AIAA SciTech Forum
8-12 January 2024, Orlando, FL
AIAA SCITECH 2024 Forum

10.2514/6.2024-0755

A Hybrid Model Reference Adaptive Control System for Multi-Rotor Unmanned Aerial Vehicles

Mattia Gramuglia*, Giri Mugundan Kumar†, and Andrea L'Afflitto‡
Virginia Tech, Blacksburg, VA, 24061

In this paper, a novel model reference adaptive control (MRAC) architecture for nonlinear, time-varying, hybrid dynamical systems is applied for the first time to design the control system of a multi-rotor unmanned aerial vehicle (UAV). The proposed control system is specifically designed to address problems of practical interests involving autonomous UAVs transporting unknown, unsteady payloads and subject to instantaneous variations both in their state and in their dynamics. These variations can be due, for instance, to the payload's dynamics, impacts between the payload and its casing, and sudden payload dropping and pickup. The proposed hybrid MRAC architecture improves the UAV's trajectory tracking performance over classical MRAC also in the presence of motor failures. The applicability of the proposed framework is validated numerically through the first use of the high-fidelity simulation environment PyChrono for autonomous UAV control system testing.

Essential Nomenclature

$\mathbb{I}, \mathbb{J}(\cdot)$	=	Inertial and body reference frames
$\Gamma_J(\cdot,\cdot)$	=	Inverse of the Jacobian matrix
Θ	=	Unknown matrix capturing parametric uncertainties
$\Phi(\cdot), \overline{\Phi}(\cdot)$	=	Regressor vector and extended regressor vector
$\phi(\cdot), \theta(\cdot), \psi(\cdot)$	=	Roll, pitch, and yaw angles in a 3-2-1 sequence
Σ, σ	=	Mode set and mode index
$\omega(\cdot)$	=	Angular velocity of the UAV
$\omega_{\text{cmd}}(\cdot)$	=	Angular velocity needed to follow the reference attitude
$\omega_{\text{ref}}(\cdot)$	=	Angular velocity of the reference model
$c_{\text{D},\sigma}$	=	Drag coefficient
$e(\cdot)$	=	Trajectory tracking error
$I_\sigma, \overline{I}_\sigma$	=	Inertia matrix and estimated inertia matrix of the UAV, including payload
$\mathcal{M}$	=	Mixer matrix
$m, \overline{m}$	=	Mass and estimated mass of the UAV
P_σ	=	Solution of a Lyapunov equation
$r(\cdot)$	=	Reference command input
$r_A(\cdot)$	=	Position of the UAV's reference point A
$\mathcal{S}_\sigma, \mathcal{S}_{\sigma,\text{ref}}$	=	Sets of resetting events in the plant model and the reference mode, respectively
s_σ	=	Generic element of a convergent series
$T(\cdot)$	=	Vector of the propellers' thrust forces
t	=	Time variable
$t_{\text{ref}}(\cdot)$	=	Resetting time of the reference model
$v_A(\cdot)$	=	Translational velocity of the UAV's reference point
$x(\cdot)$	=	Generic state vector and UAV state vector
$x_{\text{ref}}(\cdot)$	=	State of a generic reference model
$x_{\text{tran,ref}}(\cdot)$	=	State of the reference model for the translational dynamics

*Graduate Student, Department of Mechanical Engineering.
†Graduate Student, Department of Mechanical Engineering.
‡Associate Professor, Grado Department of Industrial and Systems Engineering, AIAA Senior Member.

I. Introduction

This paper presents the first application of a novel model reference adaptive control (MRAC) system for hybrid plants affected by parametric and matched uncertainties [1] to the design of a control system for autonomous multi-rotor unmanned aerial vehicles (UAVs). The use of multi-rotor UAVs in industrial problems of practical interest, such as the transportation of large and unknown goods as well as the interaction with objects, obstacles, and human operators, pose several challenges to traditional autopilots. Indeed, these applications require that the UAV's control systems are robust to uncertainties in the vehicle's and the payload's dynamical model, impulsive forces and moments, and sudden changes in the state vector capturing the vehicle's position, attitude, translational velocity, and angular velocity. For instance, dropping some large and heavy payload implies sudden changes in the UAV's inertial properties. Furthermore, by Newton's third law, the vehicle experiences a sudden change in its state vector due to the reaction forces. This problem becomes particularly significant when the sequence of discontinuities in the vehicle's dynamics is rapid and large. Commercial-off-the-shelf autopilots for UAVs are based on relatively simple control laws, which are not robust to large parametric uncertainties, and existing control systems for UAVs, including the vast majority of research-grade ones, are designed assuming that the differential equations modeling the vehicle's dynamics are at least continuous and that their solutions do not experience discontinuities of any kind.

Recently, the authors proposed the first extension of the LaSalle-Yoshizawa theorem to nonlinear, time-varying, hybrid dynamical systems, that is, to dynamical systems that experience instantaneous variations both in their dynamical model and their state. This result applies to Krasovskii solutions of hybrid dynamical systems, which are either complete or non-complete solutions, that is, which can be defined on the semi-infinite time horizon or not. This unique result has enabled the design of the first MRAC system for nonlinear, time-varying, hybrid dynamical plants affected by matched and parametric uncertainties as well as uncertainties in the uncontrolled plant's discontinuities [1].

In this paper, we apply this MRAC architecture for nonlinear, time-varying, hybrid dynamical plants to design both the inner loop and the outer loop of an X8-copter, that is, a UAV equipped with four pairs of propellers placed at the extremities of the frame of a classical quadcopter. The usefulness of these UAVs lays in their ability to produce more thrust than classical quadcopters of the same size and their enhanced compactness compared to octocopers able to produce an equivalent thrust. Furthermore, multiple motors and propellers guarantee enhanced robustness to failures compared to classical quadcopters.

The X8-copters considered in this paper are tasked with challenging missions such as, for instance, transporting some unknown payloads, whose mass is as large as half of the overall vehicle's mass, which are free to move in the payload's compartment and, hence, collide with one another and the walls of the payload compartment, and are dropped at unknown points in time over the course of the mission. Due to the unknown inertial properties of the payload, the impulsive nature of the payload-to-payload and payload-to-UAV interactions as well as of the payload dropping, and the instantaneous changes in the UAV's state, an MRAC system able to guarantee tracking of a user-defined reference trajectory is highly preferable over classical control architectures.

To verify the applicability of the proposed control system in realistic scenarios and show its improved performance over the proportional-integral-derivative (PID) control law, which underlies the majority of commercial-off-the-shelf autopilots, and a classical MRAC system [2, Ch. 9], we present the results of model-in-the-loop numerical simulations involving an X8-copter following a user-defined reference trajectory, experiencing the failure of one motor and a significant fault in an additional motor, and the dropping of large and heavy payloads at unknown time instants. The physics engine used to perform these simulations is provided by PyChrono [3, 4]. Initially developed for multi-body simulations involving a large number of bodies, specifically for terramechanics and off-road vehicles, Chrono and its Python-wrapped version PyChrono have undergone significant enhancements in recent years, making them suitable for simulating complex robotic systems. This software package supports rigid and flexible body dynamics, such as nonlinear finite element analysis (FEA), friction and contact handling, deformable terrain simulation, modules for fluid-solid interaction based on Navier–Stokes equations with smoothed-particle hydrodynamics (SPH), and granular dynamics [5]. To the authors' knowledge, Chrono has not been utilized by researchers for simulating multi-copters or other aerial vehicles. The only publicly disclosed example of the use of Chrono on aerial systems is the demo model provided called the *LittleHexy copter* [6]. While this model serves as a useful starting point, it lacks the necessary customization and flexibility required by control algorithm researchers, and has never been used as a testbed for control systems. An actual X8-copter has been designed and realized and a high-fidelity CAD model of this vehicle has been produced, imported in PyChrono, and employed for the numerical simulations. These numerical simulations show how the MRAC system for hybrid, time-varying, nonlinear dynamical plants outperforms both the classical MRAC system and the classical PID controller in trajectory tracking error, control effort, and frequency of oscillations in the required motors' thrust. The results of the numerical simulations are shown in the YouTube video [7].

This paper is organized as follows. Section II details the mathematical notation and Section III recalls key elements of hybrid systems theory. Section IV presents an MRAC architecture for hybrid, time-varying, nonlinear dynamical systems. Section V recalls the equations of motion of a multi-rotor UAV equipped with an unsteady payload and able to drop or pickup some payload, and Section VI discusses the proposed control architecture. Finally, Section VII presents the results of numerical simulations in both nominal and off-nominal scenarios, and Section VIII draws conclusions for this work and outlines future work directions.

II. Notation

Let $\mathbb{N}$ denote the *set of positive integers*, $\overline{\mathbb{N}}$ denote the *set of nonnegative integers*, $\mathbb{R}$ the *set of real numbers*, $\mathbb{R}^n$ the *set of $n \times 1$ real column vectors*, and $\mathbb{R}^{n\times m}$ the *set of $n \times m$ real matrices*. The *boundary of* $\mathcal{D} \subset \mathbb{R}^n$ is denoted by $\partial\mathcal{D}$, and the *closure of* $\mathcal{D}$ is denoted by $\overline{\mathcal{D}}$.

The *open ball of radius* $\rho > 0$ *centered at* $x \in \mathbb{R}^n$ is denoted by $\mathcal{B}_\rho(x)$. Given $\rho > 0$ and the bounded set $\mathcal{A} \subset \mathbb{R}^n$, let $\mathcal{B}_\rho(\mathcal{A}) \triangleq \cup_{x\in\mathcal{A}}\mathcal{B}_\rho(x)$ denote the union of all open balls of radius ρ centered at the points of $\mathcal{A}$.

The Lebesgue measure of the set $\mathcal{D}$ is denoted by $\mu(\mathcal{D})$. A property $\mathfrak{P}$ is verified *almost everywhere* with respect to the Lebesgue measure $\mu(\cdot)$ on a set $\mathcal{X} \subseteq \mathbb{R}^n$ if there exists $\mathcal{N} \subset \mathcal{X}$ such that $\mu(\mathcal{N}) = 0$ and $\mathfrak{P}$ is verified by all $x \in \mathcal{X} \setminus \mathcal{N}$. In this case, we write "$\mathfrak{P}$ is verified for $x \in \mathcal{X}$ a.e.." The *indicator function* of the set $\mathcal{A} \subset \mathbb{R}^n$ is denoted by $\chi_\mathcal{A} : \mathcal{A} \to \{0, 1\}$ and is defined so that if $x \in \mathcal{A}$, then $\chi_\mathcal{A}(x) = 1$, and if $x \notin \mathcal{A}$, then $\chi_\mathcal{A}(x) = 0$. Integrals are always meant in the sense of Lebesgue. The *transpose of* $B \in \mathbb{R}^{n\times m}$ is denoted by B^{T}.

The *zero vector* in $\mathbb{R}^n$ is denoted by 0_n, the *zero $n \times m$ matrix* in $\mathbb{R}^{n\times m}$ is denoted by $0_{n\times m}$, and the *identity matrix* in $\mathbb{R}^{n\times n}$ is denoted by $\mathbf{1}_n$. We write $\|\cdot\|$ for the *Euclidean vector norm* and the corresponding *equi-induced matrix norm* [8, Def. 9.4.1]. The distance between $x \in \mathbb{R}^n$ and the set $\mathcal{A} \subset \mathbb{R}^n$ is defined as $\|x\|_\mathcal{A} \triangleq \inf_{a\in\mathcal{A}} \|x - a\|$ [9, Def. 3.5].

III. Fundamentals of Hybrid Systems Theory

In this paper, we consider nonlinear, time-varying, hybrid dynamical systems, that is, nonlinear time-varying dynamical system that experience instantaneous changes both in their trajectory and in their dynamics whenever resetting events occur. Such systems are captured by a set of differential and difference equations in the form

$$\dot{x}(t) = f_\mathrm{c}(t, x(t)), \qquad (t, x(t)) \notin \mathcal{D}, \tag{1}$$

$$x(t^+) = g_\mathrm{d}(t, x(t)), \qquad (t, x(t)) \in \mathcal{D}, \tag{2}$$

with $x(t_0) = x_0$. The *initial time* is denoted by $t_0 \in [0, \infty)$. The open set, where the solutions $t \mapsto x(t)$ of (1) and (2) are defined, is denoted by $\mathcal{S} \subseteq \mathbb{R}^n$, and we assume that $0_n \in \mathcal{S}$. The *vector field* $f_\mathrm{c} : [t_0, \infty) \times \mathcal{S} \to \mathbb{R}^n$ is Lebesgue integrable, locally bounded, and such that $f_\mathrm{c}(t, 0_n) = 0_n$ for all $t \in [t_0, \infty)$. The *switching law*, also known as the *jump map*, $g_\mathrm{d} : [t_0, \infty) \times \mathcal{S} \to \mathbb{R}^n$ is continuous in its arguments and locally bounded. The *set of resetting events* is denoted by $\mathcal{D} \subset [t_0, \infty) \times (\mathcal{S} \setminus \{0\})$.

The *flow* of solutions of (1) and (2) is denoted by $s_\mathrm{c} : [t_0, \infty) \times [t_0, \infty) \times \mathcal{S} \to \mathcal{S}$. Furthermore, the *resetting time before* $t \geq t_0$ is defined iteratively as $t_1 \triangleq \min\{t \geq t_0 : (t, s_\mathrm{c}(t, t_0, x_0)) \notin \mathcal{D}\}$ and $t_k \triangleq \min\{t > t_{k-1} : (t, s_\mathrm{c}(t, t_{k-1}, x_{k-1})) \notin \mathcal{D}\}$ for all $k \in \mathbb{N} \setminus \{1\}$. We assume that the system (1) and (2) is *left-continuous*, that is, the following three conditions are verified:

$$\lim_{\tau\to t^-} x(\tau) = x(t), \qquad t \in (t_0, \infty); \tag{3}$$

if $(t_0, x_0) \notin \mathcal{D}$, then, for all $t \in [t_0, \infty) \setminus \bigcup_{k\in\mathbb{N}}\{t_k\}$, it holds that

$$\lim_{\tau\to t^+} x(\tau) = x(t), \tag{4}$$

and, for all $k \in \mathbb{N}$, it holds that

$$x(t_k^+) = \lim_{\tau\to t_k^+} x(\tau) = g_\mathrm{d}(t_k, x_k); \tag{5}$$

and if $(t_0, x_0) \in \mathcal{D}$, then (4) is verified for all $t \in (t_0, \infty) \setminus \bigcup_{k\in\mathbb{N}}\{t_k\}$ and (5) is verified for all $k \in \overline{\mathbb{N}}$. In the remainder of this paper, for brevity, we assume that $(t_0, x_0) \notin \mathcal{D}$; the case whereby $(t_0, x_0) \in \mathcal{D}$ is not addressed explicitly and can be deduced from the arguments provided.

Next, we introduce the notion of *Krasovskii solution of* (1) and (2), which is derived from [10]. For the statement of this definition, let $\mathcal{I} \subseteq [t_0, \infty)$ be connected and such that $t_0 \in \mathcal{I}$ and $\mathcal{T} \triangleq (\mathcal{I} \times \mathcal{S}) \setminus \mathcal{D}$.

Definition III.1 *Assume that $x : \mathcal{I} \to \mathcal{S}$ is piecewise absolutely continuous, has a finite number of discontinuities on any compact subinterval of $\mathcal{I}$, and is such that if $(t, x(t)) \in \overline{\mathcal{T}}$, then*

$$\dot{x}(t) \in K[f_{\mathrm{c}}](t, x(t)), \tag{6}$$

and if $(t, x(t)) \in \overline{\mathcal{D}}$, then

$$x(t^+) \in K[g_{\mathrm{d}}](t, x(t)), \tag{7}$$

where

$$K[f_{\mathrm{c}}](t,x) \triangleq \bigcap_{\delta>0} \overline{\mathrm{co}} \left(f_{\mathrm{c}} \left(\overline{\mathcal{B}}_\delta \left(\begin{bmatrix} t \\ x \end{bmatrix} \right) \cap \mathcal{T} \right) \right), \ (t,x) \in \overline{\mathcal{T}}, \tag{8}$$

$$K[g_{\mathrm{d}}](t,x) \triangleq \bigcap_{\delta>0} \overline{g_{\mathrm{d}} \left(\overline{\mathcal{B}}_\delta \left(\begin{bmatrix} t \\ x \end{bmatrix} \right) \cap \mathcal{D} \right)}, \quad (t,x) \in \overline{\mathcal{D}}, \tag{9}$$

denote the Krasovskii regularizations of (1) *and* (2)*, respectively, and $\overline{\mathrm{co}}(\cdot)$ denotes the convex closure of its argument. Then, $x(\cdot)$ is a Krasovskii solution of* (1) *and* (2)*. If there do not exist a connected set $\mathcal{J} \subseteq [t_0, \infty)$ and a Krasovskii solution $\overline{x} : \mathcal{J} \to \mathcal{S}$ of* (1) *and* (2) *such that $\mathcal{I} \subset \mathcal{J}$ and $\overline{x}(t) = x(t)$, $t \in \mathcal{I}$, then $x : \mathcal{I} \to \mathcal{S}$ is a maximal Krasovskii solution of* (1) *and* (2)*. If $\mathcal{I} = [t_0, \infty)$, then a Krasovskii solution $x : \mathcal{I} \to \mathcal{S}$ of* (1) *and* (2) *is complete.*

We assume that the points of discontinuities of Krasovskii solutions of (1) and (2) occur at the resetting times only. The existence and uniqueness of Krasovskii solutions of hybrid dynamical systems is discussed in [10]. In this paper, solutions of hybrid systems are always meant in the sense of Krasovskii. In the reminder of this work, the following assumption is made on solutions of (1) and (2).

Assumption III.1 *If $(t, x(t)) \in \overline{\mathcal{D}} \setminus \mathcal{D}$, then there exists $\varepsilon > 0$ such that, for all $\delta \in (0, \varepsilon)$, $s_{\mathrm{c}}(t + \delta, t, x(t)) \notin \mathcal{D}$. Furthermore, if $(t_k, x(t_k)) \in \partial\mathcal{D} \cap \mathcal{D}$, then there exists $\varepsilon > 0$ such that, for all $\delta \in (0, \varepsilon)$, $s_{\mathrm{c}}(t_k + \delta, t_k, x(t_k^+)) \notin \mathcal{D}$.*

Assumption III.1 guarantees that if a trajectory of (1) and (2) reaches the closure of $\mathcal{D}$ at a point that does not belong to $\mathcal{D}$, then the trajectory must move away from $\mathcal{D}$. Furthermire, Assumption III.1 guarantees that if a trajectory reaches the boundary of $\mathcal{D}$ at a point that belongs to $\mathcal{D}$, then the trajectory moves away from any resetting event, and, hence, the continuous dynamics takes over for a non-trivial time interval. Thus, since piecewise absolutely continuous solutions of (1) and (2) are considered and Krasovskii solutions of (1) and (2) are discontinuous at resetting times only, Assumption III.1 implies that the solutions of (1) and (2) can not enter the interior of $\mathcal{D}$. This assumption is targeted at the scopes of this work, namely, the design model reference adaptive control laws for plants, whose trajectory and dynamics experience instantaneous variations, and whose continuous-time dynamics resumes immediately after a resetting event.

IV. A Model Reference Adaptive Control Law for Hybrid Plants

In this section, we recall the first MRAC system for nonlinear, time-varying, hybrid plants affected by matched and parametric uncertainties. This system was originally developed in [1]. This adaptive control system allows regulating uncertain, time-varying dynamical systems, whose trajectory tracking error dynamics are captured by

$$\begin{bmatrix} \dot{e}(t) \\ \dot{\sigma}(t) \end{bmatrix} = \begin{bmatrix} A_{\mathrm{ref},\sigma(t)} e(t) + B_{\sigma(t)} \Lambda_{\sigma(t)} \left[u(t) - \Theta_{\sigma(t)}^{\mathrm{T}} \Phi_{\sigma(t)}(t, x(t)) \right] \\ 0 \end{bmatrix}, \qquad \begin{bmatrix} e(t_0) \\ \sigma(t_0) \end{bmatrix} = \begin{bmatrix} e_0 \\ \sigma_0 \end{bmatrix},$$
$$\left((t, x(t)) \notin \mathcal{S}_{\sigma(t)} \right) \wedge \left((t, x_{\mathrm{ref}}(t)) \notin \mathcal{S}_{\mathrm{ref},\sigma(t)} \right), \tag{10}$$

$$\begin{bmatrix} e(t^+) \\ \sigma(t^+) \end{bmatrix} = g_{\mathrm{d},\sigma(t)}(t, e(t)), \qquad \left((t, x(t)) \in \mathcal{S}_{\sigma(t)} \right) \vee \left((t, x_{\mathrm{ref}}(t)) \in \mathcal{S}_{\mathrm{ref},\sigma(t)} \right), \tag{11}$$

where $e : [t_0, \infty) \to \mathbb{R}^n$ denotes the *trajectory tracking error*, the index $\sigma : [t_0, \infty) \to \Sigma$ tracks changes in the system's dynamics, $\Sigma \subset \mathbb{N}$ is bounded and, hence, without loss of generality, comprises the first $\sigma_{\max}$ positive integers, the

piecewise continuous function $u : [t_0, \infty) \to \mathbb{R}^m$ denotes the *control input*, $A_{\mathrm{ref},\sigma} \in \mathbb{R}^{n\times n}$ is user-defined and Hurwitz, $B_\sigma \in \mathbb{R}^{n\times m}$ is such that $(A_{\mathrm{ref},\sigma}, B_\sigma\Lambda_\sigma)$ is controllable, $\Lambda_\sigma \in \mathbb{R}^{m\times m}$ is diagonal, positive-definite, unknown, and the mapping $\sigma \mapsto \Lambda_\sigma$ is unknown, $\Theta_\sigma \in \mathbb{R}^{N_\sigma\times m}$ is unknown and the mapping $\sigma \mapsto \Theta_\sigma$ is unknown, the *regressor vector* $\Phi_\sigma : [t_0, \infty) \times \mathbb{R}^n \to \mathbb{R}^{N_\sigma}$ is Lipschitz continuous and captures matched uncertainties, $\wedge$ denotes the conjunction logic operator *and*, $\vee$ denotes the disjunction logic operator *or*, $x(t) \triangleq e(t)+x_{\mathrm{ref}}(t)$ denotes the *plant state*, $x_{\mathrm{ref}} : [t_0, \infty) \to \mathbb{R}^n$ denotes the *reference model's trajectory* and verifies the *reference model*

$$\begin{bmatrix}\dot{x}_{\mathrm{ref}}(t)\\ \dot{\sigma}(t)\end{bmatrix} = \begin{bmatrix}A_{\mathrm{ref},\sigma(t)}x_{\mathrm{ref}}(t) + B_{\mathrm{ref},\sigma(t)}r(t)\\ 0\end{bmatrix}, \quad \begin{bmatrix}x_{\mathrm{ref}}(t_0)\\ \sigma(t_0)\end{bmatrix} = \begin{bmatrix}x_{\mathrm{ref},0}\\ \sigma_0\end{bmatrix}, \quad (t, x_{\mathrm{ref}}(t)) \notin \mathcal{S}_{\mathrm{ref},\sigma(t)}, \tag{12}$$

$$\begin{bmatrix}x_{\mathrm{ref}}(t^+)\\ \sigma(t^+)\end{bmatrix} = g_{\mathrm{d,ref},\sigma(t)}(t, e(t)), \qquad (t, x_{\mathrm{ref}}(t)) \in \mathcal{S}_{\mathrm{ref},\sigma(t)}, \tag{13}$$

in the sense of Krasovskii, $B_{\mathrm{ref},\sigma} \in \mathbb{R}^{n\times m}$ is user-defined and such that

$$B_{\mathrm{ref},\sigma} = B_\sigma K_{r,\sigma}^{\mathrm{T}} \tag{14}$$

for some $K_{r,\sigma} \in \mathbb{R}^{m\times m}$, and the user-defined *reference command input* $r : [t_0, \infty) \to \mathbb{R}^m$ is bounded and piecewise continuous. The resetting events $\{\mathcal{S}_\sigma\}_{\sigma\in\Sigma}$ are unknown and are to be interpreted as the set of resetting events of the plant to control. Such resetting events are such that if $x_{\mathrm{ref}}(t) \equiv 0$, $t \geq t_0$, then Assumption III.1 is verified by (10) and (11) with any piecewise continuous control input $u(\cdot)$. The definitions of the resetting events $\left\{\mathcal{S}_{\mathrm{ref},\sigma}\right\}_{\sigma\in\Sigma}$, which are considered design variables, and of the associated sequence of resetting times are provided in the following. The switching law $g_{\mathrm{d},\sigma}(\cdot,\cdot)$, $\sigma \in \Sigma$, is assumed to be a known, uncontrollable property of the system's dynamics.

To present adaptive control laws that regulate (10) and (11), let $\Sigma_1, \ldots, \Sigma_p \subseteq \Sigma$, $p \in \{1, \ldots, \sigma_{\max}\}$, denote partitions of Σ, define $\overline{\Phi}_\sigma(t,x) \triangleq \left[\chi_{\Sigma_1}(\sigma)\Phi_1^{\mathrm{T}}(t,x), \ldots, \chi_{\Sigma_p}(\sigma)\Phi_p^{\mathrm{T}}(t,x)\right]^{\mathrm{T}}$, $(\sigma,t,x) \in \Sigma \times [t_0,\infty) \times \mathbb{R}^n$, and define $\Theta \triangleq \left[\Theta_1^{\mathrm{T}}, \ldots, \Theta_p^{\mathrm{T}}\right]^{\mathrm{T}}$ and $N \triangleq \sum_{\sigma=1}^p N_\sigma$. In this case, (10) is equivalent to

$$\begin{bmatrix}\dot{e}(t)\\ \dot{\sigma}(t)\end{bmatrix} = \begin{bmatrix}A_{\mathrm{ref},\sigma(t)}e(t) + B_{\sigma(t)}\left[u(t) - \Theta^{\mathrm{T}}\overline{\Phi}_{\sigma(t)}(t, x(t))\right]\\ 0\end{bmatrix},$$

$$\begin{bmatrix}e(t_0)\\ \sigma(t_0)\end{bmatrix} = \begin{bmatrix}e_0\\ \sigma_0\end{bmatrix}, \qquad \left((t, x(t)) \notin \mathcal{S}_{\sigma(t)}\right) \wedge \left((t, x_{\mathrm{ref}}(t)) \notin \mathcal{S}_{\mathrm{ref},\sigma(t)}\right). \tag{15}$$

Thus, we consider the *control law*

$$\eta(\hat{\Theta}, \overline{\Phi}_\sigma(t,x)) = \hat{\Theta}^{\mathrm{T}}\overline{\Phi}_\sigma(t,x), \qquad (\sigma,t,x,\hat{\Theta}) \in \Sigma \times [t_0,\infty) \times \mathbb{R}^n \times \mathbb{R}^{N\times m}, \tag{16}$$

and the *adaptive law*

$$\dot{\hat{\Theta}}(t) = \Gamma\overline{\Phi}_{\sigma(t)}(t, x(t))e^{\mathrm{T}}(t)P_{\sigma(t)}B_{\sigma(t)}, \qquad \hat{\Theta}(t_0) = \hat{\Theta}_0, \qquad t \geq t_0, \tag{17}$$

where $\Gamma \in \mathbb{R}^{N\times N}$ is user-defined, symmetric, and positive-definite, $P_\sigma \in \mathbb{R}^{n\times n}$, $\sigma \in \Sigma$, denotes the symmetric, positive-definite solution of the *algebraic Lyapunov equation*

$$0_{n\times n} = A_{\mathrm{ref},\sigma}^{\mathrm{T}}P_\sigma + P_\sigma A_{\mathrm{ref},\sigma} + Q_\sigma, \tag{18}$$

and $Q_\sigma \in \mathbb{R}^{n\times n}$ is user-defined, symmetric, and positive-definite. Remarkably, if B_σ, $\overline{\Phi}_\sigma(\cdot,\cdot)$, and P_σ were constant with $\sigma \in \Sigma$, then (16) and (17) reduces to the classical control law and adaptive law of MRAC, respectively [2, Ch. 9].

Next, let

$$W(e) = \overline{\lambda}_{\min}\left(\{Q_\sigma\}_{\sigma\in\Sigma}\right)\|e\|^2, \qquad e \in \mathbb{R}^n, \tag{19}$$

where $\overline{\lambda}_{\min}\left(\{Q_\sigma\}_{\sigma\in\Sigma}\right) \triangleq \min\{\lambda_{\min}(Q_\sigma), \sigma \in \Sigma\}$ and $\lambda_{\min}(Q_\sigma)$, $\sigma \in \Sigma$, denotes the smallest eigenvalue of Q_σ. We note that the right-hand side of (17) is piecewise Lipschitz continuous with points of discontinuity at the resetting times t_k, $k \in \mathbb{N}$. Thus, (17) is a switched dynamical system, and we consider Carathéodory solutions of (17); recall that this

class of solutions is absolutely continuous on $[t_0, \infty)$. Consequently, any instantaneous variation of $V(t, e(t), \hat{\Theta}(t))$, $t \geq t_0$, is due to variations of $e^{\mathrm{T}}(t)P_{\sigma(t)}e(t)$ across resetting events.

We let $\{t_k\}_{k\in\mathbb{N}} = \{t_{\mathrm{s},i}\}_{i\in\mathbb{N}} \cup \left(\bigcup_{i\in\mathbb{N}}\{t_{\mathrm{ref},i_w}\}_{w\in\mathbb{N}}\right)$, that is, we partition the set of resetting times of (10) and (11) into the union of the resetting times due to $\{\mathcal{S}_\sigma\}_{\sigma\in\Sigma}$ and the resetting times due to $\left\{\mathcal{S}_{\mathrm{ref},\sigma}\right\}_{\sigma\in\Sigma}$. The i-th resetting time of the resetting event $\mathcal{S}_{\sigma_{i-1}}$, $(i,\sigma) \in \mathbb{N}\times\Sigma$, is given by $t_{\mathrm{s},i} = \min\{t \geq t_{\mathrm{s},i-1} : (t, s_{\mathrm{c},\sigma_{i-1}}(t, t_{\mathrm{s},i-1}, x_{i-1})) \notin \mathcal{S}_{\sigma_{i-1}}\}$. The resetting events of the reference model are defined so that $\mathcal{S}_{\mathrm{ref},\sigma_{i_w}} \triangleq \{t_{\mathrm{ref},i_w}\}\times\mathbb{R}^n$, $(i,w) \in \mathbb{N}\times\mathbb{N}$, where

$$t_{\mathrm{ref},i_w} \triangleq \inf\left\{t > \max\{t_{\mathrm{s},i}, t_{\mathrm{ref},i_w-1}\} : \int_{t_0}^{t} W(e(\tau))\mathrm{d}\tau \geq \sum_{j=1}^{k-1}\left[e^{\mathrm{T}}(t_j^+)P_{\sigma(t_j^+)}e(t_j^+) - e^{\mathrm{T}}(t_j)P_{\sigma(t_j)}e(t_j)\right]\right\} \tag{20}$$

denotes the *w-th resetting time due to the reference model after the i-th resetting event of the plant*, k denotes the index for the generic resetting times, i denotes the index for the resetting times due to $\{\mathcal{S}_\sigma\}_{\sigma\in\Sigma}$, and i_w denotes the index for the resetting times due to $\left\{\mathcal{S}_{\mathrm{ref},\sigma}\right\}_{\sigma\in\Sigma}$ after the i-th resetting time. The switching law $g_{\mathrm{d,ref},\sigma}(t,e)$, $(\sigma,t,e) \in \Sigma\times[t_0,\infty)\times\mathbb{R}^n$, is defined so that

$$x_{\mathrm{ref}}(t_{\mathrm{ref},i_w}^+) = x(t_{\mathrm{ref},i_w}) - \sqrt{\frac{e^{\mathrm{T}}(t_{\mathrm{ref},i_w})P_{\sigma(t_{\mathrm{ref},i_w})}e(t_{\mathrm{ref},i_w}) - s_{\mathrm{ref},i_w}}{e^{\mathrm{T}}(t_{\mathrm{ref},i_w})P_{\sigma(t_{\mathrm{ref},i_w})}e(t_{\mathrm{ref},i_w})}}P^{-\frac{1}{2}}_{\sigma(t^+_{\mathrm{ref},i_w})}P^{\frac{1}{2}}_{\sigma(t_{\mathrm{ref},i_w})}e(t_{\mathrm{ref},i_w}), \quad (i,w) \in \mathbb{N}\times\mathbb{N}, \tag{21}$$

where $s_{\mathrm{ref},i_w} \in \left(0, e^{\mathrm{T}}(t_{\mathrm{ref},i_w})P_{\sigma(t_{\mathrm{ref},i_w})}e(t_{\mathrm{ref},i_w})\right)$ is user-defined, $\sum_{i=1}^{\infty}\sum_{w=1}^{\infty} s_{\mathrm{ref},i_w}$ is convergent, and $P_\sigma^{\frac{1}{2}} \in \mathbb{R}^{n\times n}$, $\sigma\in\Sigma$, is symmetric, positive-definite, and such that $P_\sigma = P_\sigma^{\frac{1}{2}}P_\sigma^{\frac{1}{2}}$. Details on how to construct s_{ref,i_w}, $(i,w)\in\mathbb{N}\times\mathbb{N}$, can be found in [1].

From (20), we deduce that if $\int_{t_0}^{t} W(e(\tau))\mathrm{d}\tau$, $t \geq t_0$, does not increase sufficiently fast over $\bigcup_{j\in\{1,\ldots,i\}}(t_{\mathrm{s},j}, t_{\mathrm{s},j+1}]$ for any $i\in\mathbb{N}$, then t_{ref,i_w}, $w\in\mathbb{N}$, can not be defined. However, the collection of positive-definite matrices $\{Q_\sigma\}_{\sigma\in\Sigma}$ is a user-defined parameter, and the smallest eigenvalues of these matrices can be set arbitrarily large for t_{ref,i_w}, $w\in\mathbb{N}$, to be defined. In light of this consideration, in the remainder of this paper, we consider the following assumption verified.

Assumption IV.1 *For each $i\in\mathbb{N}$, the set of reference model's resetting times $\{t_{\mathrm{ref},i_w}\}_{w\in\mathbb{N}}$ is always defined.*

The following theorem provides the main result of this section, namely a model reference adaptive control law for hybrid dynamical models in the same form as (15) and (11).

Theorem IV.1 *Consider the trajectory tracking error dynamics* (15) *and* (11)*, the control law* (16)*, the adaptive law* (17)*, and the reference model* (12) *and* (13)*. If $u(t) = \eta(\hat{\Theta}(t), \overline{\Phi}_\sigma(t, x(t)))$, $t \geq t_0$, the matching condition* (14) *is verified, and $\{Q_\sigma\}_{\sigma\in\Sigma}$ are chosen so that Assumption IV.1 is verified, then both the trajectory tracking error $e(\cdot)$ and the adaptive gain matrix $\hat{\Theta}(\cdot)$ are bounded uniformly in $\{t_k\}_{k\in\overline{\mathbb{N}}}$, and $e(t)\to 0$ as $t\to\infty$ for all $e_0\in\mathbb{R}^n$ uniformly in $\{t_k\}_{k\in\overline{\mathbb{N}}}$.*

Theorem IV.1 proves the effectiveness of the proposed MRAC system by showing that both the trajectory tracking error and the adaptive gains are bounded, and the trajectory tracking error asymptotically converges to zero, uniformly in the initial time and the sequence of resetting times. We remark how the adaptive gains are continuous functions of time. This feature eases the implementation of the proposed system to problems of practical interest.

Equation (20) requires the ability to compute the set of resetting times $\{t_k\}_{k\in\overline{\mathbb{N}}}$. However, in most problems of practical interest, it is difficult or impossible to detect the resetting times of the plant dynamics $\{t_{\mathrm{s},i}\}_{i\in\overline{\mathbb{N}}} \subseteq \{t_k\}_{k\in\overline{\mathbb{N}}}$. Furthermore, whereas the trajectory tracking error $e(\cdot)$ is assumed to be measurable at any time instant, in problems of practical interest, the reference trajectory $x_{\mathrm{ref}}(\cdot)$ is computed numerically as a solution of (12) and (13). Thus, in this paper, we employ the following algorithm to compute $\bigcup_{i\in\mathbb{N}}\{t_{\mathrm{ref},i_w}\}_{w\in\mathbb{N}}$. Let $\left\{t_p\right\}_{p\in\overline{\mathbb{N}}} \supseteq \{t_k\}_{k\in\mathbb{N}}$ denote the sequence of points over which $x_{\mathrm{ref}}(\cdot)$ is computed numerically, and let $t_p - t_{p-1}$ be sufficiently small for all $p\in\mathbb{N}$. Given $p\in\mathbb{N}$, let $\mathfrak{P}_{\mathrm{increasing}} \subset \mathbb{N}$ be such that if $e^{\mathrm{T}}(t_p)P_{\sigma(t_p)}e(t_p) > e^{\mathrm{T}}(t_{p-1})P_{\sigma(t_{p-1})}e(t_{p-1})$, then $p\in\mathfrak{P}_{\mathrm{increasing}}$. Thus, we compute $\int_{t_0}^{t_p} W(e(\tau))\mathrm{d}\tau$ by means of a quadrature method over $\{t_p\}_{p\notin\mathfrak{P}_{\mathrm{increasing}}}$, and we approximate $\sum_{j=1}^{k-1}\left[e^{\mathrm{T}}(t_j^+)P_{\sigma(t_j^+)}e(t_j^+) - e^{\mathrm{T}}(t_j)P_{\sigma(t_j)}e(t_j)\right]$ by $\sum_{p\in\mathfrak{P}_{\mathrm{increasing}}}\left[e^{\mathrm{T}}(t_p)P_{\sigma(t_p)}e(t_p) - e^{\mathrm{T}}(t_{p-1})P_{\sigma(t_{p-1})}e(t_{p-1})\right]$. For the rationale of this approximation, see the proof of Theorem 4 in [1].

V. Equations of Motion of an X8-Copter with Unsteady Payload

In this section, we present the equations of motion of an X8-copter with an unsteady, unknown payload. Recalling these equations is essential to justify the control system architecture presented in Section VI below and assess its underlying assumptions. These results are drawn from [11].

Consider the orthonormal, inertial reference frame $\mathbb{I} \triangleq \{O; X, Y, Z\}$, centered in $O \in \mathbb{R}^3$ and with axes $X, Y, Z \in \mathbb{R}^3$. Consider also the orthonormal reference frame $\mathbb{J}(\cdot) \triangleq \{A(\cdot); x(\cdot), y(\cdot), z(\cdot)\}$ fixed with the vehicle's frame, centered at a point $A : [t_0, \infty) \to \mathbb{R}^3$ conveniently chosen, and with axes $x, y, z : [t_0, \infty) \to \mathbb{R}^3$; in this paper, we refer to $\mathbb{J}(\cdot)$ as the *body reference frame*. If a vector $a \in \mathbb{R}^3$ is expressed in the reference frame $\mathbb{I}$, then it is denoted by $a^{\mathbb{I}}$; if a vector is expressed in $\mathbb{J}(\cdot)$, then no superscript is used.

The *UAV's mass*, including its payload, is denoted by $m_\sigma : [t_0, \infty) \to \mathbb{R}$, $\sigma \in \Sigma$. The resetting events tracked by the index $\sigma \in \Sigma$ include instantaneous changes in the UAV's mass due to sudden payload dropping or pickup events. Additional resetting events include instantaneous changes in the vehicle's translational and angular velocities due to impacts of the payload with its casing. The *position* of the reference point $A(\cdot)$ with respect to O is denoted by $r_A^{\mathbb{I}} : [t_0, \infty) \to \mathbb{R}^3$, and the *velocity* of $A(\cdot)$ with respect to the reference frame $\mathbb{I}$ is denoted by $v_A^{\mathbb{I}} : [t_0, \infty) \to \mathbb{R}^3$. The orientation of $\mathbb{J}(\cdot)$ with respect to $\mathbb{I}$ is captured by a 3-2-1 sequence of implicit Tait-Bryan angles [12, Ch.1] such that $\phi : [t_0, \infty) \to [0, 2\pi)$ denotes the *roll angle*, $\theta : [t_0, \infty) \to \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ denotes the *pitch angle*, and $\psi : [t_0, \infty) \to [0, 2\pi)$ denotes the *yaw angle*.

The UAV's translational kinematics are captured by

$$\dot{r}_A^{\mathbb{I}}(t) = R(\phi(t), \theta(t), \psi(t)) v_A(t), \qquad r_A^{\mathbb{I}}(t_0) = r_{A,0}^{\mathbb{I}}, \qquad t \geq t_0, \tag{22}$$

where

$$R(\phi, \theta, \psi) \triangleq \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix},$$
$$(\phi, \theta, \psi) \in [0, 2\pi) \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \times [0, 2\pi), \tag{23}$$

denotes the rotation matrix associated to a 3-2-1 rotation sequence of implicit Tait-Bryan angles. Between resetting events, the vehicle's translational dynamics are captured by

$$F_{\sigma(t)}(t) + F_{\mathrm{g},\sigma(t)}(t, \phi(t), \theta(t)) + T_{\mathrm{exhaust},\sigma(t)}(t) + T_{\mathrm{thrust}}(t)$$
$$= m_{\sigma(t)}(t) \left[\dot{v}_A(t) + \omega^{\times}(t) v_A(t) + \ddot{r}_C(t) + \dot{\omega}^{\times}(t) r_C(t) + 2\omega^{\times}(t)\dot{r}_C(t) + \omega^{\times}(t)\omega^{\times}(t) r_C(t)\right],$$
$$v_A(t_0) = v_{A,0}, \qquad t \geq t_0, \tag{24}$$

where $F_\sigma : [t_0, \infty) \to \mathbb{R}^3$, $\sigma \in \Sigma$, denotes the sum of the *external forces*, such as aerodynamic forces, acting on the vehicle,

$$F_{g,\sigma}(t, \phi, \theta) \triangleq m_\sigma(t) g \begin{bmatrix} -\sin\theta \\ \cos\theta \sin\phi \\ \cos\theta \cos\phi \end{bmatrix}, \qquad (t, \phi, \theta) \in [t_0, \infty) \times [0, 2\pi) \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right), \tag{25}$$

denotes the *weight of the UAV*, including its payload, $g > 0$ denotes the *gravitational acceleration*,

$$T_{\mathrm{exhaust},\sigma}(t) \triangleq \int_{\mathcal{V}_\sigma} v_{\mathrm{exh}}(t) \delta\dot{m}_{\sigma(t)}(t) \tag{26}$$

denotes the force generated according to Newton's third law for deploying some mass with exhaust velocity $v_{\mathrm{exh}} : [t_0, \infty) \to \mathbb{R}^3$ from the control volume $\mathcal{V}_\sigma$ that encloses the UAV and its payload,

$$T_{\mathrm{thrust}}(t) \triangleq -[0, 0, u_1(t)]^{\mathrm{T}}, \tag{27}$$

$u_1 : [t_0, \infty) \to [u_{\min}, u_{\max}] \subset \mathbb{R}$ denotes the the *thrust force* produced by the propellers, $u_{\min} > 0$ captures the fact that commercial-off-the-shelf motors for UAVs produce neither negative nor arbitrarily small thrust forces, $u_{\max} > u_{\min}$ captures the saturation in the thrust produced by the motors, and $r_C : [t_0, \infty) \to \mathbb{R}^3$ denotes the position of the center of mass $C(\cdot)$ of the UAV, including its payload, relative to the reference point $A(\cdot)$, where the reference frame $\mathbb{J}(\cdot)$ is

centered. Additional information about the user-defined bounds $u_{\min}$ and $u_{\max}$ are postponed to Section VI.B for clarity of exposition.

The UAV's rotational kinematics are captured by

$$\begin{bmatrix}\dot{\phi}(t)\\ \dot{\theta}(t)\\ \dot{\psi}(t)\end{bmatrix} = \Gamma_J(\phi(t),\theta(t))\omega(t), \qquad \begin{bmatrix}\phi(t_0)\\ \theta(t_0)\\ \psi(t_0)\end{bmatrix} = \begin{bmatrix}\phi_0\\ \theta_0\\ \psi_0\end{bmatrix}, \qquad t \geq t_0, \tag{28}$$

where

$$\Gamma_J(\phi,\theta) \triangleq \begin{bmatrix}1 & \sin\phi\tan\theta & \cos\phi\tan\theta\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta\end{bmatrix} \tag{29}$$

denotes the inverse of the *Jacobian matrix*. Between resetting events, the UAV's rotational dynamics are given by

$$\begin{aligned} M_{\sigma(t)}(t) + M_{\text{exhaust},\sigma(t)}(t) + M_{\text{thrust}}(t) = {} & m_{\sigma(t)}(t) r_C^{\times}(t)[\dot{v}_A(t) + \omega^{\times}(t) v_A(t)] + I_{\sigma(t)}(t)\dot{\omega}(t) + \omega^{\times}(t) I_{\sigma(t)}(t)\omega(t) \\ & + \dot{h}_{\sigma(t)}(t) + \omega^{\times}(t) h_{\sigma(t)}(t) + J_{\sigma(t)}(t)\omega(t) + \dot{I}_{\sigma(t)}(t)\omega(t), \\ & \omega(t_0) = \omega_0, \qquad t \geq t_0, \end{aligned} \tag{30}$$

where $M_\sigma : [t_0,\infty) \to \mathbb{R}^3$, $\sigma \in \Sigma$, denotes the *moment of the external forces*,

$$M_{\text{exhaust},\sigma}(t) \triangleq \int_{\mathcal{V}_\sigma} r_{mA}^{\times}(t) v_{\text{exh}}(t) \delta\dot{m}_{\sigma(t)}(t), \tag{31}$$

denotes the moment of $T_{\text{exhaust},\sigma}(\cdot)$, $M_{\text{thrust}} : [t_0,\infty) \to \mathbb{R}^3$ denotes the *moment of the thrust force* produced by the propellers by means of differential thrust and differential drag,

$$I_\sigma(t) \triangleq -\int_{\mathcal{V}_\sigma} r_{mA}^{\times}(t) r_{mA}^{\times}(t) \delta m_{\sigma(t)}(t) \tag{32}$$

denotes the *inertia matrix of the UAV* and its payload with respect to the reference point $A(\cdot)$,

$$J_\sigma(t) \triangleq \int_{\mathcal{V}_\sigma} r_{mA}^{\times}(t) r_{mA}^{\times}(t) \delta\dot{m}_{\sigma(t)}(t) \tag{33}$$

captures the variation in the moment of inertia due to the variation of the vehicle's unsteady mass over time, and

$$h_\sigma(t) \triangleq \int_{\mathcal{V}_\sigma} r_{mA}^{\times}(t) \dot{r}_{mA}(t) \delta m_{\sigma(t)}(t) \tag{34}$$

captures the contribution of moving parts, such as propellers and shifting payload, to the UAV's *angular momentum*. Consistently with the literature on UAV control, in the following, we denote $M_{\text{thrust}}(t) = [u_2(t), u_3(t), u_4(t)]^{\mathrm{T}}$.

VI. Control Architecture for X8-Copters

In this section, we present a classical control architecture for coaxial multi-rotor UAVs such as X8-copters. This classical architecture assumes that the UAV's reference point $A(\cdot)$ coincides with its center of mass $C(\cdot)$, and the UAV is a rigid body. The effects of moving components on the UAV dynamics, which have been detailed in Section V, will need to be compensated for by the control algorithms implemented in this proposed architecture. These effects, however, will be accounted for by the simulator employed in Section VII below.

We assume that the user provides a piecewise twice continuously differentiable *reference trajectory* for the UAV's reference point, which is denoted by $r_{\text{user}} : [t_0,\infty) \to \mathbb{R}^3$, and a piecewise twice continuously differentiable *reference yaw angle* $\psi_{\text{user}} : [t_0,\infty) \to [0, 2\pi)$. The control architecture presented here involves both an outer loop and an inner loop. The outer loop determines the thrust force that the propellers should generate $T_{\text{thrust,ideal}}(\cdot)$ and, hence, $u_1(\cdot)$. The outer loop also computes both the *reference pitch angle* $\phi_{\text{d}} : [t_0,\infty) \to [0, 2\pi)$ and the *reference roll angle* $\theta_{\text{d}} : [t_0,\infty) \to [0, 2\pi)$ so that the direction of the actual thrust produced by the propellers $T_{\text{thrust}}(\cdot)$ is in the same direction as $T_{\text{thrust,ideal}}(\cdot)$. The inner loop computes the moment of the propeller's thrust force so that the X8 attitude tracks the reference attitude captured by $\phi_{\text{d}}(\cdot)$, $\theta_{\text{d}}(\cdot)$, and $\psi_{\text{user}}(\cdot)$. The set of resetting events in the UAV dynamics $\bigcup_{\sigma\in\Sigma} \mathcal{S}_\sigma$, which is not characterized explicitly, comprises events such as payload dropping and pickup, the collisions between payloads, and the collisions between the payload its casing.

A. Outer Loop Design: Determination of the Total Thrust Force

In this section, we discuss how to compute the propellers' thrust force $T_{\text{thrust,ideal}} : [t_0, \infty) \to \mathbb{R}^3$ that should be produced by the UAV so that if $T_{\text{trust}}(t) = T_{\text{trust,ideal}}(t)$, $t \geq t_0$, then the reference point $A(\cdot)$ follows the user-defined trajectory $r_{\text{user}}(t)$. To this goal, we assume that $A(t) \equiv C(t)$, $t \geq t_0$, and, hence, $r_C(t) \equiv 0$. Under these assumptions, (22) and (24) reduce to

$$m_{\sigma(t)} \ddot{r}^{\mathbb{I}}_A(t) = F^{\mathbb{I}}_{\sigma(t)}(t) + F^{\mathbb{I}}_{\text{g},\sigma(\text{t})}(t) + T^{\mathbb{I}}_{\text{thrust}}(t), \qquad r^{\mathbb{I}}_A(t_0) = r^{\mathbb{I}}_{A,0} \qquad \dot{r}^{\mathbb{I}}_A(t_0) = v^{\mathbb{I}}_{A,0}, \qquad t \geq t_0, \tag{35}$$

where

$$F_\sigma(t) = -\frac{1}{2}\rho S_\sigma c_{\text{D},\sigma} \|v_A(t)\| v_A(t), \qquad (\sigma, t) \in \Sigma \times [t_0, \infty), \tag{36}$$

$\rho > 0$ denotes the *air density*, which is assumed unknown, $S_\sigma > 0$ denotes the UAV's *cross-section area*, which is assumed unknown, and $c_{\text{D},\sigma} \in \mathbb{R}^{3\times 3}$ is symmetric, positive-definite, and captures the *matrix of aerodynamic coefficients*.

Replacing $T_{\text{trust}}(\cdot)$ with $T_{\text{trust,ideal}}(\cdot)$, the continuous-time dynamics of the equations of motion are given by

$$\dot{x}_{\text{tran}}(t) = A x_{\text{tran}}(t) + B_{\text{tran},\sigma} \Lambda_{\sigma(t)} \left[T^{\mathbb{I}}_{\text{trust,ideal}}(t) + \Theta^{\text{T}}_{\text{tran},\sigma(\text{t})} \Phi_{\text{tran},\sigma(\text{t})}(t, x_{\text{tran}}(t)) \right], \qquad x_{\text{tran}}(t_0) = \begin{bmatrix} r^{\mathbb{I}}_{A,0} \\ v^{\mathbb{I}}_{A,0} \end{bmatrix}, \qquad t \geq t_0, \tag{37}$$

where $x_{\text{tran}}(t) \triangleq \left[r^{\mathbb{I},\text{T}}_A(t), v^{\mathbb{I},\text{T}}_A(t) \right]^{\text{T}}$, $A = \begin{bmatrix} 0_{3\times 3} & \mathbf{1}_3 \\ 0_{3\times 3} & 0_{3\times 3} \end{bmatrix}$, $B_{\text{tran},\sigma} = \begin{bmatrix} 0_{3\times 3} \\ \mathbf{1}_3 \end{bmatrix}$, $\Lambda_\sigma = m_\sigma^{-1} \mathbf{1}_3$,

$$\Theta_{\text{tran},\sigma} \triangleq \left[-\frac{1}{2}\rho S_\sigma c_{\text{D},\sigma}, m_\sigma g \mathbf{1}_3 \right]^{\text{T}}, \qquad \sigma \in \Sigma, \tag{38}$$

$$\Phi_{\text{tran},\sigma}(t, x_{\text{tran}}) \triangleq \begin{bmatrix} \|v_A\| R(\phi(t), \theta(t), \psi(t)) v_A \\ e_3 \end{bmatrix}, \qquad (t, x_{\text{tran}}) \in [t_0, \infty) \times \mathbb{R}^6, \tag{39}$$

and $e_3 \triangleq [0, 0, 1]^{\text{T}}$.

To apply the MRAC framework, we consider the *reference trajectory* $x_{\text{tran,ref}}(t) = \left[r^{\mathbb{I},\text{T}}_{\text{ref}}(t), v^{\mathbb{I},\text{T}}_{\text{ref}}(t) \right]^{\text{T}}$, $t \geq t_0$, whose continuous-time dynamics are given by

$$\begin{aligned} \dot{x}_{\text{tran,ref}}(t) = A x_{\text{tran,ref}}(t) + B_{\text{ref,tran},\sigma(\text{t})} \Big[& \overline{m}_{\sigma(t)} \ddot{r}^{\mathbb{I}}_{\text{user}}(t) - \overline{m}_{\sigma(t)} K_{\text{tran},P,\sigma(t)} \left(r^{\mathbb{I}}_{\text{ref}}(t) - r^{\mathbb{I}}_{\text{user}}(t) \right) \\ & - \overline{m}_{\sigma(t)} K_{\text{tran},D,\sigma(t)} \left(v^{\mathbb{I}}_{\text{ref}}(t) - \dot{r}^{\mathbb{I}}_{\text{user}}(t) \right) - \overline{m}_{\sigma(t)} K_{\text{tran},I,\sigma(t)} \int_{t_0}^{t} \left(r^{\mathbb{I}}_{\text{ref}}(\tau) - r^{\mathbb{I}}_{\text{user}}(\tau) \right) \text{d}\tau \Big], \\ & x_{\text{tran,ref}}(t_0) = x_{\text{tran,ref},0}, \end{aligned} \tag{40}$$

where $B_{\text{ref,tran},\sigma} = \left[0_{3\times 3}, \overline{m}_\sigma^{-1} \mathbf{1}_3 \right]^{\text{T}}$, $\sigma \in \Sigma$, $\overline{m}_\sigma > 0$ is user-defined and denotes the *estimated mass*, and $K_{\text{tran},P,\sigma}$, $K_{\text{tran},I,\sigma}$, and $K_{\text{tran},D,\sigma} \in \mathbb{R}^{3\times 3}$ are user-defined, symmetric, and positive-definite gain matrices that capture a PID control system to steer $x_{\text{tran,ref}}(\cdot)$ toward the vector $\left[r^{\mathbb{I},\text{T}}_{\text{user}}(\cdot), \dot{r}^{\mathbb{I},\text{T}}_{\text{user}}(\cdot) \right]^{\text{T}}$. The reference model continuous-time dynamics (40) can be equivalently expressed as

$$\dot{x}_{\text{tran,ref}}(t) = A_{\text{ref,tran},\sigma(\text{t})} x_{\text{tran,ref}}(t) + B_{\text{ref,tran},\sigma(\text{t})} r_{\text{tran}}(t), \qquad x_{\text{tran,ref}}(t_0) = x_{\text{tran,ref},0}, \qquad t \geq t_0, \tag{41}$$

where $A_{\text{ref,tran},\sigma} \triangleq \begin{bmatrix} 0_{3\times 3} & \mathbf{1}_3 \\ -K_{\text{tran},P,\sigma} & -K_{\text{tran},D,\sigma} \end{bmatrix}$, $\sigma \in \Sigma$. It is apparent how (41) could be further simplified by removing $\overline{m}_\sigma$ from both the definition of $B_{\text{ref,tran},\sigma}$ and reference command input for the translational dynamics is given by

$$\begin{aligned} r_{\text{tran}}(t) \triangleq & \overline{m}_{\sigma(t)} \ddot{r}^{\mathbb{I}}_{\text{user}}(t) + \overline{m}_{\sigma(t)} K_{\text{tran},P,\sigma(t)} r^{\mathbb{I}}_{\text{user}}(t) + \overline{m}_{\sigma(t)} K_{\text{tran},D,\sigma(t)} \dot{r}^{\mathbb{I}}_{\text{user}}(t) \\ & - \overline{m}_{\sigma(t)} K_{\text{tran},I,\sigma(t)} \int_{t_0}^{t} \left(r^{\mathbb{I}}_{\text{ref}}(\tau) - r^{\mathbb{I}}_{\text{user}}(\tau) \right) \text{d}\tau, \qquad t \geq t_0. \end{aligned} \tag{42}$$

However, (41) may be preferable for its relative ease of interpretation since each term in the reference command input captures a force.

The continuous-time reference model (41) can be readily reduced to the same form as (12). Furthermore, defining the *translational trajectory tracking error* $e_{\text{tran}}(t) \triangleq x_{\text{tran}}(t) - x_{\text{tran,ref}}(t)$, $t \geq t_0$, the continuous-time translational trajectory tracking error dynamics can be reduced to the same form as (10); this passage is omitted for brevity. Thus, the control input $T^{\mathbb{I}}_{\text{thrust,ideal}}(\cdot)$ can be determined directly by applying the MRAC system for time-varying hybrid plants presented in Section IV.

MRAC systems are usually very aggressive, and, to reduce this effect, it is customary to employ a baseline controller. This way, the adaptive control input only needs to compensate for the lack of effectiveness of the baseline controller. In this paper, we let

$$T^{\mathbb{I}}_{\text{thrust,ideal}}(t) = T^{\mathbb{I}}_{\text{thrust,ideal,baseline}}(t) + T^{\mathbb{I}}_{\text{thrust,ideal,adaptive}}(t), \qquad t \geq t_0, \tag{43}$$

where

$$\begin{aligned} T^{\mathbb{I}}_{\text{thrust,ideal,baseline}}(t) = {} & \underbrace{-\overline{m}_{\sigma(t)} K_{\text{tran},P,\sigma(t)} \left(r^{\mathbb{I}}(t) - r^{\mathbb{I}}_{\text{ref}}(t) \right) - \overline{m}_{\sigma(t)} K_{\text{tran},D,\sigma(t)} \left(v^{\mathbb{I}}(t) - v^{\mathbb{I}}_{\text{ref}}(t) \right)}_{\text{PD baseline}} \\ & \underbrace{+\frac{1}{2} \overline{\rho} \overline{S}_{\sigma(t)} \overline{c}_{\text{D},\sigma(t)} \| v_A(t) \| R(\phi(t), \theta(t), \psi(t)) v_A(t)}_{\text{dynamic inversion baseline}} \\ & \underbrace{+\overline{K}_x^{\text{T}} x_{\text{tran}}(t) - \overline{\Theta}^{\text{T}}_{\text{tran},\sigma(t)} \Phi_{\text{tran},\sigma(t)}(t, x_{\text{tran}}(t)) + \overline{K}_r^{\text{T}} r_{\text{tran}}(t)}_{\text{MRAC-like baseline}} + \overline{m}_{\sigma(t)} \ddot{r}^{\mathbb{I}}_{\text{ref}}(t), \end{aligned} \tag{44}$$

$\overline{K}_x \in \mathbb{R}^{6\times 3}$, $\overline{K}_r \in \mathbb{R}^{3\times 3}$, and $\overline{\Theta}_{\text{tran},\sigma} \in \mathbb{R}^{6\times 3}$ are user-defined, $\overline{\rho} > 0$ is user-defined and captures the *estimated air density*, $\overline{S}_\sigma > 0$, $\sigma \in \Sigma$, is user-defined and captures the *estimated UAV's cross-section area*, $\overline{c}_{\text{D},\sigma} \in \mathbb{R}^{3\times 3}$ is user-defined, positive-definite, and captures the *estimated matrix of aerodynamic coefficients*, and $T^{\mathbb{I}}_{\text{thrust,ideal,adaptive}} : [t_0, \infty) \to \mathbb{R}^3$. Thus, embedding $T^{\mathbb{I}}_{\text{thrust,ideal,baseline}}(\cdot)$ in the regressor vector, $T^{\mathbb{I}}_{\text{thrust,ideal,adaptive}}(\cdot)$ can be computed according to the adaptive control scheme discussed in Section IV.

The baseline control input (44) comprises a proportional-derivative (PD) control input , a dynamic inversion control input, and a control input that is structured in the same form a the MRAC input, but employing constant, user-defined gains. It is apparent how several terms appear twice in (44). Although these repetitions are unnecessary, they also simplify the gain tuning process for the relative ease of interpretation of (44).

B. Outer Loop Design: Determination of the Reference Roll and Pitch Angles

Having determined $T_{\text{thrust,ideal}}(\cdot)$, we let its realization be given by

$$T^{\mathbb{I}}_{\text{thrust,ideal}}(t) = -u_1(t) R(\phi_{\text{d}}(t), \theta_{\text{d}}(t), \psi_{\text{user}}(t)) e_3, \qquad t \geq t_0, \tag{45}$$

where

$$u_1(t) \triangleq \text{sat}\left(\| T_{\text{thrust,ideal}}(t) \|, u_{\min}, u_{\max} \right), \tag{46}$$

and

$$\text{sat}(\alpha, \alpha_{\min}, \alpha_{\max}) \triangleq \min\left\{ \alpha_{\max}, \max\left\{ \alpha, \alpha_{\min} \right\} \right\}, \quad (\alpha, \alpha_{\min}, \alpha_{\max}) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}, \tag{47}$$

with $\alpha_{\min} < \alpha_{\max}$. Expanding (45), we note that

$$-T_{\text{thrust},Z}(t) \tan\theta_{\text{d}}(t) = -\left(T_{\text{thrust},X}(t) \cos\psi_{\text{user}}(t) + T_{\text{thrust},Y}(t) \sin\psi_{\text{user}}(t) \right), \qquad t \geq t_0, \tag{48}$$

$$u_1(t) \sin\phi_{\text{d}}(t) = -\left(T_{\text{thrust},X}(t) \sin\psi_{\text{user}}(t) - T_{\text{thrust},Y}(t) \cos\psi_{\text{user}}(t) \right), \tag{49}$$

where $T^{\mathbb{I}}_{\text{thrust,ideal}}(t) = \left[T_{\text{thrust},X}(t), T_{\text{thrust},Y}(t), T_{\text{thrust},Z}(t) \right]^{\text{T}}$. Thus, we let

$$\theta_{\text{d}}(t) = \text{atan2}\Bigg(- \text{sat}\left(T_{\text{thrust},X}(t), T_{\min,X}, T_{\max,X} \right) \cos\psi_{\text{user}}(t) - \text{sat}\left(T_{\text{thrust},Y}(t), T_{\min,Y}, T_{\max,Y} \right) \sin\psi_{\text{user}}(t),$$

$$- T_{\text{thrust},Z}(t)\Big), \qquad t \geq t_0, \tag{50}$$

$$\phi_{\text{d}}(t) = \text{atan2}\left(\mathfrak{T}(t), \sqrt{1-\mathfrak{T}^2(t)}\right) \tag{51}$$

where the *signed arc-tangent function* is defined as

$$\text{atan2}(y,x) \triangleq \begin{cases} \tan^{-1}\left(\frac{y}{x}\right) & \text{if } x > 0, \\ \tan^{-1}\left(\frac{y}{x}\right) + \pi & \text{if } x < 0 \text{ and } y \geq 0, \\ \tan^{-1}\left(\frac{y}{x}\right) - \pi & \text{if } x < 0 \text{ and } y < 0, \\ \frac{\pi}{2} & \text{if } x = 0 \text{ and } y > 0, \\ -\frac{\pi}{2} & \text{if } x = 0 \text{ and } y < 0, \\ \text{undefined} & \text{if } x = 0 \text{ and } y = 0, \end{cases} \qquad (x,y) \in \mathbb{R} \times \mathbb{R}, \tag{52}$$

$\tan^{-1}(\cdot)$ denotes the classical arc-tangent function,

$$\mathfrak{T}(t) \triangleq -\frac{\text{sat}\left(T_{\text{thrust},X}(t), T_{\text{min},X}, T_{\text{max},X}\right)\sin\psi_{\text{user}}(t) - \text{sat}\left(T_{\text{thrust},Y}(t), T_{\text{min},Y}, T_{\text{max},Y}\right)\cos\psi_{\text{user}}(t)}{u_1(t)}, \tag{53}$$

$\left[T_{\text{min},X}, T_{\text{min},Y}, T_{\text{min},Z}\right]^{\text{T}}, \left[T_{\text{max},X}, T_{\text{max},Y}, T_{\text{max},Z}\right]^{\text{T}} \in \mathbb{R}^3$ are user-defined and such that $\left[T_{\text{min},X}, T_{\text{min},Y}, T_{\text{min},Z}\right]^{\text{T}} \leq\leq T^{\mathbb{I}}_{\text{thrust,ideal}}(t) \leq\leq \left[T_{\text{max},X}, T_{\text{max},Y}, T_{\text{max},Z}\right]^{\text{T}}$, $\left[T_{\text{min},X}, T_{\text{min},Y}, T_{\text{min},Z}\right]^{\text{T}} << 0_3$, $\left[T_{\text{max},X}, T_{\text{max},Y}, T_{\text{max},Z}\right]^{\text{T}} >> 0_3$, the symbols $\leq\leq$, $<<$, $\geq\geq$, and $>>$ denote the component-wise inequalities, and

$$u_{\min} \triangleq \sqrt{T^2_{\text{min},X} + T^2_{\text{min},Y} + T^2_{\text{min},Z}}, \tag{54}$$

$$u_{\max} \triangleq \sqrt{T^2_{\text{max},X} + T^2_{\text{max},Y} + T^2_{\text{max},Z}}. \tag{55}$$

It is worthwhile remarking how both $\left[T_{\text{min},X}, T_{\text{min},Y}, T_{\text{min},Z}\right]^{\text{T}}$ and $\left[T_{\text{max},X}, T_{\text{max},Y}, T_{\text{max},Z}\right]^{\text{T}}$ can be tuned to design the ranges of both $\phi_{\text{d}}(\cdot)$ and $\theta_{\text{d}}(\cdot)$.

C. Inner Loop Design: Determination of the Moment of the Thrust Force

In this section, we discuss how to compute the moment of the propellers' thrust force $M_{\text{thrust}} : [t_0, \infty) \to \mathbb{R}^3$ that should be produced by the UAV so that $\lim_{t\to\infty} \|\eta(t) - \eta_{\text{d}}(t)\| = 0$ uniformly in $t_0 \geq 0$, where $\eta(t) \triangleq [\phi(t), \theta(t), \psi(t)]^{\text{T}}$, $t \geq t_0$, and $\eta_{\text{d}}(t) \triangleq [\phi_{\text{d}}(t), \theta_{\text{d}}(t), \psi_{\text{user}}(t)]^{\text{T}}$. For the statement of this result, firstly, we assume that $\dot{\phi}_{\text{d}}(t)$, $t \geq t_0$, $\dot{\theta}_{\text{d}}(t)$, $\ddot{\phi}_{\text{d}}(t)$, $\ddot{\theta}_{\text{d}}(t)$ can be computed in real time by a differentiator. Thus, we define *command angular velocity*

$$\omega_{\text{cmd}}(t) \triangleq \Gamma_J^{-1}(\phi(t), \theta(t))\left[\dot{\eta}_{\text{d}}(t) - K_{\text{rot},P}\left(\eta(t) - \eta_{\text{d}}(t)\right)\right], \qquad t \geq t_0, \tag{56}$$

where $K_{\text{rot},P} \in \mathbb{R}^{3\times 3}$ is symmetric, positive-definite, and user-defined. This way, if $\omega(t) = \omega_{\text{cmd}}(t)$, $t \geq t_0$, then $\lim_{t\to\infty} \|\eta(t) - \eta_{\text{d}}(t)\| = 0$ uniformly in $t_0 \geq 0$.

Next, we compute $M_{\text{thrust}}(\cdot)$ so that $\lim_{t\to\infty} \|\eta(t) - \eta_{\text{d}}(t)\| = 0$ and $\lim_{t\to\infty} \|\omega(t) - \omega_{\text{cmd}}(t)\| = 0$ uniformly in $t_0 \geq 0$. To this goal, we assume that $m_\sigma(\cdot)$ is constant for each $\sigma \in \Sigma$, $I_\sigma(\cdot)$ is constant and diagonal, $r_C(t) \triangleq 0_3$, $t \geq t_0$, $J_\sigma(t) \equiv 0_{3\times 3}$, and $h_\sigma(t) \equiv 0_3$. Finally, we assume that

$$M_\sigma(t) = \delta^{\times}_\sigma F_\sigma(t), \qquad (\sigma, t) \in \Sigma \times [t_0, \infty), \tag{57}$$

where $\delta_\sigma \in \mathbb{R}^3$ denotes the position of the aerodynamic center relative to the reference point $A(\cdot) \equiv C(\cdot)$. Under these assumptions, between resetting events, (30) reduces to

$$I_{\sigma(t)}\dot{\omega}(t) = M_{\sigma(t)}(t) + M_{\text{thrust}}(t) - \omega^{\times}(t) I_{\sigma(t)}(t)\omega(t), \qquad \omega(t_0) = \omega_0, \qquad t \geq t_0, \tag{58}$$

which can be equivalently expressed as

$$\dot{\omega}(t) = I^{-1}_{\sigma(t)}\left[M_{\text{thrust}}(t) + \Theta^{\text{T}}_{\text{rot},\sigma}\Phi_{\text{rot},\sigma}(t,\omega)\right], \qquad \omega(t_0) = \omega_0, \qquad t \geq t_0, \tag{59}$$

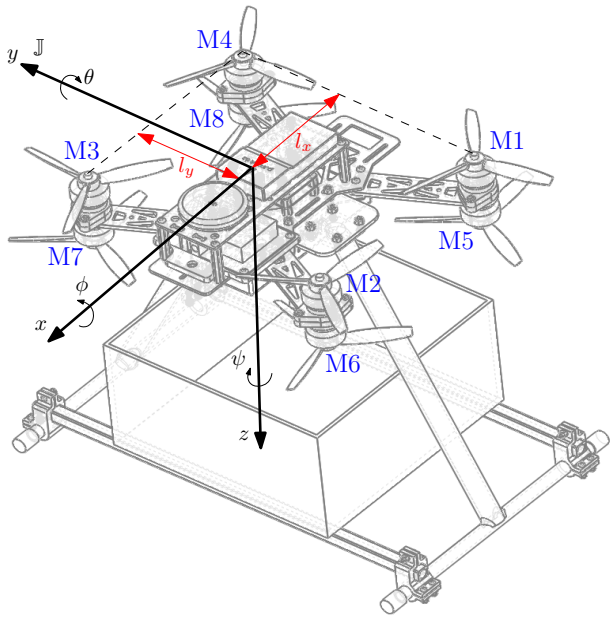

Fig. 1 CAD model of the X8-copter imported in PyChrono and employed for the numerical simulations. Motors are a distance l_x from the pitch axis and l_y from the roll axis. The motors are labeled as M1, M2, . . . , M8. The yaw-axis points downward. Motors M1, M3, M6, and M8 spin counter-clockwise, and Motors M2, M4, M5, and M7 spin clockwise

where, for each $\sigma \in \Sigma$,

$$\Theta_{\text{rot},\sigma} \triangleq \left[-\frac{1}{2}\rho S_\sigma \delta^\times c_{\text{D},\sigma}, \text{diag}\left(\left[I_{zz} - I_{yy}, I_{xx} - I_{zz}, I_{yy} - I_{xx}\right]^{\text{T}}\right)\right]^{\text{T}}, \tag{60}$$

$$\Phi_{\text{rot},\sigma}(t,\omega) \triangleq \begin{bmatrix} \|v_A(t)\| R(\phi(t),\theta(t),\psi(t)) v_A(t) \\ \omega_y\omega_z \\ \omega_x\omega_z \\ \omega_x\omega_y \end{bmatrix}, \qquad (t,\omega) \in [t_0,\infty) \times \mathbb{R}^3, \tag{61}$$

$\omega = \left[\omega_x, \omega_y, \omega_z\right]^{\text{T}}$, $I_\sigma = \text{diag}\left(\left[I_{xx,\sigma}, I_{yy,\sigma}, I_{zz,\sigma}\right]^{\text{T}}\right)$, and $\text{diag} : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ produces a diagonal matrix from its vector argument.

To apply the MRAC framework, we consider the *reference angular velocity* $\omega_{\text{ref}}(\cdot)$, whose continuous-time dynamics are given by

$$\dot{\omega}_{\text{ref}}(t) = -K_{P,\omega_{\text{ref}}}\omega_{\text{ref}}(t) + r_{\text{rot}}(t), \qquad x_{\text{ref}}(t_0) = x_{\text{ref},0}, \qquad t \geq t_0, \tag{62}$$

where $K_{P,\omega_{\text{ref}}} \in \mathbb{R}^{3\times3}$ is positive-definite and user-defined, and

$$r_{\text{rot}}(t) \triangleq K_{P,\omega_{\text{ref}}}\omega_{\text{cmd}}(t) + \dot{\omega}_{\text{cmd}}(t) \tag{63}$$

denotes the reference command input for the rotational dynamics. The continuous-time reference model (62) is in the same form as the continuous-time dynamics in (12) since $-K_{P,\omega_{\text{ref}}}$ is Hurwitz.

The *rotational trajectory tracking error* can be expressed as $e_{\text{rot}}(t) \triangleq \omega(t) - \omega_{\text{ref}}(t)$, $t \geq t_0$, whose continuous time-dynamics is omitted for brevity. Thus, the control input $M_{\text{thrust}}(\cdot)$ can be either computed directly applying the framework presented in Section IV or as

$$M_{\text{thrust}}(t) = M_{\text{thrust,baseline}}(t) + M_{\text{thrust,adaptive}}(t), \qquad t \geq t_0, \tag{64}$$

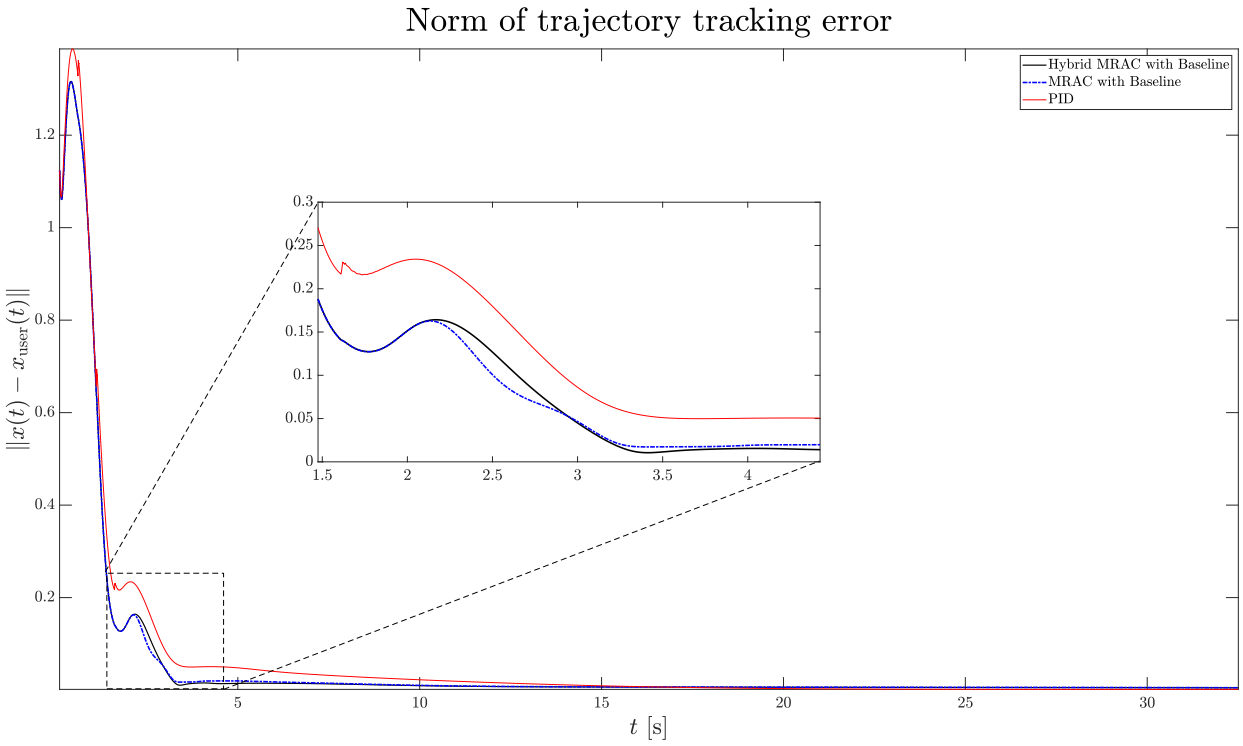

Fig. 2 Trajectory tracking errors obtained applying a PID controller, an MRAC controller, and a hybrid MRAC controller assuming that the UAV's inertial and aerodynamic properties are known. The three control systems have been tuned to attain similar performances in terms of maximum overshoots and settling times

where

$$M_{\text{thrust,baseline}}(t) = \underbrace{-\overline{I}_{\sigma(t)}\left[K_{\text{rot},P,\text{baseline}}\left(\omega(t) - \omega_{\text{ref}}(t)\right) + K_{\text{rot},I,\text{baseline}}\int_{t_0}^{t}\left(\omega(\tau) - \omega_{\text{ref}}(\tau)\right)\mathrm{d}\tau\right]}_{\text{PI baseline}}$$
$$\underbrace{+\omega^{\times}(t)\overline{I}_{\sigma(t)}\omega(t) + \frac{1}{2}\overline{\rho}\overline{S}_{\sigma(t)}\overline{\delta}^{\times}_{\sigma(t)}\overline{c}_{\text{D},\sigma(\text{t})}\|v_A(t)\|R(\phi(t),\theta(t),\psi(t))v_A(t) + \overline{I}_{\sigma(t)}\dot{\omega}_{\text{ref}}(t)}_{\text{dynamic inversion baseline}}, \quad (65)$$

$K_{\text{rot},P,\text{baseline}}, K_{\text{rot},I,\text{baseline}} \in \mathbb{R}^{3\times3}$ are user-defined, symmetric, positive-definite, and capture the gains of a baseline proportional-integral (PI) control input, $\overline{I}_{\sigma} \in \mathbb{R}^{3\times3}$, $\sigma \in \Sigma$, is user-defined, diagonal, positive-definite, and captures the estimated matrix of inertia, $\overline{\delta}_{\sigma} \in \mathbb{R}^3$ is user-defined and captures the *estimated position of the aerodynamic center* relative to $A(\cdot)$, and $M_{\text{thrust,adaptive}} : [t_0, \infty) \to \mathbb{R}^3$. Embedding the baseline controller into the regressor vector, $M_{\text{thrust,adaptive}}(\cdot)$ can be computed employing the MRAC framework for time-varying hybrid plants discussed in Section IV.

D. Thrust Allocation

Having determined the desired total thrust (46) and the desired moment of the trust (64), the thrust produced by each motors is computed as the solution of the optimization problem

$$\min_{T(t)\in\mathbb{R}^8} \|T(t)\|^2 \quad (66)$$
$$\text{s.t.} \begin{bmatrix} u_1(t) \\ M_{\text{thrust}}(t) \end{bmatrix} = \mathcal{M}T(t), \quad (67)$$

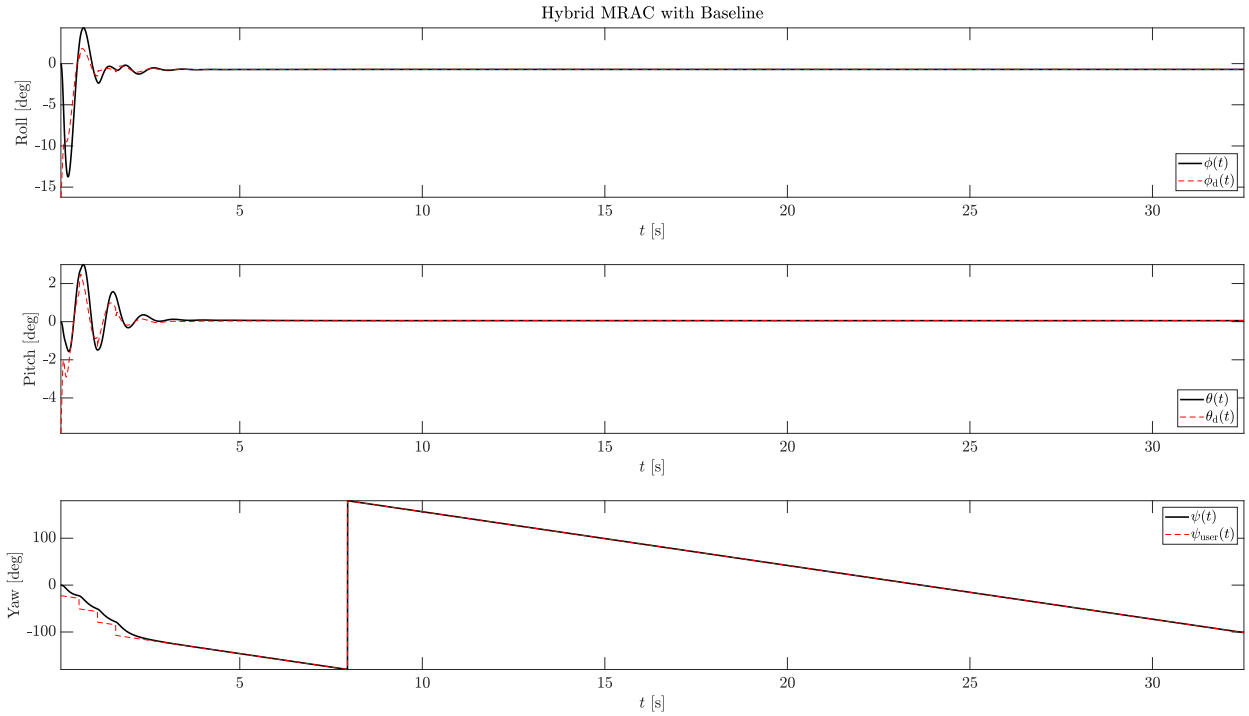

Fig. 3 Attitude and reference attitude of the UAV controller by the hybrid MRAC system. The UAV's attitude rapidly converges to the reference attitude. The reference roll and pitch angles are determined by the outer loop. The user-defined yaw angle has three sudden variations in the interval $[0, 1.64]$ **s**

for each $t \geq t_0$, where $T(t) \triangleq [T_1(t), T_2(t), \ldots, T_8(t)]^{\mathrm{T}}$, $T_i : [t_0, \infty) \to \mathbb{R}$ denotes the thrust force of the i-th motor, $i = 1, \ldots, 8$,

$$\mathcal{M} \triangleq \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ l_y & l_y & -l_y & -l_y & l_y & l_y & -l_y & -l_y \\ -l_x & l_x & l_x & -l_x & -l_x & l_x & l_x & -l_x \\ -c_T & c_T & -c_T & c_T & c_T & -c_T & c_T & -c_T \end{bmatrix}, \tag{68}$$

denotes the *mixer matrix* assuming that the motors are numbered as in Figure 1, Motors M1, M3, M6, and M8 spin counter-clockwise, and Motors M2, M4, M5, and M7 spin clockwise, $l_x, l_y > 0$ denote the distance of the motors from the pitch and roll axes, respectively, and $c_T > 0$ denotes the propellers' drag coefficient. The analytical solution of this optimization problem is given by

$$T(t) = \mathcal{M}^{+} \begin{bmatrix} u_1(t) \\ M_{\mathrm{thrust}}(t) \end{bmatrix}, \qquad t \geq t_0, \tag{69}$$

where

$$\mathcal{M}^{+} \triangleq \frac{1}{8} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ l_y^{-1} & l_y^{-1} & -l_y^{-1} & -l_y^{-1} & l_y^{-1} & l_y^{-1} & -l_y^{-1} & -l_y^{-1} \\ -l_x^{-1} & l_x^{-1} & l_x^{-1} & -l_x^{-1} & -l_x^{-1} & l_x^{-1} & l_x^{-1} & -l_x^{-1} \\ -c_T^{-1} & c_T^{-1} & -c_T^{-1} & c_T^{-1} & c_T^{-1} & -c_T^{-1} & c_T^{-1} & -c_T^{-1} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{8 \times 4}, \tag{70}$$

denotes the *Moore-Penrose pseudoinverse* of $\mathcal{M}$.

Equation (69) does not assure that $\sum_{i=1}^{8} T_i(t) \in [u_{\min}, u_{\max}]$ for all $t \geq t_0$. This point will be addressed in future research. In the numerical examples presented in Section VII below, the thrust force generated by each motor is

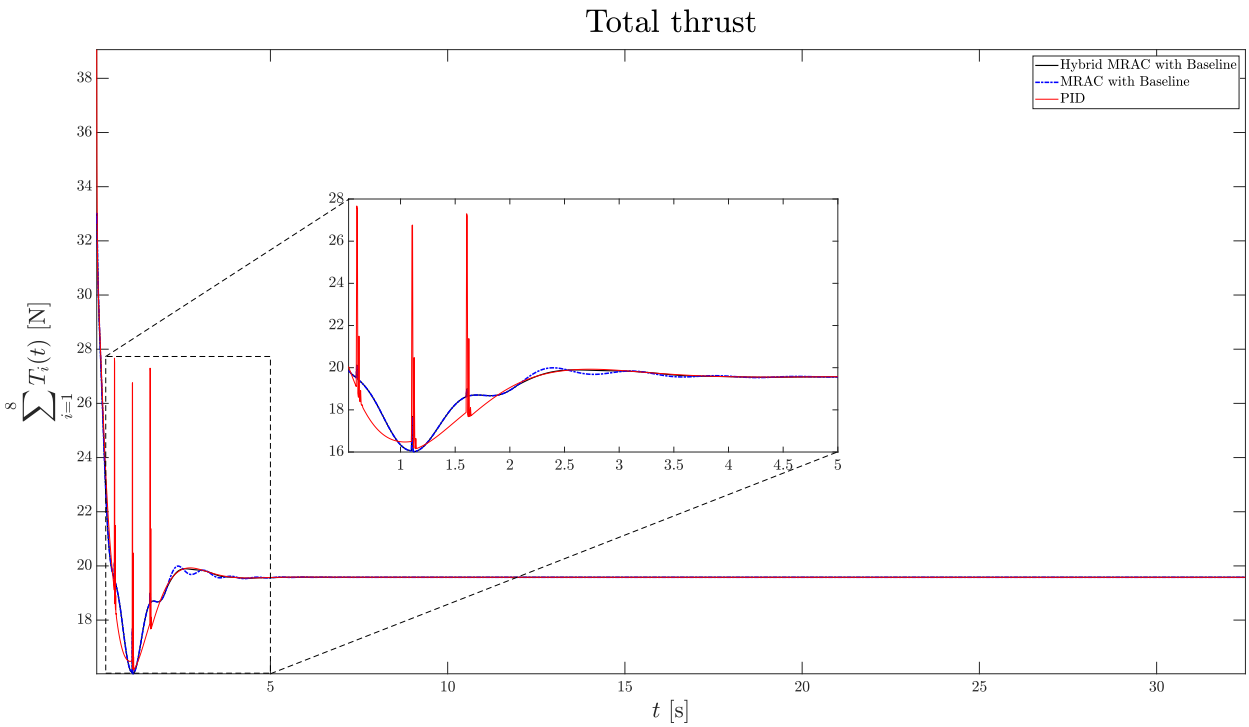

Fig. 4 Sum of the thrust forces exerted by each motor in nominal conditions. The thrust force required by the three controllers is substantially similar. However, the PID controller requires a larger initial thrust. Furthermore, the PID controller requires some sudden variations in the thrust force to cope with the three sudden variations in the user-defined yaw angle shown in Figure 3

computed as

$$\text{sat}\left(\|T_i(t)\|, T_{i,\min}, T_{i,\max}\right) \tag{71}$$

where $T_{i,\min} = 0.6275$ N denotes the motor's minimum realizable thrust force and $T_{i,\max} = 10.625$ N denotes the motor's maximum realizable thrust force. Future work directions involve correlating $T_{\min,X}$, $T_{\min,Y}$, $T_{\min,Z}$, $T_{\max,X}$, $T_{\max,Y}$, and $T_{\max,Z}$ with $T_{i,\max}$ and $T_{i,\min}$ with for each $i = 1, \ldots, 8$.

VII. Numerical Simulations

In this section, we present the results of numerical simulations performed using the high-fidelity environment PyChrono [3, 4]. Three control systems are tested, namely a system based on the PID controller, a system based on an MRAC controller [2, Ch. 9], and a system based on the proposed MRAC controller for nonlinear, time-varying hybrid plants. The PID controller has been chosen because a vast number of commercial-off-the-shelf microcontrollers for UAVs employ this control technique. The MRAC controller has been chosen because it the most suitable for a fair comparison with the results in Section IV and because MRAC has been extensively applied to control multi-rotor UAVs in the past decade [13]. The results of the numerical simulations presented in this section are shown also in the YouTube video [7].

A. Actual and Simulated X8-Copter Details

To simulate last-mile delivery missions, we designed and realized a Group 1 UAV [14] equipped with a payload compartment; see Figure 6. The X8-copter configuration has been chosen for its compactness, redundancy of thrust in the case of fault or failure of a motor, and higher thrust-to-weight ratio than classical quadcopters. The mass of the UAV frame is approximately 0.450 kg, and its dimensions are approximately 0.345 m $\times$ 0.245 m $\times$ 0.300 m. The battery we employed has an 8000 mAh capacity, 25 C discharge rating, 50 C burst rating, and a 14.8 V nominal voltage. The

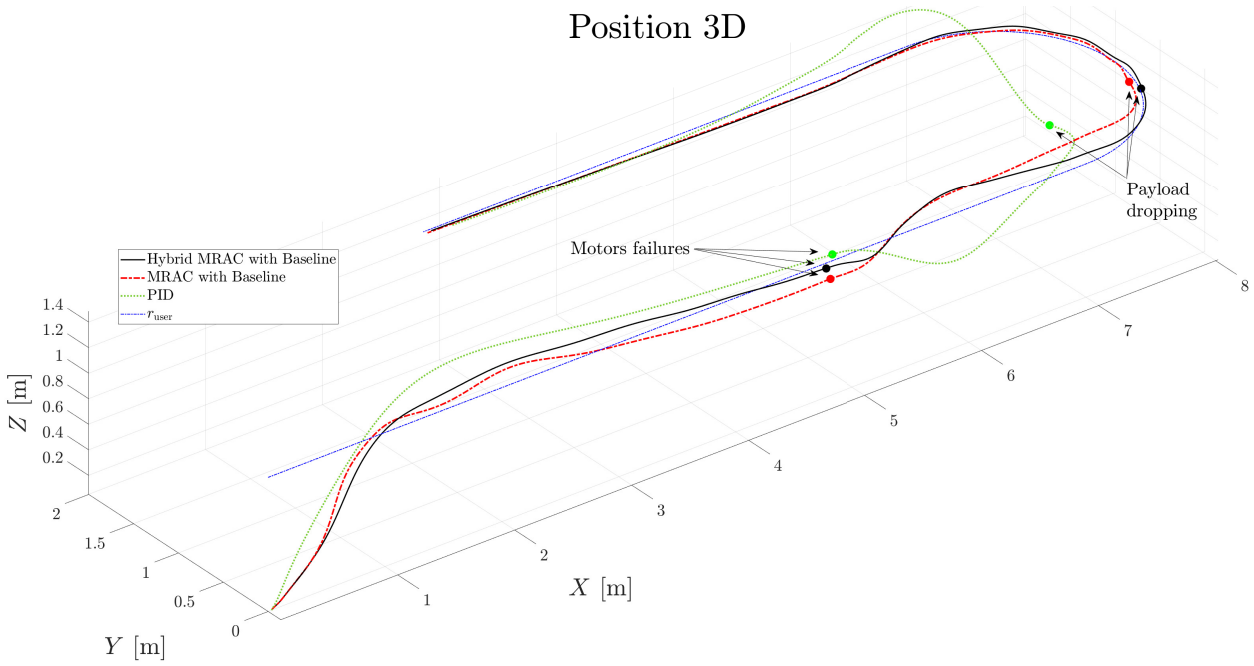

Fig. 5 Trajectories of the UAV in off-nominal conditions. The motor failure occurs half-way through the first rectilinear segment of the stadium. The payloads are dropped while traversing the circular segments. Despite the knowledge of the payload's mass, the PID controller is unable to follow the user-defined trajectory. The two MRAC systems allow considerably better trajectory following. However, as shown in Figures 9 and 10, the adaptive controller for hybrid systems outperforms the classical adaptive controller

motors operate at full load at a maximum current of 37.8 A, maximum angular rotational speed of 28800 rpm, and 620 W of maximum power. The propellers chosen in this research have three blades, a diameter of 124 mm, and a pitch of 109 mm. From acquired data on the configuration of coaxial motors, with two motors and two propellers, we measured an effective thrust of 1.800 kg. The companion computer is an ODroid XU4 and the microcontroller is a Pixhawk 6C. In flight, the companion computer will be responsible for executing the control algorithms. The microcontroller will provide the estimates on the UAV's state and will interface the companion computer with the propellers through the ESCs (electronic speed controllers).

As part of the design process of the actual UAV, a CAD (computer aided design) model of the UAV was being produced through an iterative process. When the X8-copter was completely realized, the CAD model reproduced the actual features of the UAV in detail. Each part of the CAD model was endowed with the proper material properties so that the mass of each component matched the mass of the same component in the actual vehicle. Imported in PyChrono, the CAD model allowed not only to simulate the vehicle's dynamics, but also to account for collisions between the payload and its casing. Each motor's thrust force has been simulated by applying a force vector modeled as (36) at the hub of each motor. Aerodynamic forces and moments have been simulated by adding a vector force at the vehicle's center of mass; future research directions involve the refinement of these assumptions. The UAV's position is computed as the position of the microcontroller's reference point. Remarkably, this point does not coincide with the UAV's center of mass, as assumed in Section VI.

B. Simulations in Nominal Conditions

In this section we show how, in nominal conditions, that is, the UAV's aerodynamic properties are known and no payload is transported, the X8-copter can follow the user-defined trajectory with similar performances using a PID controller, a classical MRAC system, and the proposed hybrid MRAC system. The controllers' tunable parameters have been tuned to obtain comparable performances in terms of maximum overshoots and settling times in a specific mission. In this mission, the UAV is tasked with following a circular reference trajectory of radius 3 m at a height of 1 m with a constant translational velocity of 0.6 m/s. The UAV's roll axis needs to be tangential to the user-defined trajectory at all

Fig. 6 X8-copter, whose detailed CAD model, shown in Figure 1, has been employed for high-fidelity simulations. As in the numerical simulations presented in this paper, two steel balls of 50.8 mm diameter are placed in a transparent rigid container that serves as payload compartment

times. At take-off, the UAV's roll axis points in a radial direction relative to the user-defined trajectory.

Figure 2 shows the trajectory tracking error measured as the Euclidean norm of the difference between the UAV's position and velocity $x(\cdot)$ and the desired position and velocity $x_{\text{user}}(\cdot)$ defined by the user. The controllers were tuned assuming that $\overline{m} = m_\sigma = 2.025$ kg for all $\sigma \in \Sigma$, and

$$\overline{I}_\sigma = I_\sigma = \begin{bmatrix} 2.2720 \cdot 10^{-2} & -6.5570 \cdot 10^{-6} & -1.0035 \cdot 10^{-3} \\ -6.5570 \cdot 10^{-6} & 2.2020 \cdot 10^{-2} & 5.6584 \cdot 10^{-6} \\ -1.0035 \cdot 10^{-3} & 5.6584 \cdot 10^{-6} & 1.6147 \cdot 10^{-2} \end{bmatrix} \text{kg} \cdot \text{m}^2; \tag{72}$$

other tunable parameters are omitted for brevity.

Figure 3 shows the UAV's attitude and reference attitude obtained by applying the proposed MRAC system for hybrid plants. It is worthwhile noting that the user-defined reference yaw angle experiences three sudden variations in the time interval $[0, 1.64]$ s. Figure 4 shows how the thrust forces required by the three controllers are substantially similar. However, at $t = 0$, the PID controller requires a stronger initial thrust. Furthermore, because of the three sudden variations in the user-defined refernece yaw angle over the time interval $[0, 1.64]$ s, the PID controller requires three sudden variations in the thrust force. Such rapid variations in the thrust force are not required by the adaptive controllers. In these simulations, the resetting condition (20) is not verified, and, hence, the classical MRAC system and the hybrid MRAC system perform identically.

C. Simulations in Off-Nominal Conditions

In this section, we present the results of numerical simulations obtained by applying the same controllers as in Section VII.B in a challenging mission scenario. The first of these challenges, which are aimed at testing the controllers' reliability, consist in tasking the X8-copter to follow a trajectory that is shaped like a stadium at an altitude of 1.1 m. The rectilinear segments of the stadium are 7 m long, and the radii of the circular segments are 1 m long, which is considerably smaller than the radius of the circular trajectory used to tune the three controllers considered herein. The user-defined translational velocity is 1.2000 m/s, and the UAV's roll axis is tasked with being tangential to the reference trajectory at all times. At the junction between the rectilinear and the semi-circular segments, the user-defined acceleration experiences an instantaneous variation of 1.44 m/s^2 in the direction of the axis Y of the inertial reference frame $\mathbb{I}$.

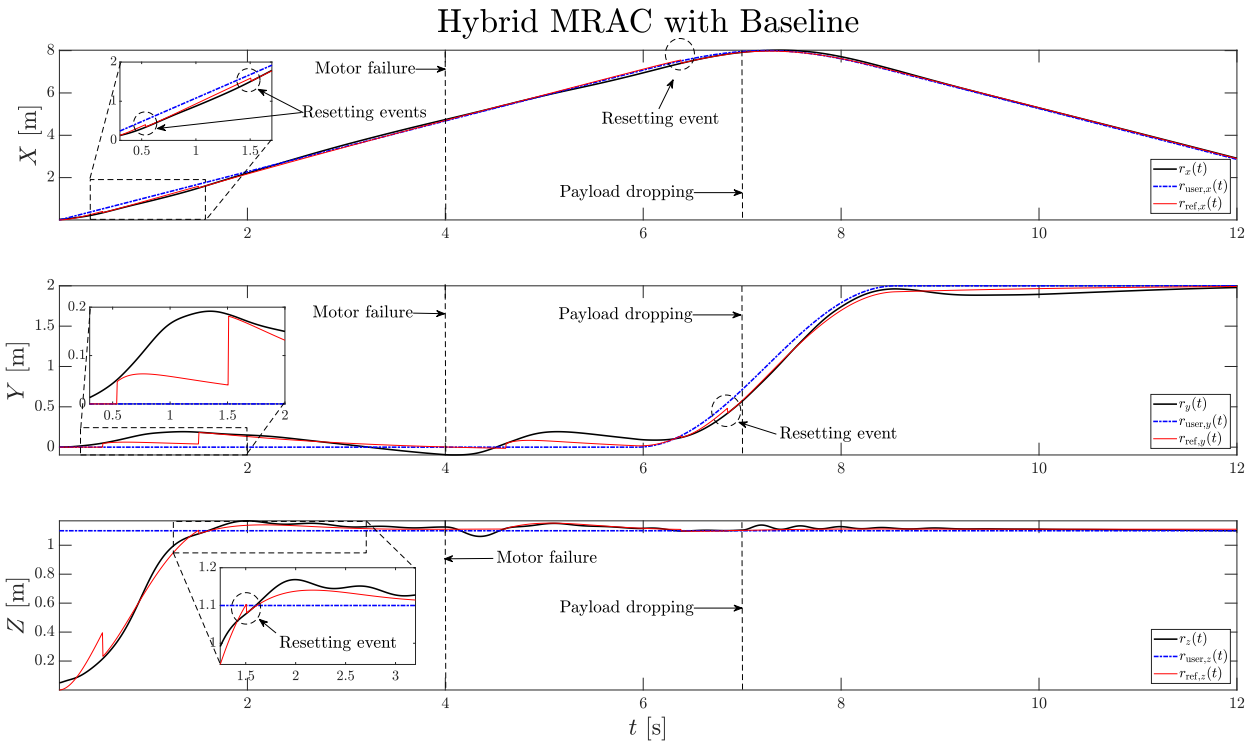

Fig. 7 Position of the UAV controlled by means of the hybrid MRAC system, reference model's position, and user-defined reference position. Multiple resetting events in the reference position, which are captured by (21), are marked

The second challenge consists in taking off with two steel balls, whose mass is 0.5388 kg each and whose radius is 0.0508 m each. Since the UAV's payload compartment is 0.2450 m wide and 0.2000 m large, these balls are free to move throughout the flight, and, hence vary key inertial properties of the overall systems such as position of the center of mass and inertia matrix, and produce impulses when hitting the walls of the payload compartment. Placing both balls in the same corner of the payload compartment, the inertia matrix of the UAV and its payload is

$$\begin{bmatrix} 1.191 \cdot 10^{-1} & 5.709 \cdot 10^{-3} & -2.551 \cdot 10^{-2} \\ 5.709 \cdot 10^{-3} & 1.193 \cdot 10^{-1} & -1.865 \cdot 10^{-2} \\ -2.551 \cdot 10^{-2} & -1.865 \cdot 10^{-2} & 3.120 \cdot 10^{-2} \end{bmatrix} \text{kg} \cdot \text{m}^2. \tag{73}$$

The $(2,2)$-induced norm of this matrix is $1.344 \cdot 10^{-1}$ kg$\cdot$ m^2, whereas the $(2,2)$-induced norm of the inertia matrix of the UAV without its payload, which is captured by (72), is $0.2286 \cdot 10^{-1}$ kg$\cdot$ m^2. Therefore, the introduction of the payload may produce a relevant variation in some of the inertial properties of this Group 1 UAV. The two MRAC controllers employ the same tunable parameter as in Section VII.B. However, setting $\overline{m} = 2.025$ kg, the PID controller is unable to enable take off. Thus, since the mass of a payload can be readily determined, and to enable a comparison between the MRAC controllers and the popular PID controller, in this section, the PID controller is executed assuming that $\overline{m} = 3.1026$ kg.

The third challenge consists in simulating the failure of motor M3 and a fault in motor M8 at 4 s; for details, see Figure 1. Specifically, for all $t \geq 4$ s, $T_3(t) = 0$ and $T_8(t) = 0.5T_{8,\text{nominal}}(t)$, where $T_{8,\text{nominal}}(\cdot)$ denotes the thrust the eight motor should produce. The controllers are unaware of when the motors stop performing as expected. The fourth challenge consists in dropping the payload at 7 s. The controllers are unaware of when the payload is dropped.

Figure 5 shows the UAV's trajectories, where the motors fail, and where the payloads are dropped. Despite the knowledge of the payload's mass, the PID controller is unable to follow the user-defined trajectory. Figure 7 shows the position of the UAV as a function of time, the reference model's position, and the user-defined reference position. Figure 8 shows the velocity of the UAV as a function of time, the reference model's velocity, and the user-defined reference velocity. The resetting condition given by (20) is verified on multiple occasions, and the reference model's position and

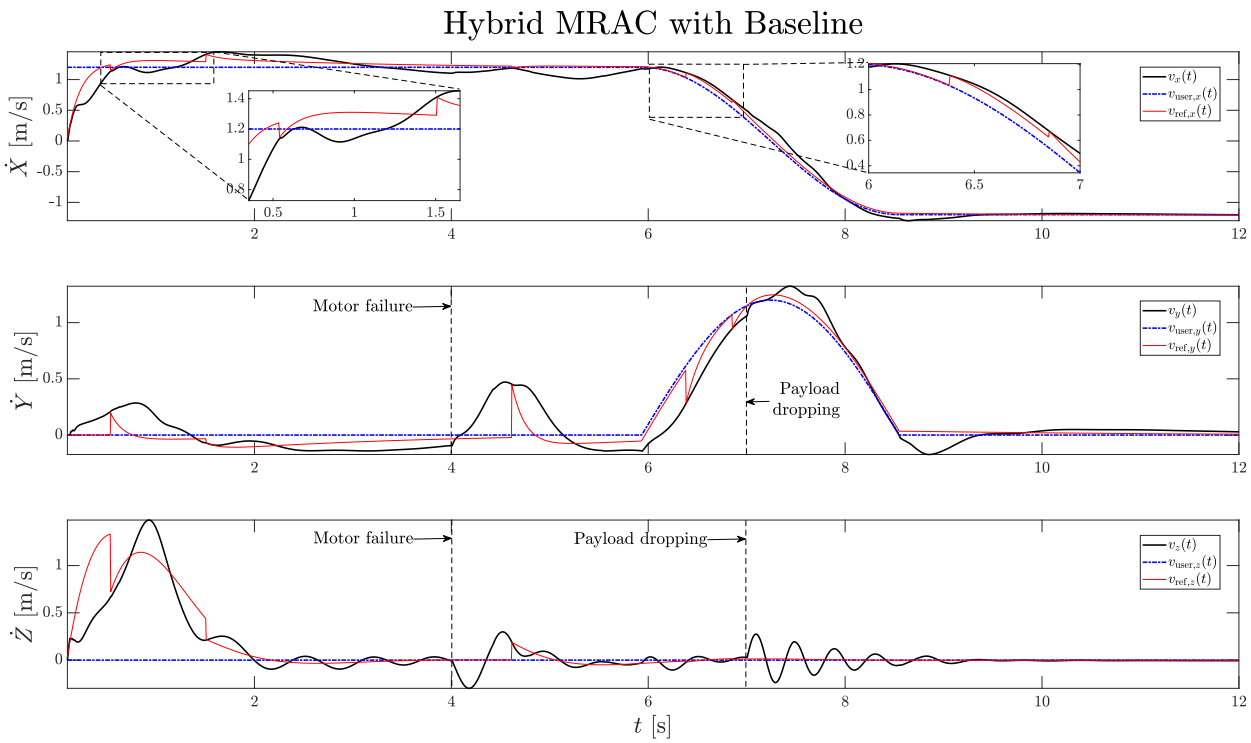

Fig. 8 Velocity of the UAV controlled by means of the hybrid MRAC system, reference model's velocity, and user-defined reference velocity. Multiple resetting events in the reference position occur to reduce the error in tracking the user-defined velocity

velocity experience instantaneous changes captured by (21). As shown in Figures 7 and 8, some of these jumps in the reference trajectory are substantial, and some others are less relevant. As discussed in the following, these resetting events in the reference model allow the X8-copter to follow the user-defined position and velocity more closely and with a smaller effort than the classical MRAC system and the PID-based system. Resetting events of the reference model appear to be stronger in the velocity tracking problem than in the position tracking problem.

Figure 9 shows the trajectory tracking error measured as the Euclidean norm of the difference between the UAV's position and velocity $x(\cdot)$ and the position and velocity $x_{\text{user}}(\cdot)$ defined by the user. The MRAC controllers clearly outperform the PID controller despite the fact that this classical control system is implemented assuming that the payload's mass is known, whereas the adaptive controllers are implemented considering the UAV's mass, without its payload, to be the system's actual mass. In general, the hybrid MRAC system performs considerably better than the classical MRAC system.

Figure 10 shows the sum of the thrust forces exerted by each motor. The hybrid MRAC controller requires smaller levels of control effort, and lower-frequency oscillations in the total thrust. The PID controller requires sudden variations in the thrust force despite the fact that the payload's mass is known since the user-defined reference acceleration is discontinuous at the junction between the rectilinear and the semi-circular segments of the user-defined reference trajectory.

VIII. Conclusion

This paper provided the first application of a novel model reference adaptive control architecture for nonlinear, time-varying, hybrid dynamical systems to the design of control systems for multi-rotor UAVs, such as X8-copters. This control architecture allows to account for model uncertainties, motor faults and failures, unknown and unsteady payloads, and instantaneous variations in the vehicle's state and dynamics due to collisions or the unmodeled dynamics of the payloads. Numerical simulations performed employing the high-fidelity solver PyChrono illustrate the applicability and the advantages of the proposed results. This numerical analysis involved comparing the performance of the hybrid

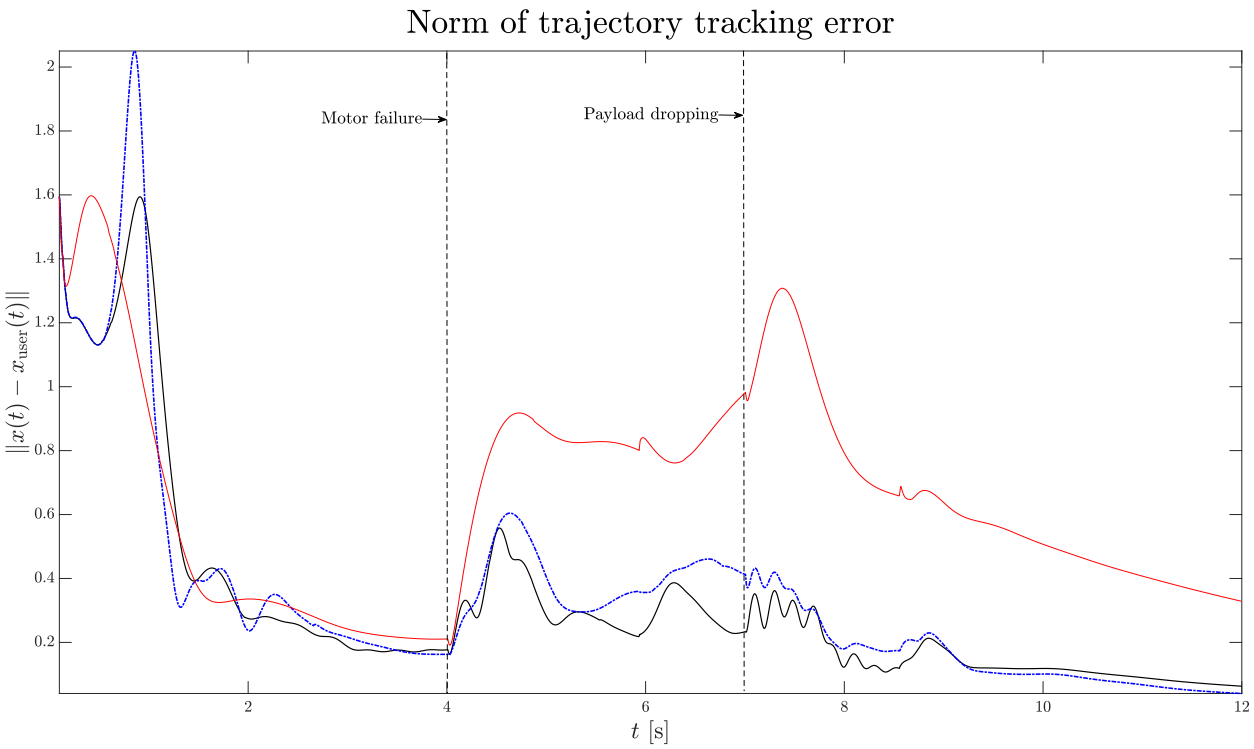

Fig. 9 Trajectory tracking error applying the hybrid MRAC system, a classical MRAC system, and a PID control system. The tracking error is measured as the difference between the UAV's translational position and velocity $x(\cdot)$ and the user-defined reference position and velocity $x_{\text{user}}(\cdot)$. The hybrid MRAC system outperforms both the classical MRAC system and the PID-based control system

MRAC controller with the performance of a classical MRAC system and a PID controller. The classical MRAC system has been chosen for a fair comparison, and the PID controller has been chosen for the popularity of this control scheme in commercial-off-the-shelf microcontrollers for autonomous UAVs. It is apparent how the hybrid MRAC system allows to recover more rapidly from off-nominal conditions, such as faults and failures of multiple motors, the presence of heavy, unknown payloads, and the sudden dropping of this payload at unknown times. These improved performances are attained by applying thrust forces, whose magnitude and oscillations were smaller than those required by the classical MRAC system and the PID controller.

Future work directions are multiple. Some of these include an experimental validation of the results produced by PyChrono. Furthermore, PyChrono will be used for extensive computational analyses over large numbers of simulations, each representing a different mission scenario.

Acknowledgments

This work has been partly supported funded by the National Science Foundation under Grant no. 2137159, and the US Army Research Lab under Grant no. W911QX2320001.

References

[1] L'Afflitto, A., "Model reference adaptive control for nonlinear time-varying hybrid dynamical systems," *International Journal of Adaptive Control and Signal Processing*, Vol. 37, No. 8, 2023, pp. 2162–2183. https://doi.org/https://doi.org/10.1002/acs.3631.

[2] Lavretsky, E., and Wise, K., *Robust and Adaptive Control: With Aerospace Applications*, Springer, London, UK, 2012.

[3] Tasora, A., Serban, R., Mazhar, H., Pazouki, A., Melanz, D., Fleischmann, J., Taylor, M., Sugiyama, H., and Negrut, D., "Chrono: An open source multi-physics dynamics engine," *High Performance Computing in Science and Engineering – Lecture Notes in Computer Science*, edited by T. Kozubek, Springer, 2016, pp. 19–49.

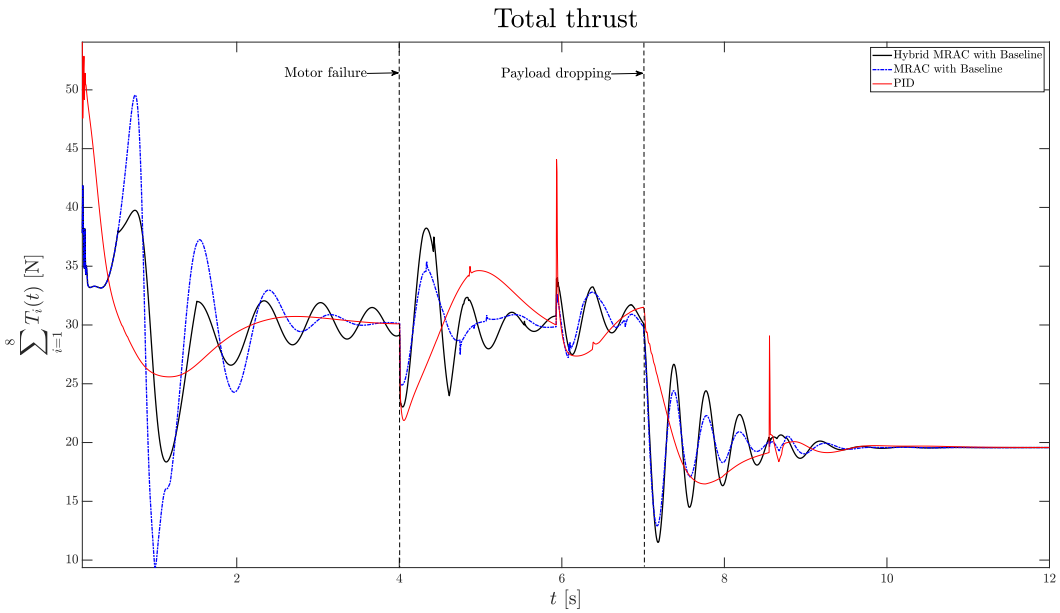

Fig. 10 Sum of the thrust forces exerted by each motor in off-nominal conditions. The hybrid MRAC controller requires smaller levels of control effort, and lower-frequency oscillations in the total thrust. The PID controller requires sudden variations in the thrust force despite the fact that the payload's mass is known since the user-defined reference acceleration is discontinuous at the junction between the rectilinear and the semi-circular segments of the user-defined reference trajectory

[4] Project Chrono, "Chrono: An Open Source Framework for the Physics-Based Simulation of Dynamic Systems," https://projectchrono.org/, 2014. Accessed: 2023-10-14.

[5] Liu, C. K., and Negrut, D., "The Role of Physics-Based Simulators in Robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, Vol. 4, No. 1, 2021, pp. 35–58. https://doi.org/10.1146/annurev-control-072220-093055.

[6] Project Chrono, "LittleHexy copter model," https://api.projectchrono.org/group__robot__models__copter.html, 2014. Accessed: 2023-10-14.

[7] Gramuglia, M., Kumar, G. M., and L'Afflitto, A., "A Hybrid Model Reference Adaptive Control System for Multi-Rotor Unmanned Aerial Vehicles," YouTube video https://youtu.be/TL-FlCxEhCM, 2023. Last accessed 11/28/2023.

[8] Bernstein, D. S., *Matrix Mathematics: Theory, Facts, and Formulas, Second Edition*, Princeton University Press, Princeton, NJ, 2009.

[9] Goebel, R., Sanfelice, R., and Teel, A., *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*, Princeton University Press, 2012.

[10] Sanfelice, R. G., Goebel, R., and Teel, A. R., "Generalized solutions to hybrid dynamical systems," *Control, Optimisation and Calculus of Variations*, Vol. 14, No. 4, 2008, pp. 699–724. https://doi.org/10.1051/cocv:2008008.

[11] L'Afflitto, A., and Mohammadi, K., "Robust observer-based control of nonlinear dynamical systems with state constraints," *Journal of the Franklin Institute*, Vol. 354, No. 16, 2017, pp. 7385–7409. https://doi.org/https://doi.org/10.1016/j.jfranklin.2017.09.007.

[12] L'Afflitto, A., *A Mathematical Perspective on Flight Dynamics and Control*, Springer, London, UK, 2017.

[13] Marshall, J. A., Sun, W., and L'Afflitto, A., "A survey of guidance, navigation, and control systems for autonomous multi-rotor small unmanned aerial systems," *Annual Reviews in Control*, Vol. 52, 2021, pp. 390–427. https://doi.org/https://doi.org/10.1016/j.arcontrol.2021.10.013.

[14] Federal Aviation Administration, "Unmanned Aircraft Systems Categorization Chart," https://www.faa.gov/air_traffic/publications/atpubs/aim_html/chap11_section_3.html, 2011. Accessed: 2023-11-8.