



MIT Open Access Articles

Accurate Tracking of Aggressive Quadrotor Trajectories Using Incremental Nonlinear Dynamic Inversion and Differential Flatness

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

As Published	10.1109/TCST.2020.3001117
Publisher	Institute of Electrical and Electronics Engineers (IEEE)
Version	Author's final manuscript
Citable link	https://hdl.handle.net/1721.1/133387
Terms of Use	Creative Commons Attribution-Noncommercial-Share Alike
Detailed Terms	http://creativecommons.org/licenses/by-nc-sa/4.0/

Accurate Tracking of Aggressive Quadrotor Trajectories using Incremental Nonlinear Dynamic Inversion and Differential Flatness

Ezra Tal, *Student Member, IEEE*, Sertac Karaman, *Member, IEEE*

Abstract—Autonomous unmanned aerial vehicles (UAVs) that can execute aggressive (*i.e.*, high-speed and high-acceleration) maneuvers have attracted significant attention in the past few years. This paper focuses on accurate tracking of aggressive quadcopter trajectories. We propose a novel control law for tracking of position and yaw angle and their derivatives of up to fourth order, specifically, velocity, acceleration, jerk, and snap along with yaw rate and yaw acceleration. Jerk and snap are tracked using feedforward inputs for angular rate and angular acceleration based on the *differential flatness* of the quadcopter dynamics. Snap tracking requires direct control of body torque, which we achieve using closed-loop motor speed control based on measurements from optical encoders attached to the motors. The controller utilizes *incremental nonlinear dynamic inversion (INDI)* for robust tracking of linear and angular accelerations despite external disturbances, such as aerodynamic drag forces. Hence, prior modeling of aerodynamic effects is not required. We rigorously analyze the proposed control law through response analysis, and we demonstrate it in experiments. The controller enables a quadcopter UAV to track complex 3D trajectories, reaching speeds up to 12.9 m/s and accelerations up to 2.1g, while keeping the root-mean-square tracking error down to 6.6 cm, in a flight volume that is roughly 18 m by 7 m and 3 m tall. We also demonstrate the robustness of the controller by attaching a drag plate to the UAV in flight tests and by pulling on the UAV with a rope during hover.

Index Terms—Flight control, robust control, drone racing, aggressive maneuvering, trajectory following, differential flatness, incremental control, nonlinear dynamic inversion, quadcopter.

SUPPLEMENTAL MATERIAL

A video of the experiments can be found at <https://youtu.be/K15INBAKDCs>.

I. INTRODUCTION

HIGH-SPEED aerial navigation through complex environments has been a focus of control theory and robotics research for decades. More recently, *drone racing*, in which remotely-operated rotary-wing aircraft are piloted through challenging, obstacle-rich courses at very high speeds, has further inspired and popularized this research direction. Development of fully-autonomous drone racers requires accurate control of aircraft during aggressive, *i.e.*, high-speed and agile,

This work was partly supported by Office of Naval Research (ONR) grant N00014-17-1-2670. A preliminary version of this article was presented at 57th IEEE Conference on Decision and Control (CDC 2018) [1].

E. Tal and S. Karaman are with the Department of Aeronautics and Astronautics and the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology (MIT), Cambridge, MA 02139, USA. (e-mail: {atal,sertac}@mit.edu)

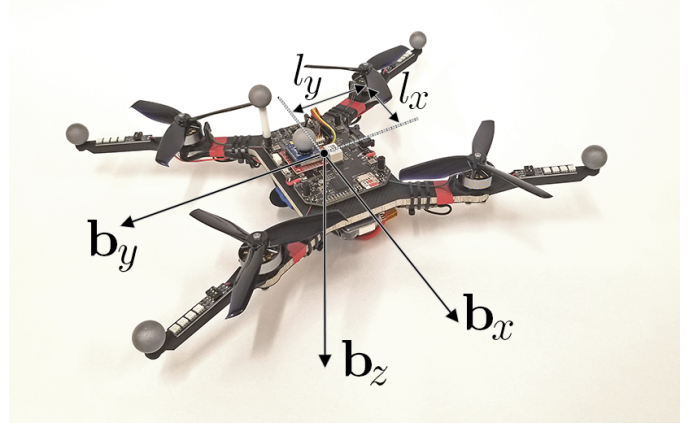


Fig. 1: Quadrotor with body-fixed reference system and motor arm definitions.

maneuvers. At high speeds, aerodynamic drag, which is hard to model, becomes a dominant factor. This poses an important challenge in control design. Additionally, accurate tracking of a reference trajectory with fast-changing acceleration requires considering its higher-order time derivatives, *i.e.*, jerk and snap. In contrast, control design for rotary-wing, vertical take-off and landing (VTOL) aircraft at low speeds typically neglects both aerodynamics and higher-order derivatives.

In this paper, we propose a novel control design for accurate tracking of aggressive trajectories using a quadcopter aircraft, such as the one shown in Fig. 1. The proposed controller generates feedforward control inputs based on differential flatness of the quadcopter dynamics, and uses incremental nonlinear dynamic inversion (INDI) to handle external disturbances, such as aerodynamic drag.

Nonlinear dynamic inversion (NDI), also called feedback linearization, enables the use of a linear control law by transforming the nonlinear dynamics into a linear input-output map [2]–[4]. Although variants of NDI were quickly developed for flight control [5]–[9], it is well known that exact dynamic inversion inherently suffers from lack of robustness [10]. As a result, other nonlinear control methods, *e.g.*, adaptive sliding mode [10]–[12] and backstepping designs [13], have been considered in order to achieve robustness in flight control. More recently, an incremental version of nonlinear dynamic inversion has been developed [14], [15], based on earlier derivations [16], [17], which provide robustness by incrementally applying control inputs based on inertial measurements. In existing literature, the INDI technique has been applied to quadcopters for stabilization, *e.g.*, for robust hovering [18],

[19], but not for trajectory tracking.

Differential flatness, or feedback linearizability, of a dynamics system allows expressing all state and input variables in terms of a set of flat outputs and its derivatives [20]–[24]. In the context of flight control, this property enables reformulation of the trajectory tracking problem as a state tracking problem [9], [25]. Specifically, it enables consideration of higher-order derivatives of the reference trajectory through feedforward state and input references, which has also been applied to control [26]–[29].

Quadcopter aircraft are relatively easy to maneuver and experiment with. Arguably, these qualities make them ideal for drone racing events. For the same reasons, they have been heavily used as experimental platforms in robotics and control theory research since the start of this century [30]–[33]. Complex trajectory tracking control systems have been designed and demonstrated for aircraft in motion capture rooms, where the position and the orientation of the aircraft can be obtained with high accuracy [34]–[40]. Agile maneuvers for quadcopter aircraft have also been demonstrated [41], [42]. Despite being impressive, these demonstrations have showcased complex trajectories only at relatively slow speeds, *e.g.*, less than 2 m/s, so that aerodynamic forces and moments may be neglected. At higher speeds, aerodynamic effects heavily influence the vehicle dynamics. This has been addressed in recent research through modeling [29], [43], [44], estimation [45], [46], and learning [47] of aerodynamic drag effects towards tracking control in high-speed flight.

The main contribution of this paper is a trajectory tracking control design that achieves accurate tracking during high-speed and high-acceleration maneuvers without depending on modeling or estimation of aerodynamic drag parameters. The design exploits differential flatness of the quadcopter dynamics to generate feedforward control terms based on the reference trajectory and its derivatives up to fourth order, *i.e.*, velocity, acceleration, jerk, and snap. Modeling inaccuracies and disturbances due to aerodynamic drag are compensated for using incremental control based on the INDI technique. This control design is novel in the following ways. Firstly, the design incorporates direct tracking of reference snap through accurate control of the motor speeds using optical encoders attached to each motor. We recognize that snap is directly related to vehicle angular acceleration, and thus to the control torque acting on the quadcopter. Accurate application of torque commands is achieved by precise closed-loop control of the motor speeds using measurements from the optical encoders. To the best of our knowledge, the direct control over snap using motor speed measurements is novel. In contrast, trajectory tracking control based on body rate inputs, *e.g.*, using a typical inner-loop flight controller, is incapable of truly considering reference snap. Secondly, we develop a novel INDI control design for quadcopter trajectory tracking. Thrust and torque commands are applied incrementally for robustness against significant external disturbances, such as aerodynamic drag, without the need to model or estimate said disturbances. As far as we are aware, the proposed controller is the first design that is tailored for trajectory tracking, as existing INDI flight control designs focus on state regulation, *e.g.*, for maintaining

hover under external disturbances. Thirdly, we provide and evaluate a novel implementation of INDI angular acceleration control that includes nonlinear computation of the control increments, as opposed to the existing implementations that use inversion of linearized control effectiveness equations. Finally, we demonstrate the proposed controller in experiments, and we rigorously analyze the benefits of the key aspects of our control design through response analysis. In our experiments, the proposed control law enables a unmanned aerial vehicle (UAV) to track complex 3D trajectories, reaching speeds up to 12.9 m/s and accelerations up to 2.1g, while keeping the root-mean-square (RMS) tracking error down to 6.6 cm, in a flight volume that is roughly 18 m long, 7 m wide, and 3 m tall. We also demonstrate the robustness of the controller by attaching a drag plate to the UAV in flight tests and by pulling on the UAV using a tensioned wire during hover. The improved performance due to the tracking of reference jerk and snap through feedforward angular velocity and angular acceleration inputs is also demonstrated both in theoretical analysis and in experiments.

A preliminary version of this paper was previously presented at 57th IEEE Conference on Decision and Control (CDC 2018) [1]. The significant extensions introduced in the current work include a reformulation of the controller using quaternion attitude representation, a more elaborate description of its architecture, the response analysis section in its entirety, and new experimental results at increased speeds. The application of the singularity-free quaternion representation enables tracking of very aggressive trajectories that would incur singular states if an Euler angle representation were used.

The paper is structured as follows: Nomenclature is presented in Table I. In Section II, the quadrotor model is specified, and we show how differential flatness is used to formulate feedforward control inputs in terms of the reference trajectory. In Section III, we describe the architecture of the trajectory tracking controller, and its individual components. Analysis in Section IV illustrates the robustness of INDI and the effect of the feedforward control inputs through response analysis. Finally, we give experimental results from real-life flights in Section V.

II. PRELIMINARIES

In this section, we describe the quadrotor dynamics model, and its differential flatness property. Specifically, we show how the control system utilizes this property to track the reference trajectory jerk and snap through feedforward angular rate and angular acceleration inputs.

A. Quadrotor Model

We consider a 6 degree-of-freedom (DOF) quadrotor, as shown in Fig. 1. The unit vectors depicted in the figure are the basis of the body-fixed reference frame and form the rotation matrix $\mathbf{R} = [\mathbf{b}_x \ \mathbf{b}_y \ \mathbf{b}_z] \in SO(3)$, which gives the transformation from the body-fixed reference frame to the inertial reference frame. The basis of the north-east-down (NED) inertial reference frame consists of the columns of the identity matrix $[\mathbf{i}_x \ \mathbf{i}_y \ \mathbf{i}_z]$.

TABLE I: Nomenclature. The subscript *ref* is used to indicate elements of the reference trajectory function and its time derivatives, as well as feedforward variables directly obtained from the reference trajectory function. The subscript *c* is used for commanded values that are obtained from a feedback control loop. Low-pass filtered measurements and signals obtained from such measurements are indicated by the subscript *f*.

\circ	Hamilton quaternion product	p	polynomial relating motor speeds to throttle inputs
$\bullet^{\circ n}$	n -th Hadamard (element-wise) power	q	vehicle pitch rate around \mathbf{b}_y -axis, rad/s
$[\bullet]_{\times}$	cross-product matrix	\mathbf{r}_{yb}	yaw direction vector in inertial frame
\mathbf{a}	linear acceleration in inertial frame, m/s ²	\mathbf{R}	body-fixed to inertial frame rotation matrix
\mathbf{a}^b	linear acceleration including gravitational acceleration in body-fixed frame, <i>i.e.</i> , as measured by IMU, m/s ²	s	Laplace variable
$\mathbf{b}_x, \mathbf{b}_y, \mathbf{b}_z$	basis vectors of body-fixed frame	s	snap in inertial frame, m/s ⁴
C^n	n -th order differentiability class	\mathbf{S}	angular rate to yaw rate transformation
\mathbf{f}_{ext}	external disturbance force vector in inertial frame, N	$SO(3)$	three-dimensional special orthogonal group
g	gravitational acceleration, m/s ²	t	time, s
\mathbf{G}_1	propeller speed control effectiveness matrix	T	thrust, N
\mathbf{G}_2	propeller acceleration control effectiveness matrix	\mathbb{T}	circle group
$H(s)$	low-pass filter transfer function	\mathbf{v}	velocity in inertial frame, m/s
$\mathbf{i}_x, \mathbf{i}_y, \mathbf{i}_z$	standard basis vectors	\mathbf{x}	position in inertial frame, m
\mathbf{j}	jerk in inertial frame, m/s ³	α	vehicle pitch acceleration around \mathbf{b}_y -axis, rad/s ²
\mathbf{J}	vehicle moment of inertia matrix, kg·m ²	Δ	modeling error parameter
J_{yy}	vehicle moment of inertia around \mathbf{b}_y -axis, kg·m ²	ζ	throttle command vector
J_{rz}	motor rotor and propeller moment of inertia, kg·m ²	θ	vehicle pitch angle, rad
k_θ, k_q	scalar control gains	$\boldsymbol{\mu}$	control moment vector, N·m
k_G	linearized pitch control effectiveness, kg·m ² /(rad·s)	$\boldsymbol{\mu}_{ext}$	external disturbance moment vector, N·m
k_{μ_z}	propeller torque coefficient, kg·m ² /rad ²	$\boldsymbol{\xi}$	normed quaternion attitude vector
k_τ	propeller thrust coefficient, kg·m/rad ²	$\xi^w, \xi^x, \xi^y, \xi^z$	elements of $\boldsymbol{\xi}$
$\mathbf{K}_x, \mathbf{K}_v,$	diagonal control gain matrices	ξ_c	incremental command relative to current attitude
$\mathbf{K}_a, \mathbf{K}_\xi$		ξ_e	vector of error angles in body-fixed frame
$\mathbf{K}_\Omega, \mathbf{K}_{I\omega}$		$\sigma_{ref}(t)$	reference trajectory function, m, rad
l_x	moment arm component parallel to \mathbf{b}_x -axis, m	τ	specific thrust, m/s ²
l_y	moment arm component parallel to \mathbf{b}_y -axis, m	τ_m	motor dynamics time constant, s
m	vehicle mass, kg	ψ	vehicle yaw angle, rad
$M(s)$	motor (control) dynamics transfer function	ω	deviation from hover state motor rotation speed, rad/s
NI	transfer function corresponding to non-incremental controller	ω_0	hover state motor rotation speed, rad/s
		$\boldsymbol{\omega}$	vector of four motor rotation speeds, rad/s
		Ω	vehicle angular velocity in body-fixed frame, rad/s

The vehicle translational dynamics are given by

$$\dot{\mathbf{x}} = \mathbf{v}, \quad (1)$$

$$\dot{\mathbf{v}} = g\mathbf{i}_z + \tau\mathbf{b}_z + m^{-1}\mathbf{f}_{ext}, \quad (2)$$

where \mathbf{x} and \mathbf{v} are the position and velocity in the inertial reference frame, respectively. Equation (2) includes three contributions to the linear acceleration. Firstly, the gravitational acceleration g in downward direction. Secondly, the specific thrust τ , which is the ratio of the total thrust T and the vehicle mass m . Note that the thrust vector is always aligned with the \mathbf{b}_z -axis, so that the quadrotor must pitch or roll to accelerate forward, backward or sideways. Finally, the external disturbance force vector \mathbf{f}_{ext} accounts for all other forces acting on the vehicle, such as aerodynamic drag.

The rotational dynamics are given by

$$\dot{\boldsymbol{\xi}} = \frac{1}{2}\boldsymbol{\xi} \circ \boldsymbol{\Omega}, \quad (3)$$

$$\dot{\boldsymbol{\Omega}} = \mathbf{J}^{-1}(\boldsymbol{\mu} + \boldsymbol{\mu}_{ext} - \boldsymbol{\Omega} \times \mathbf{J}\boldsymbol{\Omega}), \quad (4)$$

where $\boldsymbol{\Omega}$ is the angular velocity in the body-fixed reference frame, and $\boldsymbol{\xi} = [\xi^w \ \xi^x \ \xi^y \ \xi^z]^T$ is the normed quaternion attitude vector, so that $\mathbf{R}\mathbf{x} = \boldsymbol{\xi} \circ \mathbf{x} \circ \boldsymbol{\xi}^{-1}$ with \circ the Hamilton product. Note that a zero magnitude element is implied when multiplying three-element vectors with quaternions. The matrix \mathbf{J} is the vehicle moment of inertia tensor. The control moment vector is indicated by $\boldsymbol{\mu}$, and the external disturbance moment vector by $\boldsymbol{\mu}_{ext}$. The third term of (4) accounts for the conservation of angular momentum.

Each propeller axis is assumed to be aligned perfectly with the \mathbf{b}_z -axis, so that all motor speeds are described by the four-element vector $\boldsymbol{\omega} > 0$. The total thrust T and control moment vector in body-reference frame $\boldsymbol{\mu}$ are given by

$$\begin{bmatrix} \boldsymbol{\mu} \\ T \end{bmatrix} = \mathbf{G}_1\boldsymbol{\omega}^{\circ 2} + \mathbf{G}_2\dot{\boldsymbol{\omega}}, \quad (5)$$

where \circ indicates the Hadamard power;

$$\mathbf{G}_1 = \begin{bmatrix} l_y k_\tau & -l_y k_\tau & -l_y k_\tau & l_y k_\tau \\ l_x k_\tau & l_x k_\tau & -l_x k_\tau & -l_x k_\tau \\ -k_{\mu_z} & k_{\mu_z} & -k_{\mu_z} & k_{\mu_z} \\ -k_\tau & -k_\tau & -k_\tau & -k_\tau \end{bmatrix}, \quad (6)$$

with l_x and l_y the moment arms indicated in Fig. 1, k_τ the propeller thrust coefficient, and k_{μ_z} the propeller torque coefficient; and

$$\mathbf{G}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -J_{rz} & J_{rz} & -J_{rz} & J_{rz} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (7)$$

with J_{rz} the rotor and propeller moment of inertia. The second term in (5) represents the control torque directly due to motor torques. Due to their relatively small moment of inertia, the contribution of the motors to the total vehicle angular momentum may be neglected.

B. Differential Flatness

The controller aims to accurately track the reference trajectory defined by the following function:

$$\boldsymbol{\sigma}_{ref}(t) = [\mathbf{x}_{ref}(t)^T \psi_{ref}(t)]^T, \quad (8)$$

which consists of four differentially flat outputs, *i.e.*, the quadrotor position in the inertial reference frame $\mathbf{x}_{ref}(t) \in \mathbb{R}^3$, and the vehicle yaw angle $\psi_{ref}(t) \in \mathbb{T}$, where \mathbb{T} denotes the circle group. Henceforward, we do not explicitly write the time argument t everywhere.

For (8) to be dynamically feasible, it is required that \mathbf{x}_{ref} is of differentiability class C^4 , *i.e.*, its first four derivatives exist and are continuous, and that ψ_{ref} is of class C^2 . The temporal derivatives of \mathbf{x}_{ref} are successively the reference velocity \mathbf{v}_{ref} , the reference acceleration \mathbf{a}_{ref} , the reference jerk \mathbf{j}_{ref} , and the reference snap \mathbf{s}_{ref} , all in the inertial reference frame. Similarly, temporal differentiation of ψ_{ref} gives the yaw rate $\dot{\psi}_{ref}$, and the yaw acceleration $\ddot{\psi}_{ref}$.

The quadcopter dynamics are differentially flat, so that we can express its states and inputs as a function of $\boldsymbol{\sigma}_{ref}(t)$ and its derivatives. This enables reformulation of the trajectory tracking problem as a state tracking problem. In this section, we derive expressions for the angular rate reference $\boldsymbol{\Omega}_{ref}$, and the angular acceleration reference $\dot{\boldsymbol{\Omega}}_{ref}$ in terms of trajectory jerk, snap, yaw rate, and yaw acceleration. These reference states will be applied as feedforward inputs in the trajectory tracking control design.

Taking the derivative of (2) yields the following expression for jerk:

$$\mathbf{j} = \tau \mathbf{R} [\mathbf{i}_z]_{\times}^T \boldsymbol{\Omega} + \dot{\tau} \mathbf{b}_z, \quad (9)$$

where $[\bullet]_{\times}$ indicates the cross-product matrix, and variations in the unmodeled external force \mathbf{f}_{ext} are neglected. This external force consists chiefly of body drag and rotor drag [48], [49]. Both contributions can be included in the differential flatness transform [29], but the resulting controller will depend on a vehicle-specific aerodynamics model. Instead, we forgo modeling of the external force and instead use sensor-based control to directly compensate for it. Therefore, our controller is able to handle external disturbances without depending on a vehicle-specific model, as described in the next section.

By taking the derivative once more, the following expression for snap is found:

$$\mathbf{s} = \mathbf{R} \left(\ddot{\tau} \mathbf{i}_z + (2\dot{\tau} + \tau [\boldsymbol{\Omega}]_{\times}) [\mathbf{i}_z]_{\times}^T \boldsymbol{\Omega} + \tau [\mathbf{i}_z]_{\times}^T \dot{\boldsymbol{\Omega}} \right). \quad (10)$$

According to typical aerospace convention, we define yaw as the angle between \mathbf{i}_x and the vector

$$\mathbf{r}_{\psi} = [b_x^1 \quad b_x^2 \quad 0]^T \quad (11)$$

with superscripts indicating individual elements of \mathbf{b}_x . Taking the derivative of (11) using $\dot{\mathbf{R}} = \mathbf{R}[\boldsymbol{\Omega}]_{\times}$, we obtain the following expression for the yaw rate:

$$\dot{\psi} = \frac{\mathbf{r}_{\psi} \times \dot{\mathbf{r}}_{\psi}}{\mathbf{r}_{\psi}^T \mathbf{r}_{\psi}} = \underbrace{\frac{\begin{bmatrix} -b_x^2 & b_x^1 \\ \mathbf{r}_{\psi}^T \mathbf{r}_{\psi} \end{bmatrix}}{\mathbf{r}_{\psi}^T \mathbf{r}_{\psi}} \begin{bmatrix} 0 & -b_z^1 & b_y^1 \\ 0 & -b_z^2 & b_y^2 \end{bmatrix}}_{\mathbf{S}} \boldsymbol{\Omega} = \mathbf{S} \boldsymbol{\Omega}, \quad (12)$$

and, by the product rule, the following expression for the yaw acceleration:

$$\ddot{\psi} = \mathbf{S} \dot{\boldsymbol{\Omega}} + \dot{\mathbf{S}} \boldsymbol{\Omega}. \quad (13)$$

An expression for the derivative $\dot{\mathbf{S}}$ is omitted here for brevity, but can be obtained by applying the product rule to the expression for \mathbf{S} given in (12). From (9) and (12), we obtain the angular rate reference

$$\begin{bmatrix} \boldsymbol{\Omega}_{ref} \\ \dot{\tau}_{ref} \end{bmatrix} = \begin{bmatrix} \tau \mathbf{R} [\mathbf{i}_z]_{\times}^T & \mathbf{b}_z \\ \mathbf{S} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{j}_{ref} \\ \dot{\psi}_{ref} \end{bmatrix}, \quad (14)$$

and from (10) and (13) the angular acceleration reference

$$\begin{bmatrix} \dot{\boldsymbol{\Omega}}_{ref} \\ \ddot{\tau}_{ref} \end{bmatrix} = \begin{bmatrix} \tau \mathbf{R} [\mathbf{i}_z]_{\times}^T & \mathbf{b}_z \\ \mathbf{S} & 0 \end{bmatrix}^{-1} \left(\begin{bmatrix} \mathbf{s}_{ref} \\ \dot{\psi}_{ref} \end{bmatrix} - \begin{bmatrix} \mathbf{R}(2\dot{\tau} + \tau [\boldsymbol{\Omega}]_{\times}) [\mathbf{i}_z]_{\times}^T \boldsymbol{\Omega} \\ \dot{\mathbf{S}} \boldsymbol{\Omega} \end{bmatrix} \right). \quad (15)$$

Note that these expressions also contain reference signals for the first and second derivatives of specific thrust. However, as we are unable to command the corresponding first and second derivatives of the motor speed, these references remain unused by the controller.

III. TRAJECTORY TRACKING CONTROL

The control design consists of several components based on various control methods. Table II gives an overview of the components with their respective methodology, references, and control outputs. The control architecture is visualized in three block diagrams. Figure 2 shows the outer-loop position and velocity controller as described in Section III-A. The intermediate control loop shown in Fig. 3 controls linear acceleration, attitude and angular rate, and angular acceleration as described in Section III-B, III-C, and III-D, respectively. Finally, vehicle moment and thrust are directly controlled through closed-loop motor speed control in the inner loop, shown in Fig. 4 and described in Section III-E.

The controller utilizes a vehicle state estimate consisting of position, velocity, and attitude. Additionally, motor speed measurements are obtained from optical encoders, and linear acceleration and angular rate measurements are obtained from the inertial measurement unit (IMU). For the application of incremental angular acceleration control, angular acceleration measurements are obtained by numerical differentiation of the measured angular rate. A low-pass filter (LPF) is required to alleviate the effects of noise, *e.g.*, airframe vibrations, on measurements obtained directly from the IMU. We denote the LPF outputs using the subscript f , *e.g.*, by $\boldsymbol{\Omega}_f$ and $\dot{\boldsymbol{\Omega}}_f$ for the angular rate output and its derivative, respectively. The gravity-corrected LPF acceleration output in the inertial reference frame is obtained as follows:

$$\mathbf{a}_f = (\mathbf{R} \mathbf{a}^b + g \mathbf{i}_z)_f. \quad (16)$$

A. PD Position and Velocity Control

Position and velocity control is based on two cascaded proportional-derivative (PD) controllers. The resulting con-

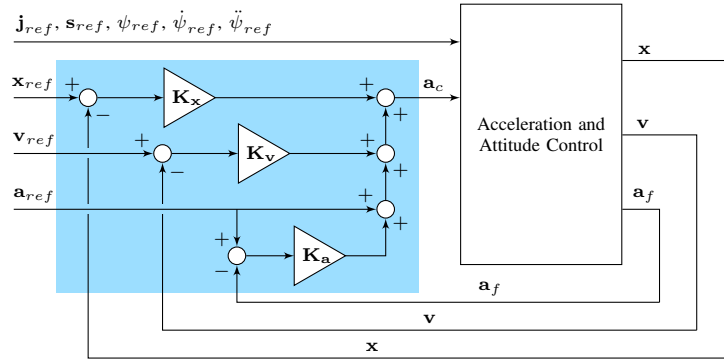


Fig. 2: Position and Velocity Control. The blue area contains the PD control design as described in Section III-A.

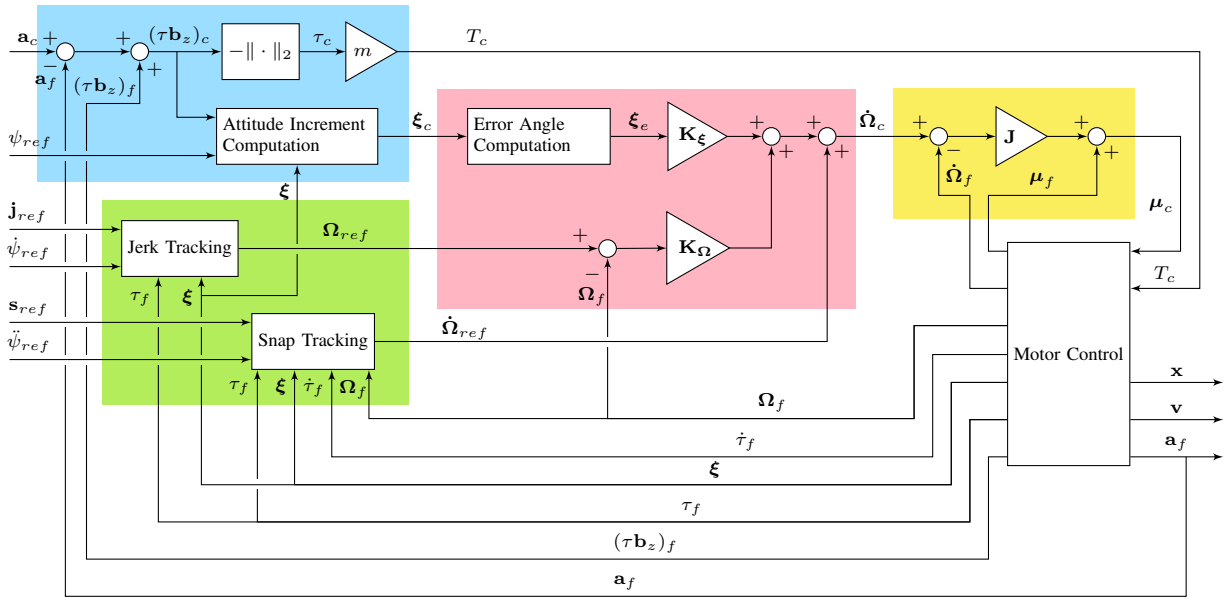


Fig. 3: Acceleration and Attitude Control. The blue area contains the INDI linear acceleration and yaw control as described in Section III-B. The green area contains the computation of angular rate and angular acceleration references based on differential flatness as described in Section II-B. The red area contains the attitude and angular rate control as described in Section III-C. The yellow area contains the INDI angular acceleration control as described in Section III-D.

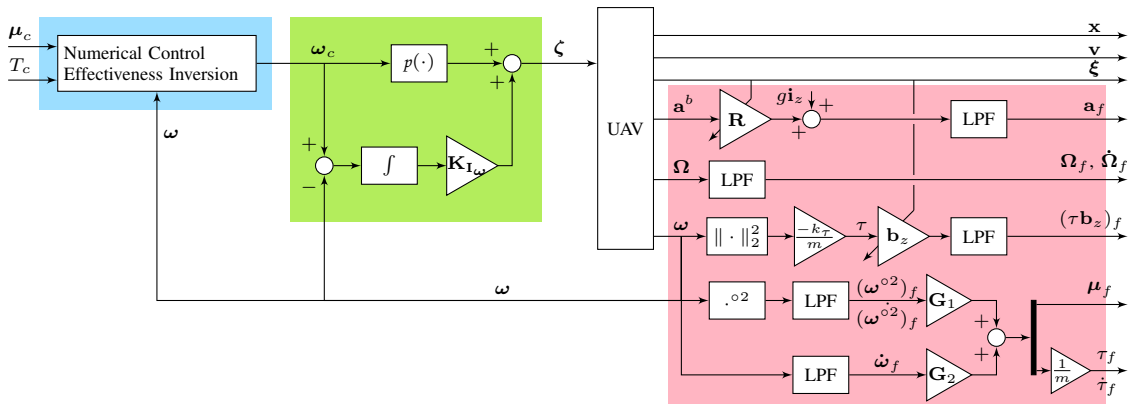


Fig. 4: Motor Control and Computation of Filtered Signals. The blue and green areas contain the moment and thrust control (including motor speed command saturation resolution), and the motor speed control, respectively. Both are described in Section III-E. The UAV block represents the UAV hardware, including ESCs, motors, and sensors. The red area contains the computation of filtered signals based on IMU and optical encoder measurements.

TABLE II: Overview of trajectory tracking controller components.

Component	Methodology	Reference	Control Output	Description
Position and Velocity Control	PD	$\mathbf{x}_{ref}, \mathbf{v}_{ref}, \mathbf{a}_{ref}$	\mathbf{a}_c	Section III-A
Linear Acceleration and Yaw Control	INDI	\mathbf{a}_c, ψ_{ref}	ξ_c, T_c	Section III-B
Jerk and Snap Tracking	Differential Flatness	$\dot{\mathbf{j}}_{ref}, \mathbf{s}_{ref}, \dot{\psi}_{ref}, \ddot{\psi}_{ref}$	$\Omega_{ref}, \dot{\Omega}_{ref}$	Section II-B
Attitude and Angular Rate Control	PD	$\xi_c, \Omega_{ref}, \dot{\Omega}_{ref}$	$\dot{\Omega}_c$	Section III-C
Angular Acceleration Control	INDI	$\dot{\Omega}_c$	μ_c	Section III-D
Moment and Thrust Control	Inversion	μ_c, T_c	ω_c	Section III-E
Motor Speed Control	Integrative	ω_c	ζ	Section III-E

troller is mathematically equivalent to the following single expression:

$$\mathbf{a}_c = \mathbf{K}_x (\mathbf{x}_{ref} - \mathbf{x}) + \mathbf{K}_v (\mathbf{v}_{ref} - \mathbf{v}) + \mathbf{K}_a (\mathbf{a}_{ref} - \mathbf{a}_f) + \mathbf{a}_{ref} \quad (17)$$

with \mathbf{K}_\bullet indicating diagonal gain matrices. The subscript *ref* is used to indicate values obtained directly from the reference trajectory. In contrast, the subscript *c* indicates commanded values that are computed in one of the control loops. For example, \mathbf{a}_{ref} is obtained directly from the reference trajectory as the second derivative of \mathbf{x}_{ref} , while \mathbf{a}_c is computed based on (17) and includes terms based on the position, velocity, and acceleration deviations. The first three terms in (17) ensure tracking of position and velocity references, while the final term serves as a feedforward input to ensure tracking of the reference acceleration. The control utilizes the inertial reference frame with — in our implementation — identical gains for the horizontal \mathbf{i}_x - and \mathbf{i}_y -directions, but separately tuned gains for the vertical \mathbf{i}_z -direction. The commanded acceleration is used to calculate thrust and attitude commands, as will be shown in the next section.

B. INDI Linear Acceleration and Yaw Control

Existing literature presents the derivation of an INDI linear acceleration controller using Taylor series approximation [19]. In this section, we arrive at equivalent control equations through an intuitive derivation that follows the practical working of the INDI notion based on estimation of the external force acting on the quadrotor.

An expression for the external force in terms of measured acceleration and specific thrust is obtained by rewriting (2), as follows:

$$\mathbf{f}_{ext} = m (\mathbf{a}_f - (\tau \mathbf{b}_z)_f - g \mathbf{i}_z), \quad (18)$$

where τ is the specific thrust calculated according to (5) using motor speed measurements. Identical LPFs must be used to ensure that equal phase lag is incurred by acceleration and thrust measurements [18]. Note that the specific thrust vector and the linear acceleration (cf. (16)) are both transformed to the inertial reference frame prior to filtering. This order is appropriate because the external force in the inertial reference frame \mathbf{f}_{ext} is assumed to be slow-changing relative to the LPF dynamics, as described in Section II-B. Substitution of

(18) into (2) gives the following expression for the current acceleration:

$$\begin{aligned} \mathbf{a} &= \tau \mathbf{b}_z + g \mathbf{i}_z + m^{-1} \mathbf{f}_{ext} \\ &= \tau \mathbf{b}_z + g \mathbf{i}_z + m^{-1} (m (\mathbf{a}_f - (\tau \mathbf{b}_z)_f - g \mathbf{i}_z)) \\ &= \tau \mathbf{b}_z - (\tau \mathbf{b}_z)_f + \mathbf{a}_f. \end{aligned} \quad (19)$$

The specific thrust vector command that results in the commanded acceleration prescribed by (17) can be computed using the following incremental relation based on (19):

$$(\tau \mathbf{b}_z)_c = (\tau \mathbf{b}_z)_f + \mathbf{a}_c - \mathbf{a}_f. \quad (20)$$

The incremental nature of (20) enables the controller to achieve the commanded acceleration despite possible disturbances or modeling errors. If the commanded value is not obtained immediately, the thrust and attitude commands will be incremented further in subsequent control updates. This principle eliminates the need for integral action anywhere in the control design.

The thrust magnitude command is obtained as

$$T_c = -m \|(\tau \mathbf{b}_z)_c\|_2 \quad (21)$$

with the negative sign following from the definition that thrust is positive in \mathbf{b}_z -direction. The incremental attitude command ξ_c represents the rotation from the current attitude to the commanded attitude and is obtained in two steps: first, the minimum rotation to align $-\mathbf{b}_z$ with the thrust vector command $(\tau \mathbf{b}_z)_c$ is obtained; second, a rotation around \mathbf{b}_z is added to satisfy the yaw reference ψ_{ref} . For the first step, we transform the normalized thrust vector command to the current body-fixed reference frame, as follows:

$$(-\mathbf{b}_z)_c^b = \xi^{-1} \circ (-\mathbf{b}_z)_c \circ \xi. \quad (22)$$

The appropriate rotation to align the current $-\mathbf{b}_z$ with $(\tau \mathbf{b}_z)_c$ is then given by

$$\bar{\xi}_c = \left[\begin{array}{c} 1 - \widehat{\mathbf{i}_z^T (-\mathbf{b}_z)_c^b} \\ -\mathbf{i}_z \times (-\mathbf{b}_z)_c^b \end{array} \right], \quad (23)$$

where hat refers to quaternion normalization, *i.e.*, $\hat{\xi} = \xi / \|\xi\|_2$. For the second step, the yaw reference normal vector is first transformed to the intermediate attitude command frame, as follows:

$$\begin{aligned} \bar{\mathbf{n}}_{\psi_{ref}} &= (\xi \circ \bar{\xi}_c)^{-1} \\ &\circ \left[\begin{array}{ccc} \sin \psi_{ref} & -\cos \psi_{ref} & 0 \end{array} \right]^T \circ (\xi \circ \bar{\xi}_c). \end{aligned} \quad (24)$$

Next, we obtain the following rotation that makes \mathbf{b}_x coincide with the plane defined by normal vector $\bar{\mathbf{n}}^{\psi_{ref}}$:

$$\xi_\psi = \begin{bmatrix} 1 & 0 & 0 & -\frac{\bar{n}_x^1 \psi_{ref}}{\bar{n}_z^2 \psi_{ref}} \end{bmatrix}^T. \quad (25)$$

Equation (25) implicitly selects between tracking of ψ_{ref} and $\psi_{ref} + \pi$ rad based on minimizing the magnitude of rotation. Due to continuity of ψ_{ref} this does not cause any unwanted switching, but it does prevent unwanted discontinuities such as a π rad rotation around \mathbf{b}_z to maintain yaw tracking when pitching through $\pm\pi/2$ rad. Note that (23) and (25) incur singularities if $\mathbf{i}_z = (-\mathbf{b}_z)_c^b$ and $\bar{n}_z^2 \psi_{ref} = 0$, respectively. However, by computing the attitude command relative to the current attitude we move these singularities far away from the nominal trajectory. Moreover, they are straightforwardly detected and resolved by selecting any direction of rotation. Finally, the incremental attitude command is obtained as

$$\xi_c = \bar{\xi}_c \circ \xi_\psi. \quad (26)$$

C. PD Attitude and Angular Rate Control

In this section, we describe the attitude and angular rate controller. This controller specifies the angular rate command and is thus solely based on angular kinematics. This has two major advantages compared to incorporating control torque or motor speeds. Firstly, the attitude controller does not take into account any model-specific parameters, such as the vehicle inertia matrix \mathbf{J} . Therefore the control design avoids discrepancies due to model mismatches and has vehicle-independent gains. Secondly, accurate torque control cognizant of the external moment μ_{ext} can be performed separately using sensor-based INDI, as described in Section III-D. This eliminates the need to incorporate a complicated disturbance model in the attitude controller, which further improves controller robustness and simplicity.

The three-element angle vector ξ_e associated with the incremental attitude command ξ_c is computed as follows:

$$\xi_e = \frac{2 \arccos \xi_c^w}{\sqrt{1 - \xi_c^w \xi_c^w}} \begin{bmatrix} \xi_c^x & \xi_c^y & \xi_c^z \end{bmatrix}^T. \quad (27)$$

Using these error angles, the angular acceleration command is obtained as

$$\dot{\Omega}_c = \mathbf{K}_\xi \xi_e + \mathbf{K}_\Omega (\Omega_{ref} - \Omega_f) + \dot{\Omega}_{ref}, \quad (28)$$

where Ω_{ref} and $\dot{\Omega}_{ref}$ are the angular velocity and angular acceleration feedforward terms defined in (14) and (15), respectively. The resulting attitude controller not only tracks the attitude command, but also angular rate and acceleration. This enables tracking of trajectory jerk and snap, which is essential for accurate tracking of aggressive trajectories, as will be shown analytically in Section IV and experimentally in Section V. In contrast, trajectory tracking control based on body rate inputs, *e.g.*, using an off-the-shelf flight controller, is incapable of truly considering reference snap, because snap corresponds to the vehicle angular acceleration, as shown in (15).

D. INDI Angular Acceleration Control

Robust tracking of the angular acceleration command $\dot{\Omega}_c$ is achieved through INDI control. We rewrite (4) into the following expression for the external moment based on the measured angular rate, angular acceleration, and control moment:

$$\mu_{ext} = \mathbf{J} \dot{\Omega}_f - \mu_f + \Omega_f \times \mathbf{J} \Omega_f \quad (29)$$

with μ_f the control moment in the body-fixed reference frame, obtained from the measured motor speeds by (5) and low-pass filtering. Analogous to the external force in Section III-B, the external moment μ_{ext} is assumed slow-changing with regard to the LPF dynamics. Substitution of (29) into (4) then gives:

$$\begin{aligned} \dot{\Omega} &= \mathbf{J}^{-1}(\mu + \mu_{ext} - \Omega \times \mathbf{J} \Omega) \\ &= \mathbf{J}^{-1}(\mu + (\mathbf{J} \dot{\Omega}_f - \mu_f + \Omega_f \times \mathbf{J} \Omega_f) - \Omega \times \mathbf{J} \Omega) \\ &= \dot{\Omega}_f + \mathbf{J}^{-1}(\mu - \mu_f). \end{aligned} \quad (30)$$

In (30), it is assumed that the difference between the gyroscopic angular momentum term and its filtered counterpart is sufficiently small to be neglected, because the term is relatively slow changing compared to the angular acceleration and control moment, and moreover is second-order. By inversion of the final line, we obtain the following incremental expression for the commanded control moment:

$$\mu_c = \mu_f + \mathbf{J} (\dot{\Omega}_c - \dot{\Omega}_f). \quad (31)$$

E. Inversion-Based Moment and Thrust Control, and Integrative Motor Speed Control

In Section III-B and Section III-D, we have found expressions for the commanded thrust T_c and control moment μ_c , respectively. Tracking of these commands requires control of the motor speeds, as evidenced by the direct relation given in (5). State-of-the-art INDI implementations for quadrotors are based on linearization of this relation and do not accurately model transient behavior [18], [19]. Our proposed implementation is based on a nonlinear inversion of the control effectiveness and explicitly incorporates the motor response time constant, as such it provides a more accurate computation of control inputs.

In order to achieve fast and accurate closed-loop motor speed control, we employ optical encoders that measure the motor speeds. The availability of motor speed measurements furthermore enables accurate calculation of the thrust and control moment, as required by the INDI controller in (20) and (31). In practice, the optical encoder, shown in Fig. 5, measures the motor rotational speed by detecting the passage of stripes on a reflective strip attached to the motor hub. As such, the optical encoder provides a high-rate, accurate, lightweight, and unintrusive manner to obtain the motor speed.

The motor speed corresponding to the commanded thrust and control moment is found by inverting the nonlinear control effectiveness equation (5). In order to do so, we estimate the effect of the motor speed command on the motor speed derivative using the following first-order model:

$$\dot{\omega} = \tau_m^{-1}(\omega_c - \omega) \quad (32)$$

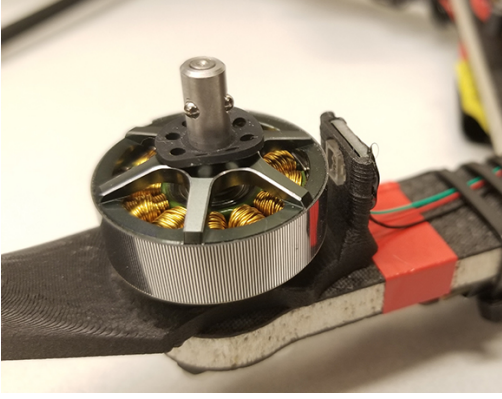


Fig. 5: Motor (propeller removed) with optical encoder for rotational speed measurement. Note the optical encoder lens on the right, and the accompanying reflective strip on the motor hub.

with τ_m the motor dynamics time constant. After equating to the control moment and thrust commands, the resulting equation,

$$\begin{bmatrix} \mu_c \\ T_c \end{bmatrix} = \mathbf{G}_1 \omega_c^2 + \tau_m^{-1} \mathbf{G}_2 (\omega_c - \omega), \quad (33)$$

can be solved numerically, *e.g.*, using Newton's method. Inversion of this nonlinear control effectiveness relation improves the accuracy of thrust and control moment tracking, when compared to the linearized inversion that does not consider the motor transient response as given by (52).

Inversion of (33) may lead to infeasible, *i.e.*, saturated, motor speed commands. We address this first by altering the control moment around \mathbf{b}_z . Since the control effectiveness is relatively much smaller around this axis, this is most likely to resolve the command saturation. Moreover, it typically least affects vehicle stability and position tracking, since rotation purely around the \mathbf{b}_z -axis does not alter the thrust vector. Let $\underline{\omega}$ and $\bar{\omega}$ be respectively the minimum and maximum feasible motor speeds, then the set of \mathbf{b}_z control momenta — excluding J_{r_z} contributions — that result in feasible motor speed commands is

$$\begin{aligned} & \max \left\{ \frac{k_{\mu_z}}{k_\tau} \left(4k_\tau \omega^2 + T_c \pm \left(\frac{\mu_c^y}{l_x} - \frac{\mu_c^x}{l_y} \right) \right), \right. \\ & \quad \left. - \frac{k_{\mu_z}}{k_\tau} \left(4k_\tau \bar{\omega}^2 + T_c \pm \left(\frac{\mu_c^y}{l_x} + \frac{\mu_c^x}{l_y} \right) \right) \right\} \leq \mu_c^z \\ & \leq \min \left\{ \frac{k_{\mu_z}}{k_\tau} \left(4k_\tau \bar{\omega}^2 + T_c \pm \left(\frac{\mu_c^y}{l_x} - \frac{\mu_c^x}{l_y} \right) \right), \right. \\ & \quad \left. - \frac{k_{\mu_z}}{k_\tau} \left(4k_\tau \omega^2 + T_c \pm \left(\frac{\mu_c^y}{l_x} + \frac{\mu_c^x}{l_y} \right) \right) \right\}. \quad (34) \end{aligned}$$

If this set is non-empty, we set μ_c^z to equal the boundary closest to the original moment command. The motor speed command is then obtained as

$$\omega_c = \left(\mathbf{G}_1^{-1} \begin{bmatrix} \mu_c \\ T_c \end{bmatrix} \right)^{\circ \frac{1}{2}}. \quad (35)$$

Note that due to J_{r_z} contributions the actual \mathbf{b}_z control moment will not exactly be equal to μ_c^z . However, we still obtain

the feasible control moment that is closest to the original commanded moment, because k_{μ_z} and J_{r_z} have identical signs in (6) and (7), respectively. If there exists no μ_c^z that results in feasible motor commands, we consider a reduction or limited increase in the thrust magnitude command T_c based on the reasoning that application of thrust is only effective in the correct direction, *i.e.*, at the correct vehicle pitch and roll. Since adjustment of T_c results in equal magnitude shift of the constraints, it is straightforward to verify whether there exists an acceptable value of T_c such that the lower and upper boundaries in (34) coincide. If so, T_c is set to this value and μ_c^z to the feasible point, after which (35) is used to compute the motor speed commands. If not, μ_c^z is set to the average of the lower and upper boundaries in (34), and any infeasible motor speed commands resulting from (35) are clipped.

Finally, the throttle vector ζ that contains the motor electronic speed control (ESC) commands is obtained as follows:

$$\zeta = p(\omega_c) + \mathbf{K}_{I\omega} \int \omega_c - \omega dt \quad (36)$$

with p a vector-valued polynomial function relating motor speeds to throttle inputs. This function was obtained by regression analysis of static test data. Integral action is added to account for changes in this relation due to decreasing battery voltage. The measured motor speed signal ω remains unfiltered here to minimize phase lag.

IV. RESPONSE ANALYSIS

Incremental control, and the tracking of high-order reference derivatives are two key aspects of our control design. In this section, we theoretically verify the advantages of these features. Namely, the improved robustness of incremental control in comparison to non-incremental control, and the improved trajectory tracking accuracy due to the consideration of high-order reference trajectory derivatives, *i.e.*, jerk and snap. The purpose of this section is to provide an intuitive understanding of how these aspects improve tracking performance. In order to analyze the behavior of the closed-loop system, we use linearized dynamics and control equations, as the resulting simplifications allow for easier qualitative interpretation. However, the observations in this section also apply to the full, nonlinear dynamics and control equations. Our findings are validated and quantitatively assessed using real-life flights in Section V.

We consider forward and pitch movement around the hover state. The subscript x indicates the forward component, *e.g.*, $a_{x,ref} = \mathbf{a}_{ref}^T \mathbf{i}_x$, and the subscript y the pitch component, *e.g.*, $\mu_y = \boldsymbol{\mu}^T \mathbf{i}_y$. In hover condition, $\tau = -g$, $\theta = 0$, and $\boldsymbol{\Omega} = \mathbf{0}_{3 \times 1}$, so that (2) and (4) can be linearized to obtain

$$a_x = -g\theta + m^{-1} f_{x,ext}, \quad (37)$$

$$J_{yy} \dot{q} = \mu_y + \mu_{y,ext}, \quad (38)$$

where θ is the pitch angle, and J_{yy} is the vehicle moment of inertia about the \mathbf{b}_y -axis. Similarly, the INDI linear acceleration control law (20) is linearized to obtain the error angle

$$-g\theta_e = -g\theta_f + a_{x,ref} - (a_x)_f + g\theta, \quad (39)$$

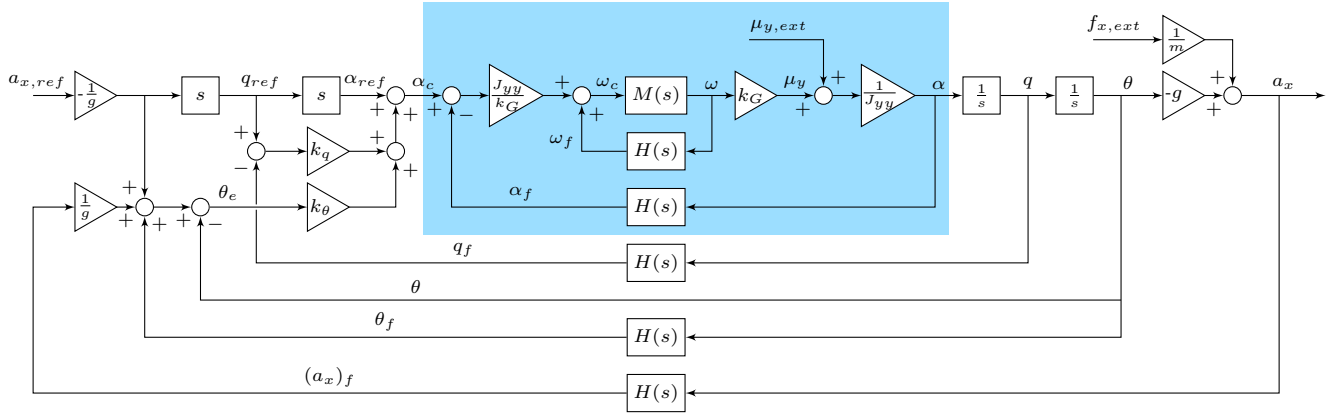


Fig. 6: Linearized closed-loop forward acceleration dynamics, with pitch acceleration dynamics in blue area.

where $-g\theta_f$ represents the forward component of the specific thrust vector, and $(a_x)_f$ represents the filtered forward acceleration as obtained by (16). The commanded pitch acceleration α_c is obtained by taking the pitch component of (28), as follows:

$$\alpha_c = k_\theta \theta_e + k_q (q_{ref} - q_f) + \alpha_{ref}, \quad (40)$$

where $q_{ref} = -\frac{j_{x,ref}}{g}$ and $\alpha_{ref} = -\frac{s_{x,ref}}{g}$ by linearization of (14) and (15). The scalar control gains k_θ and k_q are obtained by selecting the pitch elements from the corresponding control gain matrices described in Section III.

Next, we linearize the angular acceleration and moment control laws. The four motors can be modeled collectively, as the system is linearized around the hover state where all motors have identical angular speeds. The scalar value ω refers to the deviation from the hover state motor speed ω_0 , or, equivalently, to half of the angular speed difference between the front and rear motor pairs. Equating (31) and (33), and isolating the pitch channel gives

$$\omega_c = \sqrt{((\omega_0 + \omega)^2)_f + J_{yy}(4l_x k_\tau)^{-1}(\alpha_c - \alpha_f)} - \omega_0. \quad (41)$$

with the factor 4 due to the number of motors. Linearization around the hover state gives

$$\omega_c = \omega_f + J_{yy} k_G^{-1} (\alpha_c - \alpha_f), \quad (42)$$

with the linearized control effectiveness gain $k_G = 8\omega_0 l_x k_\tau$, so that $\mu_y = k_G \omega$.

In order to analyze the robustness properties provided by the proposed incremental controller, it is compared to a regular, *i.e.*, non-incremental, controller with linearized equations (cf. (39) and (42))

$$\theta_{c,NI} = -\frac{a_{x,ref}}{g}, \quad \omega_{c,NI} = \frac{J_{yy}}{k_G} \alpha_c, \quad (43)$$

where α_c is still given by (40) using $\theta_e = \theta_c - \theta$, and the subscript *NI* is used to indicate the non-incremental controller.

A. Robustness against Disturbance Forces and Moments

An overview of the resulting linearized closed-loop acceleration dynamics is given in Fig. 6. From the blue area, we obtain the following pitch acceleration dynamics:

$$\frac{\alpha}{\alpha_c}(s) = \frac{J_{yy} k_G^{-1} \frac{M(s)}{1-M(s)H(s)} k_G J_{yy}^{-1}}{1 + J_{yy} k_G^{-1} \frac{M(s)}{1-M(s)H(s)} k_G J_{yy}^{-1} H(s)} = M(s), \quad (44)$$

$$\frac{\alpha}{\mu_{y,ext}}(s) = \frac{J_{yy}^{-1}}{1 + \frac{H(s)M(s)}{1-H(s)M(s)}} = J_{yy}^{-1} (1 - H(s)M(s)) \quad (45)$$

with $\alpha(s)$ the pitch acceleration, *i.e.*, $\alpha(s) = sq(s) = s^2\theta(s)$. The LPF transfer function is denoted by $H(s)$, *e.g.*, $\frac{\alpha_f}{\alpha}(s) = H(s)$, and the motor (control) dynamics are denoted by $M(s)$, *i.e.*, $\frac{\omega}{\omega_c}(s) = M(s)$. In (44), we observe that the closed-loop angular acceleration dynamics are solely determined by the motor dynamics [18]. Hence, the aggressiveness of trajectories that can be tracked is theoretically limited by only the bandwidth of the motor response. This is also the case for a non-incremental version of the controller.

The disturbance moment $\mu_{y,ext}$ is fully counteracted using incremental control based on the two feedback loops in the blue shaded area of Fig. 6: the expected angular acceleration from the motor speeds, *i.e.*, $\frac{k_G}{J_{yy}}\omega_f$, and the measured angular acceleration α_f , which includes the effects of the disturbance moment. As shown in (45), the counteraction depends on $H(s)$ and $M(s)$ so that the ability to reject disturbances is limited by the bandwidth of both the LPFs and the motors. To the contrary, in a non-incremental controller the α_f and ω_f feedback loops are not present, so that $\alpha = J_{yy}^{-1}\mu_{y,ext}$ (cf. (45)). The disturbance moment now propagates undamped to the attitude and position control loops, as there is no closed-loop angular acceleration control that directly evaluates the moments acting on the vehicle.

We obtain similar results for the disturbance force $f_{x,ext}$, which is corrected for incrementally using the difference between the acceleration due to thrust, *i.e.*, $-g\theta_f$, and the true acceleration including the disturbance force, *i.e.*, $(a_x)_f$. All in all, the proposed incremental controller maintains identical nominal reference tracking performance for both angular and linear accelerations, while achieving superior disturbance

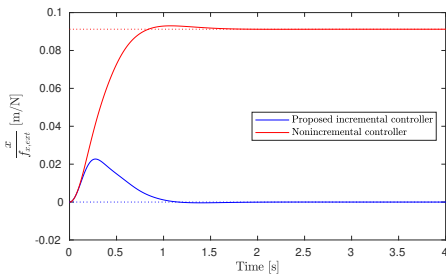
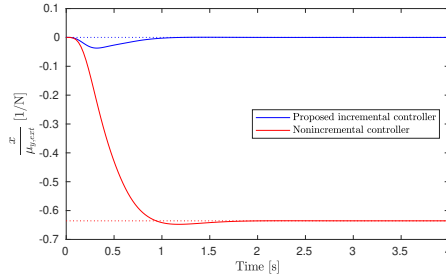
(a) Position response to $f_{x,ext}$ step input.(b) Position response to $\mu_{y,ext}$ step input.

Fig. 7: Simulated disturbance response using the proposed incremental controller, and a non-incremental controller.

TABLE III: Trajectory tracking controller gains.

Gain	Value
\mathbf{K}_x	diag ([18 18 13.5])
\mathbf{K}_v	diag ([7.8 7.8 5.9])
\mathbf{K}_a	diag ([0.5 0.5 0.3])
\mathbf{K}_ξ	diag ([175 175 82])
$\mathbf{K}_{\dot{\xi}}$	diag ([19.5 19.5 19.2])

rejection of external moments and forces when compared to the non-incremental controller.

In order to evaluate the effect of the disturbance force and moment on the position tracking error, we close the loop around Fig. 6 using the position and velocity controller given by (17). Figure 7 shows the resulting step responses for both incremental and non-incremental control. The response was simulated using the platform-independent control gains given in Table III, a second-order Butterworth filter with cut-off frequency equal to 188.5 rad/s (30 Hz), and the first-order motor model given by (32) with τ_m set to 20 ms. It can be seen that the proposed incremental controller is able to counteract the disturbances and reaches zero steady-state error, while the non-incremental controller is unable to do so. In order to null the steady-state errors due to force and moment disturbances, integral action must be added to the non-incremental controller. This is not necessary in the case of INDI, so that our proposed control design is able to quickly and wholly counteract disturbance forces and moments, while avoiding the negative effects that integral action typically has on the tracking performance, *e.g.*, degraded stability, and increased overshoot and settling time.

B. Robustness against Modeling Errors

The proposed control design requires only few vehicle-specific parameters. Nonetheless, it is desirable that tracking performance is maintained if inaccurate parameters are used, *e.g.*, because control effectiveness data obtained from static tests may not be representative for the entire flight envelope. The linearized control equations described above incorporate the ratio of the moment of inertia J_{yy} and the linearized control effectiveness k_G . We denote the values used in the

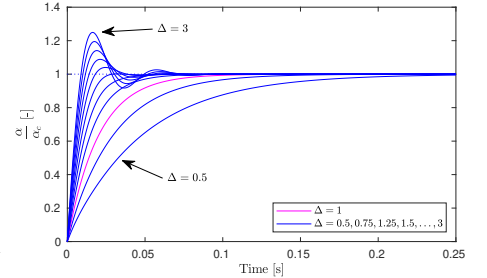


Fig. 8: Simulated angular acceleration step response for various modeling errors using the proposed incremental controller.

controller \bar{J}_{yy} and \bar{k}_G , and define the modeling error Δ such that

$$\frac{\bar{J}_{yy}}{\bar{k}_G} = \Delta \frac{J_{yy}}{k_G}. \quad (46)$$

This leads to the following pitch acceleration dynamics for the proposed incremental NDI controller, and the non-incremental controller described above:

$$\frac{\alpha}{\alpha_c}(s) = \frac{\Delta M(s)}{(\Delta - 1)H(s)M(s) + 1}, \quad (47)$$

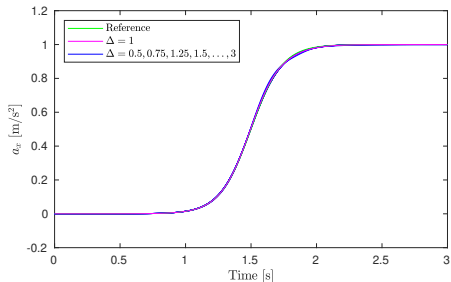
$$\frac{\alpha}{\alpha_{cNI}}(s) = \Delta M(s). \quad (48)$$

It can be seen that the error acts as a simple gain in the non-incremental controller, leading to an incorrect angular acceleration. On the contrary, the proposed incremental controller compares the expected angular acceleration from the motor speeds, *i.e.*, $\frac{\bar{k}_G}{\bar{J}_{yy}}\omega_f$, to the measured angular acceleration, *i.e.*, α_f , to implicitly correct for the modeling error. The corresponding angular acceleration responses for several values of Δ are shown in Fig. 8. The figure shows that the modeling error affects the transient response, but that the incremental controller is able to correct for it and quickly reaches the commanded acceleration value even for very large model discrepancies.

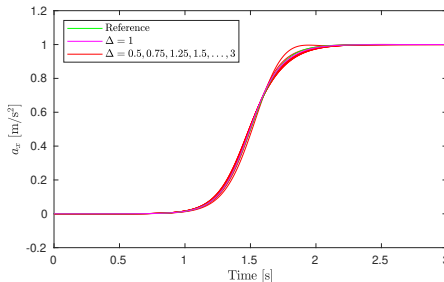
In order to assess the effect of modeling errors on acceleration tracking, we simulate the time response to the following acceleration reference:

$$a_{x,ref}(t) = \frac{1}{2} \tanh\left(\frac{4}{3}\pi t - 2\pi\right) + \frac{1}{2}, \quad (49)$$

which is C^2 , *i.e.*, the corresponding jerk and snap signals are continuous, and has boundary conditions $a_{x,ref}(0) = \dot{a}_{x,ref}(0) = \ddot{a}_{x,ref}(0) = \ddot{a}_{x,ref}(3) = 0$ and $a_{x,ref}(3) = 1 \text{ m/s}^2$. The responses for various values of Δ are shown in Fig. 9. It can be seen that the incremental controller is able to accurately track the reference signal even when large modeling errors are present. When non-incremental control is used, the tracking performance declines more severely with growing modeling error.



(a) Using the proposed incremental controller.



(b) Using a non-incremental controller.

Fig. 9: Simulated linear acceleration tracking response for various modeling errors.

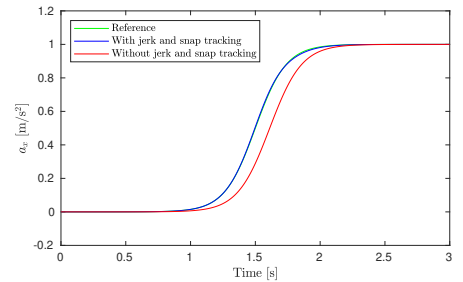


Fig. 10: Simulated linear acceleration tracking response using the proposed controller with and without jerk and snap tracking.

C. Jerk and Snap Tracking

Jerk and snap tracking is a crucial aspect of the proposed controller design that enables tracking of fast-changing acceleration references. It is embodied by the feedforward terms $k_q s$ and s^2 in the nominator of the acceleration response transfer function

$$\frac{a_x}{a_{x,ref}}(s) = \frac{M(s)(s^2 + k_q s + k_\theta)}{s^2 + k_q H(s)M(s) + k_\theta M(s)}. \quad (50)$$

These feedforward terms add two zeros to the closed-loop transfer function. These zeros — in combination with the LPF — act essentially as a lead compensator and help improve the transient response of the system. Effective placement of the zeros through tuning of k_q leads to improved tracking of a rapidly changing acceleration input signal, *e.g.*, during aggressive flight maneuvers.

Figure 10 shows the simulated acceleration responses with and without jerk and snap tracking to the reference signal defined in (49). It can be seen that the inclusion of jerk and snap tracking causes a faster response, resulting in more accurate acceleration tracking. In the next section, we show that the improvement is also achieved in practice.

V. EXPERIMENTAL RESULTS

In this section, experimental results for high-speed, high-acceleration flight are presented. A video of the experiments is available at <https://youtu.be/K15INBAKDCs>. We evaluate the performance of the trajectory tracking controller on two trajectories that include yawing, tight turns with acceleration up to over 2g, and high-speed straights at up to 12.9 m/s.

TABLE IV: 3D trajectory tracking performance for experiments with forward yaw and constant yaw.

	Forward yaw	Constant yaw
RMS $\ \mathbf{x} - \mathbf{x}_{ref}\ _2$ [cm]	6.6	6.1
max $\ \mathbf{x} - \mathbf{x}_{ref}\ _2$ [cm]	10.8	11.9
RMS $ \psi - \psi_{ref} $ [deg]	5.1	1.9
max $ \psi - \psi_{ref} $ [deg]	12.8	6.4
RMS $\ \mathbf{v}\ _2$ [m/s]	6.8	5.6
max $\ \mathbf{v}\ _2$ [m/s]	12.9	11.3
RMS $\ \mathbf{a} - g\mathbf{i}_z\ _2$ [m/s ²]	14.4	12.5
max $\ \mathbf{a} - g\mathbf{i}_z\ _2$ [m/s ²]	20.8	20.0

Furthermore, we examine the effect of the feedforward inputs based on the reference trajectory jerk and snap. We establish the independence of any model-based drag estimate by attaching a drag-inducing cardboard plate that more than triples the frontal area of the vehicle. Robustness against external disturbance forces is further displayed by pulling on a string attached to the quadcopter in hover. Finally, we compare the proposed nonlinear INDI angular acceleration control to its linearized counterpart.

A. Experimental Setup

Experiments were performed in an indoor flight room using the quadcopter shown in Fig. 1. The quadrotor body is machined out of carbon fiber composite with balsa wood core. The propulsion system consists of T-Motor F35A ESCs and F40 Pro II Kv 2400KV motors with Gemfan Hulkie 5055 propellers. Adjacent motors are mounted 18 cm apart. The quadcopter is powered by a single 4S LiPo battery. Its total flying mass is 609 g.

Control computations are performed at 2000 Hz using an onboard STM32H7 400 MHz microcontroller running custom firmware. On this platform, the total computation time of a control update at 32-bit floating point precision is 16 μ s. Linear acceleration and angular rate measurements are obtained from an onboard Analog Devices ADIS16477-3 IMU at 2000 Hz, while position, velocity, and orientation measurements are obtained from an OptiTrack motion capture system at 360 Hz with an average latency of 18 ms. The latency is corrected for by propagating motion capture data using integrated IMU measurements. Motor speed measurements are obtained from

TABLE V: Roulette curve trajectory tracking performance for: (i) the proposed controller; (ii) jerk and snap tracking disabled; and (iii) drag plate attached.

	(i)	(ii)	(iii)
RMS $\ \mathbf{x} - \mathbf{x}_{ref}\ _2$ [cm]	9.0	16.8	7.6
max $\ \mathbf{x} - \mathbf{x}_{ref}\ _2$ [cm]	14.3	28.9	14.2
RMS $ \psi $ [deg]	1.8	5.3	12.6
max $ \psi $ [deg]	5.3	14.9	51.7
RMS $\ \mathbf{v}\ _2$ [m/s]	3.7	4.3	3.8
max $\ \mathbf{v}\ _2$ [m/s]	7.3	8.2	7.7
RMS $\ \mathbf{a} - g\mathbf{i}_z\ _2$ [m/s ²]	14.0	15.2	14.2
max $\ \mathbf{a} - g\mathbf{i}_z\ _2$ [m/s ²]	19.1	21.3	20.4

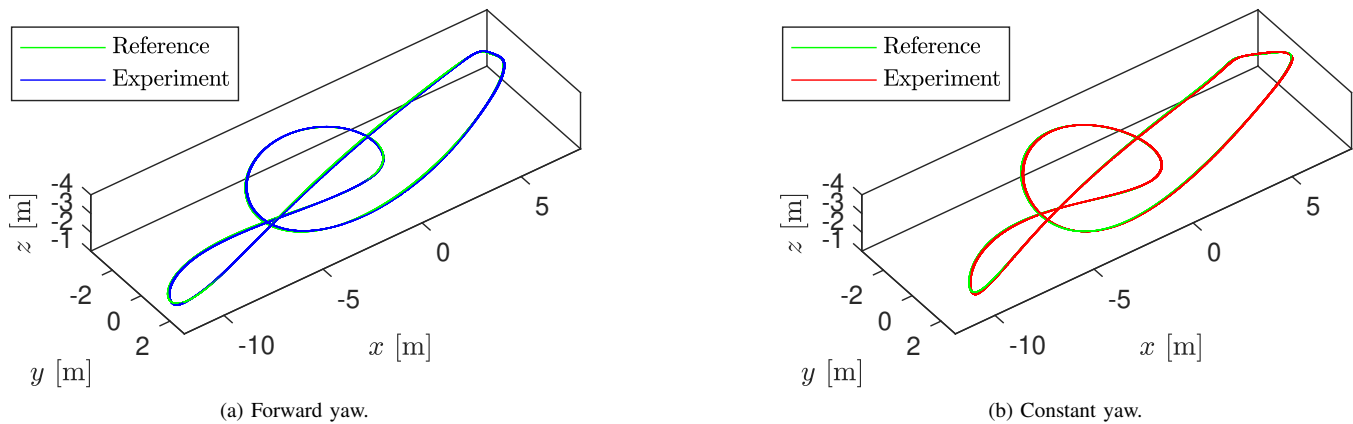


Fig. 11: Experimental flight results for 3D trajectory.

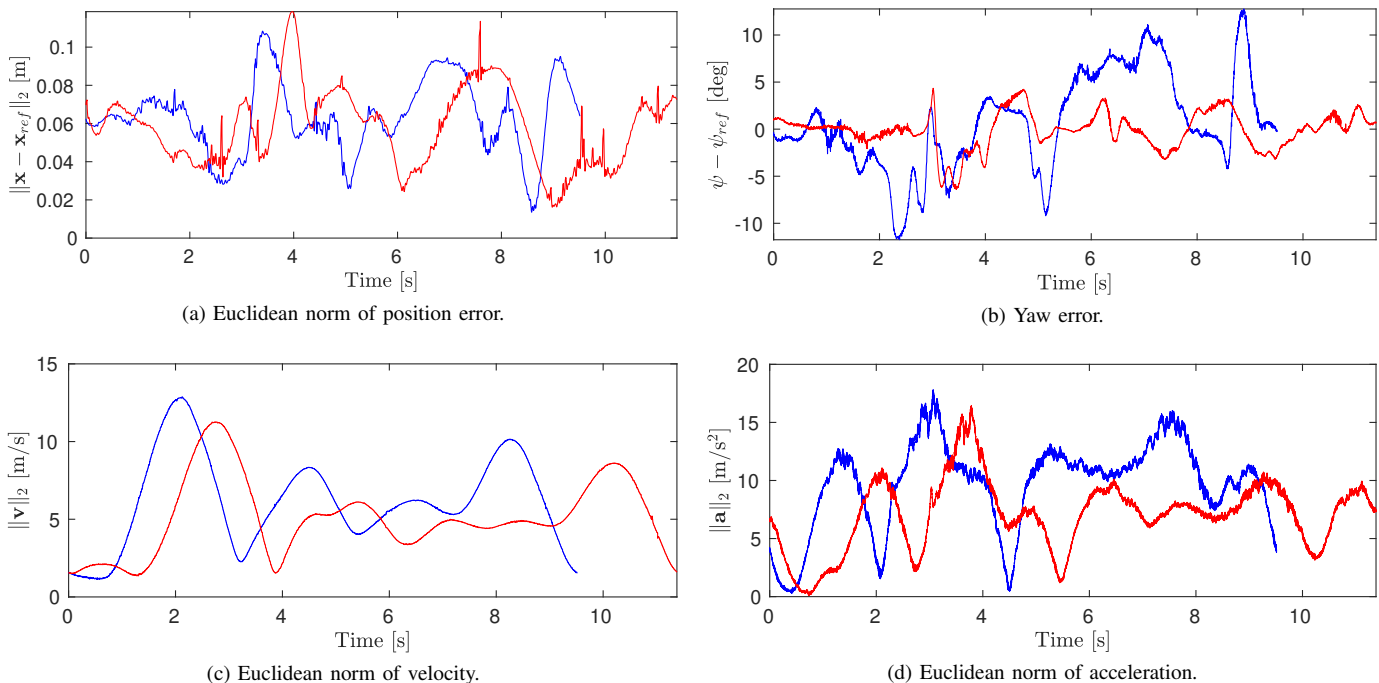


Fig. 12: Experimental flight results for 3D trajectory: forward yaw (blue), and constant yaw (red).

the optical encoders at approximately 5000 Hz. The motor speed and IMU measurements are low-pass filtered using a software second-order Butterworth filter with cutoff frequency 188.5 rad/s (30 Hz).

The platform-independent controller gains listed in Table III were used. Additionally, the controller requires several platform-specific parameters, namely: vehicle mass m , moment of inertia \mathbf{J} , motor time constant τ_m , control effectiveness matrices \mathbf{G}_1 and \mathbf{G}_2 , and the gain and polynomial fit used by the motor speed controller. We obtained control effectiveness data from static tests. In experiments, it was found that using the controller on a different quadcopter (with different dynamic properties, inertial sensors, and propulsion system) required no changes to controller algorithms or gains. After updating only the aforementioned platform-specific pa-

rameters, the controller performed without loss of tracking accuracy.

B. Evaluation of Proposed Controller

In this section, we evaluate the performance of the trajectory tracking controller on two trajectories: a 3D trajectory that includes a high-speed straight and fast turns, and a roulette curve trajectory consisting of fast successive turns resulting in high jerk and snap. The 3D trajectory is generated from a set of waypoints using the method described in [50]. The trajectory is flown with two yaw references: forward yaw, *i.e.*, with the \mathbf{b}_x -axis in the velocity direction, and constant yaw set to zero. Figure 11 shows the corresponding reference trajectories, along with experimental results. The forward yaw trajectory is flown in slightly shorter time. Performance data

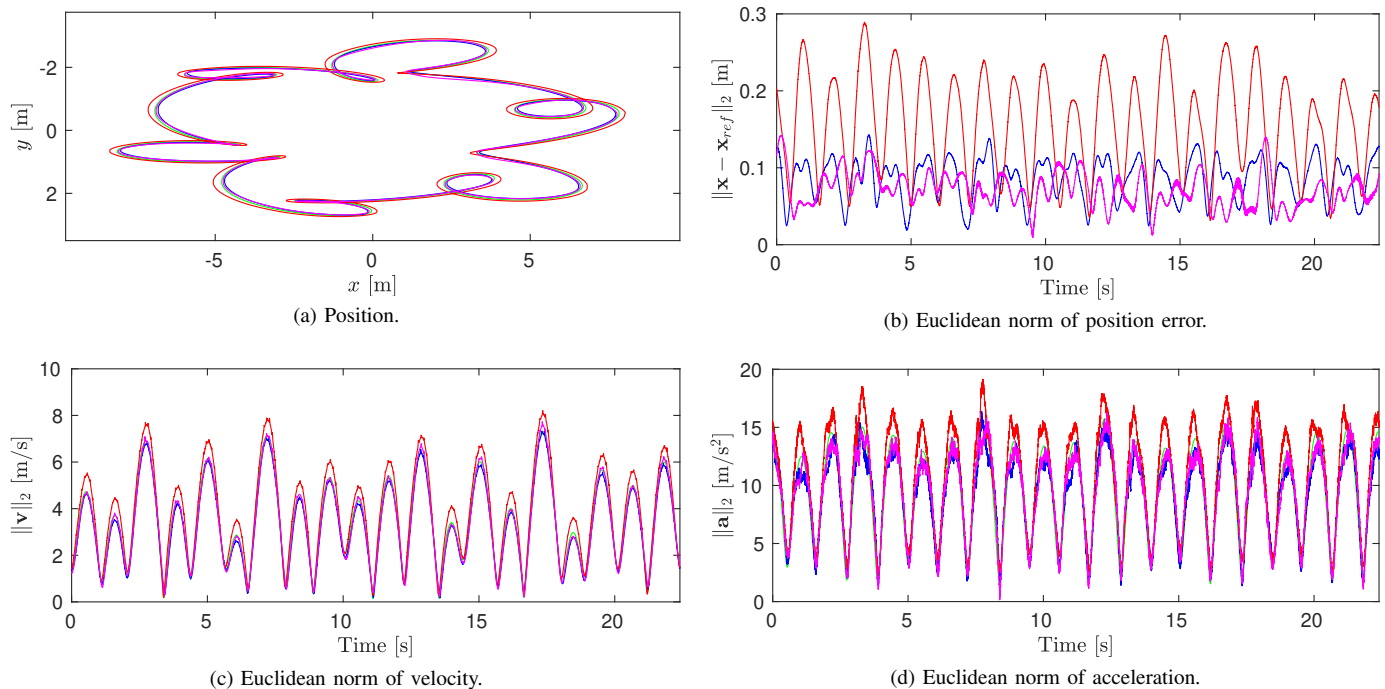


Fig. 13: Experimental flight results for roulette curve trajectory: reference trajectory (green), proposed controller (blue), without jerk and snap tracking (red), and with drag plate attached (magenta).

for both trajectories are given in Table IV and shown in Fig. 12. Over the forward yaw trajectory a maximum speed of 12.9 m/s is achieved, while the RMS tracking error is limited to 6.6 cm. The vehicle attains a maximum proper acceleration of 20.8 m/s² (2.12g). Similar values can be observed for the constant yaw trajectory. The most significant difference is a reduction in yaw tracking error from an RMS value of 5.1 deg to 1.9 deg and from a maximum value of 13 deg to 6.4 deg.

The second, roulette curve trajectory is defined as

$$\sigma_{ref}(t) = \begin{bmatrix} r_1 \cos k_1 t + r_2 \cos k_2 t + r_3 \sin k_3 t \\ r_4 \sin k_1 t + r_5 \sin k_2 t + r_6 \cos k_3 t \\ r_z \\ 0 \end{bmatrix} \quad (51)$$

with $r_1 = 6$ m, $r_2 = 1.8$ m, $r_3 = 0.6$ m, $r_4 = -2.25$ m, $r_5 = -0.3$ m, $r_6 = -0.45$ m, $k_1 = 0.28$ rad/s, $k_2 = 2.8$ rad/s, $k_3 = 1.4$ rad/s, and r_z a constant offset. The trajectory, shown in Fig. 13(a), contains fast, successive turns. Accurate tracking is particularly demanding as it requires fast changes in acceleration, *i.e.*, large jerk and snap, requiring high angular rates and angular accelerations. A single lap is traversed in 22.4 s. The position tracking error is shown in blue in Fig. 13(b), and tracking performance metrics are given in the first column of Table V. Comparison of the position tracking error to the values in Table IV confirms that the controller achieves consistent performance across trajectories. Due to its arduous nature, the roulette curve trajectory is particularly suitable to expose differences in tracking performance. Therefore, we use the trajectory defined by (51) to examine several modifications in subsequent sections. In all cases, the trajectory parameters are identical to those given above.



Fig. 14: Quadrotor with 16 cm × 32 cm cardboard drag plate.

C. Jerk and Snap Tracking

The red curves in Fig. 13 correspond to our proposed control design, but with jerk and snap tracking disabled, *i.e.*, $\Omega_{ref} = \dot{\Omega}_{ref} = \mathbf{0}_{3 \times 1}$. Examination of the figures shows the significant improvement in trajectory tracking performance obtained through the tracking of the jerk and snap feedforward terms. This observation is confirmed by comparing the first two columns of Table V. It can be seen that the RMS position tracking error increases from 9.0 cm to 16.8 cm when jerk and snap tracking are disabled. In Section IV, it was shown that lead compensation provided by jerk and snap tracking results in improved performance when tracking fast-changing acceleration commands. This effect can also be observed in Fig. 13. It can be seen that the system response has less overshoot when jerk and snap tracking are enabled, conform the analytical response of the linearized system.

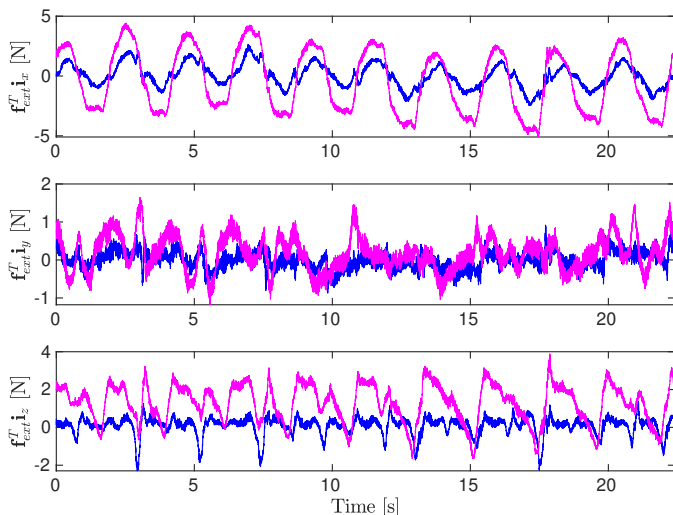


Fig. 15: Estimated external disturbance force for roulette curve trajectory: proposed controller (blue), and with drag plate attached (magenta).

D. Increased Aerodynamic Drag

The magenta curves in Fig. 13 correspond to the trajectory tracking controller as described in this paper, but using the quadcopter with attached drag plate. The drag plate is a 16 cm \times 32 cm cardboard plate that is attached to the bottom of the quadrotor, as shown in Fig. 14. The plate more than triples the frontal surface area of the quadrotor, and as such has a significant effect on the aerodynamic force and moment that act on the vehicle, especially during high-speed flight, and fast pitch and yaw motion. The flight controller is not adapted in any way to account for either these aerodynamic effects, or the changes in mass and moment of inertia.

Comparison of columns (i) and (iii) in Table V shows that the drag plate does not significantly affect position tracking performance. Yaw tracking performance is also consistent, except when the drag plates generates an external yaw moment that causes motor speed saturation and very large momentary yaw tracking error. The consistent tracking performance demonstrates the robustness property of INDI. Controllers that depend on the estimation of drag forces based on velocity, such as [29] and [43], may suffer from much larger loss of tracking performance when the aerodynamic properties of the vehicle are modified. Instead of depending on a model-based drag estimate, INDI counteracts the disturbance force and moment by sensor-based incremental control. The controller implicitly estimates the external force by (18). In Fig. 15, it can be seen that the drag plate has a significant effect on the external disturbance force: its estimated magnitude is approximately tripled. In order to counteract the greater external force, commanded thrust and vehicle pitch increase when the drag plate is attached.

E. Nonlinear Control Effectiveness Inversion

We also compare our proposed nonlinear inversion of the control effectiveness (33), with linearized INDI as presented in

[19]. In the latter case, control moment and thrust commands are tracked using linearized inversion of (5), as follows:

$$\omega_c = \omega_f + (2\mathbf{G}_1 + \Delta t^{-1}\mathbf{G}_2) \left(\begin{bmatrix} \boldsymbol{\mu}_c - \boldsymbol{\mu}_f \\ T_c - T_f \end{bmatrix} + \Delta t^{-1}\mathbf{G}_2 B(\omega_c - \omega_f) \right), \quad (52)$$

where B is the one-sample backshift operator and Δt is the controller update interval. This linearized inversion does not take into account local nonlinearity of (5), nor does it consider the transient response of the motors. Therefore, nonlinear inversion of (33) — as described in Section III-E — theoretically results in improved tracking of the angular acceleration command and thereby in improved trajectory tracking performance.

In experimental flights we found that the difference between nonlinear and linearized inversion does not lead to significant differences in tracking performance for our quadrotor system. However, we found that the failure to properly consider the transient response of the motors in (52) can be detrimental for controller performance. In particular, if the motor time constant τ_m and the controller interval Δt differ greatly, this may result in fast yaw oscillations. Consideration of the motor time constant τ_m , as in (33), resolves this issue.

F. Hover with Disturbance Force

For a constant σ_{ref} input, *i.e.*, hover, the controller consistently achieves sub-centimeter position tracking error if no external disturbance is purposely applied. In this section, we present results for hover with an external disturbance force through a tensioned wire. One end of the wire is attached to the bottom plate of the quadrotor. We pull on the other end of the wire to drag the vehicle away from its hover position.

In Fig. 16, it can be seen that the quadrotor maintains its position to within at most 4 cm, while a changing disturbance force is applied through the wire. The largest position error occurs around 10 s when an external force of approximately 3.7 N is applied. Figure 17 shows the estimated external disturbance force, computed according to (18). The force component in the \mathbf{i}_z -direction has a small steady-state value due to discrepancy between true and estimated thrust. Comparison to Fig. 18 shows that the direction of the estimated external disturbance force vector corresponds to the direction of the wire. For example, at 22 s, Fig. 17 shows that the external force

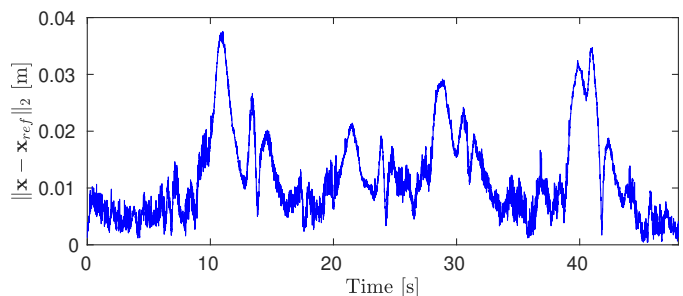


Fig. 16: Euclidean norm of position error for hover with disturbance force through tensioned wire.

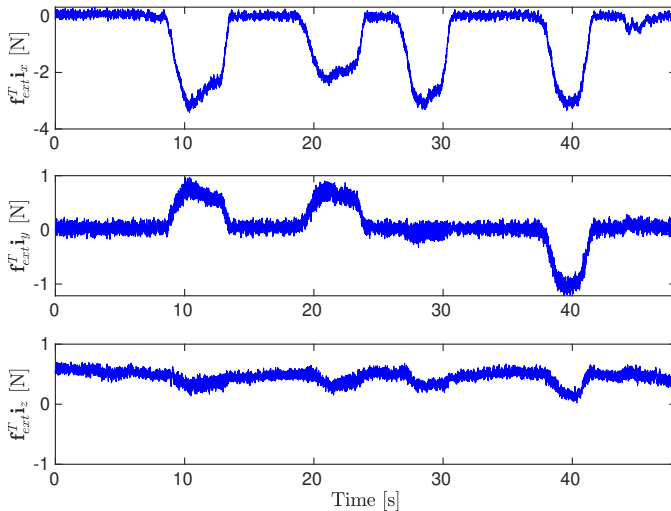


Fig. 17: Estimated external disturbance force for hover with disturbance force through tensioned wire.

has a negative component in the i_x -direction and a positive component in the i_y -direction, and in Fig. 18(a) the wire is indeed tensioned in negative i_x - and positive i_y -direction.

VI. CONCLUSIONS

In this paper, we proposed a novel control system for the tracking of aggressive, *i.e.*, fast and agile, trajectories for quadrotor vehicles. Our controller tracks reference position and yaw angle with their derivatives of up to fourth order, specifically, the position, velocity, acceleration, jerk, and snap along with the yaw angle, yaw rate and yaw acceleration using incremental nonlinear dynamic inversion and differential flatness. The tracking of snap was enabled by closed-loop control of the propeller speeds using optical encoders attached to each motor hub. The resulting control system achieves 6.6 cm RMS position tracking error in agile and fast flight, reaching a top speed of 12.9 m/s and acceleration of 2.1g, in an 18 m long, 7 m wide, and 3 m tall flight volume. Our analysis and experiments demonstrated the robustness of the control design against external disturbances, making it particularly suitable for high-speed flight where significant aerodynamic effects occur. The proposed controller does not require any modeling or estimation of aerodynamic drag parameters.

ACKNOWLEDGMENT

The authors thank Gilhyun Ryou for help with the experiments.

REFERENCES

- [1] E. Tal and S. Karaman, "Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness," in *IEEE Conference on Decision and Control (CDC)*, 2018, pp. 4282–4288.
- [2] J.-J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice Hall, Englewood Cliffs, 1991.
- [3] A. Isidori, *Nonlinear control systems*. Springer, New York, 1995.
- [4] S. Sastry, *Nonlinear systems: Analysis, stability, and control*. Springer-Verlag, New York, 1999.
- [5] S. A. Snell, D. F. Enns, and W. L. Garrard, "Nonlinear inversion flight control for a supermaneuverable aircraft," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 15, no. 4, pp. 976–984, 1992.
- [6] D. J. Bugajski and D. F. Enns, "Nonlinear control law with application to high angle-of-attack flight," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 15, no. 3, pp. 761–767, 1992.
- [7] J. Hauser, S. Sastry, and G. Meyer, "Nonlinear control design for slightly non-minimum phase systems: Application to V/STOL aircraft," *Automatica*, vol. 28, no. 4, pp. 665–679, 1992.
- [8] D. F. Enns, D. J. Bugajski, R. Hendrick, and G. Stein, "Dynamic inversion: An evolving methodology for flight control design," *International Journal of Control*, vol. 59, no. 1, pp. 71–91, 1994.
- [9] T. Koo and S. Sastry, "Output tracking control design of a helicopter model based on approximate linearization," in *IEEE Conference on Decision and Control (CDC)*, 1998, pp. 3635–3640.
- [10] D. Lee, H. J. Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *International Journal of Control, Automation and Systems*, vol. 7, no. 3, pp. 419–428, 2009.
- [11] R. Xu and U. Ozguner, "Sliding mode control of a quadrotor helicopter," in *IEEE Conference on Decision and Control (CDC)*, 2006, pp. 4957–4962.
- [12] T. Madani and A. Benallegue, "Backstepping sliding mode control applied to a miniature quadrotor flying robot," in *IEEE Conference on Industrial Electronics (IECON)*, 2006, pp. 700–705.
- [13] E. Frazzoli, M. A. Dahleh, and E. Feron, "Trajectory tracking control design for autonomous helicopters using a backstepping algorithm," in *American Control Conference (ACC)*, 2000, pp. 4102–4107.
- [14] S. Sieberling, Q. Chu, and J. Mulder, "Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 33, no. 6, pp. 1732–1742, 2010.
- [15] P. Simplício, M. Pavel, E. Van Kampen, and Q. Chu, "An acceleration measurements-based approach for helicopter nonlinear flight control using incremental nonlinear dynamic inversion," *Control Engineering Practice*, vol. 21, no. 8, pp. 1065–1077, 2013.
- [16] P. Smith, "A simplified approach to nonlinear dynamic inversion based flight control," in *AIAA Atmospheric Flight Mechanics Conference*, 1998, pp. 4461–4469.
- [17] B. Bacon and A. Ostroff, "Reconfigurable flight control using nonlinear dynamic inversion with a special accelerometer implementation," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2000, pp. 4565–4579.
- [18] E. J. Smeur, Q. P. Chu, and G. C. de Croon, "Adaptive incremental nonlinear dynamic inversion for attitude control of micro air vehicles," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 38, no. 12, pp. 450–461, 2015.
- [19] E. J. Smeur, G. C. de Croon, and Q. P. Chu, "Cascaded incremental nonlinear dynamic inversion control for MAV disturbance rejection," *Control Engineering Practice*, vol. 73, pp. 79–90, 2018.
- [20] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Sur les systèmes non linéaires différentiellement plats," *CR Acad. Sci. Paris*, pp. 619–624, 1992.
- [21] P. Martin, "Contribution à l'étude des systèmes différentiellement plats," Ph.D. dissertation, École Nationale Supérieure des Mines de Paris, 1992.
- [22] M. Van Nieuwstadt, M. Rathinam, and R. Murray, "Differential flatness and absolute equivalence of nonlinear control systems," *SIAM Journal on Control and Optimization*, vol. 36, no. 4, pp. 1225–1239, 1998.
- [23] P. Martin, R. M. Murray, and P. Rouchon, "Flat systems, equivalence and trajectory generation," California Institute of Technology, Tech. Rep. 2003.008, 2003.
- [24] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: Introductory theory and examples," *International Journal of Control*, vol. 61, no. 6, pp. 1327–1361, 1995.
- [25] P. Martin, "Aircraft control using flatness," in *IMACS/IEEE-SMC Multi-conference and CESA Symposium on Control, Optimization and Supervision*, 1996, pp. 194–199.
- [26] G. Rivera and O. Sawodny, "Flatness-based tracking control and non-linear observer for a micro aerial quadcopter," in *AIP International Conference on Numerical Analysis and Applied Mathematics (ICNAAM)*, 2010, pp. 386–389.
- [27] J. Ferrin, R. Leishman, R. Beard, and T. McLain, "Differential flatness based control of a rotorcraft for aggressive maneuvers," in *IEEE/RSS International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 2688–2693.
- [28] T. Engelhardt, T. Konrad, B. Schäfer, and D. Abel, "Flatness-based control for a quadrotor camera helicopter using model predictive control

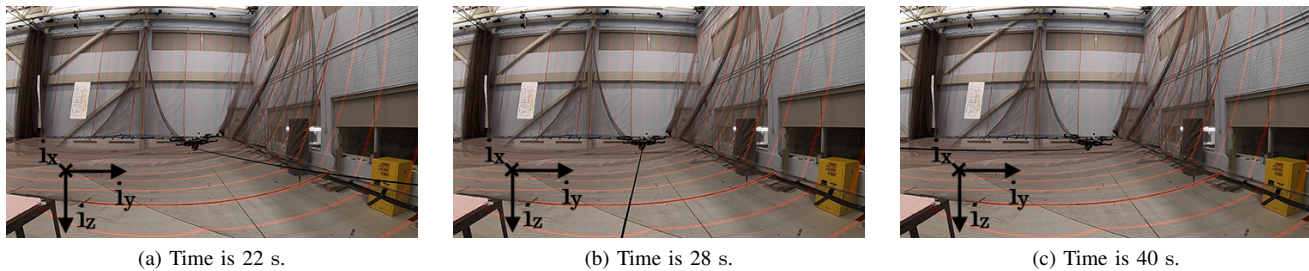
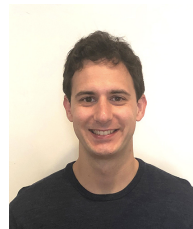


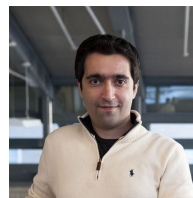
Fig. 18: Quadrotor in hover with disturbance force through tensioned wire.

- trajectory generation,” in *Mediterranean Conference on Control and Automation (MED)*, 2016, pp. 852–859.
- [29] M. Faessler, A. Franchi, and D. Scaramuzza, “Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 620–626, 2018.
- [30] G. Hoffmann, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. S. Jang, and C. J. Tomlin, “The Stanford testbed of autonomous rotorcraft for multi agent control (STARMAC),” in *Digital Avionics Systems Conference (DASC)*, 2004, pp. 12.E.4.1–10.
- [31] I. Kroo, F. Prinz, M. Shantz, P. Kunz, G. Fay, S. Cheng, T. Fabian, and C. Partridge, *The Mesicopter: A miniature rotorcraft concept. Phase II interim report*. Stanford University, 2000.
- [32] P. Pounds, R. Mahony, P. Hynes, and J. M. Roberts, “Design of a four-rotor aerial robot,” in *Australasian Conference on Robotics and Automation (ACRA)*, 2002, pp. 145–150.
- [33] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski, “Dynamic modelling and configuration stabilization for an X4-flyer,” in *15th IFAC World Congress*, 2002, pp. 217–222.
- [34] M. Valenti, B. Bethke, G. Fiore, J. How, and E. Feron, “Indoor multi-vehicle flight testbed for fault detection, isolation, and recovery,” in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006, pp. 6200–6217.
- [35] J. P. How, B. Behnhke, A. Frank, D. Dale, and J. Vian, “Real-time indoor autonomous vehicle test environment,” *IEEE Control Systems*, vol. 28, no. 2, pp. 51–64, 2008.
- [36] S. Zarovy, M. Costello, A. Mehta, G. Gremillion, D. Miller, B. Ranganathan, J. S. Humbert, and P. Samuel, “Experimental study of gust effects on micro air vehicles,” in *AIAA Atmospheric Flight Mechanics Conference*, 2010, pp. 7818–7844.
- [37] G. Ducard and R. D’Andrea, “Autonomous quadrotor flight using a vision system and accommodating frames misalignment,” in *IEEE International Symposium on Industrial Embedded Systems (SIES)*, 2009, pp. 261–264.
- [38] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, “The GRASP multiple micro-UAV testbed,” *IEEE Robotics & Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.
- [39] M. Ol, G. Parker, G. Abate, and J. Evers, “Flight controls and performance challenges for MAVs in complex environments,” in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008, pp. 6508–6529.
- [40] S. Bieniawski, D. Halaas, and J. Vian, “Micro-aerial vehicle flight in turbulent environments: Use of an indoor flight facility for rapid design and evaluation,” in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008, pp. 6512–6522.
- [41] D. Mellinger, N. Michael, and V. Kumar, “Trajectory generation and control for precise aggressive maneuvers with quadrotors,” *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.
- [42] M. Müller, S. Lupashin, and R. D’Andrea, “Quadcopter ball juggling,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 5113–5120.
- [43] J. Svacha, K. Mohta, and V. Kumar, “Improving quadrotor trajectory tracking by compensating for aerodynamic effects,” in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 860–866.
- [44] J.-M. Kai, G. Allibert, M.-D. Hua, and T. Hamel, “Nonlinear feedback control of quadrotors exploiting first-order drag effects,” in *20th IFAC World Congress*, 2017, pp. 8189–8195.
- [45] H. Liu, D. Li, Z. Zuo, and Y. Zhong, “Robust three-loop trajectory tracking control for quadrotors with multiple uncertainties,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2263–2274, 2016.
- [46] H. Huang, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, “Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 3277–3282.
- [47] A. P. Schoellig, F. L. Mueller, and R. D’Andrea, “Optimization-based iterative learning for precise quadcopter trajectory tracking,” *Autonomous Robots*, vol. 33, no. 1, pp. 103–127, 2012.
- [48] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, “Quadrotor helicopter flight dynamics and control: Theory and experiment,” in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2007, pp. 6461–6480.
- [49] P. Martin and E. Salaün, “The true role of accelerometer feedback in quadrotor control,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 1623–1629.
- [50] G. Ryou, E. Tal, and S. Karaman, “Multi-fidelity black-box optimization for time-optimal quadrotor maneuvers,” in *Robotics: Science and Systems (RSS)*, 2020.



Ezra Tal received BSc and MSc degrees from the Faculty of Aerospace Engineering, Delft University of Technology in 2012 and 2015, respectively. He is currently pursuing a PhD degree at the Massachusetts Institute of Technology. In 2012 Ezra was a visiting student at the Technion – Israel Institute of Technology, and in 2015 he visited NASA Ames Research Center (ARC) as a graduate research intern. He is the recipient of a Huygens Talent Scholarship, and a NASA Group Achievement Award as part of the Adaptive Aeroelastic Wing Shaping Control

Team at ARC. His current research interests include differential games and robust control theory, particularly for applications in planning and control of robotics vehicles.



Sertac Karaman is an Associate Professor of Aeronautics and Astronautics at the Massachusetts Institute of Technology (MIT). He obtained an S.M. degree in mechanical engineering in 2009, and a Ph.D. degree in electrical engineering and computer science in 2012, both from MIT. He studies the applications of probability theory, stochastic processes, stochastic geometry, formal methods, and optimization for the design and analysis of high-performance cyber-physical systems. The application areas of his research include driverless cars, unmanned aerial

vehicles, distributed aerial surveillance systems, air traffic control, certification and verification of control systems software, and many others. He is the recipient of an IEEE Robotics and Automation Society Early Career Award in 2017, an Office of Naval Research Young Investigator Award in 2017, Army Research Office Young Investigator Award in 2015, National Science Foundation Faculty Career Development (CAREER) Award in 2014.