Model Predictive Control for Quadcopters with Almost Global Trajectory Tracking Guarantees

A.R.P. Andriën, E. Lefeber, D. Antunes, W.P.M.H. Heemels

***Abstract*— This paper provides a new method for trajectory tracking for quadcopters following a cascaded control approach with formal closed-loop tracking guarantees. An outer-loop model predictive controller generates twice differentiable acceleration references, which provide attitude, angular velocity and acceleration references for a nonlinear inner-loop controller. The model predictive controller allows for tracking of references while explicitly considering that the thrust of the quadcopter is upper and lower limited. It is proven that the overall strategy renders the trajectory tracking errors uniformly almost globally asymptotically stable. Via a numerical case study the advantages of the novel method are highlighted.**

***Index Terms*— global stability, model predictive control, quadcopters, trajectory tracking**

I. Introduction

QUADCOPTERS are now widespread in the consumer market [1] and are used in many applications, such as agriculture [2], surveillance [3], wildlife monitoring [4], construction [5], (medicine) delivery [6], [7] and even extraterrestrial exploration [8], [9]. From a research point of view, quadcopters have received much attention as well, due to the challenging nonlinear dynamics required to properly describe them [10], their under-actuated configuration [11] and high maneuverability [12]. In particular, for the control of quadcopters many avenues have been explored, such as sliding mode control [13], iterative learning control [14], nonlinear control [15], [16], reinforcement learning [17], to name a few. However, despite the broad range of approaches in quadcopter control, it is still hard to find in the literature a control approach with the following highly desired features:

1) Ability to anticipate on future reference information;
2) Explicit handling of constraints on the states and inputs;
3) Real-time implementability on embedded hardware;
4) Stability or tracking error convergence guarantees.

Model Predictive Control (MPC) can potentially provide these features, due to its ability to anticipate using future reference information, handle constraints explicitly and the availability of well-established theoretical results that can be used to provide guarantees on closed-loop behavior [18]. Moreover, the typically high computational burden of (nonlinear) MPC (3) can potentially be overcome by using the differential flatness property of quadcopters as motivated in [19]. Yet, to the best of the authors' knowledge there are no MPC strategies currently available that provide all these favorable features. Indeed, MPC setups have been used for quadcopters in, for example, [20], exploiting differential flatness to achieve a convex optimization problem that is solved in real-time; however no stability guarantees are provided. In [21], a hierarchical MPC strategy was developed that uses linearization around the trajectory to make the problem computationally feasible, however tracking error convergence guarantees are not given. A common element in many quadcopter control approaches is a cascaded control structure, where the control of the orientation of the quadcopter, known as the inner loop, is separated from the position and velocity control, referred to as the outer loop [22]. A strategy combining a thrust prioritizing inner-loop controller with an outer-loop controller that satisfies constraints on some of the states was presented in [23], however, again without stability and convergence guarantees. Control strategies following other approaches also do not provide all the desired features. For instance, the strategies provided in [24], [25] meet 3) and 4), however 1) and 2) are not satisfied.

Nonlinear MPC (NMPC) has also been used in more recent papers, due to the increase in onboard computational power of most quadcopters. A cascaded approach is used in [26], where NMPC and LQR control with integral action are used for the inner and outer loop, respectively. NMPC is used in the outer loop in [27] and used for obstacle avoidance. In [28] NMPC is used to avoid obstacles in cluttered environments while performing high speed trajectories, respecting both input and state constraints. NMPC is compared with differential-flatness-based control in [29], both with and without the addition of an inner loop controller using the incremental nonlinear dynamic inversion (INDI) method for both cases, showing that NMPC outperforms DFBC at the expense of a much higher computational load. They also show the importance of considering the aerodynamic drag in both methods, where they consider a linear drag model that captures the major effects [30]. Although all these methods provide impressive results in terms of tracking high speed trajectories in real experiments, none of them provide stability guarantees, resulting in crashes for aggressive trajectories [29]. Moreover, they do not show guaranteed feasibility of the NMPC algorithm, resulting in unpredictable and sometimes unstable behavior.

Motivated by this gap in the literature, in this paper a new

(*Corresponding author: A.R.P. Andriën*)
A.R.P. Andriën, D. Antunes and W.P.M.H. Heemels are with the Control Systems Technology Group and E. Lefeber is with Dynamics and Control Group, Dep. of Mech. Engineering, Eindhoven University of Technology, 5612 AE Eindhoven, The Netherlands (emails: [A.R.P.Andrien, A.A.J.Lefeber, D.Antunes, W.P.M.H.Heemels]@tue.nl).

MPC approach is proposed based on a general fourth-order quadcopter model as presented in [30]. A cascaded control design is employed, in which the control of the translational system, consisting of position and velocity kinematics and dynamics, is re-formulated as a linear problem by considering a virtual acceleration as input. This is referred to as the outer-loop tracking problem and an MPC strategy is designed that allows for meeting the desired features mentioned above. In order for the system to track the desired virtual acceleration generated by the MPC in the outer loop, a desired thrust vector is generated, which is converted into a desired attitude that is tracked by using the attitude controller presented in [16]. The attitude tracking problem is referred to as the inner loop, and the adopted controller requires that the desired virtual acceleration is twice differentiable, which is ensured by considering a linear fourth-order model for the outer loop, as we will show. Indeed, it is shown how the constraints for the original nonlinear model can be translated into constraints for the linear fourth-order model, although this translation involves the introduction of some degree of conservatism, as will be carefully explained. Considering a given class of reference inputs, it is shown that the outer-loop MPC control strategy results in uniform global asymptotic stability (UGAS) for the tracking error. This convergence proof relies on new technical contributions that rely on state and input transformations and on recent results on globally stable MPC strategies for linear systems with input constraints. Moreover, we explicitly consider the discrete-time nature of the MPC controller and show that constraint satisfaction of the continuous-time system is still guaranteed in between samples. The advantages of the proposed cascaded control scheme are shown in a numerical case study, where we show that the quadrotor is able to recover from an upside-down starting attitude and converges to a challenging reference trajectory.

Compared to the preliminary results presented in [31], here: (i) a more complete, fourth-order model of the quadcopter as proposed in [30] is considered, (ii) trajectory tracking guarantees are provided for the full cascaded system rather than setpoint guarantees only for the outer-loop controller, and (iii) all the proofs and extensive explanations, not available in [31], are provided.

The remainder of this paper is structured as follows. First, the notation used in this paper is introduced in Section II, followed by the dynamic model and the problem definition in Section III, after which the method is outlined in Section IV. The inner- and outer-loop controllers are presented in Sections V and VI, respectively, which are combined to provide the overall controller in Section VII, together with the proofs of the main results. Simulations results are provided in Section VIII. Section IX provides concluding remarks.

II. Preliminaries

In this section the notation used in this paper is introduced. Let $e_i \in \mathbb{R}^3$ for $i \in \{1,2,3\}$ denote the standard unit vectors. The trace of a matrix A is denoted by $\mathrm{tr}\,(A)$ for a square matrix $A \in \mathbb{R}^{n\times n}$. The Euclidean, or two-norm, of a vector is denoted by $\|v\|$ for $v \in \mathbb{R}^n$, i.e., $\|v\| = \sqrt{v^\top v}$. The induced Euclidean matrix norm for $A \in \mathbb{R}^{n\times n}$ is denoted similarly by $\|A\|$, which is equal to $\sqrt{\lambda_{\max}(A^\top A)}$, where $\lambda_{\max}(A^\top A)$ denotes the largest eigenvalue of $A^\top A$. A positive definite and semi-positive definite matrix A are denoted by $A \succ 0$ and $A \succeq 0$, respectively. A diagonal matrix with scalar entries a, b on the diagonal is denoted as $\mathrm{diag}(a,b)$.

Consider the non-autonomous system

$$\dot{x} = f(t,x), \tag{1}$$

with state x taking values in $\mathbb{R}^n$, time $t \in \mathbb{R}_{\geq 0}$ and $f : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \to \mathbb{R}^n$ is piecewise continuous in t and locally Lipschitz in x. We assume that the origin $x = 0$ is an equilibrium point of (1), meaning that

$$f(t,0) = 0, \text{ for all } t \in \mathbb{R}_{\geq 0}. \tag{2}$$

In this paper uniform global asymptotic stability (UGAS), uniform global exponential stability (UGES) and uniform local exponential stability (ULES) are considered, for which definitions are given in [32] and repeated here for convenience.

Definition 1 (cf. [32]): Consider (1) and let $x = 0$ be an equilibrium point of (1). This equilibrium is

- stable, if for each $t_0 \geq 0$ and each $\epsilon > 0$, there is $\delta = \delta(\epsilon, t_0) > 0$ such that
$$\|x(t_0)\| < \delta \implies \|x(t)\| < \epsilon, \text{ for all } t \geq t_0. \tag{3}$$
- unstable, if it is not stable.
- uniformly stable, if for each $\epsilon > 0$, there is $\delta = \delta(\epsilon) > 0$, independent of t_0, such that (3) is satisfied for all $t_0 \geq 0$.
- asymptotically stable, if it is stable and for all $t_0 \geq 0$ there is a positive constant $c = c(t_0)$ such that $x(t) \to 0$ as $t \to \infty$, for all $\|x(t_0)\| < c$.
- uniformly asymptotically stable, if it is uniformly stable and there is a positive constant c, independent of t_0, such that for all $t_0 \geq 0$, all $\|x(t_0)\| < c$ and each $\eta > 0$, there is $T = T(\eta) > 0$ such that
$$\|x(t)\| < \eta, \text{ for all } t \geq t_0 + T(\eta). \tag{4}$$
- uniformly globally asymptotically stable (UGAS), if it is uniformly asymptotically stable, where $\delta(\epsilon)$ can be chosen to satisfy $\lim_{\epsilon\to\infty} \delta(\epsilon) = \infty$, and, for each pair of positive numbers η and c, there is $T = T(\eta, c) > 0$ such that for all $t_0 \geq 0$ it holds that
$$\|x(t)\| < \eta, \text{ for all } t \geq t_0 + T(\eta, c), \text{ for all } \|x(t_0)\| < c. \tag{5}$$
- uniformly locally exponentially stable (ULES), if there exist positive constants c, k, and λ such that for all $t_0 \geq 0$, all $t \geq t_0$ and all $\|x(t_0)\| < c$ it holds that
$$\|x(t)\| \leq k \|x(t_0)\| e^{-\lambda(t-t_0)}. \tag{6}$$
- uniformly globally exponentially stable (UGES), if (6) is satisfied for all initial states $x(t_0) \in \mathbb{R}^n$, all $t_0 \geq 0$ and all $t \geq t_0$.

Global stabilization of a quadcopter involves global stabilization of its attitude on $SO(3)$ and global stabilization of its linear position. Since global stabilization on $SO(3)$ using a

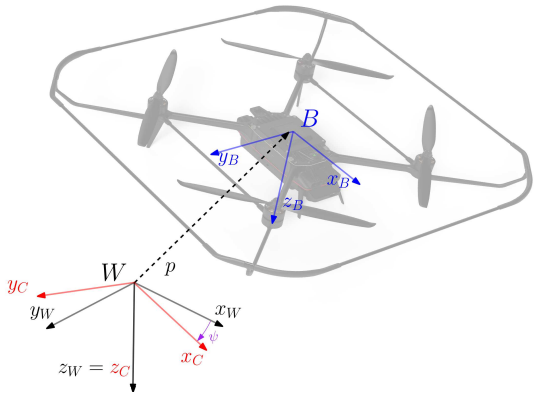

Fig. 1. The inertial reference frame $\boldsymbol{W}$ and body fixed frame $\boldsymbol{B}$.

continuous control input can not be achieved, cf. [33], we need to relax such stability notion. In fact, uniform *almost* global asymptotic stability is the aim in this paper, which is defined as follows:

Definition 2: Consider (1) and let $x = 0$ be an equilibrium point. This equilibrium is uniformly almost globally asymptotically stable (UaGAS), if it is UGAS, except for initial conditions in a set of measure zero. That is, (3) and (5) hold for every $x(t_0) \in \mathbb{R}^n \setminus M$, where M is a set of measure zero, and $\delta(\epsilon)$ can be chosen to satisfy $\lim_{\epsilon\to\infty} \delta(\epsilon) = \infty$.

III. Quadcopter Dynamics and Problem Formulation

In this section a model of the quadcopter dynamics is first introduced, followed by the introduction of the notion of feasible reference trajectories. Based on these two ingredients, a formal problem statement is provided.

A. Dynamic model of quadcopter

The model that is used here is based on [15]. However, in contrast to [15], stiff rotors and no external wind are considered here (as was also done in [30]). To present the resulting model, suitable coordinate frames are needed. To introduce them, let W denote a right-handed inertial (or world) frame according to the North-East-Down (NED) convention, with unit vectors along the axes denoted by $\{x_W, y_W, z_W\}$, forming an orthonormal basis. Let B denote a right-handed body-fixed frame with unit vectors $\{x_B, y_B, z_B\}$ forming an orthonormal basis, where these vectors are the axes of B with respect to W. The origin of the body-fixed frame coincides with the center of mass of the quadrotor, and z_B is aligned with z_W and the gravitational vector when the quadrotor is at hover, see Figure 1. The orientation of B with respect to W is represented by the rotation matrix $R = [x_B, y_B, z_B] \in SO(3)$. Let $\omega = [\omega_1, \omega_2, \omega_3]^\top$ denote the angular velocities of B relative to W, expressed in B. The position and linear velocity of the center of mass of the quadrotor with respect to W are denoted by $p = [p_x, p_y, p_z]^\top$ and $v = [v_x, v_y, v_z]^\top$, respectively.

Using the above variables, the model can now be described by the equations

$$\dot{p} = v, \tag{7a}$$
$$\dot{v} = gz_W - Tz_B - RDR^\top v, \tag{7b}$$
$$\dot{R} = RS(\omega), \tag{7c}$$
$$J\dot{\omega} = S(J\omega)\omega - \tau_g - AR^\top v - C\omega + \tau. \tag{7d}$$

The forces acting on the translational dynamics ((7a)-(7b)) of the quadrotor consist of the gravity, given by gz_W, where g is the gravitational constant, the thrust force $-Tz_B$, where $T \geq 0$ denotes the magnitude of the combined thrust of the four propellers (mass-normalized), and a drag force as a result of rotor drag $-RDR^\top v = -d(R)v$, where $D = \text{diag}(d_x, d_y, d_z)$, $d_x, d_y, d_z > 0$ are the mass-normalized rotor drag coefficients and $d(R) = RDR^\top$. Note that similarly to [27], [34]–[36] we consider a linear drag model.

The rotation of the quadcopter is characterized by the attitude kinematics given in (7c), where $S(a)$ represents a skew-symmetric matrix such that $S(a)b = a \times b$ for any vectors $a, b \in \mathbb{R}^3$ and the dynamics given in (7d), where $J \in \mathbb{R}^{3\times3}$ is the inertia matrix, $\tau_g \in \mathbb{R}^3$ are torques resulting from gyroscopic effects, A and C are constant matrices and $\tau = [\tau_1, \tau_2, \tau_3]^\top \in \mathbb{R}^3$ is the torque input.

The thrust is considered to be non-negative and limited according to

$$0 \leq T(t) \leq T_{\max}, \text{ for all } t \in \mathbb{R}_{\geq 0}, \tag{8}$$

where $T_{\max} > g$ is the maximal thrust. This is a physical restriction dictated by the fact that the propellers can only generate limited thrust upwards and must be capable of counteracting the gravitational force. Note that the thrust T and the torque τ are considered as the control inputs of the quadcopter. Effects such as motor dynamics and propeller aerodynamics are omitted.

B. Reference trajectory

In this paper we focus on a certain class of feasible reference trajectories. In fact, a reference trajectory $(\bar{p}, \bar{v}, \bar{R}, \bar{\omega}, \bar{T}, \bar{\tau}) : \mathbb{R}_{\geq 0} \to \mathbb{R}^3 \times \mathbb{R}^3 \times SO(3) \times \mathbb{R}^3 \times \mathbb{R} \times \mathbb{R}^3$ is called *feasible*, if it satisfies the dynamics (7) in the sense that for all $t \in \mathbb{R}_{\geq 0}$

$$\dot{\bar{p}} = \bar{v}, \tag{9a}$$
$$\dot{\bar{v}} = gz_W - \bar{T}\bar{z}_B - \bar{R}D\bar{R}^\top\bar{v}, \tag{9b}$$
$$\dot{\bar{R}} = \bar{R}S(\bar{\omega}), \tag{9c}$$
$$J\dot{\bar{\omega}} = S(J\bar{\omega})\bar{\omega} - \tau_g - A\bar{R}^\top\bar{v} - C\bar{\omega} + \bar{\tau}, \tag{9d}$$

and

$$0 < \underline{\epsilon} \leq \bar{T}(t) \leq T_{\max} - \bar{\epsilon}, \tag{10}$$

for fixed $\bar{\epsilon}, \underline{\epsilon} > 0$. Note that the thrust of a feasible reference is required to be strictly greater than zero and have a maximum that is strictly smaller than the maximal thrust $T_{\max}$ of the actual quadcopter. This is required in order for the actual quadcopter to be able to follow the reference.

C. Problem statement

Given a feasible reference trajectory, the error coordinates can be defined as

$$\tilde{p} = \bar{p} - p, \tag{11a}$$
$$\tilde{v} = \bar{v} - v, \tag{11b}$$
$$\tilde{R} = \bar{R}^\top R, \tag{11c}$$
$$\tilde{\omega} = \omega - \tilde{R}^\top \bar{\omega}, \tag{11d}$$

which can be used to formulate the main problem considered in this paper as follows.

Problem 1: Given a feasible reference trajectory $(\bar{p}, \bar{v}, \bar{R}, \bar{\omega}, \bar{T}, \bar{\tau})$, find control laws

$$\begin{aligned} T &= T(p, v, R, \omega, \bar{p}, \bar{v}, \bar{R}, \bar{\omega}, \bar{T}, \bar{\tau}), \\ \tau &= \tau(p, v, R, \omega, \bar{p}, \bar{v}, \bar{R}, \bar{\omega}, \bar{T}, \bar{\tau}), \end{aligned} \tag{12}$$

such that (8) holds and such that for the closed-loop system (7), (9), (12)

$$\lim_{t\to\infty} (\tilde{p}(t), \tilde{v}(t), \tilde{R}(t), \tilde{\omega}(t)) = (0, 0, I, 0), \tag{13}$$

for all initial conditions $(p(0), v(0), R(0), \omega(0)) \in \mathbb{R}^3 \times \mathbb{R}^3 \times SO(3) \setminus M \times \mathbb{R}^3$, where $M \subset SO(3)$ is a set of measure zero. In fact, we aim for the closed-loop system to satisfy a UaGAS property, see Definition 2.

Assumption 1: It is assumed that the effects of rotation in the drag force are negligible, i.e., that $d(R) = RDR^\top \approx D$, for all trajectories considered.

Remark 1: While Problem 1 assumes, for simplicity, a continuous-time controller implementation by making T and τ at time t a function of the state and reference at time t, the actual proposed controller will take a sample and hold form. This is required due to the MPC approach that we will follow. More formally, we can state that T, τ are functions of the state and reference from time 0 to time t.

IV. Methodology

In order to solve the nonlinear tracking problem defined in Problem 1, a cascaded controller design is employed. The cascade consists of an outer loop and an inner loop, that contain the translational ((7a)-(7b)) and rotational ((7c)-(7d)) subsystems, respectively, as depicted in Figure 2. First the setup of the cascade is discussed, followed by the resulting constraints on the subsystems that follow from the setup and the thrust constraint (8). These constraints also ensure that the variables used in this section are well-defined. The section is concluded with the problem definitions related to the inner loop and outer loop, that are solved in the subsequent sections.

A. Cascaded trajectory tracking setup

Considering the definition of the position and the velocity errors in ((11a)-(11b)), their dynamics can be found by subtracting ((9a)-(9b)) from ((7a)-(7b)), giving

$$\begin{aligned} \dot{\tilde{p}} &= \tilde{v}, \\ \dot{\tilde{v}} &= -D\tilde{v} + Tz_B - \bar{T}\bar{z}_B. \end{aligned}$$

Introducing a new virtual input $a_d \in \mathbb{R}^3$, referred to as the desired acceleration (error), the actual acceleration error is replaced by a_d, i.e.,

$$a_d = Tz_B - \bar{T}\bar{z}_B, \tag{14}$$

which leads to

$$\dot{\tilde{p}} = \tilde{v}, \tag{15a}$$
$$\dot{\tilde{v}} = -D\tilde{v} + a_d. \tag{15b}$$

In Section VI an MPC strategy is used to find a desired acceleration a_d such that the dynamics in (15) are stabilized and the constraint in (8) is satisfied. Based on this desired acceleration, the inputs τ and T are then used to have the actual acceleration converge to the desired acceleration, i.e., to have the error

$$\tilde{a} = a_d - Tz_B + \bar{T}\bar{z}_B. \tag{16}$$

converge to zero. By setting (16) to zero, it follows that

$$Tz_B = a_d + \bar{T}\bar{z}_B, \tag{17}$$

and by setting the thrust as the magnitude of the vector on the right-hand side, i.e.,

$$T = \left\| a_d + \bar{T}\bar{z}_B \right\|, \tag{18}$$

the first input is determined. Note that in order to ensure the correct direction of the vector on the right-hand side in (17), the vector z_B can be used. However, this is not a direct control input of the quadcopter. Therefore, instead, the desired rotation of the quadcopter, denoted by R_d, is determined next. First, note that pre-multiplication of (17) with $\bar{R}^\top$ results in

$$T\bar{R}^\top R e_3 = \bar{R}^\top a_d + \bar{T}\bar{R}^\top \bar{R} e_3,$$

or, equivalently,

$$T\tilde{R} e_3 = \bar{R}^\top a_d + \bar{T} e_3. \tag{19}$$

The desired thrust direction is then set to

$$z_{B,d} = \begin{bmatrix} z_{B,d_1} \\ z_{B,d_2} \\ z_{B,d_3} \end{bmatrix} = \frac{\bar{R}^\top a_d + \bar{T} e_3}{\left\| \bar{R}^\top a_d + \bar{T} e_3 \right\|}, \tag{20}$$

and the remaining columns of the desired quadcopter orientation are set to

$$y_{B,d} = \frac{z_{B,d} \times e_1}{\| z_{B,d} \times e_1 \|} = \begin{bmatrix} 0 & \frac{z_{B,d_3}}{\theta} & \frac{-z_{B,d_2}}{\theta} \end{bmatrix}^\top,$$
$$x_{B,d} = y_{B,d} \times z_{B,d} = \begin{bmatrix} \theta & -\frac{z_{B,d_1} z_{B,d_2}}{\theta} & -\frac{z_{B,d_1} z_{B,d_3}}{\theta} \end{bmatrix}^\top,$$

for $\theta = \sqrt{z_{B,d_2}^2 + z_{B,d_3}^2}$. The desired attitude of the quadcopter is then given by

$$R_d = \begin{bmatrix} x_{B,d}, & y_{B,d}, & z_{B,d} \end{bmatrix}. \tag{21}$$

Roughly speaking, this ensures that when the errors in (15) converge to zero, a_d converges to zero and $z_{B,d} \to e_3$, making $R_d \to I$.

Note that the Euclidean norm is invariant under rotation, so that the thrust defined in (18) can be written as

$$T = \left\| a_d + \bar{T}\bar{z}_B \right\| = \left\| \bar{R}^\top a_d + \bar{T} e_3 \right\|.$$

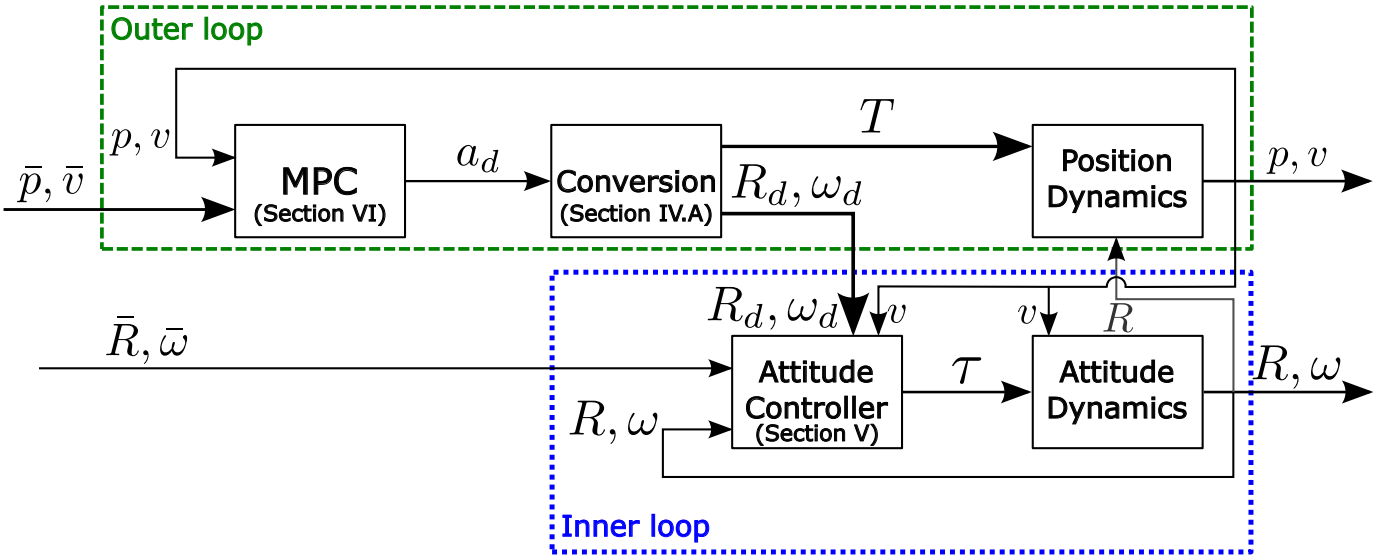

Fig. 2. Overview of the proposed control strategy. In the outer loop, MPC is used to generate desired accelerations $\boldsymbol{a_d}$, as will be discussed in Section VI. As will be discussed in Section IV-A, these accelerations are subsequently converted to thrust inputs $\boldsymbol{T}$ that are applied to the quadcopter as well as desired attitudes $\boldsymbol{R_d}$ and angular velocities $\boldsymbol{\omega_d}$. These are then combined with the reference and measured attitudes and angular velocities, $\bar{\boldsymbol{R}}, \bar{\boldsymbol{\omega}}$ and $\boldsymbol{R}, \boldsymbol{\omega}$, respectively, in the attitude controller in the inner loop, resulting in torque inputs $\boldsymbol{\tau}$ that are applied to the quadcopter, see Section V. The outer-loop and inner-loop strategies are combined in Section VII, where the stability of the overall system is discussed as well.

Combined with (20) and (21), this shows that (19) can be written as

$$T\tilde{R}e_3 = \bar{R}^\top a_d + \bar{T}e_3 = TR_d e_3. \tag{22}$$

From this it becomes clear that by making $\tilde{R}$ converge to R_d, the desired acceleration is achieved. This will be achieved using the torque τ, which is generated by the attitude tracking controller presented in Section V. This controller will require differentiable setpoints for the desired angular velocity, which are found by first noting that from $\dot{R}_d = R_d S(\omega_d)$ it follows that

$$\dot{x}_{B,d} = \omega_{d_3} y_{B,d} - \omega_{d_2} z_{B,d}, \tag{23a}$$
$$\dot{y}_{B,d} = -\omega_{d_3} x_{B,d} + \omega_{d_1} z_{B,d}, \tag{23b}$$
$$\dot{z}_{B,d} = \omega_{d_2} x_{B,d} - \omega_{d_1} y_{B,d}. \tag{23c}$$

Pre-multiplying (23b) with $x_{B,d}^\top$ and (23c) with $x_{B,d}^\top$ and $y_{B,d}^\top$ results in an expression for the desired angular velocity as

$$\omega_d = \begin{bmatrix} -y_{B,d}^\top \dot{z}_{B,d} \\ x_{B,d}^\top \dot{z}_{B,d} \\ -x_{B,d}^\top \dot{y}_{B,d} \end{bmatrix}. \tag{24}$$

The above derivations show that the desired acceleration a_d can be achieved by generating the desired attitude in (21) and using the thrust as in (18). Next, it is discussed how the constraint on the thrust in (8) can be converted to constraints on the desired acceleration a_d, which is to be generated by the MPC. Moreover, the constraints on the desired acceleration will also ensure that the variables in this section are well-defined.

B. Constraints

The magnitude of the thrust vector is constrained according to (8) and for (20) to be defined it is required that $T \neq 0$. Furthermore, in order for $y_{B,d}$ to be well-defined it is required that $z_{B,d}$ and e_1 are never parallel, which is achieved by requiring $z_{B,d_3} > 0$, which also ensures $T \neq 0$. Note that from (20) it follows that ensuring $z_{B,d_3} > 0$ is equivalent to ensuring $\bar{z}_B^\top a_d + \bar{T} > 0$. To this effect, $a_d \in \mathcal{A}$ is constrained and the set of admissible values that the desired acceleration can take is defined as

$$\mathcal{A}(\bar{R}, \bar{T}) := \{a_d \in \mathbb{R}^3 | 0 < \left\| a_d + \bar{T}\bar{z}_B \right\| \leq T_{\max}, \\ \bar{z}_B^\top a_d + \bar{T} > 0\}. \tag{25}$$

The inner-loop controller requires setpoints for the desired angular velocity and its derivative, which means that the desired acceleration needs to be twice differentiable, see (24).

C. Cascaded problem definition

The original tracking control problem as defined in Problem 1 is now split into two subproblems, namely an outer-loop and an inner-loop problem. The outer-loop problem is formulated as

Problem 2 (Outer-loop problem): Find a twice differentiable virtual acceleration control law $a_d = a_d(p, v, \bar{p}, \bar{v}, \bar{T})$, such that the origin $(\tilde{p}(t), \tilde{v}(t)) = (0, 0)$ of the system ((15a)-(15b)) is UGAS and such that $a_d \in \mathcal{A}(\bar{R}, \bar{T})$, for all $t \in \mathbb{R}_{\geq 0}$, with $\mathcal{A}(\bar{R}, \bar{T})$ as defined in (25).

Since it is desired to steer $\tilde{R}$ to R_d, the attitude error and angular velocity error considered in the inner-loop problem are defined as

$$R_e = R_d^\top \tilde{R} \tag{26}$$

and

$$\omega_e = \omega - \tilde{R}^\top \bar{\omega} - R_e^\top \omega_d, \tag{27}$$

respectively. The inner-loop problem is now formulated as

Problem 3 (Inner-loop problem): Find a control law $\tau = \tau(R, \omega, \bar{R}, \bar{\omega}, R_d, \omega_d)$, such that the origin $(R_e(t), \omega_e(t)) = (I, 0)$ of the system ((26)-(27)) is UaGAS.

In Section VII it will be shown that by solving Problems 2 and 3 a solution to Problem 1 can be obtained.

Remark 2: Similarly to Remark 1, we can state that the control laws in Problem 2 and 3 are rather a function of state and reference variables from time 0 to time t, to emphasize the sampled-data approach.

V. Inner-loop tracking

As mentioned in the introduction, many controllers for stabilizing the attitude dynamics of quadcopters have been proposed over the years. Here, a controller similar to the one proposed in [16] is employed, because it provides ULES and UaGAS for the attitude dynamics. The dynamics of the error variables in ((26),(27)) are given by

$$\begin{aligned}\dot{R}_e &= R_e S(\omega_e),\\ J\dot{\omega}_e &= S(J\omega)\omega - \tau_g - AR^\top v - C\omega + \tau\\ &\quad - J\tilde{R}^\top J^{-1}\left(S(J\bar{\omega})\bar{\omega} - \tau_g - A\bar{R}^\top\bar{v} - C\bar{\omega} + \bar{\tau}\right)\\ &\quad + J\left[\left(S(\omega)\tilde{R}^\top - \tilde{R}^\top S(\bar{\omega})\right)\bar{\omega} + S(\omega_e)R_e^\top\omega_d - R_e^\top\dot{\omega}_d\right],\end{aligned}$$

which, combined with the input

$$\begin{aligned}\tau &= -K_\omega\omega_e + K_R\sum_{i=1}^{3}k_i(e_i\times R_e^\top e_i)\\ &\quad - S(J\omega)\omega + \tau_g + AR^\top v + C\omega\\ &\quad + J\tilde{R}^\top J^{-1}\left(S(J\bar{\omega})\bar{\omega} - \tau_g - A\bar{R}^\top\bar{v} - C\bar{\omega} + \bar{\tau}\right)\\ &\quad - J\left[\left(S(\omega)\tilde{R}^\top - \tilde{R}^\top S(\bar{\omega})\right)\bar{\omega} + S(\omega_e)R_e^\top\omega_d - R_e^\top\dot{\omega}_d\right],\end{aligned} \tag{28}$$

result in the closed-loop system

$$\dot{R}_e = R_e S(\omega_e), \tag{29a}$$

$$J\dot{\omega}_e = -K_\omega\omega_e + K_R\sum_{i=1}^{3}k_i(e_i\times R_e^\top e_i), \tag{29b}$$

with distinct $k_i > 0$, $K_\omega \succ 0$ and $K_R \succ 0$. The following theorem asserts stability for this closed-loop system:

Theorem 1 (cf. [16], Theorem 4): The system described in ((29a),(29b)) has $(I, 0)$ as a ULES and UaGAS equilibrium. That is, let $E_c = \{I, \mathrm{diag}(1,-1,-1), \mathrm{diag}(-1,1,-1), \mathrm{diag}(-1,-1,1)\}$, then R_e converges to E_c and ω_e converges to zero. The equilibria $(R_e, 0)$ of ((29a),(29b)), where $R_e \in E_c \setminus \{I\}$, are unstable and the set of all initial conditions converging to the equilibria $(R_e, 0)$, where $R_e \in E_c \setminus \{I\}$ form a lower dimensional manifold.

Since R_e converges to I for almost all initial conditions it follows from (26) that $\tilde{R}$ converges to R_d for almost all initial conditions. Moreover, as $\omega_e \to 0$ it follows from (27) that $\tilde{\omega} \to R_e^\top\omega_d$, which combined with $R_e \to I$ results in $\tilde{\omega} \to \omega_d$. This solves Problem 3.

VI. Outer-loop tracking

For the outer-loop control problem defined in Problem 2 a model predictive control (MPC) strategy is used, that allows for the desired acceleration to be constrained to the set defined in (25), while still providing appropriate stability guarantees. As is common for many MPC strategies, the MPC law used is formulated in discrete time. An overview of the MPC strategy is provided in Figure 3. In this section, the actual optimal control problem (OCP) that is solved is presented in a stepwise manner, starting with the discretization of the dynamics ((15a)-(15b)) in Section VI-A. In this same section, the constraints on the continuous states are replaced by constraints on the discrete input, which still ensure the continuous-time satisfaction of the constraint on the desired acceleration a_d in (25). An MPC law is formulated based on the resulting discretized system in Section VI-C. This MPC law uses the fact that a globally stabilizing control law is known, which is introduced just before, in Section VI-B. In Section VI-D it will be shown that stability and constraint satisfaction is achieved for the continuous time system as well.

A. Discretization and input transformation

As mentioned, the MPC law used is formulated in discrete time, and in order to be able to provide twice differentiable desired accelerations for the inner loop the following extended version of (15) is considered:

$$\dot{\tilde{p}} = \tilde{v}, \tag{30a}$$

$$\dot{\tilde{v}} = -D\tilde{v} + a_d, \tag{30b}$$

$$\dot{a}_d = -\frac{1}{\gamma}(a_d + \eta), \tag{30c}$$

$$\dot{\eta} = -\frac{1}{\gamma}(\eta + s), \tag{30d}$$

where $\gamma > 0$, $\tilde{p}, \tilde{v}, a_d, \eta \in \mathbb{R}^3$ and $s \in \mathbb{R}^3$ is considered as the input. By designing a piecewise constant (zero-order-hold (ZOH)) control law for s in this model, instead of for a_d directly, it is ensured that a_d is twice differentiable for the original model (15). Recall that twice differentiability of the desired acceleration a_d is required by the inner-loop controller. Note that for the purpose of ensuring differentiability, integrator dynamics would have sufficed, however, the first-order filter dynamics will be used at the end of this subsection to guarantee constraint satisfaction in between sample times.

Next, a_d is constrained to lie in a more conservative set than strictly necessary, i.e., contained in the set defined in (25). However, with the benefit that this idea will result in an easier control design to generate a_d by decoupling the constraint in (25). First note that

$$\left\|a_d + \bar{T}\bar{z}_B\right\| \le \|a_d\| + \bar{T}.$$

Hence, if

$$\|a_d\| + \bar{T} \le T_{\max},$$

then the upper-bound in (25), i.e., $\left\|a_d + \bar{T}\bar{z}_B\right\| \le T_{\max}$, is met. Furthermore, it also required that $\bar{z}_B^\top a_d + \bar{T} > 0$, which can be ensured by requiring

$$\|a_d\| \le \bar{T} - \delta,$$

for some small $0 < \delta < \underline{\epsilon}$. These last two conditions describe two spheres for the desired acceleration a_d to be in of radius $T_{\max} - \bar{T}$ and $\bar{T} - \delta$, respectively. However, similarly to [20],

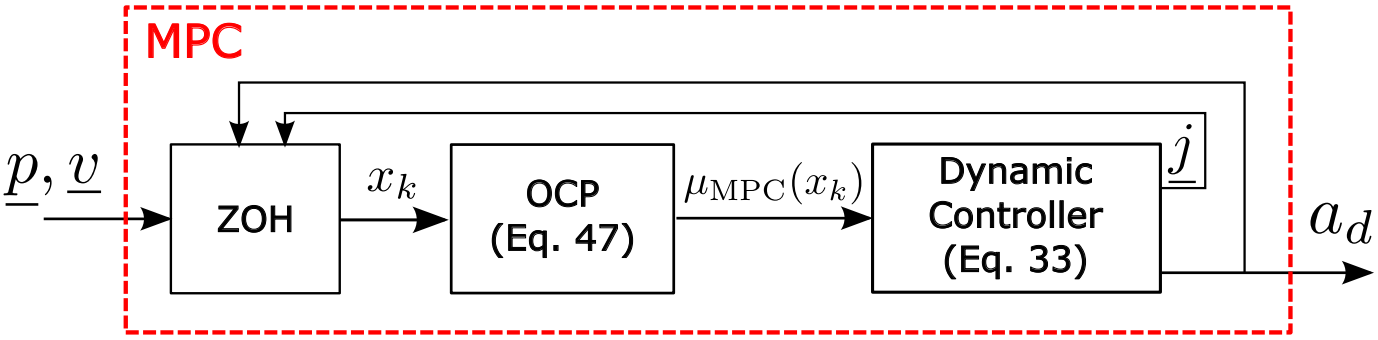

Fig. 3. Overview of the proposed MPC strategy, for one of the three axes. The position and velocity error dynamics are extended into a dynamic controller structure in (33), which is subsequently discretized using exact discretization with a ZOH input. This discretized system forms the basis for the optimal control problem (OCP) that is solved for the MPC strategy in (45).

a more conservative, box approximation $\mathcal{A}_L(\bar{T}) \subset \mathcal{A}(\bar{R}, \bar{T})$ is considered, where $\mathcal{A}(\bar{R}, \bar{T})$ as defined in (25) and

$$\mathcal{A}_L(\bar{T}(t)) := \{a_d \in \mathbb{R}^3 | -L \leq a_{d,i} \leq L, i \in \{1,2,3\}\} \quad (31)$$

with

$$L(t) = \frac{1}{\sqrt{3}} \min\left(\bar{T}(t) - \delta, T_{\max} - \bar{T}(t)\right), \quad (32)$$

which can be viewed as the largest cube that fits in the sphere that is most restricting, see Figure 4 for an illustration. Note that $a_d \in \mathcal{A}_L(\bar{T})$ implies $a_d \in \mathcal{A}(\bar{R}, \bar{T})$. This allows for

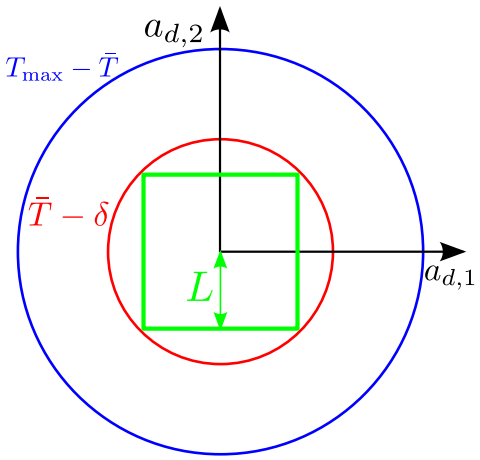

Fig. 4. Schematic depiction of the constraints on the desired acceleration $\boldsymbol{a_d}$ in 2D (so that circles and squares are considered, instead of spheres and cubes, respectively). The blue and red circles have radii of $\boldsymbol{T_{\max} - \bar{T}}$ and $\boldsymbol{\bar{T} - \delta}$, respectively. Note that the radius of each circle changes as the reference thrust $\boldsymbol{\bar{T}}$ changes, so that it is not always the case that the red circle is smaller than the blue circle. In this case however, the red circle is the smallest, and thus the most restricting for the desired acceleration $\boldsymbol{a_d}$ to have to lie in. In this circle the largest square that fits is depicted in green, and the constraint on the desired acceleration can now be decoupled by limiting each of the components to smaller than $\boldsymbol{\pm L}$.

the consideration of three separate, constrained, scalar systems given by

$$\dot{\underline{p}}^i = \underline{v}^i, \quad (33a)$$
$$\dot{\underline{v}}^i = -d^i \underline{v}^i + \underline{a}^i, \quad (33b)$$
$$\dot{\underline{a}}^i = -\frac{1}{\gamma}(\underline{a}^i + \underline{\eta}^i), \quad (33c)$$
$$\dot{\underline{\eta}}^i = -\frac{1}{\gamma}(\underline{\eta}^i + \underline{s}^i), \quad (33d)$$

with $i \in \{1,2,3\}$, $\underline{p}^i, \underline{v}^i, \underline{a}^i, \underline{\eta}^i, \underline{s}^i \in \mathbb{R}$, d^i the corresponding component of D and $\underline{a}^i \in \underline{\mathcal{A}}_L(\bar{T}(t)) := \{a \in \mathbb{R} | -L(t) \leq a \leq L(t)\}$. The index i will be omitted from here on whenever possible to improve readability, since the three scalar systems are of the same form.

Using exact discretization with a ZOH input, i.e.,

$$\underline{s}(t) = \underline{s}(t_k), \quad t \in [t_k, t_{k+1}), \quad (34)$$

with $t_k = kh$, $k \in \mathbb{N}_{\geq 0}$, and $h > 0$ the sampling period, results in the discrete-time system

$$x_{k+1} = Ax_k + Bu_k, \quad (35)$$

where $x_k = x(t_k)$, $x = \begin{bmatrix} \underline{p} & \underline{v} & \underline{a} & \underline{\eta} \end{bmatrix}^\top$ taking values in $\mathbb{R}^4$, $u_k = \underline{s}(t_k)$ taking values in $\mathbb{R}$ and $k \in \mathbb{N}$. The system matrices A and B are displayed in Table I. The time-varying bound in (32) can be lower bounded by a positive constant as

$$0 < \Delta = \inf_{t \in \mathbb{R}_{\geq 0}} L(t), \quad (37)$$

since $\bar{T}(t)$ is limited as in (8) and $\delta < \underline{\epsilon}$. The constraint $\underline{a} \in \underline{\mathcal{A}}_L(\bar{T}(t))$ is then ensured by initializing the controller such that $|\underline{a}(0)| \leq \Delta$, $|\underline{\eta}(0)| \leq \Delta$ and restricting the input as

$$-\Delta \leq u_k \leq \Delta, \text{ for all } k \in \mathbb{N}. \quad (38)$$

Indeed, this choice of controller saturation results in the satisfaction of the constraint (31) as can be seen by considering the evolution of the acceleration and the jerk of the system (33) in between sample times, which is given by

$$\underline{a}(t) = \alpha(t)\underline{a}_k - \beta(t)\underline{\eta}_k + [1 - \alpha(t) - \beta(t)]u_k, \quad (39)$$
$$\underline{\eta}(t) = \alpha(t)\underline{\eta}_k + [\alpha(t) - 1]u_k, \quad (40)$$

for $t \in [t_k, t_{k+1})$, where $\alpha(t) = e^{-t/\gamma}$ and $\beta(t) = \frac{t}{\gamma}e^{-t/\gamma}$. Note that $0 \leq \alpha(t) \leq 1$, $0 \leq \beta(t) \leq e^{-1}$, and $0 \leq \alpha(t) +$

$$A = \begin{bmatrix} 1 & \frac{1-e^{-dh}}{d} & \frac{\gamma(e^{-dh}+d\gamma-d\gamma e^{-h/\gamma}-1)}{d(d\gamma-1)} & \frac{\gamma(e^{-dh}+2d\gamma-d^2\gamma^2-2d\gamma e^{-h/\gamma}-dhe^{-h/\gamma}+d^2\gamma^2e^{-h/\gamma}+d^2\gamma he^{-h/\gamma}-1}{d(d\gamma-1)^2} \\ 0 & e^{-dh} & \frac{\gamma(e^{-h/\gamma}-e^{-dh})}{d\gamma-1} & \frac{\gamma e^{-h/\gamma}+he^{-h/\gamma}-\gamma e^{-dh}-d\gamma he^{-h/\gamma}}{(d\gamma-1)^2} \\ 0 & 0 & e^{-h/\gamma} & -\frac{h}{\gamma}e^{-h/\gamma} \\ 0 & 0 & 0 & e^{-h/\gamma} \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{e^{-dh}+dh+3d^2\gamma^2-2d^3\gamma^3+d^3\gamma^2h-3d^2\gamma^2e^{-h/\gamma}+2d^3\gamma^3e^{-h/\gamma}-2d^2\gamma h+d^3\gamma^2he^{-h/\gamma}-d^2\gamma he^{-h/\gamma}-1}{d^2(d\gamma-1)^2} \\ \frac{1-e^{-dh}-2d\gamma+d^2\gamma^2+2d\gamma e^{-h/\gamma}+dhe^{-h/\gamma}-d^2\gamma^2e^{-h/\gamma}-d^2\gamma he^{-h/\gamma}}{d(d\gamma-1)^2} \\ \frac{\gamma-\gamma e^{-h/\gamma}-he^{-h/\gamma}}{\gamma} \\ e^{-h/\gamma}-1 \end{bmatrix} \tag{36}$$

TABLE I
MATRICES OF THE SYSTEM IN (35)

$\beta(t) \leq 1$. From (39) we see that at each time instance $\underline{a}(t)$ is a convex combination of $\underline{a}_k$, $-\underline{\eta}_k$ and u_k, and similarly from (40) we see that $\underline{\eta}(t)$ is a convex combination of $\underline{\eta}_k$ and $-u_k$. Therefore, from $|\underline{a}_k| \leq \Delta$, $|\underline{\eta}_k| \leq \Delta$ and $|u_k| \leq \Delta$ we obtain $|\underline{a}(t)| \leq \Delta$ and $|\underline{\eta}(t)| \leq \Delta$ for all $t \in [t_k, t_{k+1})$.

B. Stabilizing input constraints

Before introducing our MPC law, we first derive constraints on the input that guarantee the existence of a positive definite, decrescent, time-independent Lyapunov function, and therefore guarantee UGAS (cf. [32]).

Inspired by the work in [37]–[40] on stabilization of neutral linear systems with input saturation, we first note that, based on the Jordan-decomposition of matrix A in (36), we can define the change of coordinates

$$\hat{x}_k = \begin{bmatrix} 0 & 1 & \frac{\gamma}{1-d\gamma} & \frac{-\gamma}{(1-d\gamma)^2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{h}{\gamma}e^{-h/\gamma} \\ 1 & \frac{1}{d} & \frac{\gamma}{d} & -\frac{\gamma}{d} \end{bmatrix} x_k, \tag{41}$$

which makes that the dynamics in (35) can alternatively be expressed as

$$\hat{x}_{k+1} = \begin{bmatrix} e^{-dh} & 0 & 0 & 0 \\ 0 & e^{-h/\gamma} & 1 & 0 \\ 0 & 0 & e^{-h/\gamma} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \hat{x}_k + \begin{bmatrix} \frac{1-e^{-dh}}{d(1-d\gamma)^2} \\ 1-(1+\frac{h}{\gamma})e^{-h/\gamma} \\ \frac{h}{\gamma}e^{-h/\gamma}(1-e^{-h/\gamma}) \\ \frac{h}{d} \end{bmatrix} u_k. \tag{42}$$

For these transformed dynamics, we can see that the dynamics of the first three states are input-to-state stable (ISS) with respect to u_k, i.e., the first three states will converge to zero, if the input is zero. Therefore, if we are able to design a controller specifying u_k such that the (scalar) dynamics

$$\hat{x}_{k+1,4} = \hat{x}_{k,4} + \frac{h}{d}u_k \tag{43}$$

is UGAS, then, due to ISS, we have that the system (42) is UGAS, and so is the system in (35). Since the state in (43) is scalar, it follows that to have stability (not asymptotic stability) it suffices to impose the following inequalities (next to the input saturation in (38)) on the input for all $k \in \mathbb{N}$:

$$\begin{aligned} 0 \leq u_k \leq -\frac{2d}{h}\hat{x}_{k,4}, &\qquad \text{if } \hat{x}_{k,4} \leq 0, \\ -\frac{2d}{h}\hat{x}_{k,4} \leq u_k \leq 0, &\qquad \text{if } \hat{x}_{k,4} \geq 0. \end{aligned}$$

In order to tighten these constraints to achieve asymptotic stability, we introduce a class $\mathcal{K}$-function β satisfying $\beta(q) \leq \min(\frac{d}{h}q, \Delta)$, for $q \in \mathbb{R}_{\geq 0}$, together with the input constraints

$$\beta(-\hat{x}_{k,4}) \leq u_k \leq -\frac{2d}{h}\hat{x}_{k,4} - \beta(-\hat{x}_{k,4}), \quad \text{if } \hat{x}_{k,4} \leq 0, \tag{44a}$$

$$-\frac{2d}{h}\hat{x}_{k,4} + \beta(\hat{x}_{k,4}) \leq u_k \leq -\beta(\hat{x}_{k,4}), \quad \text{if } \hat{x}_{k,4} \geq 0. \tag{44b}$$

Note that from (42) we have $\hat{x}_{k,4} = \begin{bmatrix} 1 & \frac{1}{d} & \frac{\gamma}{d} & -\frac{\gamma}{d} \end{bmatrix} x_k$, so that the constraint can be expressed in terms of x_k as well. The following theorem asserts that any input satisfying these constraints guarantees UGAS.

Theorem 2: The system (35) with input u_k constrained by (44) and (38) is uniformly globally asymptotically stable.

Proof: Consider the Lyapunov function candidate $V(\hat{x}_{k,4}) = \hat{x}_{k,4}^2$, then for the dynamics in (43) we have

$$\begin{aligned} V(\hat{x}_{k+1,4}) - V(\hat{x}_{k,4}) &= (\hat{x}_{k,4} + \tfrac{h}{d}u_k)^2 - \hat{x}_{k,4}^2 \\ &\leq \left[|\hat{x}_{k,4}| - \frac{h}{d}\beta(|\hat{x}_{k,4}|)\right]^2 - \hat{x}_{k,4}^2 \\ &= -|\hat{x}_{k,4}|\frac{h}{d}\beta(|\hat{x}_{k,4}|) \\ &\quad - \left[|\hat{x}_{k,4}| - \frac{h}{d}\beta(|\hat{x}_{k,4}|)\right]\frac{h}{d}\beta(|\hat{x}_{k,4}|) \\ &\leq -|\hat{x}_{k,4}|\frac{h}{d}\beta(|\hat{x}_{k,4}|) =: -\bar{\beta}(|\hat{x}_{k,4}|) \end{aligned}$$

Note that in the last inequality we used that $|\hat{x}_{k,4}| - \frac{h}{d}\beta(|\hat{x}_{k,4}|) \geq 0$ due to the definition of β and we define $\bar{\beta}(q) := |q|\frac{h}{d}\beta(q)$, $q \in \mathbb{R}_{\geq 0}$, which is a class $\mathcal{K}_\infty$-function. Hence, we can directly conclude UGAS of the system (43) (cf. [32]). From the ISS properties of the other states in the

transformed dynamics (42) we can conclude UGAS for that system as well. Finally, we note that the state in (42) is a linear transformation of the original state in (35), so that we conclude that the system in (35) is UGAS for any input satisfying the constraints (44) and (38). ■

Remark 3: The function β is used to stay away from control actions that keep the system at a state (or its opposite sign), i.e., to ensure we have a decreasing Lyapunov function at each time step. To make this constraint only slightly restrictive, one might take for example $\beta(q) = \frac{\varepsilon q}{1+q}$, with $\varepsilon \leq \min(\frac{d}{h}, \Delta)$ some small positive number.

C. Model predictive controller

Based on the constraints that guarantee stability presented in the previous section, the MPC law is then set up as solving the following optimal control problem (OCP) at each time step:

$$\begin{aligned}
&\min_{U_k} \; J(x_k, U_k) = l_T(x_{N|k}) + \sum_{\ell=0}^{N-1} l(x_{\ell|k}, u_{\ell|k}) \\
&\text{s.t. } x_{0|k} = x_k \\
&\quad x_{\ell+1|k} = Ax_{\ell|k} + Bu_{\ell|k},\ \ell \in \{0, 1, \dots, N-1\}, \\
&\quad |u_{\ell|k}| \leq \Delta,\ \ell \in \{0, 1, \dots, N-1\}, \\
&\quad \beta(-\hat{x}_{k,4}) \leq u_{0|k} \leq -\frac{2d}{h}\hat{x}_{k,4} - \beta(-\hat{x}_{k,4}), \text{ if } \hat{x}_{k,4} \leq 0, \\
&\quad -\frac{2d}{h}\hat{x}_{k,4} + \beta(\hat{x}_{k,4}) \leq u_{0|k} \leq -\beta(\hat{x}_{k,4}), \text{ if } \hat{x}_{k,4} \geq 0.
\end{aligned} \tag{45}$$

where $U_k = \begin{bmatrix} u_{0|k} & \dots & u_{N-1|k} \end{bmatrix}^\top$ contains the predicted future control inputs. Moreover, $N \in \mathbb{N}_{\geq 1}$ is the prediction horizon, $l : \mathbb{R}^4 \times \mathbb{R} \to \mathbb{R}_{\geq 0}$ is the stage/running cost, $l_T : \mathbb{R}^4 \to \mathbb{R}_{\geq 0}$ is the terminal cost and $x_{\ell|k}, u_{\ell|k}$ denote the prediction of the state and input at time step $\ell + k$, made at time k, respectively.

The first input of the optimal U_k, denoted as $U_k^* = \begin{bmatrix} u_{0|k}^* & \dots & u_{N-1|k}^* \end{bmatrix}^\top$, is then applied to the system, yielding the MPC policy $u_k = \mu_{MPC}(x_k)$ as a nonlinear function of the state, where

$$\mu_{MPC}(x_k) = u_{0|k}^*. \tag{46}$$

The input to the trajectory tracking dynamics in ((15a)-(15b)) is now obtained by solving the OCP at each $k \in \mathbb{N}$ for each scalar system $i \in \{1, 2, 3\}$ resulting in $\mu_{MPC,i}(x_k)$. Using these inputs as $\underline{s}^i(t) = \mu_{MPC,i}(x_k)$ for $t \in [t_k, t_{k+1})$ (see also (34)) in the linear system (33) yields $\underline{a}^i(t)$ and $a_d(t) = [\underline{a}^1(t), \underline{a}^2(t), \underline{a}^3(t)]^\top$.

The following result asserts the stability of the MPC policy:

Theorem 3: The origin of the system (35) with input (46), obtained from (45), for any $l : \mathbb{R}^4 \times \mathbb{R} \to \mathbb{R}$, $l_T : \mathbb{R}^4 \to \mathbb{R}$ and $N \in \mathbb{N}_{\geq 1}$, is uniformly globally asymptotically stable.

Proof: The optimization problem is always feasible, since a known solution in the form of (44) is feasible (one could take for instance the input exactly on the boundaries of the constraints). The constraints (44) guarantee uniform global asymptotic stability by construction. ■

Remark 4: Note that only the first input of the horizon $u_{0|k}$ in (45) needs to be constrained by Δ and (44), whereas for the rest of the horizon the constraint on the input can be chosen freely, whilst still guaranteeing feasibility and stability of the closed-loop system. This is because only the first input in the horizon is applied to the system, and there always exists a solution satisfying (44).

Remark 5: Note that the choice of cost function in (45) is not mentioned in the above theorem, i.e., any cost function and any horizon $N \in \mathbb{N}_{\geq 1}$ can be chosen whilst still guaranteeing global asymptotic stability of the system (35). In fact, global stability follows from the constraints imposed in (45). However, the performance of the MPC strategy does depend on the choice of cost function, since the controller is free to generate *any* input that satisfies the input constraints. In fact, any MPC control law that renders the closed loop with (35) UGAS can be used to guarantee that the overall scheme of Figure 3 results in guaranteed (almost) global asymptotic tracking in the sense of Theorem 5 below.

D. Stability of the continuous-time system

Although UGAS has been concluded for the system (35), care must be taken in concluding UGAS for the continuous-time system in (30). The state in (35) is exactly the state (30) due to the ZOH and the exact discretization used. However, the behavior of the continuous-time system in between sampling times needs to be considered as well.

Consider the continuous-time linear system in (30), which can be written as

$$\dot{\bar{x}}(t) = \bar{A}\bar{x}(t) + \bar{B}u(t), \tag{47}$$

where $\bar{x}(t) = [\tilde{p}^\top, \tilde{v}^\top, a_d^\top, \eta^\top]^\top \in \mathbb{R}^{12}$ and the input is generated by solving (45) for each axis and setting

$$u(t) = \begin{bmatrix} \mu_{MPC,1}(\bar{x}(kh)) \\ \mu_{MPC,2}(\bar{x}(kh)) \\ \mu_{MPC,3}(\bar{x}(kh)) \end{bmatrix}, \text{ for } t \in [kh, kh+h). \tag{48}$$

Then, for $t \in [kh, kh+h]$:

$$\bar{x}(t) = e^{\bar{A}(t-kh)}\bar{x}(kh) + \int_0^{t-kh} e^{\bar{A}s}\,\mathrm{d}s\ \bar{B}u(kh),$$

and therefore

$$\begin{aligned}
\|\bar{x}(t)\| &= \|e^{\bar{A}(t-kh)}\bar{x}(kh) + \int_0^{t-kh} e^{\bar{A}s}\,\mathrm{d}s\ \bar{B}u(kh)\| \\
&\leq \|e^{\bar{A}(t-kh)}\|\|\bar{x}(kh)\| + \|\int_0^{t-kh} e^{\bar{A}s}\,\mathrm{d}s\bar{B}\|\,\|u(kh)\| \\
&\leq c_1\|\bar{x}(kh)\| + c_2\,\|u(kh)\|\,.
\end{aligned}$$

These bounds on the inter-sample behavior, together with the fact that $\bar{x}(kh)$ is UGAS in discrete-time and we have that $u(x) \leq r(|x|)$ with $r(|x|)$ a class $\mathcal{K}$-function, guarantee that the system (47) in closed loop with the ZOH-input given by (46) is UGAS as well.

VII. Cascaded trajectory tracking controller

In the previous sections a desired acceleration for asymptotic stability of the position dynamics in the outer-loop problem was derived together with a controller that uses

the torque and thrust to acquire this desired control action asymptotically in the inner-loop problem. In order to conclude stability of the closed-loop system using the proposed strategy, the cascaded system is now examined. To conclude stability the following theorem will be used:

Theorem 4 (cf. [41], [16]): Consider a cascaded system $\dot{x} = f(t,x)$ with $f(t,0) = 0$, for all $t \in \mathbb{R}_{\geq 0}$, that can be written as

$$\dot{x}_1 = f_1(t,x_1) + g(t,x_1,x_2)x_2, \tag{49a}$$
$$\dot{x}_2 = f_2(t,x_2), \tag{49b}$$

with x_1 and x_2 taking values in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively. This system is a cascade of the systems

$$\dot{x}_1 = f_1(t,x_1) \tag{50}$$

and (49b). If the origins of the systems (50) and (49b) are UGAS and the solutions to (49) remain bounded, then the origin of the system (49) is UGAS.

Remark 6: In [41] it is assumed that $f_1(t,x_1)$ is continuously differentiable in (t,x_1) and $f_2(t,x_2)$, $g(t,x_1,x_2)$ are continuous in their arguments, and locally Lipschitz in x_2 and (x_1,x_2) respectively. However, only uniqueness of solutions is used in the proof of Theorem 4, so that the same theorem can be employed here in the stability proof for this system.

Although the choice of cost function in (45) is not relevant for providing stability guarantees in Theorem 3, in order to provide uniqueness of solutions, the cost function in (45) is limited to be strictly convex in the rest of this paper:

Assumption 2: The cost function in (45) is chosen to be strictly convex.

Note that this is not a major restriction, since in practice strictly convex cost functions are typically preferred as they often lead to lower computation times.

Consider the dynamics (7) and reference (9) in closed loop with the inputs (18), (28) and (48). The closed-loop system is then given by

$$\dot{\tilde{p}} = \tilde{v}, \tag{51a}$$
$$\dot{\tilde{v}} = -D\tilde{v} + a_d + TR(I - R_e^\top)e_3, \tag{51b}$$
$$\dot{a}_d = -\frac{1}{\gamma}(a_d + \eta), \tag{51c}$$
$$\dot{\eta} = -\frac{1}{\gamma}(\eta + u), \tag{51d}$$
$$\dot{R}_e = R_e S(\omega_e), \tag{51e}$$
$$J\dot{\omega}_e = -K_\omega \omega_e + K_R \sum_{i=1}^{3} k_i (e_i \times R_e^\top e_i). \tag{51f}$$

Theorem 5: The origin $(\tilde{p},\tilde{v},R_e,\omega_e) = (0,0,I,0)$ of (51) is UaGAS.

Proof: First note that (51) is a cascade of the systems ((51a)-(51d)) and ((51e)-(51f)). Since ((51e)-(51f)) is UaGAS, we consider our stability analysis on $\mathbb{R}^6 \times G$, where $G \subset SO(3) \times \mathbb{R}^3$ is the uniformly almost global region of attraction of ((51e)-(51f)).

Since ((51e)-(51f)) and (47) are UGAS on $\mathbb{R}^6 \times G$, it is only required to show that the solutions remain bounded to conclude UGAS of (51) on $\mathbb{R}^6 \times G$ according to Theorem 4. The dynamics ((51e)-(51f)) are bounded since they are UGAS on G and for ((51a)-(51d)) the boundedness of solutions is shown in Appendix I. The result follows from Theorem 4. Moreover, the origin $(\tilde{p},\tilde{v},R_e,\omega_e) = (0,0,I,0)$ is UGAS except for initial conditions in a set of measure zero, so that UaGAS of (51) can be concluded according to Definition 2. ■

Description	Symbol	Value
Gravitational constant	g	9.81 [m/s^2]
Inertia matrix	J	$\mathrm{diag}(2.5, 2.1, 4.3)$ [gm^2]
Translational drag coefficients	D	$\mathrm{diag}(0.26, 0.28, 0.42)$ [kg/s]
Gyroscopic torques	τ_g	$[0,0,0]^\top$
Cross drag coefficients	A	$0.1I$
Rotational Drag coefficients	C	$0.5I$
Maximum thrust (mass-normalized)	$T_{\max}$	45.21 [m/s^2]
Quadratic drag coefficients	D_q	$D/50$ [kg/s]

TABLE II
QUADCOPTER PARAMETERS [34] USED IN SIMULATIONS.

Remark 7: Note that the application of Theorem 4 on the space G (instead of the whole space) is possible because the inner-loop dynamics in closed-loop in ((51e)-(51f)) are independent from the outer-loop dynamics in closed-loop in ((51a)-(51d)).

Remark 8: The fact that we have UaGAS according to Theorem 5 gives robustness against uniformly bounded perturbations, cf. Lemma 9.3 in [32]. Note that this result requires *uniform* asymptotic stability.

Finally, it is shown that when UaGAS of (51) is realized, a solution to Problem 1 is found.

Corollary 1: The controller consisting of the inputs obtained from (18), (28) and (48) solves Problem 1.

Proof: In Theorem 5 UaGAS was already shown for the closed-loop system. Furthermore, from Theorem 5 it directly follows that, for all initial conditions $(p(0),v(0),R(0),\omega(0)) \in \mathbb{R}^3 \times \mathbb{R}^3 \times SO(3) \setminus M \times \mathbb{R}^3$, where $M \subset SO(3)$ is a set of measure zero, $(\tilde{p}(t),\tilde{v}(t)) \to (0,0)$, which combined with ((15a)-(15b)) results in $a_d(t) \to 0$. Then, by using (20), it follows that $z_{B,d}(t) \to e_3$ and from (21) it then follows that $R_d(t) \to I$. Combining this with $R_e(t) \to I$ results in $\tilde{R}(t) \to I$. Finally, since $\dot{a}_d(t) \to 0$, it follows that $\omega_d(t) \to 0$, which, combined with $\omega_e(t) \to 0$ and (27), results in $\tilde{\omega}(t) \to 0$, concluding the proof. ■

VIII. NUMERICAL CASE STUDY

In this section the effectiveness of the proposed strategy is illustrated through numerical examples. The dynamics in (7) are considered with the parameters of the in-house drone developed by the Robotics and Perception Group, University of Zurich [34], provided in Table II. We consider three challenging reference trajectories given in the form

$$\bar{r}(t) = \begin{bmatrix} \bar{p}_x(t) & \bar{p}_y(t) & \bar{p}_z(t) & \psi(t) \end{bmatrix}, \tag{52}$$

where $\psi(t)$ is the angle between the projection of x_B onto the $x_W - y_W$ plane and the x_W axis, see Figure 1. This

heading angle is only well defined when the thrust vector is limited by $0 \leq z_W^\top \bar{z}_B(t) < 1$, for $t \in \mathbb{R}_{\geq 0}$, i.e., when the quadcopter is upright, which is the case for all reference trajectories considered. Note that our controller does not use the heading angle, and can handle upside-down orientations. Any trajectory given in the form (52) fully defines the states and inputs of the reference model in (9) by following the differential flatness method employed in [30].

The reference trajectories are given by

$$\begin{aligned}
\mathcal{A}: \bar{r}(t) &= \begin{bmatrix} 2\cos(4t) & 2\sin(4t) & -10 + 2\sin(2t) & 0.2t \end{bmatrix}, \\
\mathcal{B}: \bar{r}(t) &= \begin{bmatrix} \cos(t) & \sin(t) & \sin(2t) - 0.5t & 0 \end{bmatrix}, \\
\mathbf{C}: \bar{r}(t) &= \begin{bmatrix} 2.5\cos(4t) & 2.5\sin(4t) & 2\sin(t) & 0 \end{bmatrix}.
\end{aligned}$$

The initial conditions are set to

$$p(0) = \begin{bmatrix} 1.5\bar{p}_x(0) & 0.75\bar{p}_y(0) & \bar{p}_z(0) \end{bmatrix}^\top, \tag{53a}$$
$$v(0) = \bar{v}(0), \tag{53b}$$
$$R(0) = R_x\left(170\frac{\pi}{180}\right) R_y\left(30\frac{\pi}{180}\right) R_z\left(20\frac{\pi}{180}\right), \tag{53c}$$
$$\omega(0) = \bar{\omega}(0), \tag{53d}$$

where $R_x(\theta)$ denotes a rotation around the x-axis according to

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(-\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$$

and R_y, R_z are rotations about the y and z axes defined similarly.

For all simulations the inner-loop gains are set to $K_\omega = 30J$, $K_R = 70J$ with J the inertia matrix and $[k_1, k_2, k_3] = [4.5, 5, 5.1]$. Furthermore, a sample time of $h = 0.1$ seconds is used and $\gamma = 0.1$ is set for the model used in the MPC controller. The input is generated by solving the OCP in (45) and transforming back to the system using (46). The following quadratic cost function is used for each subsystem corresponding to the x, y, z axes

$$J = x_{N|k}^\top Q_N x_{N|k} + \sum_{\ell=0}^{N-1} x_{\ell|k}^\top Q x_{\ell|k} + u_{\ell|k}^\top R u_{\ell|k},$$

with $N = 20$, $Q = \text{diag}(100, 1, 1, 1)$, $Q_N = 10Q$ and $R = 0.01$. The function used for the stabilizing constraints in (44) is given by $\beta(x) = \frac{\varepsilon x}{1+x}$, with $\varepsilon = 0.1\min(\frac{d}{h}, \Delta)$. Note that this function can be chosen independently of those of the cost function as long as $\beta(x) \leq \min(\frac{d}{h}x, \Delta)$ (cf. Remark 5 and Assumption 2).

A. Quadcopter Simulation

A simulation of the cascaded system, consisting of the outer and inner loop, is performed for trajectory $\mathcal{A}$. The resulting trajectory is visualized in Figure 8, where it can be seen that the quadcopter is able to recover from an upside-down initial attitude and converges to the reference trajectory. The corresponding position and attitude errors are shown in Figures 5 and 6, respectively. For the attitude errors the metric used is the Frobenius norm, i.e., $\eta(R_1, R_2) = \|R_1 - R_2\|_F$ and the distance of both the desired attitude R_d and the reference attitude error $\tilde{R}$ to I are evaluated. Note that the simulation was performed with the dynamics presented in (7), so violating Assumption 1.

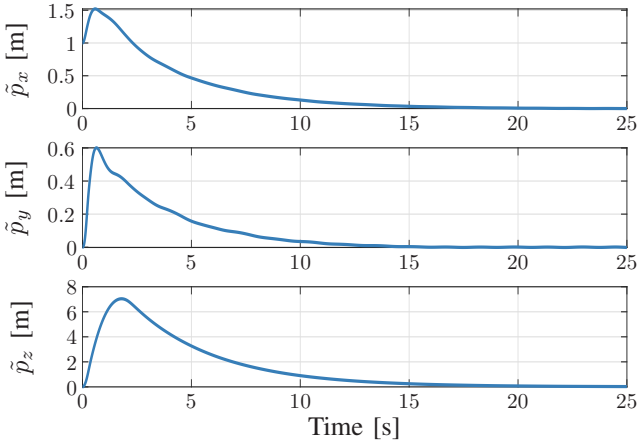

Fig. 5. Position errors $\tilde{\boldsymbol{p}} = [\tilde{\boldsymbol{p}}_{\boldsymbol{x}}, \tilde{\boldsymbol{p}}_{\boldsymbol{y}}, \tilde{\boldsymbol{p}}_{\boldsymbol{z}}]^\top$ for each axis.

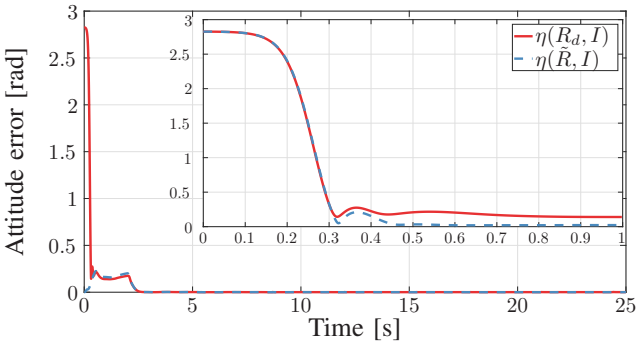

Fig. 6. Attitude errors for the desired attitude $\boldsymbol{\eta(R_d, I)} = \|\boldsymbol{R_d} - \boldsymbol{I}\|_{\boldsymbol{F}}$ (red, solid) and the reference attitude $\boldsymbol{\eta(\tilde{R}, I)} = \left\|\tilde{\boldsymbol{R}} - \boldsymbol{I}\right\|_{\boldsymbol{F}}$ (blue, dashed). The insert highlights the first second.

B. Robustness to unmodeled dynamics and measurement uncertainties

To demonstrate the robustness of our method against unmodeled dynamics and measurement uncertainties, we perform simulations in the scenario where the real system model contains a quadratic drag term on the velocity, i.e., the system simulated is that in (7) with (7b) replaced by

$$\dot{v} = gz_W - Tz_B - RDR^\top v - D_q v|v| \tag{54}$$

with D_q the quadratic drag coefficient shown in Table II. Moreover, measurement noise is added to the position and velocity vectors, i.e., we use the following measurements instead of the true states

$$\begin{aligned} p_m &= p + \rho, \\ v_m &= v + \nu, \end{aligned}$$

where $\rho \sim \mathcal{N}(0, 0.01I)$, $\nu \sim \mathcal{N}(0, 0.001I)$ and $\mathcal{N}(a, B)$ represents a Gaussian distribution with mean a and covariance matrix B. We simulate all three trajectories. The initial conditions for the translational and angular velocities are started on the reference, as in (53), however the position and rotational

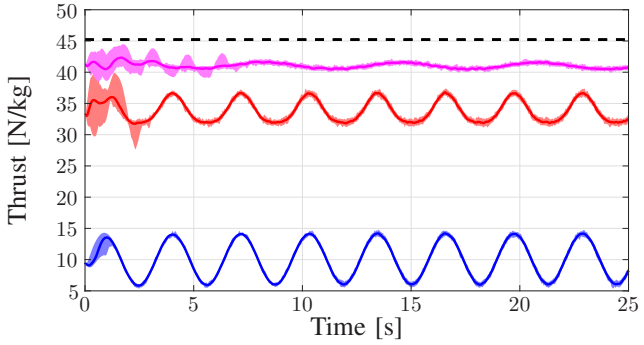

Fig. 7. Thrust T during the $\mathcal{A}$ (red), $\mathcal{B}$ (magenta) and $\mathcal{C}$ (blue) trajectories together with the maximum thrust $T_{\max}$ (black, dashed). The bold line represents the mean value over the 50 simulations, and the colored area is between the maximum and minimum value over the 50 simulations. This shows that the thrust stays below the maximum value and thus satisfies the constraint (8).

matrices are initialized according to

$$p(0) = \mathcal{N}(\bar{p}_{\text{avg}}, 2),$$
$$R(0) = R_x\left(\mathcal{N}(0,50)\frac{\pi}{180}\right) R_y\left(\mathcal{N}(0,50)\frac{\pi}{180}\right) R_z\left(\mathcal{N}(0,50)\frac{\pi}{180}\right),$$

that is, the position is started randomly according to a Gaussian distribution around the mean of the trajectory with a variance of 2 meters in each direction, and the rotation matrix is initialized in a random orientation with a variance of 50 degrees about each axis. The MPC settings remain the same, except for the horizon and sample time, which are set to $N = 5$ and $h = 0.05$ seconds, respectively. Note that we still generate the control input using (45), i.e., we use the linear drag model in our controller. We simulate each trajectory 50 times for 25 seconds each.

Figure 7 displays the thrust during each trajectory, with the mean value over the 50 simulations shown with the line and the area of the same color represents the bound within which the thrust falls for all simulations. This shows that the thrust stays below the maximum value and thus satisfies the constraint (8) for all trajectories and all simulations. In Figure 9 boxplots of the root-mean-square (RMS) errors for each trajectory and each axis are shown, where we only consider the final 2 seconds of each trajectory to compute the RMS value, ensuring that the quadcopter has converged to the trajectory. From this figure it becomes clear that despite the noisy measurements, varying initial conditions and model uncertainties (linear versus quadratic drag terms), the quadcopter always converges to the reference trajectory, and achieves a low RMS error. Finally, we note that the average and maximum computation times for the MPC algorithm during the 50 simulations of the 'Original' trajectory were 0.026 and 0.21 milliseconds, respectively, computed on a laptop with an Intel Core i7-6700HQ processor with 8 GB RAM using MATLAB R2021a, demonstrating the low computational effort required to deploy our method.

IX. Conclusions

In this paper a new cascaded controller has been presented, that enables guaranteed trajectory tracking for quadcopters while taking into account the limited thrust capabilities of quadcopters. The method uses the differential flatness property of the quadcopter dynamics in combination with a uniformly almost globally asymptotically stable inner-loop controller and a novel, MPC-based, uniformly globally asymptotically stable outer-loop controller. This combination allows for providing convergence guarantees for trajectory tracking, incorporation of future reference information and constraint handling. Moreover, the methodology used allows the MPC strategy to be formulated as three quadratic problems, each with just four states and one input, and only linear constraints on the first input of the horizon, allowing for fast computation times. The advantages of the method are shown in a numerical case study, where it is demonstrated that our method can deal with measurement uncertainties, varying initial conditions and model uncertainties (specifically, quadratic versus linear drag terms). Note that while our method is presented here for quadcopters specifically, it is applicable to any VTOL aircraft that has a configuration where the thrust provided by all the rotors can be decomposed into a body thrust and torques around the principal axes and follows the dynamics as presented in (7).

Future work includes verifying our method on a hardware quadcopter platform, as well as expanding our method, if possible, to be able to guarantee stability for a quadratic drag model. The latter would entail expanding the stability proof for the outer loop controller from a *linear*, constrained, sampled-data system with inter-sample constraint satisfaction to a *nonlinear*, constrained, sampled-data system with inter-sample constraint satisfaction, which is not straightforward.

A second avenue of future research concerns the conservatism introduced in our approach in order to achieve the formal stability guarantees. One improvement would be the development of less conservative constraints on the desired acceleration a_d, by, for example, relaxing the need for decoupling and/or providing tighter bounds on the evolution of the desired acceleration in between sample times.

Finally, an output-based variant, together with the incorporation of measurement inaccuracies in the development of the controller, will make the method more suitable for practical applications.

Appendix I
Proof of Boundedness of Solutions for ((51a)-(51d))

Consider the dynamics in (51), which, using (47), can be written as

$$\dot{\bar{x}}(t) = \bar{A}\bar{x}(t) + \bar{B}u(\bar{x}(kh)) + \begin{bmatrix} 0 \\ I_3 \\ 0 \\ 0 \end{bmatrix} TR(I - R_e^T)e_3$$

for $t \in [kh, kh + h)$ with $\bar{x}(t) \in \mathbb{R}^{12}$. Using the observation that the x, y and z dynamics were decoupled, these dynamics can be considered separately, and using the fact that T is (globally) bounded and the (R_e, ω_e) dynamics are locally exponentially stable, only boundedness (and therefore stability)

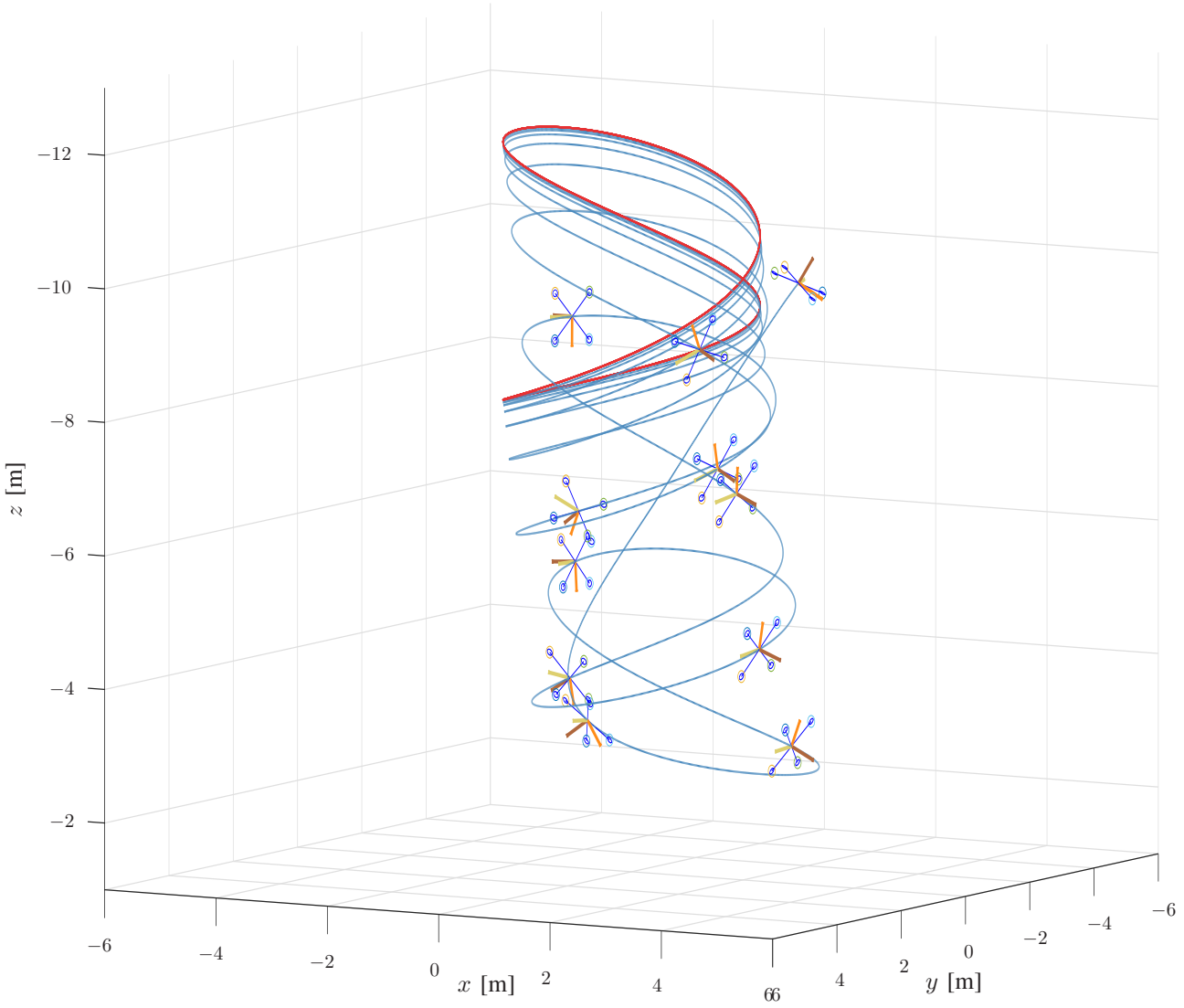

Fig. 8. Three dimensional plot of the quadcopter trajectory. The position trajectory of the reference and quadcopter are shown by the red and blue lines, respectively. Snapshots of the quadcopter during the first 3 seconds of the trajectory are shown, where the orange, yellow and brown lines depict the $\boldsymbol{x}_B$, $\boldsymbol{y}_B$ and $\boldsymbol{z}_B$ axes, respectively. The right most snapshot is the initial condition. Note that the quadcopter is able to recover from an initial upside-down attitude and converges to the reference.

of the following dynamics needs to be studied:

$$\dot{x}(t) = Ax(t) + B\mu_{MPC}(x(kh)) + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \phi(t)$$

for $t \in [kh, kh+h)$ with $x(t) \in \mathbb{R}^4$, A and B are the system matrices for a single axis and where $|\phi(t)| \leq ce^{-\lambda t}$ for some constants $\lambda > 0$ and $c > 0$, where c depends on the initial condition $(R_e(t_0), \omega_e(t_0))$.

First step is to show boundedness of $\hat{x}_{k,4}$ for the dynamics

$$\hat{x}_{k+1,4} = \hat{x}_{k,4} + \frac{h}{d} u_{\text{MPC}}(x_k) + 2\hat{c}e^{-\lambda kh}. \tag{55}$$

For the function $V_k = \hat{x}_{k,4}^2$ we obtain

$$V_{k+1} - V_k \leq 2\sqrt{V_k}\hat{c}e^{-\lambda kh} \leq 2\sqrt{V_k}\hat{c}e^{-\lambda kh} + \hat{c}^2 e^{-2\lambda kh}. \tag{56}$$

Observe that the difference equation

$$F_{k+1} = F_k + 2\sqrt{F_k}\hat{c}e^{-\lambda kh} + \hat{c}^2 e^{-2\lambda kh} \tag{57}$$

is solved by

$$F_k = \left(\hat{c}\frac{1 - e^{-\lambda kh}}{1 - e^{-\lambda h}} + \sqrt{F_0} \right)^2 \tag{58}$$

and therefore

$$\begin{aligned} V_k &= \left(\hat{c}\frac{1 - e^{-\lambda kh}}{1 - e^{-\lambda h}} + \sqrt{V_0} \right)^2 \\ &\leq \left(\frac{\hat{c}}{1 - e^{-\lambda h}} + \sqrt{V_0} \right)^2 \end{aligned}$$

This shows boundedness of $\hat{x}_{4,k}$. From the ISS properties of the other states in the transformed dynamics (42), we can conclude boundedness of $\tilde{p}$, $\tilde{v}$, a_d and η in ((51a)-(51d)) at sample times. From continuity also boundedness in between sample times follows in this case.

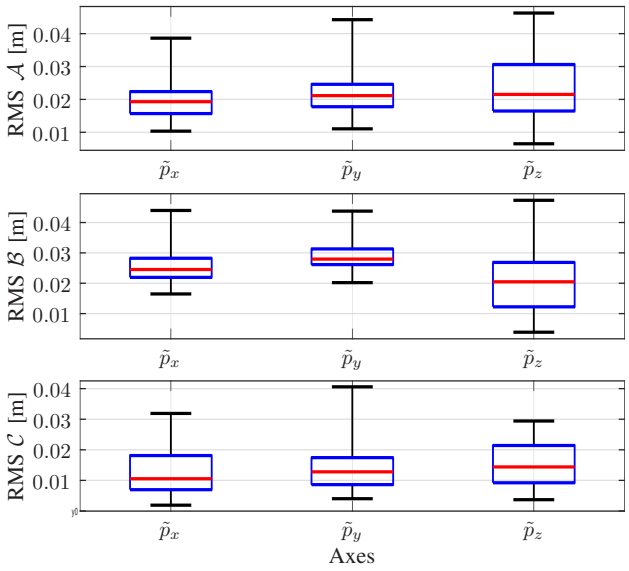

Fig. 9. Box plots of the RMS values of the position errors $\tilde{\boldsymbol{p}} = [\tilde{\boldsymbol{p}}_{\boldsymbol{x}}, \tilde{\boldsymbol{p}}_{\boldsymbol{y}}, \tilde{\boldsymbol{p}}_{\boldsymbol{z}}]^\top$ for each axis, for each trajectory. The red line represents the median, whereas the top and bottom edges of the blue box represent the 75$^{\text{th}}$ and 25$^{\text{th}}$ percentiles, respectively. The black whiskers extend to the most extreme data points.

REFERENCES

[1] B. de Miguel Molina and M. Segarra Oña, *The Drone Sector in Europe*. Cham: Springer International Publishing, 2018, pp. 7–33.

[2] J. Turner, P. Kenkel, R. B. Holcomb, and B. Arnall, "Economic Potential of Unmanned Aircraft in Agricultural and Rural Electric Cooperatives," Southern Agricultural Economics Association, 2016 Annual Meeting, February 6-9, 2016, San Antonio, Texas 230047, 2016.

[3] A. Gynnild, "The robot eye witness," *Digital Journalism*, vol. 2, no. 3, pp. 334–343, 2014.

[4] J. Linchant, J. Lisein, J. Semeki, P. Lejeune, and C. Vermeulen, "Are unmanned aircraft systems (UASs) the future of wildlife monitoring? a review of accomplishments and challenges," *Mammal Review*, vol. 45, no. 4, pp. 239–252, 2015.

[5] H. Durrant-Whyte, N. Roy, and P. Abbeel, *Construction of Cubic Structures with Quadrotor Teams*. MITP, 2012, pp. 177–184.

[6] M. Balasingam, "Drones in medicine—the rise of the machines," *International Journal of Clinical Practice*, vol. 71, no. 9, p. e12989, 2017.

[7] W. Yoo, E. Yu, and J. Jung, "Drone delivery: Factors affecting the public's attitude and intention to adopt," *Telematics and Informatics*, vol. 35, no. 6, pp. 1687–1700, 2018.

[8] P. Voosen, "NASA to fly drone on Titan," *Science*, vol. 365, no. 6448, pp. 15–15, 2019.

[9] M. Hassanalian, D. Rice, and A. Abdelkefi, "Evolution of space drones for planetary exploration: A review," *Progress in Aerospace Sciences*, vol. 97, pp. 61–105, 2018.

[10] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 20–32, 9 2012.

[11] B. J. Emran and H. Najjaran, "A review of quadrotor: An underactuated mechanical system," *Annual Reviews in Control*, vol. 46, pp. 165–180, 2018.

[12] G. Cai, J. Dias, and L. Seneviratne, "A survey of small-scale unmanned aerial vehicles: Recent advances and future development trends," *Unmanned Systems*, vol. 02, no. 02, pp. 175–199, 2014.

[13] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *2005 IEEE Int. Conference on Robotics and Automation*, 4 2005, pp. 2247–2252.

[14] M. Hehn and R. D'Andrea, "A frequency domain iterative learning algorithm for high-performance, periodic quadrocopter maneuvers," *Mechatronics*, vol. 24, no. 8, pp. 954–965, 2014.

[15] J.-M. Kai, G. Allibert, M.-D. Hua, and T. Hamel, "Nonlinear feedback control of quadrotors exploiting first-order drag effects," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8189–8195, 2017, 20th IFAC World Congress.

[16] E. Lefeber, S. J. A. M. van den Eijnden, and H. Nijmeijer, "Almost global tracking control of a quadrotor UAV on SE(3)," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 12 2017, pp. 1175–1180.

[17] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, "Control of a quadrotor with reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2096–2103, 2017.

[18] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789 – 814, 2000.

[19] M. Greeff and A. P. Schoellig, "Flatness-based model predictive control for quadrotor trajectory tracking," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 10 2018, pp. 6740–6745.

[20] M. W. Mueller and R. D'Andrea, "A model predictive controller for quadrocopter state interception," in *2013 European Control Conference (ECC)*, 2013, pp. 1383–1389.

[21] A. Bemporad, C. Pascucci, and C. Rocchi, "Hierarchical and hybrid model predictive control of quadcopter air vehicles," *IFAC Proceedings Volumes*, vol. 42, no. 17, pp. 14–19, 2009, 3rd IFAC Conference on Analysis and Design of Hybrid Systems.

[22] M. Khan, M. Zafar, and A. Chatterjee, "Barrier functions in cascaded controller: Safe quadrotor control," in *2020 American Control Conference (ACC)*, 2020, pp. 1737–1742.

[23] D. Kooijman, A. P. Schoellig, and D. J. Antunes, "Trajectory tracking for quadrotors with attitude control on $\mathcal{S}^2 \times \mathcal{S}^1$," in *2019 18th European Control Conference (ECC)*, 6 2019, pp. 4002–4009.

[24] I. Fantoni, R. Lozano, and F. Kendoul, "Asymptotic stability of hierarchical inner-outer loop-based flight controllers," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 1741 – 1746, 2008, 17th IFAC World Congress.

[25] R. Lozano, A. Sanchez, S. Salazar-Cruz, I. Fantoni, and J. Torres, "Discrete-time stabilization of integrators in cascade: Real-time stabilization of a mini-rotorcraft," in *Proceedings of the 45th IEEE Conference on Decision and Control*, 12 2006, pp. 6265–6270.

[26] M. Kamel, K. Alexis, M. Achtelik, and R. Siegwart, "Fast nonlinear model predictive control for multicopter attitude tracking on SO(3)," in *2015 IEEE Conference on Control Applications (CCA)*, 2015, pp. 1160–1166.

[27] E. Small, P. Sopasakis, E. Fresk, P. Patrinos, and G. Nikolakopoulos, "Aerial navigation in obstructed environments with embedded nonlinear model predictive control," in *2019 18th European Control Conference (ECC)*, 2019, pp. 3556–3563.

[28] J. C. Pereira, V. J. Leite, and G. V. Raffo, "Nonlinear model predictive control on SE(3) for quadrotor aggressive maneuvers," *Journal of Intelligent & Robotic Systems*, vol. 101, pp. 1–15, 2021.

[29] S. Sun, A. Romero, P. Foehn, E. Kaufmann, and D. Scaramuzza, "A comparative study of nonlinear mpc and differential-flatness-based control for quadrotor agile flight," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3357–3373, 2022.

[30] 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.

[31] A. Andriën, D. Kremers, D. Kooijman, and D. Antunes, "Model predictive tracking controller for quadcopters with setpoint convergence guarantees," in *2020 American Control Conference (ACC)*, 2020, pp. 3205–3210.

[32] H. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, New Jersey: Prentice Hall, 2002.

[33] S. P. Bhat and D. S. Bernstein, "A topological obstruction to continuous global stabilization of rotational motion and the unwinding phenomenon," *Systems & Control Letters*, vol. 39, no. 1, pp. 63–70, 2000.

[34] A. Romero, R. Penicka, and D. Scaramuzza, "Time-optimal online replanning for agile quadrotor flight," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7730–7737, 2022.

[35] A. Romero, S. Sun, P. Foehn, and D. Scaramuzza, "Model predictive contouring control for time-optimal quadrotor flight," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3340–3356, 2022.

[36] J. M. Butt, X. Ma, X. Chu, and K. W. Samuel Au, "Adaptive flight stabilization framework for a planar 4r-foldable quadrotor: Utilizing morphing to navigate in confined environments," in *2022 American Control Conference (ACC)*, 2022, pp. 1–7.

[37] X. Wang, H. F. Grip, A. Saberi, and T. A. Johansen, "A new low-and-high gain feedback design using MPC for global stabilization of

linear systems subject to input saturation," in *2012 American Control Conference (ACC)*, 6 2012, pp. 2337–2342.
[38] X. Wang, A. Saberi, A. A. Stoorvogel, and P. Sannuti, "Simultaneous global external and internal stabilization of linear time-invariant discrete-time systems subject to actuator saturation," *Automatica*, vol. 48, no. 5, pp. 699 – 711, 2012.
[39] P. Hou, A. Saberi, Z. Lin, and P. Sannuti, "Simultaneous external and internal stabilization for continuous and discrete-time critically unstable linear systems with saturating actuators," *Automatica*, vol. 34, no. 12, pp. 1547 – 1557, 1998.
[40] A. Saberi, A. A. Stoorvogel, and P. Sannuti, *Internal and External Stabilization of Linear Systems with Constraints*. Birkhäuser Boston, 2012.
[41] E. Panteley and A. Loría, "Growth rate conditions for uniform asymptotic stability of cascaded time-varying systems," *Automatica*, vol. 37, no. 3, pp. 453–460, 2001.

A.R.P. (Alex) Andriën received the M.Sc. degree in Systems and Control and the Ph.D. degree in the subject of control and robotics from the Eindhoven University of Technology, in 2016 and 2022, respectively. As part of his master's studies he performed a six-month internship at the Massachusetts Institute of Technology (MIT) in 2014.

He is currently working as Lead Control Engineer at Avular, a mobile robotics company. His research interests include optimization, as well as estimation, planning and control algorithms for robotic applications.

E. (Erjen) Lefeber received the M.Sc. degree in applied mathematics and the Ph.D. degree in the subject of tracking control of nonlinear mechanical systems from the University of Twente, Enschede, The Netherlands, in 1996 and 2000, respectively.

Since 2000, he has been an Assistant Professor with the Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands, where he was involved in the modeling and control of manufacturing systems from 2000 to 2015, and then he joined the Dynamics and Control Group in 2015. His current research interests include nonlinear control theory, in particular the control of drones and the control of platooning vehicles.

D. J. (Duarte) Antunes was born in Viseu, Portugal. He received the Licenciatura in Electrical and Computer Engineering from the Instituto Superior Técnico (IST), Lisbon, in 2005. He did his PhD (cum laude) from 2006 to 2011 in the research field of Automatic Control at the Institute for Systems and Robotics, IST, Lisbon, in close collaboration with the University of California, Santa Barbara.

From 2011 to 2013 he held a postdoctoral position at the Eindhoven University of Technology (TU/e). He is currently an Associate Professor at the Department of Mechanical Engineering of TU/e. His research interests include networked control systems, stochastic control, approximate dynamic programming, and robotics.

W.P.M.H. (Maurice) Heemels (Fellow, IEEE) received the M.Sc. degree in mathematics and the Ph.D. degree in control theory (both summa cum laude) from the Eindhoven University of Technology (TU/e), the Netherlands, in 1995 and 1999, respectively. From 2000 to 2004, he was with the Electrical Engineering Department, TU/e, as an assistant professor, and from 2004 to 2006 with the Embedded Systems Institute (ESI) as a research fellow. Since 2006, he has been with the Department of Mechanical Engineering, TU/e, where he is currently a Full Professor. He held visiting professor positions at the Swiss Federal Institute of Technology (ETH), Switzerland (2001) and at the University of California at Santa Barbara (2008). In 2004, he worked also at the company Oce, the Netherlands. His current research interests include hybrid and cyberphysical systems, networked and event-triggered control systems and constrained systems including model predictive control. Dr. Heemels served/s on the editorial boards of *Automatica*, *Nonlinear Analysis: Hybrid Systems*, *Annual Reviews in Control*, and *IEEE Transactions on Automatic Control*. He was a recipient of a personal VICI grant awarded by NWO (Dutch Research Council). He is a Fellow of the IEEE and IFAC and is currently the chair of the IFAC Technical Committee on Networked Systems (2017-2023). He was the recipient of the 2019 IEEE L-CSS Outstanding Paper Award and the 2023 Automatica Paper Prize Award. He was elected as a member of the IEEE-CSS Board of Governors (2021-2023).