Preprint version 16th Int. Symp. on Robotics Research, Singapore (2013)

Aggressive Maneuver Regulation of a Quadrotor UAV

Sara Spedicato, Giuseppe Notarstefano, Heinrich H. Bülthoff, and Antonio Franchi

Abstract In this paper we design a nonlinear controller for aggressive maneuvering of a quadrotor. We take a maneuver regulation perspective. Differently from the classical trajectory tracking approach, maneuver regulation does not require following a timed reference state, but a geometric "path" with a velocity (and possibly orientation) profile assigned on it. The proposed controller relies on three main ideas. Given a desired maneuver, i.e., a set of state trajectories equivalent under time translations, the system dynamics is decomposed into dynamics longitudinal and transverse to the maneuver. A *space-dependent* version of the transverse dynamics is derived, by using the longitudinal state, i.e., the arc-length of the path, as an independent variable. Then the controller is obtained as a function of the arc-length consisting of two terms: a feedforward term, being the nominal input to apply when on the path at the current arc-length, and a feedback term exponentially stabilizing the state-dependent transverse dynamics. Numerical computations are presented to prove the effectiveness of the proposed strategy. The controller performances are tested in presence of uncertainty of the model parameters and input noise and saturations. The controller is also tested in a realistic simulation environment validated against an experimental test-bed.

1 Introduction

In the recent years, the steadily growing number of applications involving Unmanned Aerial Vehicles (UAVs), as quadrotors, has raised attention on the execution of precise aggressive motions. This is, in fact, a fundamental requirement in several (complex) tasks. The classical approaches used in the literature to perform such motions fall into the categories of *trajectory tracking* and *path following* methods. Trajectory tracking techniques, aim at limiting the error between the actual system state and the desired state at a specified time. The system state includes position, orientation, and linear and angular velocities. Whenever an exogenous disturb (e.g., wind) forces the robot to momentarily lag behind the "moving reference" of the desired trajectory, then undesired phenomena are likely to arise, such as: (i) huge peaks of acceleration and (ii) a poor geometric tracking of the planned path that may lead to collisions with the surrounding world.

In order to highlight this last sensitive issue, let us consider the example depicted in Fig. 1. A quadrotor has to fly around an obstacle tracking a given planned trajectory, which is specified as a desired state at each time t. At $t = t_1$ strong opposing wind significantly decelerates the actual motion of the quadrotor for a few seconds. When the wind ceases, at $t = t_3$, the quadrotor recovers the full control of its motion

Sara Spedicato, Giuseppe Notarstefano
Department of Engineering, Università del Salento, Via per Monteroni, 73100 Lecce, Italy,
e-mail: sara.spedicato@unisalento.it, giuseppe.notarstefano@unisalento.it

Heinrich H. Bülthoff, Antonio Franchi
Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany,
e-mail: hhb@tuebingen.mpg.de, antonio.franchi@tuebingen.mpg.de

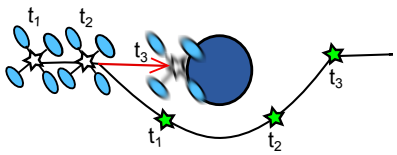


Fig. 1: Example of an undesired phenomenon arising using trajectory tracking.

and tries to quickly catch up with the moving desired state that is now on the other side of the obstacle, thus dramatically crashing into it.

These drawbacks do not arise in classical path following techniques, whose objective is to have the system position follow a geometric path without a predefined time scheduling. Since a pre-defined timing law on the path is not given, these methods consider a tracking error between the current robot position and the set of positions on the entire geometric path; while orientation, linear and angular velocities are not usually taken into account. Classical path following techniques are able to avoid undesired phenomena as the one described above. However, they can only drive the center of mass of the UAV along the path without ensuring a desired orientation and velocity profile along it.

To overcome the drawbacks that affect both these two techniques, we deal with an extended version of the path following, called *maneuver regulation*, that aims at satisfying additional requirements (such as assigning orientation, linear velocity, and angular velocity on the path).

We organize the literature on quadrotor controllers in two parts. First, a vast number of trajectory tracking techniques for quadrotors have been proposed. In [1] a dynamic feedback controller, which renders the system linear and controllable, has been developed. In [2] a full backstepping technique is presented, based on a decomposition of the dynamic model into an underactuated subsystem, and a fully-actuated subsystem. A geometric tracking control is presented in [3]. The nonlinear tracking controller is developed on the special Euclidean group $SE(3)$ and it is shown to have desirable closed loop properties that are almost global. The trajectory tracking controller in [3] has been successfully implemented in [4] to perform fast aerobatic maneuvers without exogenous disturbances and precise measures from an external motion capture system (aerobatic maneuvering is also addressed, for autonomous helicopters, in [5, 6]). A sliding mode controller is proposed in [7] in order to stabilize the quadrotor model as a class of cascaded under-actuated systems. In [8], an underactuated nonlinear $\mathscr{H}_\infty$ controller based on the six degrees of freedom dynamic quadrotor model is designed to control the attitude and altitude in an inner-loop. The outer-loop control is performed using a model-based predictive controller to track the reference trajectory. Finally a trajectory tracker based on a linear quadratic regulator (LQR) is proposed in [9].

Second, only recently, extended path following techniques for quadrotors have been presented. In [10] the problem is solved by means of a backstepping technique. In [11] the problem is addressed as the stabilization of the zero dynamics for a nonlinear control system and solved using input-output feedback linearization on an augmented quadrotor system. The technique is refined in [12] with the objective of executing a more general class of paths “in a more general manner”. The definition

of the position error as the distance between the actual quadrotor position and the desired path is used in [13], where a "commanded acceleration" is computed using a PD feedback of the position and velocity errors.

The first and main contribution of the paper is the design of a maneuver regulation controller for aggressive maneuvering of a quadrotor on a three-dimensional path with assigned orientation and velocity profiles along it. The control strategy, inspired to the one proposed in [14] for a motorcycle on a bi-dimensional path, is based on the idea of *transverse linearization* of the dynamic system. Given a desired trajectory, the system dynamics is rewritten in terms of a longitudinal and a transverse dynamics. The new system states are the arc-length s and a set of *transverse coordinates* $w(s)$ (defined by means of an appropriate distance between the current state and the desired trajectory states). Differently from the (extended) path following approaches in [10, 11, 12], we consider a *space-dependent* version of the transverse dynamics. That is, we derive a differential equation in which the arc-length s is the independent variable, so that the transverse linearization is obtained by linearizing such a space-dependent dynamics. By solving an infinite-horizon linear quadratic regulator optimal control problem, the (space-dependent) transverse dynamics linearization can be exponentially stabilized (under standard controllability assumptions). Thus, the quadrotor maneuver regulation controller is a function of the arc-length s: a feedforward term being the desired input for the given s, plus a feedback $K(s)w(s)$ to stabilize the (space-dependent) transverse dynamics. Notably, although the feedback is linear as a function of the arc-length, it turns to be a nonlinear feedback of the original system state in the time-domain as $s(t) = s(x(t))$.

As second contribution, and preliminary step for experimental tests, we test our maneuver regulation controller on a physical quadrotor simulator, in order to show the performances of the proposed maneuver regulation controller in a realistic simulation scenario. Furthermore, we perform numerical computations under uncertainties on the model parameters and input noise and saturations. The computations show how the controller well behaves in performing a fairly aggressive maneuver as a barrel roll. In fact, we verified that even if the design is executed with some nominal inertial parameters the controller is able to adapt the control effort depending to the actual inertia and to maintain the system stability even if the severe input saturation and noise do not allow a perfect tracking of the desired maneuver.

Compared to the other extended path following approaches designed for a quadrotor, [10, 11, 12], our maneuver regulation controller has three key differences. First, the other schemes only ensure the distance from the path and the error on the yaw angle to converge to zero. The resulting roll and pitch angle, although stable, cannot be assigned a priori, so that problems could arise for example in narrow passageways. Second, the scheme does not rely on structural properties of the simplified quadrotor model as, e.g., flatness or non-minimum-phase-ness. Thus, it can be applied also to more complex (possibly non-minimum phase) models. Third, the proposed maneuver regulation scheme can be decoupled into a slow time-scale block and a realtime one. The desired trajectory and the feedback gains can be computed in a slow time-scale. In particular, the desired trajectory can be computed by using trajectory optimization techniques as the ones proposed in [15, 16]. The online computation only requires the calculation of the scheduled arc-length and the application of the feedback gain. Due to this structure, the proposed controller can

be seen as a preliminary step toward the development of a receding-horizon scheme involving both trajectory generation and regulation in a coupled scheme.

The paper is organized as follows. In Section 2 the quadrotor model and the maneuver regulation problem are introduced. Section 3 addresses the design of the maneuver regulation controller for the quadrotor. Finally, in Section 4 numerical computations and physical simulations are provided in order to prove the effectiveness of the LQR based controller under parameters uncertainty and input saturation.

2 Quadrotor model and the maneuver regulation problem

In this section we present the standard quadrotor model that is instrumental to formally define the maneuver regulation problem and the proposed controller.

In the following we denote vectors using bold small symbols and matrices using capital letters. Given an inertial reference frame (with x-y-z axes oriented in a North-East-Down fashion) and a body reference frame attached to the quadrotor center of mass (with x-y-z axes oriented in a forward-right-down fashion), let $\boldsymbol{p} \in \mathbb{R}^3$ be the position vector from the origin of the inertial frame to the origin of the body frame, expressed in the inertial frame. The orientation of the body frame with respect to the inertial frame is denoted by the rotation matrix $R \in SO(3)$, which maps vectors in the body frame into vectors in the inertial frame. Let $\boldsymbol{v} \in \mathbb{R}^3$ and $\boldsymbol{\omega} \in \mathbb{R}^3$ denote respectively the linear and angular velocities expressed in the body frame.

The quadrotor is driven by four forces and torques produced by the four propellers. Each thrust force is directed along the body z-axis but pointing in the negative direction. The thrust forces produce the torques γ_1 and γ_2 around the body x-axis and y-axis respectively. A torque γ_3 around the z-axis is produced by the reaction moments acting on the propellers. The sum of the thrust forces is denoted by f and the torques vector is denoted by $\boldsymbol{\gamma} = (\gamma_1 \ \gamma_2 \ \gamma_3)^T$.

The standard quadrotor model, see, e.g., [17], is

$$\dot{\boldsymbol{p}} = R\boldsymbol{v} \tag{1}$$

$$m\dot{\boldsymbol{v}} = -m\Omega\boldsymbol{v} + mgR^T\boldsymbol{e}_3 - f\boldsymbol{e}_3 \tag{2}$$

$$\dot{R} = R\Omega \tag{3}$$

$$J\dot{\boldsymbol{\omega}} = -\Omega J\boldsymbol{\omega} + \boldsymbol{\gamma} \tag{4}$$

where m is the quadrotor mass, $J = \mathrm{diag}(j_x, j_y, j_z)$ is the inertia matrix, g is the gravity constant, Ω is the screw-symmetric matrix associated with $\boldsymbol{\omega}$ and $\boldsymbol{e}_3 = (0\ 0\ 1)^T$.

We choose to parameterize the rotation matrix R by roll-pitch-yaw angles[1]. From the inertial reference frame the first rotation is taken around the z-axis by the yaw angle ψ. The coordinate system is then rotated around the new y-axis by the pitch angle θ and finally rotated about the new x-axis by the roll angle φ. The rotation matrix is thus

[1] This parametrization of R, largely used in the literature, has a singularity when the pitch angle reaches $\pi/2$. However, the proposed techniques can be developed for any other parametrization. Thus, given the desired maneuver, the best suited parametrization can be used to avoid singularities.

$$R = \begin{pmatrix} c\psi c\theta & -s\psi c\varphi + c\psi s\theta s\varphi & s\psi s\varphi + c\psi s\theta c\varphi \\ s\psi c\theta & c\psi c\varphi + s\psi s\theta s\varphi & -s\varphi c\psi + s\psi s\theta c\varphi \\ -s\theta & s\varphi c\theta & c\theta c\varphi \end{pmatrix}, \tag{5}$$

where for a generic angle ϕ we define $c\phi := \cos(\phi)$ and $s\phi := \sin(\phi)$.

Let us define $\boldsymbol{p} = (p_1\ p_2\ p_3)^T$, $\boldsymbol{v} = (v_1\ v_2\ v_3)^T$ and $\boldsymbol{\omega} = (p\ q\ r)^T$. Using the roll-pitch-yaw parametrization equations (1-4) are

$$\dot{p}_1 = c\psi c\theta v_1 + (-s\psi c\varphi + c\psi s\theta s\varphi)v_2 + (s\psi s\varphi + c\psi s\theta c\varphi)v_3 \tag{6}$$
$$\dot{p}_2 = s\psi c\theta v_1 + (c\psi c\varphi + s\psi s\theta s\varphi)v_2 + (-s\varphi c\psi + s\psi s\theta c\varphi)v_3 \tag{7}$$
$$\dot{p}_3 = -s\theta v_1 + s\varphi c\theta v_2 + c\theta c\varphi v_3 \tag{8}$$
$$\dot{\varphi} = p + qs\varphi \tan\theta + rc\varphi \tan\theta \tag{9}$$
$$\dot{\theta} = qc\varphi - rs\varphi \tag{10}$$
$$\dot{\psi} = qs\varphi \frac{1}{c\theta} + rc\varphi \frac{1}{c\theta} \tag{11}$$
$$\dot{v}_1 = rv_2 - qv_3 - gs\theta \tag{12}$$
$$\dot{v}_2 = -rv_1 + pv_3 + gs\varphi c\theta \tag{13}$$
$$\dot{v}_3 = qv_1 - pv_2 + gc\theta c\varphi - \frac{f}{m} \tag{14}$$
$$\dot{p} = qr\left(\frac{j_y - j_z}{j_x}\right) + \frac{\gamma_1}{j_x} \tag{15}$$
$$\dot{q} = pr\left(\frac{j_z - j_x}{j_y}\right) + \frac{\gamma_2}{j_y} \tag{16}$$
$$\dot{r} = pq\left(\frac{j_x - j_y}{j_z}\right) + \frac{\gamma_3}{j_z} \tag{17}$$

The equations (6-17) represent a nonlinear, time-invariant control system of the form

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{18}$$
$$\boldsymbol{y}(t) = h(\boldsymbol{x}(t)) \tag{19}$$

with state $\boldsymbol{x} = (p_1\ p_2\ p_3\ \varphi\ \theta\ \psi\ v_1\ v_2\ v_3\ p\ q\ r)^T \in \mathbb{R}^{12}$, input $\boldsymbol{u} = (f\ \gamma_1\ \gamma_2\ \gamma_3)^T \in \mathbb{R}^4$ and output $\boldsymbol{y} = (p_1\ p_2\ p_3)^T \in \mathbb{R}^3$. The equation (18) can be also written in scalar form as

$$\dot{x}_i(t) = f_i(\boldsymbol{x}(t), \boldsymbol{u}(t)),\ \forall i = 1, ..., 12,$$

where x_i is the i-th component of $\boldsymbol{x}$ and $f_i(\cdot)$ is the scalar i-th component of the vector function $f(\cdot)$.

Given the quadrotor model, we can formalize the maneuver regulation task that we want to solve. In the approach we propose in this paper, we can decouple the task into two parts: (i) generation of a desired, or nominal, (state-control) trajectory, and (ii) regulation of the corresponding desired maneuver.

Although in this paper we focus on the regulation sub-task (ii), we briefly describe the whole task. Usually, to accomplish a complex mission, the quadrotor is required to follow a given three-dimensional path and, maybe, satisfy some (soft) constraints on the orientation (e.g., because it has to traverse a narrow passageway).

The trajectory generation sub-task (i) is the following. We suppose that a suitable output curve is assigned by the mission to satisfy some geometric constraints. A reasonable choice of output curve for the quadrotor is $\boldsymbol{y}_\xi(t) = (p_{1\xi}(t)\ p_{2\xi}(t)\ p_{3\xi}(t))^T$, $t \geq 0$. The use of the subscript ξ will be clear in the next lines. We say that an output curve $\boldsymbol{y}_\xi(\cdot)$ is admissible, if there exists a state-control trajectory $\boldsymbol{\xi} = (\boldsymbol{x}_\xi(\cdot), \boldsymbol{u}_\xi(\cdot))$, such that

$$\dot{\boldsymbol{x}}_\xi(t) = f(\boldsymbol{x}_\xi(t), \boldsymbol{u}_\xi(t)), \quad \boldsymbol{y}_\xi(t) = h(\boldsymbol{x}_\xi(t))$$

for all $t \geq 0$, $\|\dot{\boldsymbol{y}}_\xi(\cdot)\|$ is bounded away from zero and $\|\ddot{\boldsymbol{y}}_\xi(\cdot)\|$ is bounded.

State-control trajectories for the standard quadrotor model used in this paper can be generated by exploiting its differential flatness. For more general models or in case state and input constraints need to be explicitly taken into account in the desire trajectory generation, nonlinear optimal control based trajectory-generation techniques, as the ones developed in [15, 16], may be used.

Given a desired (state-control) trajectory $(\boldsymbol{x}_\xi(\cdot), \boldsymbol{u}_\xi(\cdot))$, we define a maneuver $[\boldsymbol{x}_\xi, \boldsymbol{u}_\xi]$ as the set of all the trajectories equivalent under time translation to the trajectory $(\boldsymbol{x}_\xi(\cdot), \boldsymbol{u}_\xi(\cdot))$. Given a trajectory $(\tilde{\boldsymbol{x}}_\xi(\cdot), \tilde{\boldsymbol{u}}_\xi(\cdot))$ of $f(\cdot)$, we say that it is equivalent to $(\boldsymbol{x}_\xi(\cdot), \boldsymbol{u}_\xi(\cdot))$ under time translation if and only if there exists $\Delta \in \mathbb{R}$, such that $\tilde{\boldsymbol{x}}_\xi(t) = \boldsymbol{x}_\xi(t+\Delta)$ and $\tilde{\boldsymbol{u}}_\xi(t) = \boldsymbol{u}_\xi(t+\Delta),\ \forall t \geq 0$.

We are now ready to formally define the maneuver regulation problem.

Maneuver regulation problem Let an output $\boldsymbol{y}_\xi(\cdot)$ and an associated maneuver $[\boldsymbol{x}_\xi, \boldsymbol{u}_\xi]$ of the quadrotor be given. Find a feedback control law $u = k(x; [\boldsymbol{x}_\xi, \boldsymbol{u}_\xi])$ that exponentially stabilizes the maneuver $[\boldsymbol{x}_\xi, \boldsymbol{u}_\xi]$, i.e. such that there exist $\Delta, k, \lambda > 0$ such that

$$\lim_{t\to\infty} ||x(t) - \boldsymbol{x}_\xi(t+\Delta)|| \leq k e^{-\lambda t}.$$

Remark 1. From the above definition it is clear why the maneuver regulation task protects from dangerous situations as the one described in Fig. 1. Indeed, for the task to be accomplished the quadrotor is not required to catch up a reference on the desired maneuver. In the specific scenario, when the quadrotor regains control after the disturbance has ceased, it can track a time-translated trajectory belonging to the same maneuver, whose initial condition is close to the current quadrotor state, thus avoiding to fall into the obstacle. □

3 Transverse linearization based maneuver regulation controller

In order to exponentially stabilize a desired maneuver, rather than a trajectory, we seek a controller scheduled by points on the desired output, rather than by the time. As a first step, we rewrite the quadrotor dynamics in terms of a longitudinal and a transverse dynamics.

The longitudinal dynamics describes the evolution of the system position along the curve, while the transverse dynamics describes the evolution of a suitable error between the actual state and the desired one at the current longitudinal coordinate.

We start by parametrizing the (admissible) output curve in terms of the arc-length $\sigma_\xi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$, defined as

$$\sigma_\xi(t) = \int_0^t \sqrt{\dot{p}_{1\xi}^2(\tau) + \dot{p}_{2\xi}^2(\tau) + \dot{p}_{3\xi}^2(\tau)}\, d\tau,\ \forall t \geq 0. \tag{20}$$

Defining the inverse of $\sigma_\xi(\cdot)$ as the function $\bar{t}_\xi : \mathbb{R}_0^+ \to \mathbb{R}_0^+$, the output, parametrized using the arc-length, is $\bar{\boldsymbol{y}}_\xi(\sigma) = \boldsymbol{y}_\xi(\bar{t}_\xi(\sigma))$. We shall denote the σ-parametrized curves with a bar, and the derivatives with respect to the coordinate σ with a prime symbol.

We parameterize the position of the quadrotor center of mass $\boldsymbol{y} = (p_1\ p_2\ p_3)^T$ in a tubular neighborhood of $\bar{\boldsymbol{y}}_\xi(\cdot)$ using a set of coordinates $(s, w_1, w_2) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}$. In order to do this, we construct a locally invertible function $\phi : \mathbb{R}^3 \to \mathbb{R} \times \mathbb{R} \times \mathbb{R}$ such that

$$(s, w_1, w_2) = \phi(\boldsymbol{y}), \tag{21}$$

with $\phi_i, i \in \{1,2,3\}$, the i-th scalar component of the vector function $\phi(\cdot)$, and such that $\phi(\bar{\boldsymbol{y}}_\xi(s)) = (s,0,0)$. We choose s to be the arc-length identifying the point on the desired path at minimum distance from the quadrotor center of mass. The coordinates w_1 and w_2 express the distance between the quadrotor center of mass and the point on $\bar{\boldsymbol{y}}_\xi(\cdot)$ identified by s.

Consistently, the first component of $\phi(\cdot)$ is defined as

$$\phi_1(\boldsymbol{y}) := \arg\min_{\sigma \in \mathbb{R}} \|\boldsymbol{y} - \bar{\boldsymbol{y}}_\xi(\sigma)\|^2.$$

The minimizing s is unique provided that $\bar{\boldsymbol{y}}_\xi(\cdot)$ is locally a non-intersecting C^2 curve with non-vanishing $\bar{\boldsymbol{y}}'_\xi(\cdot)$ and that $\boldsymbol{y}(t)$ is close to $\bar{\boldsymbol{y}}_\xi(\cdot)$ for all t. In order to construct the other components of $\phi(\cdot)$, let us consider the Serret-Frenet frame, which origin has $\bar{\boldsymbol{y}}_\xi(s)$ as coordinates, defined by the basis $\{\vec{\mathbf{t}}(s), \vec{\mathbf{n}}(s), \vec{\mathbf{b}}(s)\}$. The vectors $\vec{\mathbf{t}}, \vec{\mathbf{n}}, \vec{\mathbf{b}}$ are respectively the tangent, normal and bi-normal vectors and they are defined, with components in the inertial frame, as $\bar{\boldsymbol{t}}(s) := \bar{\boldsymbol{y}}'_\xi(s)$, $\bar{\boldsymbol{n}}(s) := \bar{\boldsymbol{y}}''_\xi(s)/\bar{k}(s)$, $\bar{\boldsymbol{b}}(s) := \bar{\boldsymbol{t}}(s) \times \bar{\boldsymbol{n}}(s)$, where $\bar{k}(s) := \|\bar{\boldsymbol{y}}''_\xi(s)\|$ is the curvature of $\bar{\boldsymbol{y}}_\xi(\cdot)$ at s. The position of the quadrotor center of mass $\boldsymbol{y}$ can be written as

$$\boldsymbol{y} = \bar{\boldsymbol{y}}_\xi(s) + \bar{R}_{SF}(s)\boldsymbol{d}, \tag{22}$$

where the rotation matrix $\bar{R}_{SF} = (\bar{\boldsymbol{t}}\ \bar{\boldsymbol{n}}\ \bar{\boldsymbol{b}})$ maps vectors with components in the Serret-Frenet frame into vectors with components in the world inertial frame and $\mathbf{d} := (0\ w_1\ w_2)^T$ is the vector from the origin of the Serret-Frenet frame to the origin of the body frame, with components in the Serret-Frenet frame. Using the equation (22), we define the remaining components of the function $\phi(\cdot)$ as

$$\phi_2(\boldsymbol{y}) := \bar{\boldsymbol{n}}(s)^T(\boldsymbol{y} - \bar{\boldsymbol{y}}_\xi(s)), \tag{23}$$

$$\phi_3(\boldsymbol{y}) := \bar{\boldsymbol{b}}(s)^T(\boldsymbol{y} - \bar{\boldsymbol{y}}_\xi(s)). \tag{24}$$

With the function $\phi(\cdot)$ in hand, we can provide a change of coordinates from the state $\boldsymbol{x}$ to the coordinates $(s, \boldsymbol{w})$, where $s \in \mathbb{R}$ is the *longitudinal coordinate* and $\boldsymbol{w} \in \mathbb{R}^{11}$ is the vector of *transverse coordinates* with i-th transverse coordinate, $i \in \{1, \ldots, 11\}$, denoted by w_i. The longitudinal and transverse coordinates as function

of the system state are defined as

$$\begin{aligned} s &:= \phi_1(h(\boldsymbol{x})), \quad & w_3 &:= \varphi - \bar{\varphi}_\xi(s), \quad & w_6 &:= v_1 - \bar{v}_{1\xi}(s), \quad & w_9 &:= p - \bar{p}_\xi(s), \\ w_1 &:= \phi_2(h(\boldsymbol{x})), \quad & w_4 &:= \theta - \bar{\theta}_\xi(s), \quad & w_7 &:= v_2 - \bar{v}_{2\xi}(s), \quad & w_{10} &:= q - \bar{q}_\xi(s), \\ w_2 &:= \phi_3(h(\boldsymbol{x})), \quad & w_5 &:= \psi - \bar{\psi}_\xi(s), \quad & w_8 &:= v_3 - \bar{v}_{3\xi}(s), \quad & w_{11} &:= r - \bar{r}_\xi(s). \end{aligned}$$

Furthermore, we define

$$\begin{aligned} \boldsymbol{u}_w &:= \boldsymbol{u} - \bar{\boldsymbol{u}}_\xi(s), \quad & \gamma_{1w} &:= \gamma_1 - \bar{\gamma}_{1\xi}(s), \\ f_w &:= f - \bar{f}_\xi(s), \quad & \gamma_{2w} &:= \gamma_2 - \bar{\gamma}_{2\xi}(s), \\ & & \gamma_{3w} &:= \gamma_3 - \bar{\gamma}_{3\xi}(s). \end{aligned} \tag{25}$$

The change of coordinates is used in order to write the standard quadrotor system (6-17) in $(s, \boldsymbol{w})$ coordinates. Deriving (22) with respect to time we have

$$\dot{\boldsymbol{y}} = \bar{\boldsymbol{y}}'_\xi \dot{s} + \bar{R}'_{SF} \dot{s}\, \mathbf{d} + \bar{R}_{SF} \dot{\boldsymbol{d}}. \tag{26}$$

Further on, for simplicity of notation, the dependency by s is omitted and all the bar terms are evaluated with respect to s. The term on the left side of equation (26) is

$$\dot{\boldsymbol{y}} = R\boldsymbol{v}, \tag{27}$$

according to the definition of $\boldsymbol{y}$ and the equation (1). The first term on the right side is

$$\bar{\boldsymbol{y}}'_\xi \dot{s} = \bar{R}_{SF} [\dot{s}\, 0\, 0]^T \tag{28}$$

according to the definition of the tangent vector $\bar{\boldsymbol{t}}$. Furthermore we have

$$\bar{R}'_{SF} \dot{s} = \bar{R}_{SF} \begin{bmatrix} 0 & -\bar{k}\dot{s} & 0 \\ \bar{k}\dot{s} & 0 & -\bar{\tau}\dot{s} \\ 0 & \bar{\tau}\dot{s} & 0 \end{bmatrix}, \tag{29}$$

where $\bar{\tau} := \bar{\boldsymbol{n}}\bar{\boldsymbol{b}}'$ is the torsion of $\bar{\boldsymbol{y}}_\xi(\cdot)$ at s. The expression (29) can be derived from the Serret-Frenet formulas [18] $\bar{\boldsymbol{t}}' = \bar{k}\bar{\boldsymbol{n}}$, $\bar{\boldsymbol{n}}' = -\bar{k}\bar{\boldsymbol{t}} + \bar{\tau}\bar{\boldsymbol{b}}$, $\bar{\boldsymbol{b}}' = -\bar{\tau}\bar{\boldsymbol{n}}$, using the definition of $\bar{R}_{SF}$. Multiplying both sizes of the equation (26) times $\bar{R}^T_{SF}$, using (27), (28), (29) and the coordinate transformation from $\boldsymbol{x}$ to $(s, \boldsymbol{w})$, we obtain

$$\begin{aligned} \dot{s} &= \bar{\boldsymbol{t}}^T R\boldsymbol{v}/(1 - \bar{k}w_1), \\ \dot{w}_1 &= \bar{\boldsymbol{n}}^T R\boldsymbol{v} + \bar{\tau}\dot{s}w_2, \\ \dot{w}_2 &= \bar{\boldsymbol{b}}^T R\boldsymbol{v} - \bar{\tau}\dot{s}w_1, \end{aligned} \tag{30}$$

where $\boldsymbol{v} = [(w_6 + \bar{v}_{1\xi})\ (w_7 + \bar{v}_{2\xi})\ (w_8 + \bar{v}_{3\xi})]^T$ and R is given by (5) with $\varphi = w_3 + \bar{\varphi}_\xi$, $\theta = w_4 + \bar{\theta}_\xi$ and $\psi = w_5 + \bar{\psi}_\xi$. The equations (30) are equivalent to the equations (6-8) but they are only functions of the longitudinal coordinate s, the transverse coordinates w_1, w_2 and the state trajectory $\bar{\boldsymbol{x}}_\xi(\cdot)$. Furthermore, the equations (9-17) can be expressed as function of s, $\boldsymbol{w}$ and $\bar{\boldsymbol{x}}_\xi(\cdot)$, using the coordinate transformation

from $\boldsymbol{x}$ to $(s, \boldsymbol{w})$ and equations (25). Taking the time derivative of $x_i = w_{i-1} + \bar{x}_{i\xi}$, the equations $\dot{x}_i = f_i(\boldsymbol{x}, \boldsymbol{u})$ can be written as

$$\dot{w}_{i-1} = f_i(\bar{\boldsymbol{x}}_{\xi r} + \boldsymbol{w}_r, \boldsymbol{u}_w + \bar{\boldsymbol{u}}_\xi) - \bar{x}'_{i\xi}\dot{s}, \quad \forall i = 4, ...12, \tag{31}$$

where $\bar{\boldsymbol{x}}_{\xi r}$ is the vector containing from the 4-th to the 12-th component of $\bar{\boldsymbol{x}}_\xi$ and $\boldsymbol{w}_r$ is the vector containing from the 3-rd to the 11-th component of $\boldsymbol{w}$.

The system (30-31) can be expressed in a form such that the independent variable is the longitudinal coordinate s, rather than the time t. We parameterize the transverse coordinates using s, i.e., $w_i(t) = \bar{w}_i$, $\forall i = 1, ..., 11$, and by using the chain rule, we compute the time derivatives of the transverse coordinates as $\dot{w}_i = \bar{w}'_i\dot{s}$. Thus, equations (30-31) can be written as the *transverse dynamic system*

$$\begin{aligned} \bar{w}'_1 &= (\bar{\boldsymbol{n}}^T R\boldsymbol{v})/\dot{s} + \bar{\tau}\bar{w}_2, \\ \bar{w}'_2 &= (\bar{\boldsymbol{b}}^T R\boldsymbol{v})/\dot{s} - \bar{\tau}\bar{w}_1, \\ \bar{w}'_{i-1} &= f_i(\bar{\boldsymbol{x}}_{\xi r} + \bar{\boldsymbol{w}}_r), \bar{\boldsymbol{u}}_w + \bar{\boldsymbol{u}}_\xi)/\dot{s} - \bar{x}'_{i\xi}, \quad \forall i = 4, ...12, \end{aligned} \tag{32}$$

where $\dot{s} = \bar{\boldsymbol{t}}^T R\boldsymbol{v}/(1 - \bar{k}\bar{w}_1)$, $\boldsymbol{v} = [(\bar{w}_6 + \bar{v}_{1\xi})\ (\bar{w}_7 + \bar{v}_{2\xi})\ (\bar{w}_8 + \bar{v}_{3\xi})]^T$ and R is given by (5) with $\varphi = \bar{w}_3 + \bar{\varphi}_\xi$, $\theta = \bar{w}_4 + \bar{\theta}_\xi$ and $\psi = \bar{w}_5 + \bar{\psi}_\xi$.

The system (32) is a nonlinear control system with state $\bar{\boldsymbol{w}} \in \mathbb{R}^{11}$ and input $\bar{\boldsymbol{u}}_w \in \mathbb{R}^4$, for which s-varing control laws can be developed in order to regulate the transverse state $\bar{\boldsymbol{w}}$ to zero. We compute such control law solving a linear quadratic regulator (LQR) problem.

Let us consider the transverse linearization

$$\bar{\boldsymbol{w}}' = \bar{A}_T(s)\bar{\boldsymbol{w}} + \bar{B}_T(s)\bar{\boldsymbol{u}}_w,$$

i.e., the linearization of the transverse dynamic system (32). We design a feedback matrix $\bar{K}(\cdot)$ that asymptotically stabilizes the transverse linearization by solving a linear quadratic regulator problem. If the transverse linearization is exponentially stabilized by an s-varing linear state feedback, $\bar{\boldsymbol{u}}_w = -\bar{K}(s)\bar{\boldsymbol{w}}$, then the nonlinear feedback

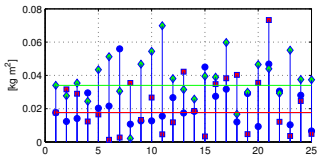
$$\boldsymbol{u} = \bar{\boldsymbol{u}}_\xi(s) - \bar{K}(s)\boldsymbol{w} \tag{33}$$

exponentially stabilizes the maneuver $[\boldsymbol{x}_\xi]$ for (6-17) [14].

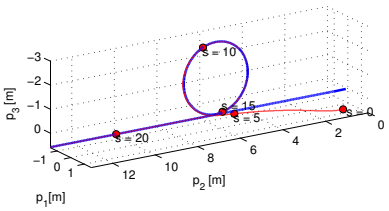
The controller in (33) can be rewritten by exploiting the dependence of s from the system output ($s = \phi_1(\boldsymbol{y})$) as

$$\boldsymbol{u} = \bar{\boldsymbol{u}}_\xi(\phi_1(\boldsymbol{y})) - \bar{K}(\phi_1(\boldsymbol{y}))\boldsymbol{w}.$$

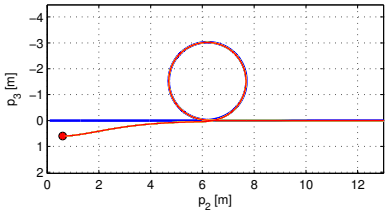
The above expression highlights the *nonlinear feedback* structure of the proposed maneuver regulation controller. In particular, the feedforward term and the feedback matrix are nonlinear functions of the system state (the output portion).



(a) Inertial parameter sets



(b) Desired vs. actual curve



(c) Desired vs. actual curves

Fig. 2: (a) Values of the inertia matrices, used to perform the Monte Carlo simulation. The blue, red and green markers represent the terms j_x, j_y, y_z, respectively. (b), (c) Closed loop trajectories for the different values of inertia matrices.

4 Numerical validation

This section provides two groups of simulations that are meant to be a preliminary step for experimental tests. The *first group* objective is to test the performance and robustness in the stabilization of an aggressive maneuver in presence of error in the initial conditions and uncertainty in the inertia matrix parameters; while the goal of the *second group* is to test the robustness on the same maneuver with respect to noise and saturation of the four propellers forces.

For both groups, we choose a *barrel roll* maneuver in the $p_2 - p_3$ plane with a constant velocity profile for which the quadrotor is subject to "significant" accelerations. The quadrotor center of mass is required to move along a circle of diameter $d = 3\ m$ at a speed $v = 8\ m/s$. The desired maneuver is designed so that the roll angle φ goes from 0 deg to 360 deg (i.e., the quadrotor performs a complete flip). As a reference for the nominal inertial parameters of the model we used the data-sheet values of the customized MK-Quadro[2] that is described in [19], i.e., $m = 0.749\ kg$, $j_x = 0.0176\ kg\ m^2$, $j_y = 0.0177\ kg\ m^2$, $j_z = 0.034\ kg\ m^2$.

In the *first group* of simulations we conduct a Monte Carlo analysis aimed at testing the robustness against imperfect initial conditions and inertial parameters. The desired barrel roll path is represented with a blue curve in Figs. 2b and 2c. In all the $25 + 1$ simulations of this first group the quadrotor starts $1.04\ m$ distant from the nominal initial position, outside the $p_2 - p_3$ plane. In particular, each component of the position vector is perturbed respectively of $0.60\ m$ from the nominal one. We opted for using the nominal mass of the MK-Quadro in every simulation. The rea-

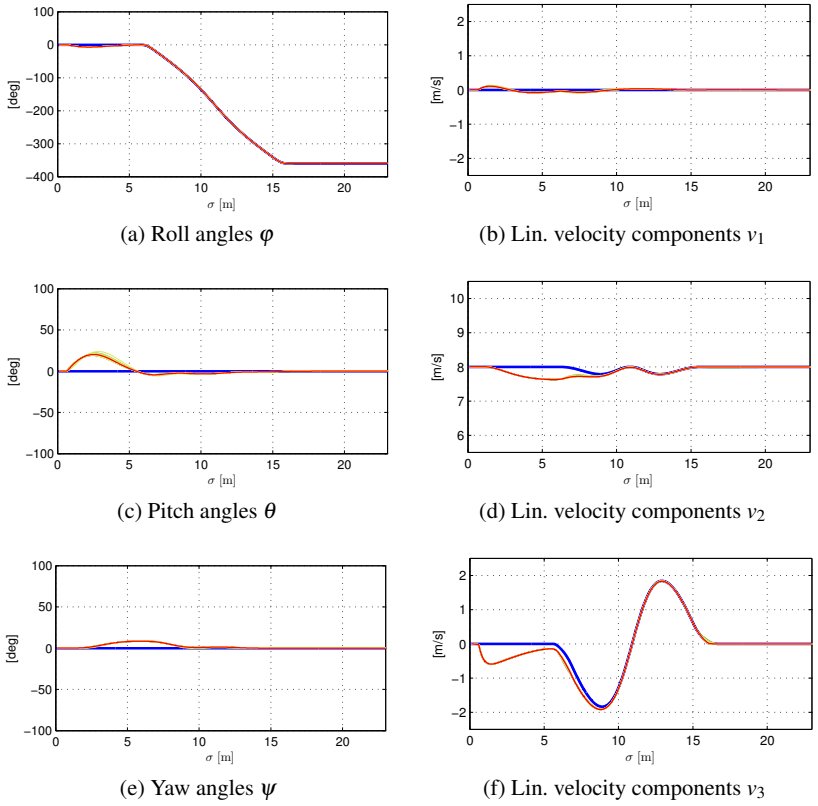
[2] www.mikrokopter.de

son is that in real applications the quadrotor mass is easily computable with high accuracy and reliability either using a scale or through simple hovering calibration. On the other hand, calibration of the inertia matrix requires a sophisticated measurement equipment and can easy become outdated if the geometrical configuration of the internal masses changes over time, e.g., whenever the battery is mounted in a slightly different place. In real applications, this is equivalent to have a random noise on the nominal parameters. In one simulation we used the nominal values for j_x, j_y, j_z while in each of the other 25 we used a different quadrotor model with set of values for j_x, j_y, j_z obtained by randomly perturbing the nominal values according to a normal distribution where the standard deviation is $\frac{0.0102}{3}$ $kg\ m^2$ (i.e., 95% of the samples are within $(j_x \pm j_z, j_y \pm j_z, j_z \pm j_z)$ being j_z the maximum among j_x, j_y, j_z). Notice that we still consider negligible the off-diagonal terms of the matrix w.r.t. the diagonal ones, which is a reasonable assumption in practice. The 25 normally distributed samples of the 3 diagonal inertia coefficients are represented in Fig. 2a, where the blue, red and green markers represent the perturbed terms j_x, j_y, j_z, respectively. The nominal values are instead represented with 3 horizontal lines (notice that j_x and j_y are almost overlapping). In each of the $25+1$ simulations the maneuver regulation controller defined in (33) employes always the same $\bar{\boldsymbol{u}}_\xi(s)$, and $\bar{K}(s)$ that are computed using the nominal values of j_x, j_y, j_z. Notice that also the desired maneuver is designed considering the nominal value. In this way we can test the robustness of the controller against model parameter uncertainties. Finally, the diagonal weight matrices employed in the LQR problem in order to compute the feedback matrix $\bar{K}(\cdot)$ are $Q = \mathrm{diag}(100,100,9,9,9,10,10,10,3,3,3)$ and $R = \mathrm{diag}(0.1,0.1,1,0.1)$ referred to the state $\bar{\boldsymbol{w}}$ and the input $\bar{\boldsymbol{u}}_w$, respectively.

Figure 2c presents the projection, on the $p_2 - p_3$ plane, of the desired position trajectory (thick blue curve), the closed loop position trajectory starting from the perturbed initial condition with nominal parameters (thick red curve), and the closed loop trajectories starting from the perturbed initial condition for all the 25 sets of perturbed inertial parameters (thin colored curve), which are actually indistinguishable from the red curve. After the initial transient, all the curves quickly converge to the desired curve right before the actual barrel roll maneuver starts. It is worth noting that even if the desired task is defined in the $p_2 - p_3$ plane, the perturbed initial condition is taken outside this plane and thus the resulting maneuver involves the whole dynamics, as can be seen from Fig. 2b.

In Figs. 3a, 3c, and 3e we show the desired roll, pith and yaw angles compared to ones achieved during the 26 closed loop trajectories (here we use the same color convention described before). The three components of the (body frame) linear velocities are instead plotted in Figs. 3b, 3e, and 3f, still with the same color convention as before. All the quantities are plotted with respect to the arc-length σ. As already pointed out, the maneuver regulation, as opposed to classical path following techniques, is able to satisfy additional constraints like assigning orientation and linear velocity. In fact, the plots clearly show how, after the transient phase these additional constraints are fulfilled. Robustness with respect to the perturbation of the model parameters is also manifest from the fact that all the 26 closed-loop trajectories are almost indistinguishable, despite the fact that the inertial parameters differ from the ones used in the controller design.

Control inputs of the first group of simulations are shown in Fig. 4. Differently from the plots presented so far, these plots show a different behavior of the torques



(a) Roll angles φ (b) Lin. velocity components v_1

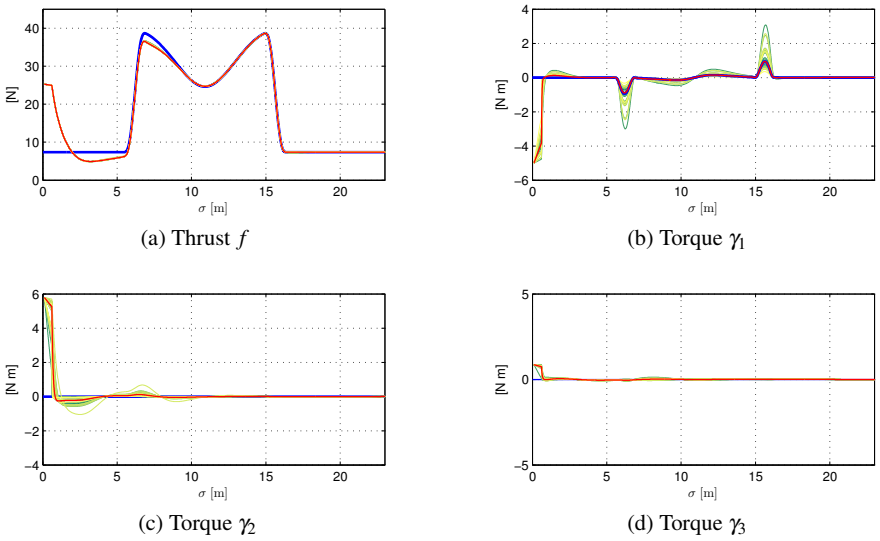
(c) Pitch angles θ (d) Lin. velocity components v_2

(e) Yaw angles ψ (f) Lin. velocity components v_3

Fig. 3: Attitude angles and velocity components for the Monte Carlo simulation. Thick blue lines are used for the desired maneuver. Thick red lines are used for closed-loop maneuver with nominal parameters. All the other lines refer to the closed-loop maneuvers with perturbed parameters.

γ_1,γ_2, and γ_3 across the 26 closed-loop simulations of this group. This happens because the proposed controller automatically adapts to the perturbation of the inertial parameters in order to track the desired trajectory. For example, the torque γ_2 in Fig. 4c, which is responsible for the flip of the roll, is either smaller or larger in order to automatically compensate for smaller or larger values of the perturbed j_y, respectively. Notice how all the thrusts show an almost indistinguishable behavior instead. This is manly due to the fact that the mass is not perturbed, for the reasons explained before, across the 26 simulations.

In the *second group* of simulations we show how the proposed maneuver regulation controller is able to ensure good performances also in case of input noise and saturations, which are common in real physical systems. In order to further validate the controller with a realistic simulator we apply the controller on SwarmSimX [20], a quadrotor simulator whose fidelity has been already validated several times with respect to the test-bed described in [19] (see, e.g., [20] for a numerical comparison between the real quadrotor and the simulated one). We encour-



(a) Thrust f (b) Torque γ_1

(c) Torque γ_2 (d) Torque γ_3

Fig. 4: Control inputs for the Monte Carlo simulation. Thick blue lines are used for the desired maneuver. Same color convention as in Fig. 3

age the reader to watch the video attachments corresponding to these simulations at antoniofranchi.com/videos/qcmaneuver.html.

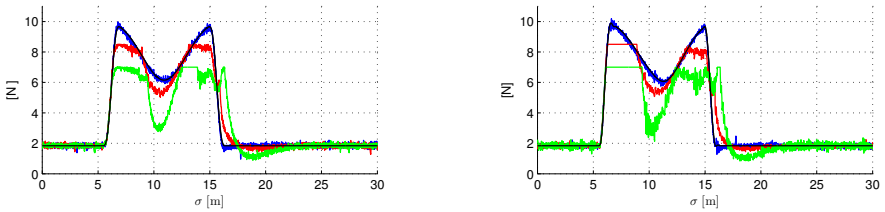
The desired maneuver is the same barrel roll trajectory that has been used before but with a negligible initial error, since we want to evaluate here the sole effect of input noise and saturation. As it happens in the real world, we apply the actual noise and saturations on each single force produced by the 4 propellers. Denoting those forces with $f_1^p, f_2^p, f_3^p, f_4^p$ we have the well-known relation [3]:

$$\begin{bmatrix} f \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -l & 0 & l \\ l & 0 & -l & 0 \\ -k & k & -k & k \end{bmatrix} \begin{bmatrix} f_1^p \\ f_2^p \\ f_3^p \\ f_4^p \end{bmatrix} \tag{34}$$

where l is the distance of the propeller from the center of mass and k is a suitable constant that mainly depends on the propeller shape.

The model parameters used in the simulations are the one of the customized MK-Quadro, i.e., mass and inertia matrix described before and l=0.30 m. We apply a Gaussian noise with standard deviation equal to 0.1 N to each propeller. Three different cases are then considered: (i) no saturation; (ii) 8.5 N saturation, and (iii) 7 N saturation. These values are consistent with the capabilities of the MK-Quadro. Considering that (see, e.g., Fig. 4a) the barrel roll maneuver needs a 40 N peak of total thrust (i.e., see (34), 10 N per propeller on average), the chosen saturations correspond to have about 15% and 30% less than the needed average total thrust.

Figure 5 shows the nominal force (thin black line) and actual forces commanded to each of the four propellers in the three saturation cases (blue, red, and green colored lines, respectively). Presence of noise and saturation is clear form the plots.



(a) Force f_1^p (b) Force f_2^p

(c) Force f_3^p (d) Force f_4^p

Fig. 5: Effect of the noise and saturation on the force produced by each propeller.

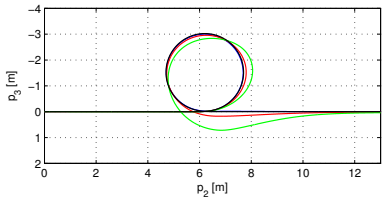
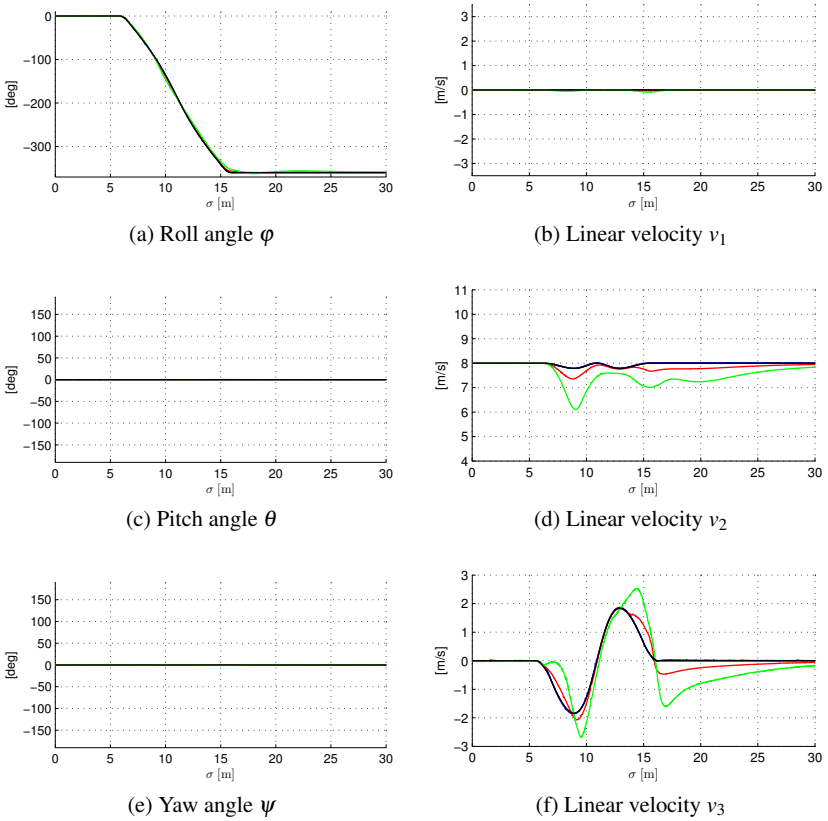


Fig. 6: Effect of the force/torque saturation of each propeller on the actual trajectory.

Figure 6 shows the projection on the $p_2 - p_3$ plane of the desired position maneuver (black line) and the three closed-loop actual position trajectories obtained for the three saturation cases (same color coding as the propeller forces). As expected, deformation of the trajectory is more pronounced for the 7 N saturation value than the 8.5 N case. However, the controller is capable of maintaining a stable behavior in all the four cases. A similar trend is observable on the evolution of the three velocities components shown in Figs. 7b,7d, and 7f as a function of the arc-length σ.

On the contrary, the achieved closed loop attitude angles are almost indistinguishable from the desired one, see Figs. 7a,7c, and 7e. The different behavior can be explained in following way. The maximum achievable γ_1 when the force of the propeller is saturated at 7 N is, from (34), equal to $2l \cdot 7$ N=4.2 Nm which is higher than the maximum nominal torque γ_1 needed by the maneuver, which is visible in Fig. 4b (blue plot). Same discussion holds for the other torques. This shows how the controller is able to let the saturation only affect the quantities whose degradation is unavoidable (position and velocity in this case).



(a) Roll angle φ (b) Linear velocity v_1

(c) Pitch angle θ (d) Linear velocity v_2

(e) Yaw angle ψ (f) Linear velocity v_3

Fig. 7: Effect of the force/torque saturation on the attitude angles and linear velocity.

5 Conclusion

In this paper we have developed an LQR based maneuver regulation controller in order to make a quadrotor UAV perform three dimensional aggressive maneuvers. This controller overcome the drawbacks that affect both path following and trajectory tracking. We have shown how our maneuver regulation controller is robust with respect to the uncertainty of model parameters and input saturations. As a preliminary step to test the controller on the real quadrotor we have performed numerical computations on a quadrotor simulator with good physical fidelity. Given the promising performances of the controller we plan to test the controller on a real quadrotor platform.

Acknowledgements

The work of G. Notarstefano and S. Spedicato is partially supported by the project SOCIAL-ROBOTS under the program “5 per mille per la ricerca”.

References

[1] V. Mistler, A. Benallegue, and N.K. M'Sirdi. Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback. In *Proc. IEEE International Workshop on Robot and Human Interactive Communication*, pages 586–593, Bordeaux, Paris, Sep. 2001.

[2] T. Madani and A. Benallegue. Control of a Quadrotor Mini-Helicopter via Full State Backstepping Technique. In *Proc. IEEE Conference on Decision and Control*, pages 1515–1520, San Diego, CA, Dec. 2006.

[3] Taeyoung Lee, M. Leoky, and N.H. McClamroch. Geometric tracking control of a quadrotor UAV on SE(3). In *Proc. IEEE Conference on Decision and Control*, pages 5420–5425, Atlanta, GA, Dec. 2010.

[4] Daniel Mellinger and Vijay Kumar. Minimum snap trajectory generation and control for quadrotors. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2520–2525. IEEE, 2011.

[5] V. Gavrilets, E. Frazzoli, B. Mettler, M. Piedmonte, and E. Feron. Aggressive maneuvering of small autonomous helicopters: A human-centered approach. *International Journal of Robotics Research*, 20(10):795–807, 2001.

[6] Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Y. Ng. An application of reinforcement learning to aerobatic helicopter flight. In *Advances in Neural Information Processing Systems 19*, 2007.

[7] Rong Xu and U. Ozguner. Sliding mode control of a quadrotor helicopter. In *Proc. IEEE Conference on Decision and Control*, pages 4957–4962, San Diego, CA, Dec. 2006.

[8] Guilherme V. Raffo, Manuel G. Ortega, and Francisco R. Rubio. Path Tracking of a UAV via an Underactuated Control Strategy. *European Journal of Control*, 17(2):194 – 213, 2011.

[9] Ian D. Cowling, Oleg A. Yakimenko, James F. Whidborne, and Alastair K. Cooke. A Prototype of an Autonomous Controller for a Quadrotor UAV. In *European Control Conference*, 2007.

[10] D. Cabecinhas, R. Cunha, and C. Silvestre. Rotorcraft path following control for extended flight envelope coverage. In *Proc. IEEE Conference on Decision and Control held jointly with the Chinese Control Conference*, pages 3460–3465, Shanghai, Dec. 2009.

[11] A. Roza and M. Maggiore. Path following controller for a quadrotor helicopter. In *American Control Conference*, pages 4655–4660, Montreal, QC, June 2012.

[12] A. Akhtar, S.L. Waslander, and C. Nielsen. Path following for a quadrotor using dynamic extension and transverse feedback linearization. In *Proc. IEEE Conference on Decision and Control*, pages 3551–3556, Maui, HI, Dec. 2012.

[13] Daniel Mellinger, Nathan Michael, and Vijay Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *The International Journal of Robotics Research*, 31(5):664–674, 2012.

[14] A. Saccon, J. Hauser, and A. Beghi. A Virtual Rider for Motorcycles: Maneuver Regulation of a Multi-Body Vehicle Model. *IEEE Transactions on Control Systems Technology*, 21(2):332–346, 2013.

[15] G. Notarstefano, J. Hauser, and R. Frezza. Computing feasible trajectories for control-constrained systems: the PVTOL example. In *Nolcos*, Pretoria, SA, August 2007.

[16] G. Notarstefano and J. Hauser. Modeling and Dynamic Exploration of a Tilt-Rotor VTOL aircraft. In *Nolcos*, Bologna, Italy, September 2010.

[17] N. Guenard, T. Hamel, and R. Mahony. A practical visual servo control for a unmanned aerial vehicle. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1342–1348, Roma, April 2007.

[18] March Peter Setterlund. *Geometric-based Spatial Path Planning*. UMI Microform, 2008.

[19] A. Franchi, C. Secchi, M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano. Shared control: Balancing autonomy and human assistance with a group of quadrotor UAVs. *IEEE Robotics & Automation Magazine, Special Issue on Aerial Robotics and the Quadrotor Platform*, 19(3):57–68, 2012.

[20] J. Lächele, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano. SwarmSimX: Real-time simulation environment for multi-robot systems. In I. Noda, N. Ando, D. Brugali, and J.J. Kuffner, editors, *3rd Int. Conf. on Simulation, Modeling, and Programming for Autonomous Robots*, volume 7628 of *Lecture Notes in Computer Science*, pages 375–387. Springer, 2012.