the closed-loop \mathbf{A} matrix are less than or equal to zero. Since $f^c \geq 0$, we need $4K_{p_p} m - K_{p_d}^2 f^c \geq 0$. This leads to $\frac{4K_{p_p} m}{K_{p_d}^2} \geq f^c$, which can be guaranteed if the control gains are chosen such that $\frac{4K_{p_p} m}{K_{p_d}^2} \geq f^{c,max}$. Assuming the gains meet this criterion, the closed-loop system is Lyapunov stable. We could achieve asymptotic stability by including Δp_x and Δp_y in the force controller, but we leave out these terms so the aircraft can achieve path following as opposed to position tracking.

3.4.4 Agile fixed-wing hover

We can go through the same steps in hovering flight as done in level flight. For hover we specify a reference orientation of 0° roll, 90° pitch, and 0° yaw. By neglecting higher order terms because of the small K_{p_p} and K_{p_d} assumption we can obtain the direction of the thrust in the inertial frame as

$$\hat{\mathbf{f}}_i = \begin{bmatrix} K_{p_p} \Delta p_x + K_{p_d} \Delta v_x \\ K_{p_p} \Delta p_y + K_{p_d} \Delta v_y \\ -1 \end{bmatrix}, \tag{47}$$

which can be substituted into Eq. (34) to obtain the velocity error dynamics

$$\Delta \dot{\mathbf{v}}_i = \begin{bmatrix} -\frac{f^c}{m} (K_{p_p} \Delta p_x + K_{p_d} \Delta v_x) \\ -\frac{f^c}{m} (K_{p_p} \Delta p_y + K_{p_d} \Delta v_y) \\ \frac{f^c}{m} - g \end{bmatrix}. \tag{48}$$

As we can see, there is no aerodynamic force (lift and drag) while hovering, because the aircraft is stationary. As done previously, the control force can be obtained from Eqs. (9 and 11), which simplifies to $f^c = mg - K_{h_p} \Delta p_z - K_v \Delta v_z$ with this hover reference orientation and neglecting higher-order terms because of the small K_{p_p} and K_{p_d} assumption. We can further simplify our velocity error dynamics to

$$\Delta \dot{\mathbf{v}}_i = \begin{bmatrix} -\frac{f^c}{m} (K_{p_p} \Delta p_x + K_{p_d} \Delta v_x) \\ -\frac{f^c}{m} (K_{p_p} \Delta p_y + K_{p_d} \Delta v_y) \\ -\frac{1}{m} (K_{h_p} \Delta p_z + K_v \Delta v_z) \end{bmatrix}. \tag{49}$$

Similar to the level flight analysis, by not substituting the control law for f^c into the x and y components, and allowing it to be any value such that $0 \leq f^c \leq f^{c,max}$, the closed-loop

translational dynamics become linear, and can be written in the form

$$\begin{bmatrix} \Delta \dot{\mathbf{p}}_i \\ \Delta \dot{\mathbf{v}}_i \end{bmatrix} = \mathbf{A} \begin{bmatrix} \Delta \mathbf{p}_i \\ \Delta \mathbf{v}_i \end{bmatrix} \tag{50}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -\frac{K_{pp} f^c}{m} & 0 & 0 & -\frac{K_{pd} f^c}{m} & 0 & 0 \\ 0 & -\frac{K_{pp} f^c}{m} & 0 & 0 & -\frac{K_{pd} f^c}{m} & 0 \\ 0 & 0 & -\frac{K_{hp}}{m} & 0 & 0 & -\frac{K_v}{m} \end{bmatrix} \tag{51}$$

We can compute the closed-loop eigenvalues of \mathbf{A} and obtain

$$\lambda_{1,2} = \frac{-K_v f^c \pm \sqrt{K_v^2 - 4K_{hp} m}}{2m} \tag{52}$$

$$\lambda_{3,4} = \frac{-K_{pd} f^c \pm \sqrt{-f^c(4K_{pp} m - K_{pd}^2 f^c)}}{2m} \tag{53}$$

$$\lambda_{5,6} = \frac{-K_{pd} f^c \pm \sqrt{-f^c(4K_{pp} m - K_{pd}^2 f^c)}}{2m} \tag{54}$$

Following the same analysis as the level flight case, we require $\frac{4K_{pp} m}{K_{pd}^2} \geq f^{c,max}$. In addition, $K_v^2 - 4K_{hp} m \leq 0$, which implies $\frac{K_v^2}{4m} \leq K_{hp}$. Thus all closed-loop eigenvalues are less than zero, proving the agile fixed-wing aircraft position is asymptotically stable in hover conditions.

3.4.5 Quadrotor hover

We can repeat the same steps for hovering quadrotor as done for the agile fixed-wing aircraft. For a hovering quadrotor we specify a reference orientation of 0° roll, 0° pitch, and 0° yaw. By neglecting higher order terms because of the small K_{pp} and K_{pd} assumption, we can obtain the direction of the thrust in the inertial frame as

$$\hat{\mathbf{f}}_i = \begin{bmatrix} K_{pp} \Delta p_x + K_{pd} \Delta v_x \\ K_{pp} \Delta p_y + K_{pd} \Delta v_y \\ -1 \end{bmatrix}, \tag{55}$$

which can be substituted into Eq. (34) to obtain the velocity error dynamics

$$\Delta \dot{\mathbf{v}}_i = \begin{bmatrix} -\frac{f^c}{m} (K_{pp} \Delta p_x + K_{pd} \Delta v_x) \\ -\frac{f^c}{m} (K_{pp} \Delta p_y + K_{pd} \Delta v_y) \\ \frac{f^c}{m} - g \end{bmatrix}. \tag{56}$$

As we can see, there is no aerodynamic force (lift and drag) for a hovering quadrotor. As done previously, the control force can be obtained from Eqs. (9 and 11), which simplifies to $f^c = mg - K_{hp} \Delta p_z - K_v \Delta v_z$ with this hover reference orientation and neglecting higher-order terms because of the small K_{pp} and K_{pd} assumption. We can further simplify our velocity error dynamics to

$$\Delta \dot{\mathbf{v}}_i = \begin{bmatrix} -\frac{f^c}{m} (K_{pp} \Delta p_x + K_{pd} \Delta v_x) \\ -\frac{f^c}{m} (K_{pp} \Delta p_y + K_{pd} \Delta v_y) \\ -\frac{1}{m} (K_{hp} \Delta p_z + K_v \Delta v_z) \end{bmatrix}. \tag{57}$$

These closed-loop dynamics are identical to the hovering agile fixed-wing aircraft. Following the same analysis for the hovering agile fixed-wing aircraft, we can show the closed-loop system is asymptotically stable.

4 Control allocation

We now need to map the control moment and body-fixed force to actuator signals and this process is platform-specific. We use simple and computationally inexpensive models to approximate this mapping; opposed to using more accurate and computationally expensive models. The inherent properties of a closed-loop control system enables the control performance to succeed despite some inaccuracies in the control allocation step. We first discuss actuator models used for most aerial systems: a propeller, a control surface, and a flapping surface. We then discuss how these actuators are used to apply a moment and body fixed-force, specifically for a quadrotor, an agile fixed-wing, a tailsitter, a flapping-wing, and a tilt-wing aircraft.

4.1 Actuator properties

Each type of actuator has an input signal denoted by u_j^s for the system's j th actuator, and this signal corresponds to a force and torque, denoted by \mathbf{u}_j^f and \mathbf{u}_j^τ , which are resolved in the body frame.

4.1.1 Propeller

For a propeller, we consider the input signal, u_j^s , to be the rotational speed of the propeller, and this can be mapped to a thrust and torque as follows:

$$\mathbf{u}_j^f = k_t (J_j) u_j^{s^2} \hat{\mathbf{f}}_b \tag{58a}$$

$$\mathbf{u}_j^\tau = \pm k_q (J_j) u_j^{s^2} \hat{\mathbf{f}}_b \tag{58b}$$

where the propeller thrust and torque coefficients are denoted by k_t and k_q (we drop the j subscript because each platform

contains only one type of propeller). Simplified thrust and torque models assume k_t and k_q are constant, which is valid for stationary propellers (i.e. hovering flight). A quadrotor is usually in a near-hover state and thus we assume k_t and k_q are constant. However, an agile fixed-wing, tailsitter, and tilt-wing aircraft all fly at higher speeds, and thus the propeller is not stationary. 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 and torque coefficients as a function of advance ratio, J , using the model presented in [13], where the advance ratio is defined as

$$J_j = \frac{\|\mathbf{v}_b\|}{2R \frac{u_{j,prev}^s}{60}} \quad (59)$$

where $u_{j,prev}^s$ is the propeller rotational speed at the previous time step, in RPM, and R is the propeller radius. The advance ratio is bounded such that $0 \leq J_j \leq 0.5$, which ensures the propeller is in its normal working state [12].

4.1.2 Control 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. These forces are typically small in magnitude but far from the aircraft center of mass; thus are ultimately used to exert moments on the aircraft. We model the force and torque generated by a control surface as follows:

$$\mathbf{u}_j^f = c_j v_{s,j}^2 \hat{\mathbf{d}}_j u_j^s \quad (60a)$$

$$\mathbf{u}_j^\tau = \mathbf{0} \quad (60b)$$

where the input signal is the deflection angle, the direction of the force is $\hat{\mathbf{d}}_j$, and the constant specific to the control surface and atmospheric conditions is denoted by c_j . In the aircraft community it is common to scale the control surface effectiveness with the square of the airspeed [7]. In conventional 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 scale the control surface effectiveness with the slipstream speed, $v_{s,j}$, instead of the aircraft speed. We can estimate this slipstream speed, $v_{s,j}$,

using momentum theory [16]:

$$v_{s,j} = \sqrt{v_{b,l}^2 + \frac{2\|\mathbf{u}_k^f\|}{\rho(\pi R^2)}} \quad (61)$$

where the aircraft longitudinal speed is denoted by $v_{b,l}$, and the commanded thrust force corresponding to the thruster inducing the slipstream is denoted by $\|\mathbf{u}_k^f\|$. For example, the agile fixed-wing aircraft has one thruster, thus the slipstream approximation is the same for each control surface. Whereas the tailsitter has two thrusters, causing the slipstream approximation for control surface 3 to be based on thruster 1, and the slipstream approximation for control surface 4 is based on thruster 2 (see Fig. 5).

The approximated slipstream is bounded to always be greater than the slipstream in a hover (calculated using Eq. (61), and setting $v_{b,l} = 0$ and $\|\mathbf{u}_k^f\| = \frac{mg}{\text{number of thrusters}}$), to avoid excessive control action at low slipstream values. In addition, we filter the approximated slipstream using a second-order low-pass filter with a 2 Hz natural frequency and .707 damping ratio, since a noisy approximated slipstream will cause abrupt changes in control surface deflections. 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.

4.1.3 Flapping surface

If we assume that we can experimentally obtain a function relating the flapping frequency of a flapping surface to the mean thrust it generates, we can also apply this universal control approach to flapping-wing vehicles. Using this assumption we can generally model a flapping surface as:

$$\mathbf{u}_j^f = g(u_j^s) \hat{\mathbf{f}}_b \quad (62a)$$

$$\mathbf{u}_j^\tau = \mathbf{0} \quad (62b)$$

where we assume this function g can be obtained experimentally, and u_j^s is the frequency of the flapping surface.

4.1.4 Actuator dynamics

The simplified models discussed above neglect the dynamics of the actuators. In reality these actuators are dynamic systems, and a propeller rotational speed, a control surface deflection, or a flapping surface frequency cannot be assigned instantaneously, but could be modeled by an ordinary differential equation. However in the context of small UAVs, the actuator dynamics occur much faster than the attitude dynamics, enabling one to neglect these effects in the control allocation step. In Sect. 5, we validate the controller in simu-

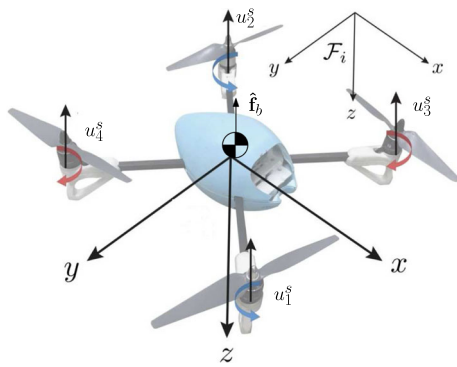


Fig. 3 Spiro quadrotor

lation (where actuator dynamics are modeled in the plant) and in flight experiments (where actuator dynamics are present), and the control strategy is shown to be effective. As the vehicles scale in size, if the actuator dynamics slow down in comparison to the attitude dynamics, it could be necessary to account for the actuator dynamics in the control allocation step.

4.2.1 Quadrotor

Consider a quadrotor with an ‘X’ configuration, such as the Pleiades Spiri shown in Fig. 3.

A quadrotor platform has four propellers; we substitute the propeller model in Eq. (58) into Eq. (63) for $j = 1, 2, 3, 4$ to obtain:

$$\begin{aligned} \mathbf{m}_b^c &= \sum_{j=1}^4 \mathbf{r}_j \times k_t u_j^{s^2} \hat{\mathbf{f}}_b \pm k_q u_j^{s^2} \hat{\mathbf{f}}_b \\ &= \sum_{j=1}^4 (\mathbf{r}_j \times k_t \hat{\mathbf{f}}_b \pm k_q \hat{\mathbf{f}}_b) u_j^{s^2} \end{aligned} \tag{64}$$

$$f^c \hat{\mathbf{f}}_b = \sum_{j=1}^4 k_t u_j^{s^2} \hat{\mathbf{f}}_b \Rightarrow f^c = \sum_{j=1}^4 k_t u_j^{s^2} \tag{65}$$

where we can obtain the applied moment and body fixed-force as a function of the input signal. We also note that the force equation simplifies to scalar form. We can rewrite Eqs. (64) and (65) in matrix form, and invert it to obtain the input signal as a function of moment and body-fixed force:

$$\begin{bmatrix} u_1^{s^2} \\ u_2^{s^2} \\ u_3^{s^2} \\ u_4^{s^2} \end{bmatrix} = \begin{bmatrix} k_t & & & \\ (\mathbf{r}_1 \times k_t \hat{\mathbf{f}}_b - k_q \hat{\mathbf{f}}_b) & k_t & & \\ (\mathbf{r}_2 \times k_t \hat{\mathbf{f}}_b - k_q \hat{\mathbf{f}}_b) & & k_t & \\ (\mathbf{r}_3 \times k_t \hat{\mathbf{f}}_b + k_q \hat{\mathbf{f}}_b) & & & k_t \\ (\mathbf{r}_4 \times k_t \hat{\mathbf{f}}_b + k_q \hat{\mathbf{f}}_b) & & & \end{bmatrix}^{-1} \begin{bmatrix} f^c \\ \mathbf{m}_b^c \end{bmatrix} \tag{66}$$

4.2 Obtaining actuator commands

We denote the position vector from the UAV’s center of mass to the j th actuator’s applied force as \mathbf{r}_j , which is resolved in the body frame. We can then compute the forces and torque’s created by the control inputs as:

$$\mathbf{m}_b^c = \sum_{j=1}^k \mathbf{r}_j \times \mathbf{u}_j^f + \mathbf{u}_j^\tau \tag{63a}$$

$$\mathbf{f}_b^c = f^c \hat{\mathbf{f}}_b = \sum_{j=1}^k \mathbf{u}_j^f \tag{63b}$$

where the system contains k actuators. We then make platform-specific assumptions that allow one to solve for the actuator signal as a function of the applied moment and body-fixed force. We discuss these details for a quadrotor, an agile fixed-wing, a tailsitter, a flapping-wing, and a tilt-wing aircraft.

4.2.2 Agile fixed-wing

Consider the McFoamy agile fixed-wing aircraft shown in Fig. 4, which has four control inputs, one propeller (u_1^s), an aileron (u_2^s), an elevator (u_3^s), and a rudder (u_4^s) deflection. The aileron is made up of two control surfaces driven by one servomotor; thus each surface deflects equal and opposite causing the input signal u_2^s to produce two forces, one on each side of the plane and denoted by subscripts $2l$ and $2r$.

Using the actuator models presented in Eqs. (58) and (60) with Eq. (63) we can obtain:

$$\begin{aligned} \mathbf{m}_b^c &= (\mathbf{r}_{2l} \times c_2 v_s^2 \hat{\mathbf{d}}_{2l} + \mathbf{r}_{2r} \times c_2 v_s^2 \hat{\mathbf{d}}_{2r}) u_2^s \\ &\quad + \mathbf{r}_3 \times c_3 v_s^2 \hat{\mathbf{d}}_3 u_3^s + \mathbf{r}_4 \times c_4 v_s^2 \hat{\mathbf{d}}_4 u_4^s \end{aligned} \tag{67}$$

$$f^c \hat{\mathbf{f}}_b = k_t u_1^{s^2} \hat{\mathbf{f}}_b \Rightarrow f^c = k_t u_1^{s^2} \tag{68}$$

which give the applied moment and body fixed-force as a function of the input signal. Note that we assume the forces

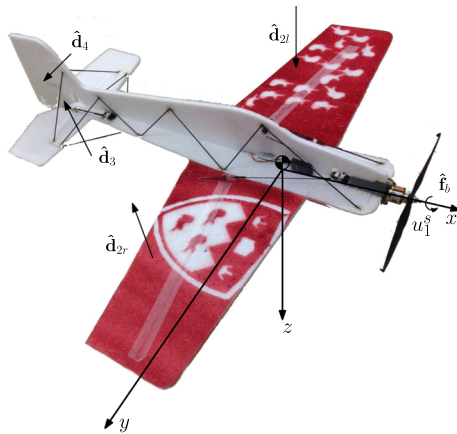


Fig. 4 McFoamy agile fixed-wing

generated by the control surfaces are negligible compared to the thruster force, which is necessary to classify this aircraft as a vehicle that can apply a moment about an arbitrary axis and a body-fixed force. We also assume that the torque generated by the propeller is negligible compared to the torque generated by the control surfaces; this assumption is not necessary, but is presented this way to be consistent with the flight testing presented later in Sect. 5.3.3. For this airframe, $\mathbf{r}_1 \times k_t \hat{\mathbf{f}}_b = \mathbf{0}$ which is why it is dropped in the equation. We again note that the force equation simplifies to scalar form. We can rewrite Eqs. (67) and (68) in matrix form, and invert it to obtain the input signal as a function of moment and body-fixed force:

$$\begin{bmatrix} u_1^{s^2} \\ u_2^s \\ u_3^s \\ u_4^s \end{bmatrix} = \begin{bmatrix} k_t & 0 & 0 & 0 \\ \mathbf{0} & (\mathbf{r}_{2l} \times c_2 v_s^2 \hat{\mathbf{d}}_{2l} + \mathbf{r}_{2r} \times c_2 v_s^2 \hat{\mathbf{d}}_{2r}) & \mathbf{r}_3 \times c_3 v_s^2 \hat{\mathbf{d}}_3 & \mathbf{r}_4 \times c_4 v_s^2 \hat{\mathbf{d}}_4 \end{bmatrix}^{-1} \begin{bmatrix} f^c \\ \mathbf{m}_b^c \end{bmatrix} \tag{69}$$

Most of the time the thrust command is simply the force command. However, in the case of a saturated control surface, the thruster can also be used to generate more slipstream to produce a larger moment, which is discussed in detail in [2].

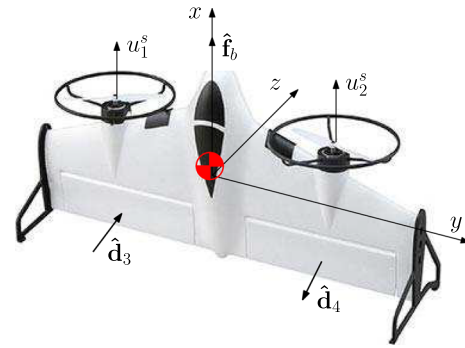


Fig. 5 Tailsitter

4.2.3 Tailsitter

Consider the tailsitter shown in Fig. 5, which has four control inputs: two propellers (u_1^s and u_2^s) and two control surfaces (u_3^s and u_4^s) called elevons.

Using the actuator models in Eqs. (58) and (60) with Eq. (63) we can obtain:

$$\mathbf{m}_b^c = (\mathbf{r}_1 \times k_t \hat{\mathbf{f}}_b + k_q \hat{\mathbf{f}}_b) u_1^{s^2} + (\mathbf{r}_2 \times k_t \hat{\mathbf{f}}_b - k_q \hat{\mathbf{f}}_b) u_2^{s^2} + \mathbf{r}_3 \times c_3 v_s^2 \hat{\mathbf{d}}_3 u_3^s + \mathbf{r}_4 \times c_4 v_s^2 \hat{\mathbf{d}}_4 u_4^s \tag{70}$$

$$f^c \hat{\mathbf{f}}_b = (k_t u_1^{s^2} + k_t u_2^{s^2}) \hat{\mathbf{f}}_b \Rightarrow f^c = k_t u_1^{s^2} + k_t u_2^{s^2} \tag{71}$$

where once again, we assume the force generated by the control surfaces is negligible compared to the thruster forces

in order to classify this vehicle as one that can apply force in a body-fixed direction. We can write this in matrix form and invert it to obtain the control inputs as a function of moment and body-fixed force:

$$\begin{bmatrix} u_1^{s^2} \\ u_2^{s^2} \\ u_3^s \\ u_4^s \end{bmatrix} = \begin{bmatrix} k_t & k_t & 0 & 0 \\ (\mathbf{r}_1 \times k_t \hat{\mathbf{f}}_b + k_q \hat{\mathbf{f}}_b) & (\mathbf{r}_2 \times k_t \hat{\mathbf{f}}_b - k_q \hat{\mathbf{f}}_b) & \mathbf{r}_3 \times c_3 v_s^2 \hat{\mathbf{d}}_3 & \mathbf{r}_4 \times c_4 v_s^2 \hat{\mathbf{d}}_4 \end{bmatrix}^{-1} \begin{bmatrix} f^c \\ \mathbf{m}_b^c \end{bmatrix} \tag{72}$$

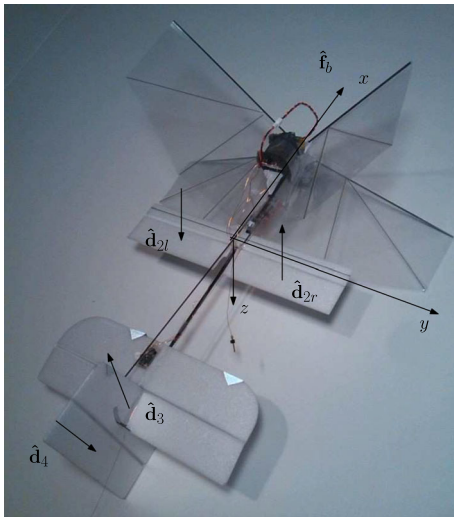


Fig. 6 Delfly flapping-wing [24]

4.2.4 Flapping-wing

Consider the Delfly flapping-wing UAV shown in Fig. 6, which has four control inputs, one flapping-wing (u_1^s), an aileron (u_2^s), an elevator (u_3^s), and a rudder (u_4^s) deflection. The aileron is made up of two control surfaces driven by one servomotor; thus each surface deflects equal and opposite causing the input signal u_2^s to produce two forces, one on each side of the plane and denoted by subscripts $2l$ and $2r$.

Using the actuator models presented in Eqs. (60) and (62) with Eq. (63) we can obtain:

$$\mathbf{m}_b^c = (\mathbf{r}_{2l} \times c_2 v_s^2 \hat{\mathbf{d}}_{2l} + \mathbf{r}_{2r} \times c_2 v_s^2 \hat{\mathbf{d}}_{2r}) u_2^s + \mathbf{r}_3 \times c_3 v_s^2 \hat{\mathbf{d}}_3 u_3^s + \mathbf{r}_4 \times c_4 v_s^2 \hat{\mathbf{d}}_4 u_4^s \tag{73}$$

$$f^c \hat{\mathbf{f}}_b = g(u_1^s) \hat{\mathbf{f}}_b \Rightarrow f^c = g(u_1^s) \tag{74}$$

which give the applied moment and body fixed-force as a function of the input signal. Similar to the previously mentioned vehicles, we assume the force generated by the control surfaces are negligible compared to the flapping-wing thrust. For this airframe, $\mathbf{r}_1 \times k_t \hat{\mathbf{f}}_b = \mathbf{0}$ which is why it is dropped in the equation. We also note that the force equation simplifies to scalar form. We can rewrite Eq. (73) in matrix form, and invert it to obtain the input signal as a function of the desired moment:

$$\begin{bmatrix} u_2^s \\ u_3^s \\ u_4^s \end{bmatrix} = \left[(\mathbf{r}_{2l} \times c_2 v_s^2 \hat{\mathbf{d}}_{2l} + \mathbf{r}_{2r} \times c_2 v_s^2 \hat{\mathbf{d}}_{2r}) \quad \mathbf{r}_3 \times c_3 v_s^2 \hat{\mathbf{d}}_3 \quad \mathbf{r}_4 \times c_4 v_s^2 \hat{\mathbf{d}}_4 \right]^{-1} \begin{bmatrix} \mathbf{m}_b^c \end{bmatrix} \tag{75}$$

To generate the desired body-fixed force, we assume that the relationship of Eq. (74) is invertible, leading to

$$u_1^s = g^{-1}(f^c) \tag{76}$$

4.2.5 Tilt-wing

Consider the Vahana tilt-wing aircraft, a single-passenger flying taxi proposed by Airbus A³ [1], shown in Figs. 7 and 8. This aircraft has eight propellers—four on the front wing, and four on the rear wing—and two control surfaces on the rear wing. The wing tilt angle, γ , is the angle between the direction of the thrust force and the body x -axis. Potentially, this angle gives an additional degree of freedom to the controller. However, since the wing tilt is likely to take place at a much slower rate than variations of thrust or control surface deflections, we can assume that the variation of γ will be determined by the trajectory generator, rather than by the feedback controller. In this case, the angle γ can be viewed by the controller as a prescribed variable, rather than an unknown, and the control allocation can be obtained by combining the actuator models of Eqs. (58) and (60) with Eq. (63) to obtain:

$$\mathbf{m}_b^c = \sum_{j=1}^8 (\mathbf{r}_j \times k_t \hat{\mathbf{f}}_b + (-1)^n k_q \hat{\mathbf{f}}_b) u_j^s + \sum_{j=9}^{10} (\mathbf{r}_j \times c_j v_{s_j}^2 \hat{\mathbf{d}}_j u_j^s) \tag{77}$$

$$f^c \hat{\mathbf{f}}_b = \sum_{j=1}^8 k_t u_j^s \hat{\mathbf{f}}_b \Rightarrow f^c = \sum_{j=1}^8 k_t u_j^s \tag{78}$$

Another difference between the Vahana platform and the preceding platforms is that it is redundantly-actuated—there are ten actuators to generate four force/moments. This results in infinite possible sets of actuator commands that can be used to generate the applied moment and body-fixed force. This indeterminacy is best resolved by formulating and solving an optimization problem, and using Eqs. (77) and (78) as constraints to that problem. For example, one could minimize an objective consisting of the weighted norm of the vector of control inputs $\sum_{j=1}^8 (w_j u_j^s)^2 + \sum_{j=9}^{10} (w_j u_j^s)^2$. In this case, the resulting optimization problem would have a quadratic objective and linear constraints in the design variables, and would be solvable in real-time. Another advantage of this optimization approach is that inequality constraints can be

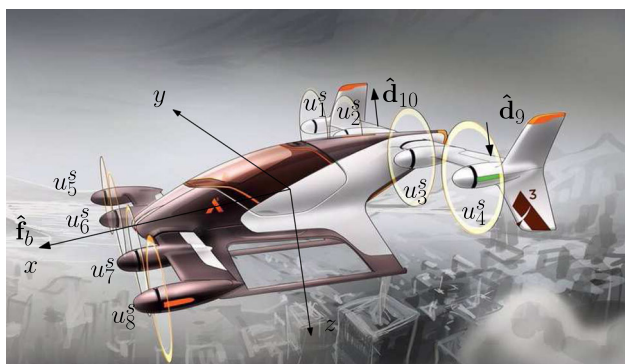


Fig. 7 Vahana tilt-wing ($\gamma = 0^\circ$) [1]



Fig. 8 Vahana tilt-wing ($\gamma = 90^\circ$) [1]

included to represent actuator limits, such as maximum control surface deflections and maximum propeller speeds.

4.2.6 Platform-specific properties

We show the values of the platform specific properties in Table 1. Not all the values are applicable to every platform, and those values are denoted by N/A. For the flapping-wing and tilt-wing we cannot specify some values because we do not have the platforms, which is denoted by NI (no info).

5 Validation

We implement our universal control approach on a quadrotor and an agile fixed-wing aircraft, depicted in Figs. 3 and 4. We specifically choose these two platforms to experimentally validate our control approach because we view them as two very different platforms, while other types of UAV’s can be viewed as a combination of these two platforms. In addition, we do not have access to the other platforms discussed in Sect. 4.

5.1 Simulation

Before testing our control algorithm in flight tests, we test the control algorithm in a Matlab/Simulink simulation envi-

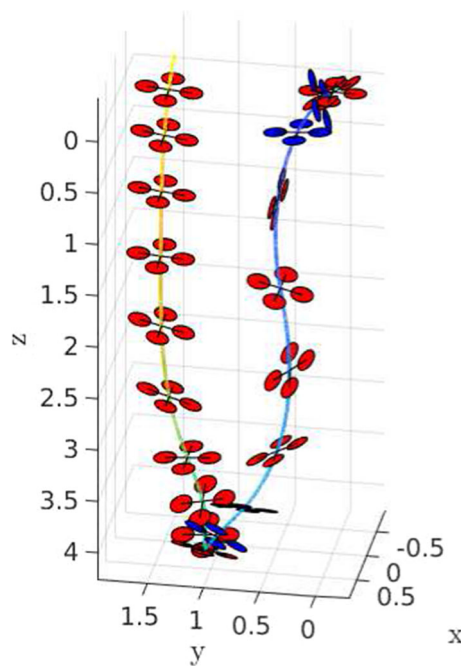


Fig. 9 Simulated quadrotor rolling flip [4]

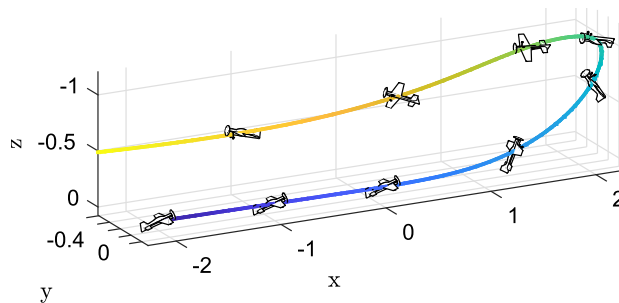


Fig. 10 Simulated fixed-wing aggressive turnaround [4]

ronment. The simulation environment is comprised of the modeling described in Sect. 2, coupled with detailed aerodynamics and thruster dynamics models presented in [12] for the agile fixed-wing aircraft, and in [15] for the quadrotor. We successfully demonstrate automatic aggressive maneuvering using our control architecture in both of these simulation environments and we present the detailed results from these simulations in [4]. We show the simulated motion of a quadrotor rolling flip presented in [4] in Fig. 9, and of a fixed-wing aggressive turnaround in Fig. 10. In these figures the trajectory line starts as blue and gets lighter with time, and in Fig. 9 the top of the quadrotor is red and the bottom is blue.

A hardware-in-the-loop (HIL) simulation is then used to initially test the hardware implementation, prior to testing on the real vehicle. In an HIL simulation, the on-board flight controller (Pixhawk) sends virtual actuator commands to a desktop computer running the UAV dynamics and sen-

Table 1 Platform details

Variable	Unit	Quadrotor	Agile fixed-wing	Tailsitter	Flapping-wing	Tilt-wing
$\hat{\mathbf{f}}_b$	-	$[0, 0, -1]^T$	$[1, 0, 0]^T$	$[1, 0, 0]^T$	$[1, 0, 0]^T$	$[\cos(\gamma), 0, -\sin(\gamma)]^T$
k_t	N/RPM^2	6.91×10^{-8}	$(-1.439J^2 - 2.212J + 2.245)10^{-7}$	NI	NI	NI
k_q	Nm/RPM^2	1.12×10^{-9}	N/A	NI	N/A	NI
c_2	$N/(^\circ m^2/s^2)$	N/A	1.0665×10^{-4}	N/A	NI	N/A
c_3	$N/(^\circ m^2/s^2)$	N/A	4.0034×10^{-4}	NI	NI	N/A
c_4	$N/(^\circ m^2/s^2)$	N/A	4.4899×10^{-4}	NI	NI	N/A
$\hat{\mathbf{d}}_{2l}$	-	N/A	$[0, 0, 1]^T$	N/A	$[0, 0, 1]^T$	N/A
$\hat{\mathbf{d}}_{2r}$	-	N/A	$[0, 0, -1]^T$	N/A	$[0, 0, -1]^T$	N/A
$\hat{\mathbf{d}}_3$	-	N/A	$[0, 0, -1]^T$	$[0, 0, 1]^T$	$[0, 0, -1]^T$	N/A
$\hat{\mathbf{d}}_4$	-	N/A	$[0, 1, 0]^T$	$[0, 0, -1]^T$	$[0, 1, 0]^T$	N/A
$\hat{\mathbf{d}}_9$	-	N/A	N/A	N/A	N/A	$[\sin(\gamma), 0, \cos(\gamma)]^T$
$\hat{\mathbf{d}}_{10}$	-	N/A	N/A	N/A	N/A	$[-\sin(\gamma), 0, -\cos(\gamma)]^T$
\mathbf{r}_1	m	$[0.1626, 0.1626, 0]^T$	$[0.25, 0, 0]^T$	$[0.05, -0.15, 0]^T$	NI	NI
\mathbf{r}_2	m	$[-0.1626, -0.1626, 0]^T$	N/A	$[0.05, 0.15, 0]^T$	N/A	NI
\mathbf{r}_{2l}	m	N/A	$[0, -0.24, 0]^T$	N/A	NI	N/A
\mathbf{r}_{2r}	m	N/A	$[0, 0.24, 0]^T$	N/A	NI	N/A
\mathbf{r}_3	m	$[0.1626, -0.1626, 0]^T$	$[-0.54, 0, 0]^T$	$[-0.15, -0.13, 0]^T$	NI	NI
\mathbf{r}_4	m	$[-0.1626, 0.1626, 0]^T$	$[-0.60, 0, 0]^T$	$[-0.15, 0.13, 0]^T$	NI	NI

Table 2 Platform properties

Platform	Parameter	Symbol	Value	Unit
Quadrotor	Mass	m	1.02	kg
	Thrust:weight	–	2.9	–
Fixed-wing	Mass	m	0.45	kg
	Thrust:weight	–	2.3	–

sensor measurement model, which in turn, sends back virtual sensor measurements to on-board hardware. For the agile fixed-wing, we use our Simulink-based dynamics and sensor model to generate the virtual sensor measurements [3]. For the quadrotor, we use the built-in HIL simulation using RotorS/Gazebo [8], with slight modifications to reflect our systems' inertial and thruster characteristics.

The HIL simulations were a useful tool in bridging the gap between the pure theoretical simulations and experimental flight tests. The initial set of control gains tuned in the Simulink simulation led to unstable flight in the HIL simulation. However, once the gains were re-tuned in the HIL simulation, little gain tuning was needed during flight testing.

5.2 Experimental setup

5.2.1 Platforms

After satisfactory results were obtained in the HIL simulations, we performed outdoor autonomous flight tests with both a quadrotor and agile fixed-wing platform. The quadrotor platform, the Pleiades Spiri, and the fixed-wing platform, WM Parkflyers McFoamy, are depicted in Figs. 3 and 4 and the physical properties are presented in Table 2.

All of the sensing and computation is done on-board each platform, using a Pixhawk flight computer running the open-source PX4 flight stack. State estimates of both UAVs are obtained from the default Extended Kalman Filter (EKF2) in PX4 that fuses the IMU, barometer, and GPS measurements. The control loop is executed at 200 Hz.

5.2.2 Outdoor environment

Our agile fixed-wing flight tests took place at the West Island Model Aeronautics Club, and our quadrotor flight tests took place at the McGill University Forbes Field, both large outdoor fields in Montreal, Canada. We only tested autonomous airborne maneuvers—we manually flew the vehicle up to altitude, and then put the vehicle in autonomous mode, performed various maneuvers autonomously, and then manually landed the vehicle.

We recorded flight data over numerous tests and days, with varying wind conditions. We were able to autonomously ex-

ecute several maneuvers with each platform, and were able to perform these maneuvers reliably and repeatedly. We specifically analyze an arbitrary attempt of two maneuvers for each platform, both attempted on days with moderate winds. Wind measurements were not recorded at the test sites but data from nearby weather stations recorded average wind speeds of 10–12 knots during the fixed-wing flight [6] and 4–6 knots during the quadrotor flight [6].

5.3 Experimental results

We specifically show two aggressive maneuvers that are performed with two vastly different platforms: two types of rolling flips with a quadrotor, and a rolling Harrier and an aggressive turnaround with an agile fixed-wing, in order to demonstrate the versatility of the control logic. To further demonstrate this versatility, each maneuver has been designed heuristically and is *not* completely dynamically feasible.

Similar control gains are used for each platform, and are shown in Table 3. While the *same* gains could be used for both platforms, which is shown in simulation in [4], better performance is achieved by tuning gains to each platform due to the uncertainty in the allocation of control forces and moments to actuator signals.

For easier understanding of the flight data, we transform the data such that origin of the position vector is the position at which the maneuver is commanded ($t = 0s$). We also, rotate the inertial frame such that the heading at the start of the maneuver is zero. This rotation makes the x axis the direction of flight for the fixed-wing aircraft. Lastly, we display the attitude in Euler angles, roll (ϕ), pitch (θ) and yaw (ψ), even though our controller is based on a quaternion representation, because the euler angles make it easier to understand the motion.

The UAV's six degrees of freedom (position and orientation) are shown alongside their reference values in Figs. 15, 16, 17 and 18. We reiterate that our controller is trying to track all six of these states with only four control inputs, and thus perfect asymptotic tracking is not possible.

5.3.1 Quadrotor: rolling flip case 1

We successfully perform a rolling flip with a quadrotor, which is shown in the supplementary video, as well as a sequence of overlaid images in Fig. 11.

For quantitative analysis of the maneuver we refer to Fig. 15 which compares the state estimates of position and orientation to the reference values, and also shows the control inputs. The control inputs are shown with normalized values, where zero corresponds to the idle motor PWM signal, and one corresponds to the maximum motor PWM signal.

Table 3 Controller gains

Gain	Symbol	Value (fixed-wing)	Value (quadrotor)	Unit
Position proportional	K_{pp}	.08	.08	rad/m
Position derivative	K_{pd}	.1	.1	rad/ $\frac{m}{s}$
Attitude proportional	\mathbf{K}_{ap}	160 $\mathbf{1}_{3 \times 3}$	110 $\mathbf{1}_{3 \times 3}$	$\frac{rad}{s^2}/rad$
Attitude derivative	\mathbf{K}_{ad}	8 $\mathbf{1}_{3 \times 3}$	20 $\mathbf{1}_{3 \times 3}$	$\frac{rad}{s^2}/\frac{rad}{s}$
Speed proportional	K_v	3	3	$\frac{m}{s^2}/\frac{m}{s}$
Height proportional	K_{hp}	5	5	$\frac{m}{s^2}/m$
Height integral	K_{hi}	0.5	0.5	$\frac{m}{s^2}/ms$



Fig. 11 Quadrotor rolling flip case 1 image sequence: the flip begins on the right side of the image, performs the flip while losing altitude, and then returns to the start of the maneuver

We show one second of hovering flight, and then start the flip maneuver at $t = 0s$. Our reference trajectory is formulated by setting a constant reference position and yaw angle to that at the start of the maneuver (causing the step in reference position at $t = 0s$), a zero pitch angle, and a roll angle which starts at zero, has a constant acceleration, followed by a coast at maximum velocity, concluding with a braking phase of constant deceleration until the roll angle reaches 360 degrees.

This reference trajectory is not dynamically feasible, as it's not possible to do a flip while remaining in the same position, although we are still able to achieve the higher-level requirements of the maneuver. As shown in Fig. 15, the quadrotor drops about 3 m in altitude while the quadrotor's thrusters are pointing downwards, and moves laterally about 1 m in x and in y . Once the flip is achieved and the quadrotor is stabilized, the quadrotor directs itself back to the reference position and remains stationary. There is a small steady-state error in the x and y position, which can be attributed to the

moderate wind gusts and lack of integral term in the control law.

Considering the aggressiveness of the flip maneuver, the attitude is tracked very well. During the flip, the maximum pitch and yaw error remain less than 20° and 10° respectively. The roll angle initially lags the reference, which is followed by a slight overshoot at the end of the flip. After the flip is complete, all of the euler angles are tracked with less than 2° error.

The actuators behave as expected, where initially the speed of rotors two and three increase, while rotors one and four decrease, causing a moment in the x direction needed to roll the vehicle. As the flip is completing, a moment in the opposite direction is needed to reduce the angular velocity, causing rotors two and three to decrease speed, while rotors one and four are increased.

5.3.2 Quadrotor: rolling flip case 2

We aim to perform the same rolling flip maneuver described in Sect. 5.3.1, except this time with a more aggressive roll reference, which ultimately speeds up the execution of maneuver, making the quadrotor lose less altitude. Two changes were made to achieve this more aggressive flip.

First, we change the proportional attitude gain to $\mathbf{K}_{ap} = 110 \text{diag}(1, 1, 0.2) \frac{rad}{s^2}/rad$, and the derivative attitude gain to $\mathbf{K}_{ad} = 20 \text{diag}(1, 1, 0.2) \frac{rad}{s^2}/\frac{rad}{s}$. This change keeps the gains associated with the roll and pitch axes the same, but reduce the gains associated with the yaw axis by a factor of 5. This reduction in yaw gain is theoretically not necessary, however, due to the simple control allocation strategy used for our quadrotor it is necessary to lower the yaw gains to prevent the motors from saturating for too long during this even more aggressive flip, and effectively losing control authority in roll and pitch.

The second change we made is turning off the position controller while the quadrotor is flipping, and turning it back on once the roll reference is back to zero. This is done in [14] and is an example of one of the benefits of a modular control architecture—the ability to easily turn on and off the



Fig. 12 Quadrotor Rolling flip case 2 image sequence: the flip begins on the left side of the image, performs the flip quickly, loses altitude, and then returns to the start of the maneuver

position controller which is discussed in Sect. 3.1. In this example, the simple reference trajectory in position contradicts the reference attitude at times, and during these times the position controller can degrade the performance of the maneuver. For example, initially the reference roll increases, causing the aircraft to roll and increase in y position, but then the position controller wants to decrease the y position, making the augmented reference roll less than reference roll, ultimately slowing the flip down.

Both of these changes allow the quadrotor to track a faster and more aggressive flip. The maneuver is shown in the supplementary video, as well as a sequence of overlaid images in Fig. 12.

For quantitative analysis of the maneuver we refer to Fig. 16. For the most part, the analysis of the maneuver is the same as in Case 1. The major difference is the flip is completed in about $0.75s$ opposed to $1s$, which causes the quadrotor to reduce the altitude drop to only $1.5m$ opposed to $3m$.

Turning off the position controller for the duration of the flip can be seen in attitude plots, where the augmented reference attitude is the same as the reference attitude from $t = 0s$ to $t = 0.75s$. After $t = 0.75s$ the position controller is turned back on, and the reference and augmented reference attitudes are no longer the same.

5.3.3 Fixed-wing: rolling harrier

We successfully demonstrate a rolling Harrier which consists of the aircraft flying along a straight line, while rolling continuously. The maneuver is shown in the supplementary video, as well as a sequence of images in Fig. 13. For quantitative analysis, we plot the state estimated and reference pose of the UAV, as well as the control inputs in Fig. 17. The control inputs (throttle, aileron, elevator, rudder) are shown

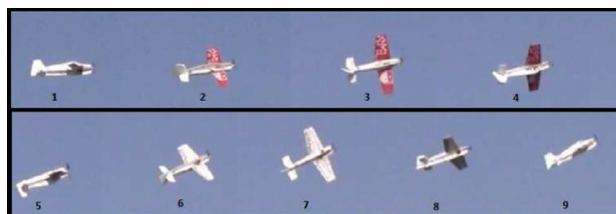


Fig. 13 Fixed-wing rolling harrier image sequence



Fig. 14 Fixed-wing aggressive turnaround image sequence: the plane starts at the bottom right in level flight, pitches until inverted flight, and then rolls into level flight with the opposite heading

with normalized values. For the control surfaces, negative one corresponds to the maximum negative deflection while one corresponds to the maximum positive deflection. For the throttle command, zero corresponds to the idle motor PWM signal, and one corresponds to the maximum motor PWM signal.

We show one second of level flight, and then start the rolling Harrier maneuver at $t = 0s$, after the plane has traveled $20m$ (about $t = 2.2s$) the plane is commanded back to level flight. The reference trajectory for the rolling Harrier is comprised of flying in a straight line while rolling at a constant five radians per second. The straight line starts at the initialization of the maneuver, and extends in the direction of the current heading, which causes a step in reference position and orientation at ($t = 0$ and $t = 2.2$). The maneuver is achieved with less than $1m$ cross-track and altitude error. The reference x position cannot be seen because it is equal to

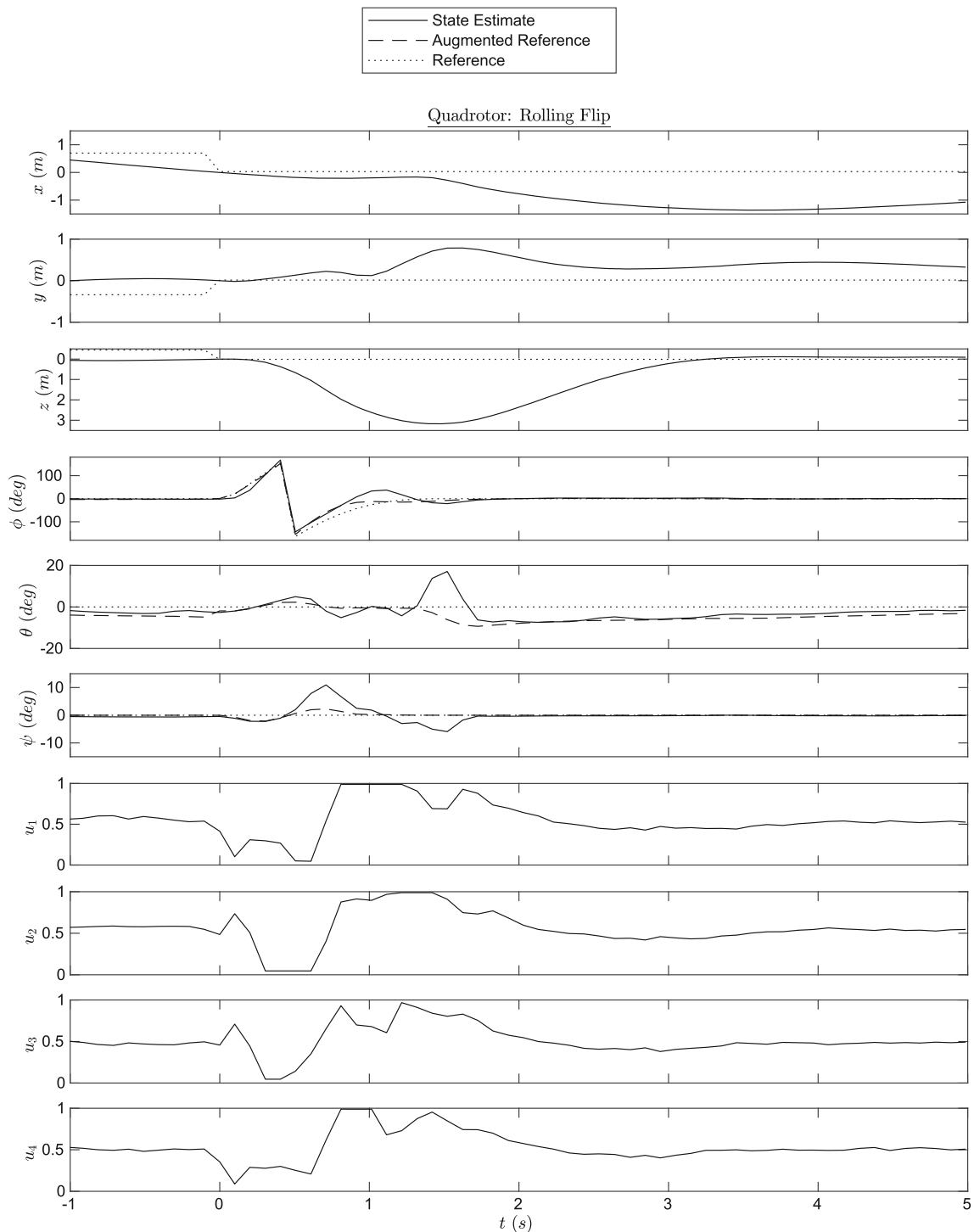


Fig. 15 Quadrotor rolling flip case 1 flight data

the estimate of the x position; but this is not perfect tracking, just due to how the reference trajectory is created.

Turning attention to the attitude results, the pitch and yaw tracking errors remain less than 20° during the rolls. The roll angle does lag behind the reference, but by less than 0.1 s.

The actuators behave as expected, the elevator and rudder (u_3 and u_4) are non-zero when correcting pitch and yaw, but have a mean value near zero. On the other hand, the mean aileron deflection (u_2) is negative, which induces a positive roll rate. At the end of the rolling harrier ($t = 2.2$ s)

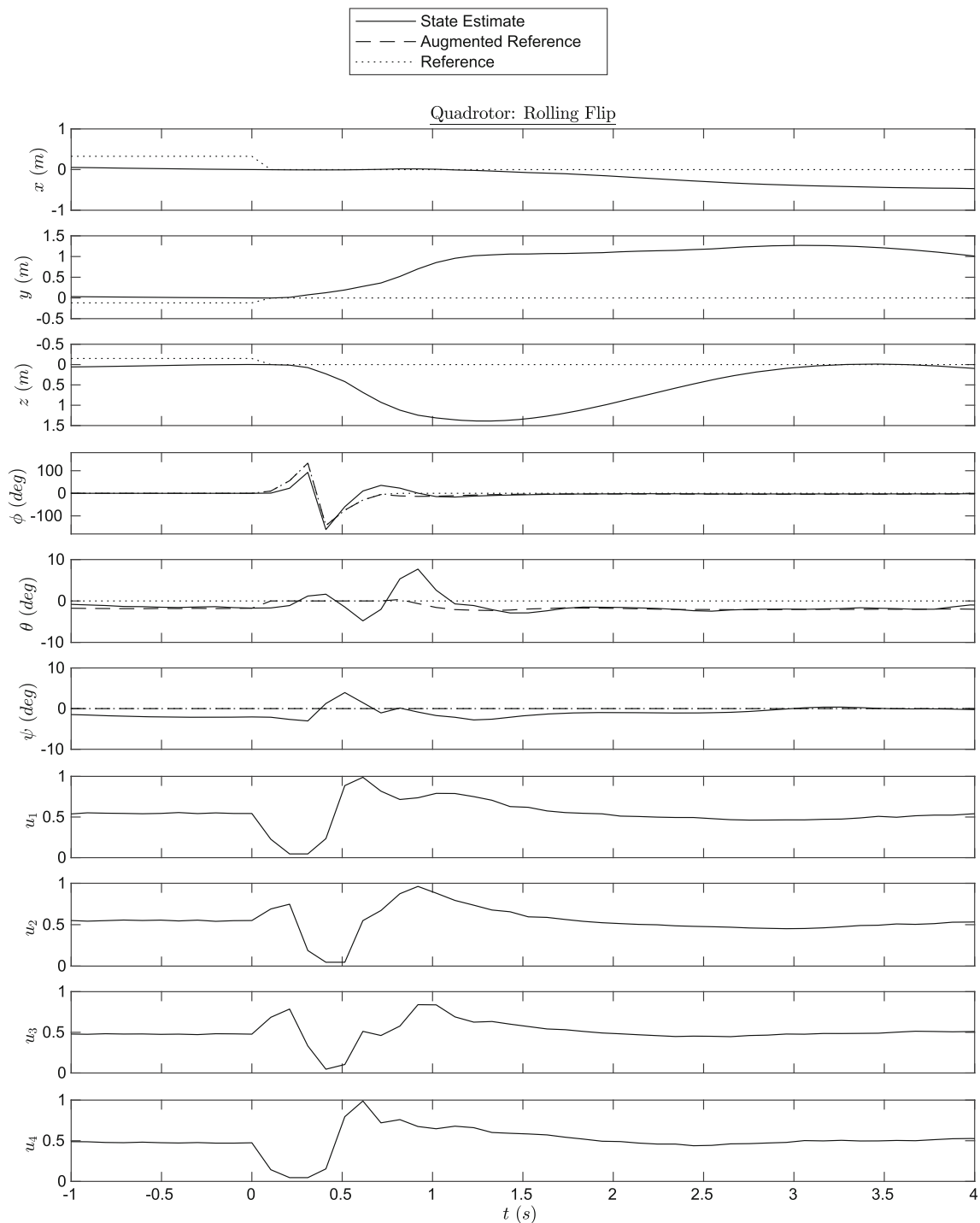


Fig. 16 Quadrotor rolling flip case 2 flight data

the reference roll angle steps up to zero, causing the aileron to saturate. The saturation of the aileron causes the thruster (u_1) to increase and also saturate, in order to increase the slipstream over the aileron and give the aircraft more attitude control authority, as mentioned in Sect. 4.2.2.

5.3.4 Fixed-wing: aggressive turnaround

We successfully demonstrate an aggressive turnaround—a maneuver which starts in level flight, and quickly reverses direction to level flight with a 180° change in heading. The

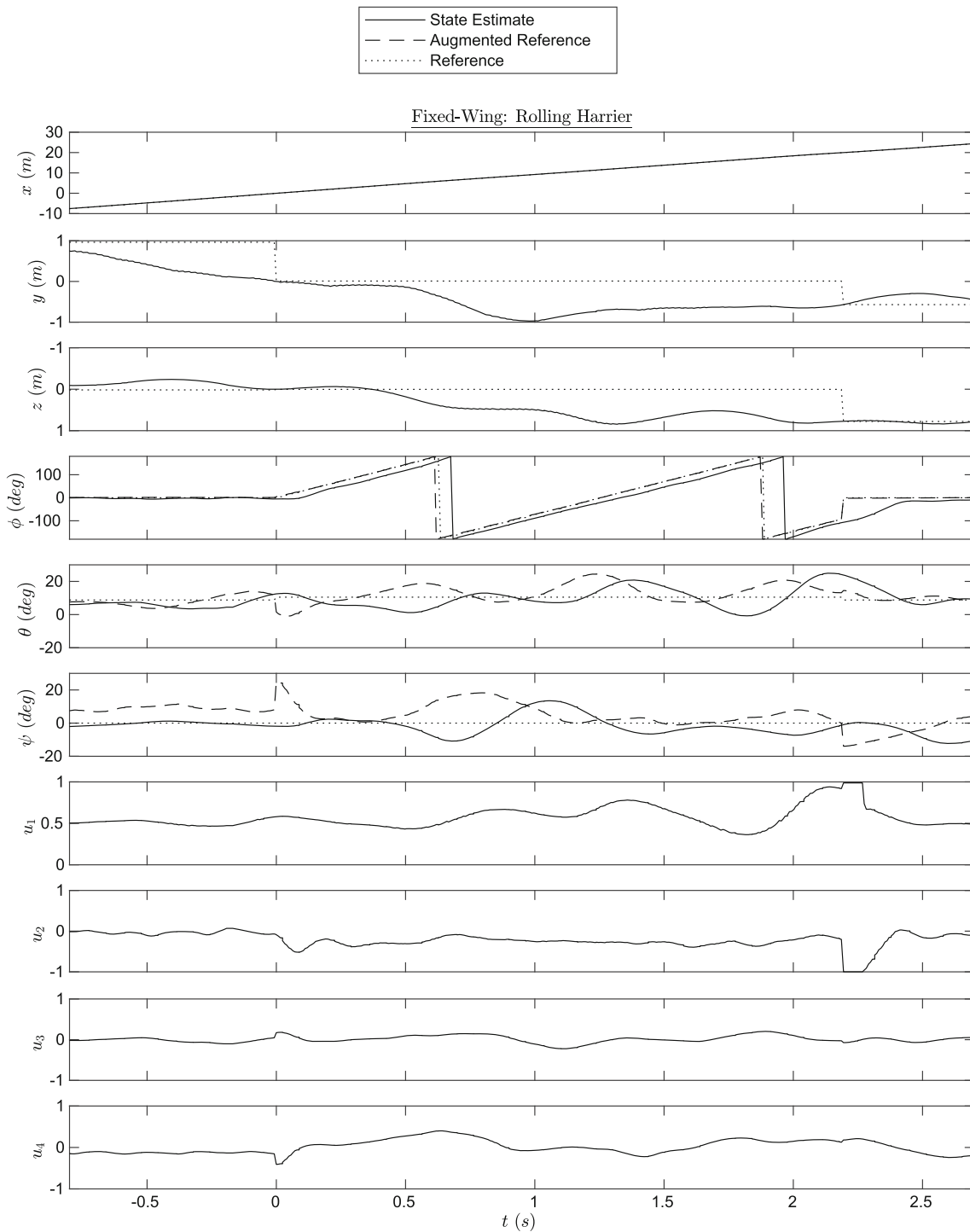


Fig. 17 Fixed-wing rolling harrier flight data

maneuver is shown in the supplementary video, as well as a sequence of overlaid images in Fig. 14. For quantitative analysis, we plot the state estimated and reference pose of the UAV, as well as the control inputs in Fig. 18.

We show a quarter second of level flight, and then start the aggressive turnaround at $t = 0s$. Again, the trajectory

generation is designed heuristically and is *not* dynamically feasible, but demonstrates the versatility of the controller. Due to the lack of feasibility, the aircraft does not perfectly track the reference, but the higher-level goals of the maneuver are achieved.

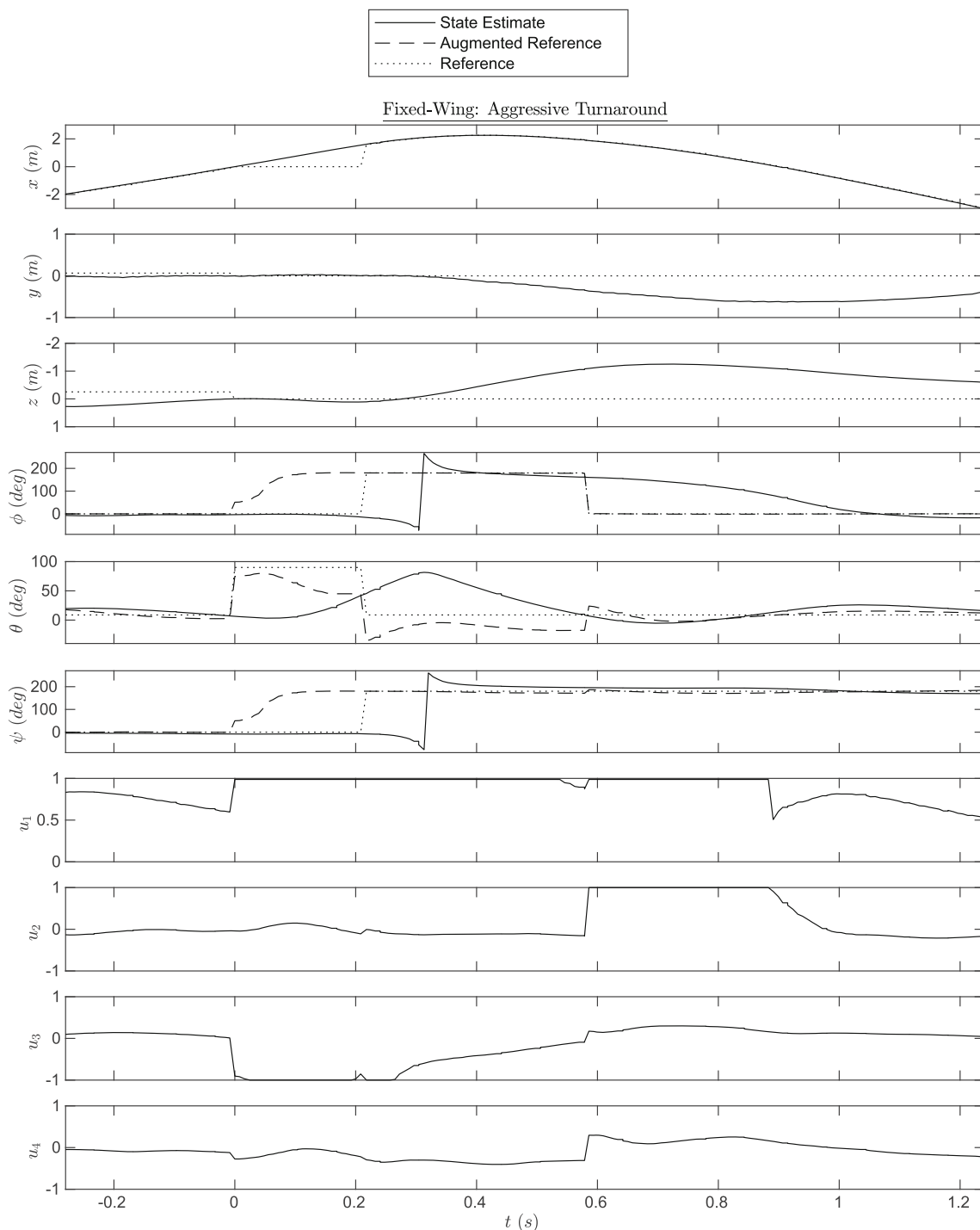


Fig. 18 Fixed-wing aggressive turnaround flight data

The reference trajectory is formed by first commanding a hover. Once the pitch angle exceeds 45° , the aircraft is commanded to inverted flight with a heading opposite to the start of the maneuver. Once the aircraft pitches down to inverted level flight, it is commanded to roll 180° , which concludes with the aircraft in level flight with opposite heading.

We can see higher level goal of the maneuver is achieved—the aircraft starts in level flight at about $8 \frac{m}{s}$, and changes its heading by 180° while only traveling 2 m in the direction of flight, 0.5 m laterally, and increasing altitude by only 1 m. Again, the 1 m altitude increase shouldn't be viewed as 1 m

tracking error, as it not possible turn the aircraft around without occupying space either laterally or vertically.

Looking at the orientation results, the reference pitch angle initially steps to 90° , while the reference roll and yaw remain zero. The position controller aims to point the thruster even further backwards, resulting in a augmented reference orientation of a pitch angle greater than 90° , with zero roll and yaw. However, since the euler angle convention limits pitch to 90° , the augmented reference orientation actually has a pitch angle less than 90° and a 180° roll and yaw. The pitch estimate climbs up to 90° at $t = 0.3s$, at this point, the pitch angle starts to decrease, but the roll and yaw angles of instantaneously jumped to 180° , due to this euler angle convention. The aircraft points itself further backwards until the it is in inverted level flight at $t = 0.6s$. At this point, the reference and augmented reference roll become zero, and the aircraft rolls into level flight by $t = 1s$.

The actuator behavior aligns with the state estimates of the aircraft. At $t = 0s$ the elevator (u_3) saturates to give a large pitching moment, and the thruster (u_1) also saturates to increase the slipstream speed over the elevator, generating an even larger moment (as mentioned in Sect. 4.2.2). Just before $t = 0.6s$, the aircraft is close to the augmented reference pitch and the elevator deflection is near zero, but now the aircraft wants to roll 180° into level flight, and therefore the aileron saturates (u_2), and to give an even larger moment, the slipstream speed is increased by saturating the thruster. At the end of the maneuver, the attitude is close to the augmented reference attitude, and all the control surfaces are near zero.

6 Conclusion

In this work we present a universal control system applicable to most types of UAVs. The control strategy can be applied to any UAV capable of exerting a moment about an arbitrary axis and a force along a body-fixed direction, which includes UAVs such as multi-copters, conventional fixed-wing, agile fixed-wing, most flying-wings, most tailsitters, some tilt-rotor/wing platforms, and some flapping-wing vehicles.

The physics-based controller is capable of tracking virtually any feasible trajectory—including extreme aerobatic maneuvers. The controller is modular, containing a position controller, an attitude controller, and a force controller, all which are platform-independent. The output of these controllers are a force and moment which are then allocated to platform-specific actuator commands.

Experimental validation of the control logic is demonstrated with autonomous aggressive maneuvers, on two vastly different platforms—two rolling flips with a quadrotor and a rolling Harrier and an aggressive turnaround with an agile fixed-wing aircraft. We demonstrate these maneuvers

in moderate winds using the *same* controller with a similar set of gains. We demonstrate satisfactory tracking of our reference pose during these aggressive maneuvers; all while dealing with the realities of a real-world aerial robotic system such as sensor noise, limited on-board computation, and large disturbances such as wind gusts.

Although we only experimentally demonstrate with two platforms, the methodology extends to other platforms, as most UAV platforms can be viewed as a combination of fixed-wing and quadrotor aircraft. In addition, we give insight on how this control architecture would be applied to a tailsitter aircraft, and flapping-wing UAV, and a tilt-wing UAV.

We believe this to be the first UAV controller with applicability to such a wide range of platforms and motions that has been demonstrated experimentally.

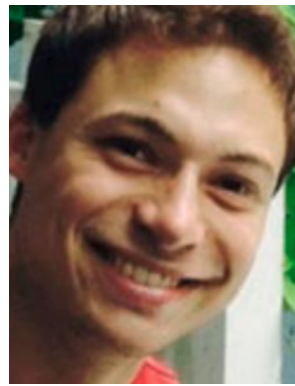
Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s10514-021-10015-8>.

Acknowledgements The authors thank Corey Miles and Walter Jothiraj for assistance during flight testing, and Professor Inna Sharf for the valuable comments and conversations. This work was supported by the Natural Sciences and Engineering Research Council (NSERC) [NSERC Discovery Grant RGPIN-2018-04547], the Fonds de Recherche du Québec—Nature et technologies (FRQNT) and by a McGill Engineering Doctoral Award (MEDA).

References

1. Airbus: Vahana. (2017). Retrieved October 16, 2018, from <https://vahana.aero>.
2. Bulka, E., & Nahon, M. (2018). Automatic control for aerobatic maneuvering of agile fixed-wing UAVs. *Journal of Intelligent & Robotic Systems*. <https://doi.org/10.1007/s10846-018-0790-z>
3. Bulka, E., & Nahon, M. (2018). Autonomous fixed-wing aerobatics: From theory to flight. In *International conference on robotics and automation (ICRA)* (pp. 6573–6580). IEEE.
4. Bulka, E., & Nahon, M. (2018). A universal controller for unmanned aerial vehicles. In *International conference on intelligent robots and systems (IROS)* (pp. 4171–4176). IEEE.
5. Dicker, G., Chui, F., & Sharf, I. (2017). Quadrotor collision characterization and recovery control. In *International conference on robotics and automation (ICRA)* (pp. 5830–5836). IEEE.
6. Environment and Climate Change Canada: Hourly data report. (2017). <http://climate.weather.gc.ca>.
7. Etkin, B. (1982). *Dynamics of flight: Stability and control*. Brisbane: Wiley.
8. Furrer, F., Burri, M., Achtelik, M., & Siegwart, R. (2016). *Robot operating system (ROS): The complete reference (Vol. 1), Chap. RotorS—A modular gazebo mav simulator framework* (pp. 595–625). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-26054-9_23.
9. Green, W. E., & Oh, P. Y. (2009). A hybrid MAV for ingress and egress of urban environments. *IEEE Transactions on Robotics*, 25(2), 253–263. <https://doi.org/10.1109/TRO.2009.2014501>.
10. Hua, M. D., Hamel, T., Morin, P., & Samson, C. (2009). Control of a class of thrust-propelled underactuated vehicles and application to a VTOL drone. In *International conference on robotics and automation (ICRA)* (pp. 972–978). IEEE.

11. Hua, M. D., Pucci, D., Hamel, T., Morin, P., & Samson, C. (2014). A novel approach to the automatic control of scale model airplanes. In *53rd Annual conference on decision and control (CDC)* (pp. 805–812). IEEE.
12. Khan, W. (2016). *Dynamics modeling of agile fixed-wing unmanned aerial vehicles*. Ph.D. thesis, McGill University, Montreal, Canada.
13. Khan, W., & Nahon, M. (2015). A propeller model for general forward flight conditions. *International Journal of Intelligent Unmanned Systems*, 3(2/3), 72–92.
14. Lee, T., Leok, M., & McClamroch, N. H. (2013). Nonlinear robust tracking control of a quadrotor UAV on SE (3). *Asian Journal of Control*, 15(2), 391–408.
15. Mahony, R., Kumar, V., & Corke, P. (2012). Multirotor aerial vehicles. *IEEE Robotics and Automation Magazine*, 19, 20–32.
16. McCormick, B. (1995). *Aerodynamics, aeronautics, and flight mechanics* (2nd ed.). Hoboken: Wiley.
17. Mellinger, D., Michael, N., & Kumar, V. (2012). Trajectory generation and control for precise aggressive maneuvers with quadrotors. *International Journal of Robotics Research*, 31(5), 664–674. <https://doi.org/10.1177/0278364911434236>
18. Moore, J., Cory, R., & Tedrake, R. (2014). Robust post-stall perching with a simple fixed-wing glider using LQR-Trees. *Bioinspiration & Biomimetics*, 9(2), 025013. <https://doi.org/10.1088/1748-3182/9/2/025013>.
19. Park, S. (2012). Autonomous aerobatics on commanded path. *Aerospace Science and Technology*, 22(1), 64–74. <https://doi.org/10.1016/j.ast.2011.06.007>
20. Pucci, D., Hamel, T., Morin, P., & Samson, C. (2015). Nonlinear feedback control of axisymmetric aerial vehicles. *Automatica*, 53, 72–78.
21. Ritz, R., & D’Andrea, R. (2017). A global controller for flying wing tailsitter vehicles. In *International conference on robotics and automation (ICRA)* (pp. 2731–2738). IEEE.
22. Snell, S. A., Enns, D. F., & Garrard, W. L., Jr. (1992). Nonlinear inversion flight control for a supermaneuverable aircraft. *Journal of Guidance, Control, and Dynamics*, 15(4), 976–984.
23. Sobolic, F. M. (2009). *Agile flight control techniques for a fixed-wing aircraft*. Thesis, Massachusetts Institute of Technology.
24. Verboom, J., Tijmons, S., De Wagter, C., Remes, B., Babuska, R., & de Croon, G. C. (2015). Attitude and altitude estimation and control on board a flapping wing micro air vehicle. In *International conference on robotics and automation (ICRA)* (pp. 5846–5851). IEEE.
25. Wie, B., Weiss, H., & Arapostathis, A. (1989). Quaternion feedback regulator for spacecraft eigenaxis rotations. *Journal of Guidance, Control, and Dynamics*, 12(3), 375–380.



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.