Attitude control of multirotor UAVs: cascade P/PID vs PI-like architecture

G. Bressan, D. Invernizzi, S. Panza and M. Lovera

Abstract This paper addresses the attitude control problem for multirotor Unmanned Aerial Vehicles with the aim of comparing two nonlinear control architectures. The first controller is based on a nonlinear cascade design with a P/PID-like structure while the second one is a PI-like nonlinear controller that has been proposed to tackle the attitude tracking problem for rigid bodies. First, a general model for the attitude dynamics of multirotor UAVs is recalled. Then, the considered controllers are reviewed on both theoretical and practical aspects, focusing on their stabilizing properties, implementation and tuning issues. Finally, the control laws are systematically tuned by applying structured H_∞ synthesis to the linearized closed-loop dynamics obtained by referring to an identified single axis model of a lightweight quadrotor.

1 Introduction

Attitude control in multirotor Unmanned Aerial Vehicles (UAVs) is of fundamental importance since their flying qualities depend significantly on the performance and stabilizing properties of their attitude controllers, the design and tuning of which must be carried out in a sensible way. Depending on the application, requirements may vary and different modeling and control law design tools have to be considered. If one is concerned with applications such as inspection, surveillance, mapping, video and photography then linear modeling and control design methods are suitable and allow to handle stringent performance requirements in a systematic way. On the other hand, when considering maneuvers involving extreme changes in attitude, linear controllers may have significant deficiencies and yield poor performance or, even worse, fail to stabilize the vehicle.

G. Bressan, D. Invernizzi, S. Panza, M. Lovera,
Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano, Italy. e-mail: gabriele1.bressan@mail.polimi.it, {davide.invernizzi, simone.panza, marco.lovera}@polimi.it

For aggressive maneuvering flight, nonlinear models and control design methods are needed, in particular, to ensure global tracking properties. To this end, minimal parametrizations of the configuration manifold in which the attitude motion evolves, namely, $\mathrm{SO}(3)$, have to be avoided since no minimal parametrization (like Euler angles) is free of singularities, which is the reason why most attitude controllers proposed in the literature are based on unit quaternions [1, 2, 3] or directly on elements of $\mathrm{SO}(3)$ [4, 5, 6] (geometric controllers). While quaternion-based controllers require only four parameters to compute attitude errors, they need special care as they doubly cover $\mathrm{SO}(3)$. On the other hand, geometric-based controllers account by design for topological obstructions of $\mathrm{SO}(3)$ and are computationally efficient since they rely on matrix operations to compute attitude errors. As per the controller architecture, standard PI control laws, which can be employed only in Euclidean spaces, have been recently extended to $\mathrm{SO}(3)$ and Lie groups in general [7, 8]. By including feed-forward terms, perfect tracking of a desired attitude motion can be guaranteed under ideal conditions, *i.e.*, when the nominal dynamics is exactly the one of a rigid body with a perfectly known inertia matrix and only constant disturbances are present. On the other hand, cascade-based designs have successfully flown onboard multirotor UAVs and are implemented in popular open-source autopilots [9]. This design exploits the cascade structure of the attitude motion in which the attitude kinematics evolves on a nonlinear manifold but is free of disturbances and in which the attitude dynamics is nonlinear and affected by disturbances but evolves on a linear manifold. Although the typical implementation of this design does not guarantee perfect attitude tracking even in ideal conditions, it is structurally more robust and has some practical advantages over the PI one.

In this work we first review the dynamical model of a multirotor UAV with coplanar propellers and then present in detail a PI-like geometric controller and a geometric version of a standard P/PID control architecture. In the analysis, emphasis is placed on theoretical as well as practical aspects, specifically implementation and tuning issues. The tuning phase is particularly challenging for nonlinear control laws that involve several parameters without a clear understanding of their contribution on the system response. Typically, gains are tuned on the basis of simple simulation models and then adjusted by trial and error on the real platform. Recently, a systematic method based on data-driven approaches has been presented to tune linear attitude controllers for multirotor UAVs [10, 11]. The development of a tuning approach for nonlinear attitude control laws is more challenging and has never been presented to the best of our knowledge. The approach presented in this work exploits linearized closed-loop systems obtained by referring to identified dynamical models. Then, structured H_∞ [12, 13] is used to tune the controller parameters. Such approach proved to be of great practical usefulness and has been widely employed in aerospace applications, in particular on rotorcraft [14, 15, 16, 17], since it guarantees robustness against model uncertainty and requirements encoded in the frequency domain can be enforced to achieve tight performance in near hovering conditions. In the last section of the paper some remarks about the expected performance of the two control architectures are reported by analyzing the corresponding complementary sensitivity functions.

2 The attitude dynamical model of multirotor UAVs

A multirotor UAV is an aerial vehicle made by a central body and n arms, each of which carries a propeller group. The propeller groups consist of a motor and a propeller and are in charge of producing the wrench (force and torque) required to control the motion of the UAV (Figure 1). The dynamics of the central body evolves on the product manifold $\mathrm{SO}(3)\times\mathbb{R}^3$, where $\mathrm{SO}(3) := \{R \in \mathbb{R}^{3\times 3} : R^T R = I_3, \det(R) = 1\}$ denotes the Special Orthogonal group of order three. The configuration of the UAV can be identified with the motion of a body-fixed frame $\mathscr{F}_B := (O_B, \{b_1, b_2, b_3\})$ with respect to a reference frame $\mathscr{F}_I := (O_I, \{e_1, e_2, e_3\})$, where b_i and e_i are unit vectors forming orthogonal triads and $O_B, O_I \in \mathbb{R}^3$ are the origins of the body and reference frame, respectively. The position vector from O_I to O_B is $x \in \mathbb{R}^3$, resolved in $\mathscr{F}_I$. We assume that the reference frame is an inertial one and, for convenience, that its axes coincide with the standard basis of $\mathbb{R}^3$, *i.e.*, $e_1 = (1,0,0)$, $e_2 = (0,1,0)$, $e_3 = (0,0,1)$. We denote $R := [\,b_1\ b_2\ b_3\,] \in \mathrm{SO}(3)$ the rotation matrix describing the attitude of the UAV, where b_i are the body axes resolved in $\mathscr{F}_I$.

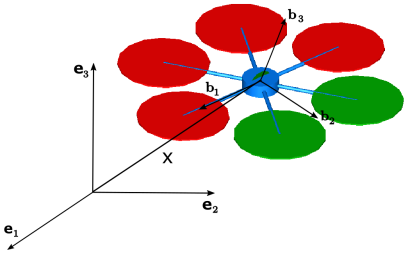

Fig. 1 Example of a multirotor UAV and definition of the frames.

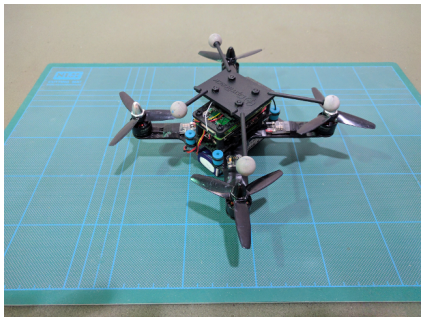

Fig. 2 Quadrotor used for the tests.

By referring to the body frame, the attitude motion of a UAV is described by the following set of equations:

$$\dot{R} = R\hat{\omega} \tag{1}$$

$$J\dot{\omega} = -\omega \times J\omega + \tau_c + \tau_e(R, x, \omega, v, \omega_r, \dot{\omega}_r, t) \tag{2}$$

where $J \in \mathbb{R}^{3\times 3}_{>0}$ is the inertia matrix with respect to O_B, $\omega \in \mathbb{R}^3$ is the body angular velocity, $v \in \mathbb{R}^3$ is the translational velocity, $\tau_c, \tau_e \in \mathbb{R}^3$ are the control and disturbance torque, respectively, and $\omega_r := (\omega_{r_1}, \ldots, \omega_{r_n}) \in (\mathbb{R}_{\geq 0})^n$ contains the angular rates of the propellers. Herein, the *hat* map $\hat{\cdot} : \mathbb{R}^3 \to \mathfrak{so}(3)$, given by

$$\omega \mapsto \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \tag{3}$$

defines an isomorphism between $\mathbb{R}^3$ and the vector space of third-order skew-symmetric matrices, *i.e.*, $\mathfrak{so}(3) := \{W \in \mathbb{R}^{3\times 3} : W = -W^T\}$. The corresponding inverse is the *vee* map $(\cdot)^\vee : \mathfrak{so}(3) \to \mathbb{R}^3$. Given $\hat{\omega} \in \mathfrak{so}(3)$, one has $\hat{\omega} y = \omega \times y$, $\forall y \in \mathbb{R}^3$, where $\times$ is the cross product.

The control torque delivered by the propellers at O_B is given by:

$$\tau_c := \sum_{i=1}^{n} x_{br_i} \times f_{p_i} + \tau_{p_i} \tag{4}$$

where f_{p_i} and $\tau_{p_i} \in \mathbb{R}^3$ are, respectively, the force and torque delivered by the *i-th* rotor and $x_{br_i} \in \mathbb{R}^3$ is the position vector from the airframe origin to the hub of the *i-th* rotor disk, all resolved in $\mathscr{F}_B$. In this work we rely on the quadratic aerodynamic model [18] and assume that the force and torque delivered by the propellers are orthogonal to the plane described by b_1, b_2 (coplanar configuration), *i.e.*,

$$f_{p_i} := k_f \omega_{r_i}^2 e_3 \tag{5}$$

$$\tau_{p_i} := -\varepsilon_i k_\tau \omega_{r_i}^2 e_3, \tag{6}$$

where ω_{r_i} is the angular rate of the *i-th* rotor, $k_f, k_\tau \in \mathbb{R}_{>0}$ are the thrust and torque coefficient, respectively, which can be obtained experimentally in static conditions, and $\varepsilon_i \in \{-1, 1\}$ defines the rotation direction of the *i-th* rotor. Note that this model is valid when considering small deviations from the hovering condition [18] although it has been successfully used even in highly acrobatic maneuvering, as documented by several experimental works [5, 19]. By defining

$$T_i := k_f \omega_{r_i}^2 \qquad i = 1, \ldots, n, \tag{7}$$

the propeller wrench can be written as $f_{p_i} = T_i e_3$ and $\tau_{p_i} = -\varepsilon_i \sigma T_i e_3$, where $\sigma := k_\tau / k_f$ is a positive constant. Then, when assuming that there are sufficiently fast low-level controllers to track any desired angular rate ω_{r_i}, $(T_1, \ldots, T_n) \in (\mathbb{R}_{\geq 0})^n$ can be considered as the input for control design. Under the assumption that the rotor hubs are placed equidistantly from the center of mass, each making an angle $\gamma_i \in [0, 2\pi)$ with respect to b_1, the control torque (4) can be written as a map $(T_1, \ldots, T_n) \mapsto \tau_c(T_1, \ldots, T_n)$, given by the following expression:

$$\tau_c(T_1, \ldots, T_n) := \sum_{i=1}^{n} \left(x_{br_i} \times T_i e_3 - \varepsilon_i \sigma T_i e_3 \right) \tag{8}$$

where $x_{br_i} := \ell R_{e_3}(\gamma_i) e_1, R_{e_3}(\gamma_i) \in \mathrm{SO}(3)$ describes a rotation of an angle $\gamma_i \in [0, 2\pi)$ about e_3 and defines the orientation of the $i-th$ arm frame with respect to $\mathscr{F}_B$. Finally, when referring to the coplanar rotors configuration, the total force is delivered by the propellers only along the positive direction of the third body axis (b_3), according to the following map:

$$T_c(T_1, \ldots, T_n) := \sum_{i=1}^{n} T_i. \tag{9}$$

The control thrust $T_c := f_c^T e_3$ is provided by a control loop when the position/altitude flight mode is engaged or directly by the pilot during manual flight. Note that T_i appears linearly in (8)-(9). Hence, the mapping can be compactly written in matrix form:

$$\begin{bmatrix} T_c \\ \tau_c \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 1 \\ \ell \sin(\gamma_1) & \cdots & \ell \sin(\gamma_n) \\ -\ell \cos(\gamma_1) & \cdots & -\ell \cos(\gamma_n) \\ -\varepsilon_1 \sigma & \cdots & -\varepsilon_n \sigma \end{bmatrix} \begin{bmatrix} T_1 \\ \vdots \\ T_n \end{bmatrix}. \tag{10}$$

It is worth remarking that since we have employed an approximated model in (5)-(6), an additional disturbance torque, dependent on the input, should be included in equation (2) to account for the induced modeling errors. The remaining part of the external torque τ_e is related to the aerodynamic interaction of the UAV main body and arms with air, to the center of mass offset with respect to O_B and to inertial and gyroscopic coupling terms related to the propellers. Therefore, a general model for the attitude dynamics would be described by:

$$J\dot{\omega} = -\omega \times J\omega + \tau_c(u) + \tau_e(R, x, \omega, v, u, \omega_r(u), \dot{\omega}_r(u), t) \tag{11}$$

where the input $u := (T_1, \ldots, T_n)$ has been introduced for compactness. Note that ω_r and $\dot{\omega}_r$ depend upon input u and that in general, the exact expression of τ_e is difficult to find.

3 The attitude tracking problem

For a sufficiently large thrust, it is always possible to obtain feasible propellers thrusts T_i (*i.e.*, positives) for any desired torque τ_c according to (10) [20]. This motivates the common assumption that the attitude dynamics for a multirotor UAV is fully actuated. As can be seen from equation (10), the coplanar multirotor UAV is an underactuated system: no force can be instantaneously delivered in the plane spanned by b_1, b_2. For controlling the position of these platforms, the standard approach exploits the assumed full actuation of the attitude subsystem to stabilize the position error dynamics. In practice, a control law for τ_c is designed to tilt the body vector b_3 in the direction of the force required for position tracking. At the same time the magnitude of the control force T_c is adjusted to match the magnitude of the required force and then, input u is computed by inverting (10). Therefore, the design of control laws for attitude tracking becomes a fundamental ingredient to ensure the stability of the overall system. If one is not interested in position tracking, the fully actuated rotational dynamics can be exploited to perform arbitrary rotational maneuvers. In this scenario the thrust magnitude is assigned to track a desired altitude motion or to make (10) always invertible, which is the case of the experiments presented later on in this work. In both flight conditions, the attitude tracking problem can be formalized as follows.

Problem 1. Consider the attitude dynamics in equations (1),(11). Given a trajectory $t \mapsto (R_d(t), \omega_d(t)) \in \mathrm{SO}(3) \times \mathbb{R}^3$, where $t \mapsto \omega_d(t) := (R_d^T(t)\dot{R}_d(t))^\vee$ is a continuously differentiable and bounded function of time, find a control torque τ_c such that $(R, \omega) \to (R_d, \omega_d)$ as $t \to \infty$ when assuming that full state information is available.

Recent results [21] in trajectory tracking for underactuated UAVs show that the attitude dynamics should be made at least asymptotically stable to guarantee asymptotic stability of the closed-loop system including the position error dynamics. To this end, the majority of works on multirotor UAV design control laws either on the linearized dynamics, to seek high performance and robustness in near hovering conditions, or on the full nonlinear dynamics, simplified by treating as constant or even by neglecting the disturbance torque, when large attitude maneuvers are planned. In this work we present two control laws that tackle the attitude tracking problem by accounting for nonlinearities as well as disturbances. Both control laws are developed in a geometric setting without resorting to parametrizations of $\mathrm{SO}(3)$. The first design is based on a cascade architecture [22] whereas the second one is a slightly modified version of the geometric PI-like control law proposed in [7].

3.1 Geometric cascade P/PID-like architecture

The cascade strategy is motivated by the structure of the equations (1), (11) when treating τ_e as an exogenous disturbance. Under this assumption, the attitude dynamics (11) is independent from the kinematics one (1). Therefore, the angular velocity ω can be considered as a virtual input to track the desired attitude $t \mapsto R_d(t)$. Then, the torque τ_c can be designed according to different control strategies to track the virtual angular velocity, relying on the full actuation assumption. For instance, one can pursue a back-stepping or a feedback linearization approach. While back-stepping requires higher order derivatives [23] of the desired reference and perfect knowledge of the dynamics for exact tracking, linear control designs can be used to enforce satisfactory tracking performance as well as good disturbance rejection capabilities. For the latter case, a popular choice is a PID law, which stands out for its simplicity, and is the one considered in this work. As mentioned above, in the cascade strategy the outer loop is in charge of providing a reference angular velocity to the inner loop. To achieve this, a quaternion based control law was proposed in [24], together with a switching strategy to handle the unwinding phenomenon. This law guarantees global tracking (when only the attitude kinematics is considered) but is discontinuous and may be subjected to chattering [3]. Instead, in this work we consider a smooth geometric control law that exploits configuration errors directly defined on $\mathrm{SO}(3)$. Such control law can only almost globally track any desired attitude due to the topological obstructions of $\mathrm{SO}(3)$ (the set of initial conditions converging to undesired equilibria is of zero measure, in the sense of Lebesgue, with respect to the manifold $\mathrm{SO}(3) \times \mathbb{R}^3$). This result, however, is valid only in ideal conditions without disturbances on the dynamics and does not pose

difficulties in practice. Moreover, the control law provides a continuous reference to the inner loop which is a much more desirable property.

In this work the following geometric cascade control law is proposed (see also Figure 3):

$$\omega_v := -e_R(R_e) \tag{12}$$

$$\tau_c(s) := PI(s)\,(\omega_v - \omega) - D(s)\omega, \tag{13}$$

where $R_e := R_d^T R \in \mathrm{SO}(3)$, $e_R := \frac{1}{2}\left(K_R R_e - R_e^T K_R\right)^\vee$ is the left trivialized derivative of $\Psi(R_e) := \frac{1}{2}\mathrm{tr}(K_R(I - R_e))$, a continuously differentiable, positive definite function on $\mathrm{SO}(3)$, $PI(s) := K_p + K_i\frac{1}{s}$ and $D(s) := K_d\frac{s}{1+\frac{s}{N}}$ are transfer functions on the Laplace domain defining, respectively, a proportional-integral and (filtered) derivative action. Herein, K_R, K_p, K_i and $K_d \in \mathbb{R}^{3\times 3}$ are positive definite diagonal matrices while $N \in \mathbb{R}_{>0}$ is the derivative filter constant; K_{x_i} refers to the i-th element on the diagonal of the generic matrix K_x.

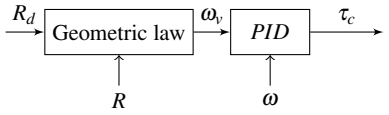

Fig. 3 Cascade attitude control loops.

For small attitude errors with respect to hovering, namely, when $R \approx I_3 + \hat{\theta}$, where $\theta \in \mathbb{R}^3$ is a vector containing small angles, or when the attitude motion is mainly occurring about one axis, the closed-loop system can be approximated as:

$$s\theta = \omega \tag{14}$$

$$sJ\omega = PI(s)\tilde{K}_R(\theta_d - \theta) - PID(s)\omega + \tau_e \tag{15}$$

in which

$$\tilde{K}_R := \begin{bmatrix} \frac{K_{R_2}+K_{R_3}}{2} & 0 & 0 \\ 0 & \frac{K_{R_1}+K_{R_3}}{2} & 0 \\ 0 & 0 & \frac{K_{R_1}+K_{R_2}}{2} \end{bmatrix} \tag{16}$$

and $PID(s) := PI(s) + D(s)$. The block diagram of the linearized control architecture is depicted in Figure 4.

Remark 1. The control law (12)-(13) does not solve Problem 1, *i.e.*, it does not guarantee perfect tracking, even in ideal conditions when the dynamics evolves according to $J\dot{\omega} + \omega \times J\omega = \tau_c + \tau_e$ with τ_e constant and J perfectly known. Nonetheless, it can be tuned to achieve tight performance, in particular, when identified models $G_{\omega_i}(s)$ from τ_{c_i} to ω_i are available and used instead of the simple integrator model of (15). In this case the inner-closed-loop function (from reference angular rate ω_{v_i}

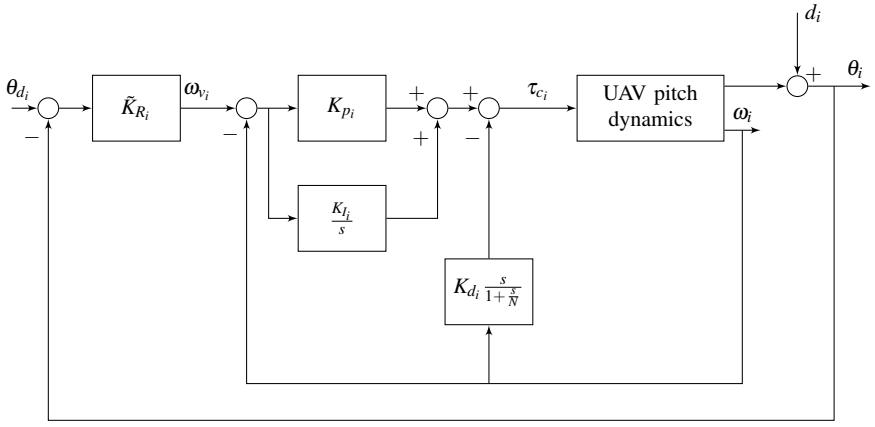

Fig. 4 Linearized cascade architecture (single axis).

to ω_i) reads:

$$s\theta_i = \omega_i \tag{17}$$

$$\frac{\omega_i}{\omega_{v_i}} = \frac{G_{\omega_i}(s)PI(s)}{1+G_{\omega_i}(s)PID(s)} = T_{\omega_i}(s) \tag{18}$$

and the outer-closed-loop function, also referred to as the complementary sensitivity function, reads

$$\frac{\theta_i}{\theta_{d_i}} = \frac{\frac{1}{s}\tilde{K}_{R_i}T_{\omega_i}(s)}{1+\frac{1}{s}\tilde{K}_{R_i}T_{\omega_i}(s)} = T_{\theta_i}(s) \tag{19}$$

and the sensitivity function (which can be interpreted as the closed-loop transfer function from d_i to θ_i, see Figure 4) is:

$$S_{\theta_i}(s) = 1 - T_{\theta_i}(s). \tag{20}$$

At this point, one can either tune the inner loop gains first and then select the gain $\tilde{K}_{R_i}$, according to the cascade structure in (17)-(18), or directly work with the sensitivity and complementary sensitivity function in equations (20) and (19). In this work the latter approach has been followed and structured H_∞ synthesis has been used to tune the gains.

3.2 Geometric PI-like architecture

The second law that we consider is inspired by [7]. It is derived by means of Lyapunov arguments by referring to the dynamics (1), (11) in which the disturbance torque is assumed to be unknown but constant. Several version of this control law

have been proposed and experimentally validated [7, 25]. In this paper we propose the following PI-like geometric control law:

$$\tau_c := -e_R(R_e) - K_\omega e_\omega - K_I e_I + JR_e^T \dot{\omega}_d + (R_e^T \omega_d) \times JR_e^T \omega_d \tag{21}$$

$$\dot{e}_I := e_R(R_e) + K_{\omega_I} e_\omega \tag{22}$$

where $e_R(R_e)$ is defined as for (12), $e_\omega := \omega - R_e^T \omega_d$ and K_ω, $K_I, K_{\omega_I} \in \mathbb{R}^{3\times 3}$ are diagonal positive definite matrices. By exploiting Lyapunov arguments [7], it is possible to show that there are diagonal gain matrices $K_R, K_\omega, K_I, K_{\omega_I}$ such that the closed-loop equilibrium $(R_e, e_\omega, e_I) = (I_3, 0, K_I^{-1}\tau_e)$ of the error dynamics

$$\dot{R}_e = R_e \hat{e}_\omega \tag{23}$$

$$J\dot{e}_\omega = -K_\omega e_\omega - e_R(R_e) + (Je_\omega + hR_e^T \omega_d) \times e_\omega - K_I e_I + \tau_e, \tag{24}$$

$$\dot{e}_I = e_R(R_e) + K_{\omega_I} e_\omega, \tag{25}$$

where $h := 2J - \mathrm{tr}(J)I_3$, is locally exponentially stable. Note that working directly with (21) is hard during the tuning phase as there is no clear understanding of the effect of changing the different gains. There is also an unconventional integration of the angular velocity error in (22), which, contrary to the linear case, is not just equivalent to increase the proportional gain in (21) unless small attitude errors are considered. Indeed, when the gyroscopic terms are neglected in (11) and the equations are linearized around the equilibrium point for $R_e \approx I_3 - \hat{\theta}_e$, $e_\omega \approx 0 - \omega_e$[1], the linearized closed-loop error system can be approximated to yield:

$$s\theta_e = \omega_e \tag{26}$$

$$sJ\omega_e = -K_\omega \omega_e - \tilde{K}_R \theta_e + K_I e_I - \tau_e \tag{27}$$

$$se_I = -\tilde{K}_R \theta_e - K_{\omega_I} \omega_e \tag{28}$$

in which $\tilde{K}_R$ is defined as in (16). To determine the local stability behavior of the closed-loop system, one can refer to the corresponding characteristic polynomial related to the i-th axis ($i = 1, 2, 3$), namely,

$$s^3 J_i + s^2 K_{\omega_i} + s(\tilde{K}_{R_i} + K_{I_i} K_{\omega_{I_i}}) + K_{I_i} \tilde{K}_{R_i} \tag{29}$$

in which τ_e can be neglected since it is an exogenous signal and does not contribute to the stability of the closed-loop system. The gains of the diagonal matrices are selected in order to make the polynomial in (29) a Hurwitz polynomial for $i = 1, 2, 3$.

A different path is to exploit the knowledge of identified models G_{ω_i}, as done for the cascade architecture, and tune the gains in order to achieve satisfactory performance when referring to the sensitivity function from d_i to θ_i, given by:

[1] Notice that the negative sign in the definition of angular velocity error is introduced to be consistent with the definition of $\theta_e = \theta_d - \theta$ and to have an error kinematic equation (26) with the same sign on both members.

$$S_{\theta_i}(s) = \frac{1}{1 + \frac{1}{s} T_{\omega_i}(s) C_{\theta_i}(s)} = \frac{\theta_i}{d_i} \tag{30}$$

where

$$T_{\omega_i}(s) = \frac{G_{\omega_i}(s)}{1 + G_{\omega_i}(s) C_{\omega_i}(s)} \tag{31}$$

$$C_{\omega_i}(s) = K_{\omega_i} + \frac{1}{s} K_{I_i} K_{\omega_{I_i}} \tag{32}$$

$$C_{\theta_i}(s) = \tilde{K}_{R_i} \left(1 + \frac{1}{s} K_{I_i} \right). \tag{33}$$

The block diagram of the linearized geometric PI-like architecture is depicted in Figure 5.

Remark 2. The control torque in equation (21) requires both the desired angular velocity and acceleration to be implemented. In the position tracking scenario depicted in Section 2 this means that the position controller should provide not only a desired attitude R_d but ω_d and $\dot{\omega}_d$ as well, whose analytical derivation requires continuously differentiable position trajectories up to the fourth order [21]. This not only makes the controller structure more complex but has potentially negative effects when only a roughly estimated inertia matrix is available. Furthermore, in case the vehicle is manually piloted, the pilot sends commands in terms of desired angles to the on-board controller and the corresponding angular velocity and acceleration must be somehow computed on-line, unlike the scenario in which both the attitude and its derivatives are provided, for instance, by a ground control computer. One option is to set $\omega_d, \dot{\omega}_d = 0$, namely, set-point tracking. This, however, represents a limitation in achieving the desired performance in terms of attitude response to reference, resulting in a sluggish response. In practice this work adopts the approach of [18, 26], where (21) is implemented with good approximation as $\tau_c \approx -e_R(R_e) - K_\omega e_\omega - K_I e_I$ which does not depend upon the UAV inertia and does not require the reference $\dot{\omega}_d$. Note that when the approximated torque is linearized, it yields exactly the closed-loop equation (27). Such control law still requires the desired angular velocity ω_d to compute e_ω, therefore a continuously differentiable signal with its derivative must be provided to the controller. This issue is addressed in detail in the next section.

3.3 Reference signal related issues

In Remark 2 it has been pointed out that the geometric PI-like architecture requires the desired attitude R_d and at least the corresponding angular velocity ω_d in the approximated case which neglects the feed-forward contribution in (21). Following up on the comments presented above, a smooth trajectory generator is therefore needed to provide the controllers with that information and to carry out a fair comparison

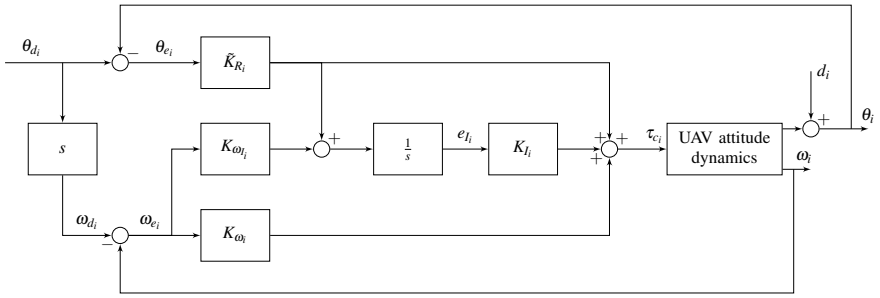

Fig. 5 Linearized geometric PI-like control architecture (single axis).

between the two architectures. The trajectory generator has been implemented onboard in the form of a filter so that the existing software architecture should not be modified: the idea is to pass the pilot/computer command through a suitable filter, so that a continuously differentiable signal and its derivative (at least) can be provided to the controllers. Since both the controllers are geometric, the filter should be developed directly on $\mathrm{SO}(3)$. To this end, we propose the geometric counterpart of the Euclidean second-order filter $\ddot{\theta}_d^f = -\omega_n^2(\theta_d^f - \theta_d) - 2\xi\omega_n\dot{\theta}_d^f$ $(\omega_n, \xi \in \mathbb{R}_{>0})$ which, in transfer function form, this can be written as:

$$\begin{bmatrix} \theta_d^f \\ \omega_d^f \end{bmatrix} = \begin{bmatrix} F_\theta(s) \\ sF_\theta(s) \end{bmatrix} \theta_d = F(s)\theta_d \qquad F_\theta(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}. \tag{34}$$

Specifically, the following geometric filter has been developed:

$$\dot{R}_d^f = R_d^f \hat{\omega}_d^f \tag{35}$$

$$\dot{\omega}_d^f = -\omega_n^2 e_R(R_e^f) - 2\xi\omega_n\omega_d^f \tag{36}$$

where $R_e^f := R_d^T R_d^f \in \mathrm{SO}(3)$ and $e_R(\cdot)$ is defined as (13). It can be verified that (34) is the linearized version of (35)-(36) for a small attitude motion $R_d(t) \approx I_3 + \hat{\theta}_d(t)$.

4 Tuning

The linearized version of the two control architectures presented in Sections 3.1 and 3.2 respectively were used to carry out tuning of the gains. This allows to resort to tuning methods for linear systems; in particular, the structured H_∞ approach is considered. The tuned gains are then plugged in the respective nonlinear control law architectures for validation in the time domain.

In this work, focus was put on the pitch axis of the vehicle. In order to ease the notation, the subscript i will be dropped, referring to the pitch axis (*i.e.*, $i = 2$).

4.1 Hardware set-up

In this work, the quadrotor depicted in Figure 2 was considered; it is a lightweight custom model with a distance of 160mm between opposite rotor axes and an overall take-off weight of about 230g. The relevant parameters are reported in Table 1.

Table 1 Main quadrotor parameters.

Variable	Value
Frame Config.	X
Propellers	Gemfan Bullnose 3055 3 blade
Arm length b	80 mm
Take-off weight m	230 g
Motors	QAV1306-3100kV brushless
ESC	ZTW Spider series 18A
Battery	Turnigy nano-tech 950mAh LIPO

The flight control unit is a Pixfalcon board, an open autopilot shield suitable for remotely controlled vehicles such as multirotors and fixed wing aircraft. It is equipped with a 3 axes accelerometer, a 3 axes gyroscope, a magnetometer and a pressure sensor. The firmware running on the Pixfalcon board is the open-source software PX4 Pro Autopilot. The firmware features attitude and position controllers and estimators, and has been customized to allow replacing the baseline attitude controller with a user-defined controller.

4.2 Model identification

A black-box model of the UAV pitch attitude dynamics was identified with the PBSID subspace model identification algorithm using closed-loop experimental data [27] and was used as the basis for control law tuning. The identified model, from the (adimensionalized) pitch moment input M to the pitch rate output q, is of order five and contains a time delay. Figure 6 shows the frequency response of the identified model against the estimate of the non-parametric frequency response function (computed accounting for the bias introduced by feedback in the closed-loop experiment); the coherence function indicates that the experimental data are valid in the frequency range $[10-100][rad/s]$, which is consistent with the expected attitude control bandwidth. Both the magnitude and phase of the identified model feature excellent fit to the nonparametric frequency response in this frequency range. The model was validated against flight data collected in a manually piloted experiment: Figure 7 shows the measured response (blue line) to the pitch attitude reference signal (red line), against the simulated closed-loop attitude angle response (black line) obtained with the identified model, showing a close match to the measured data.

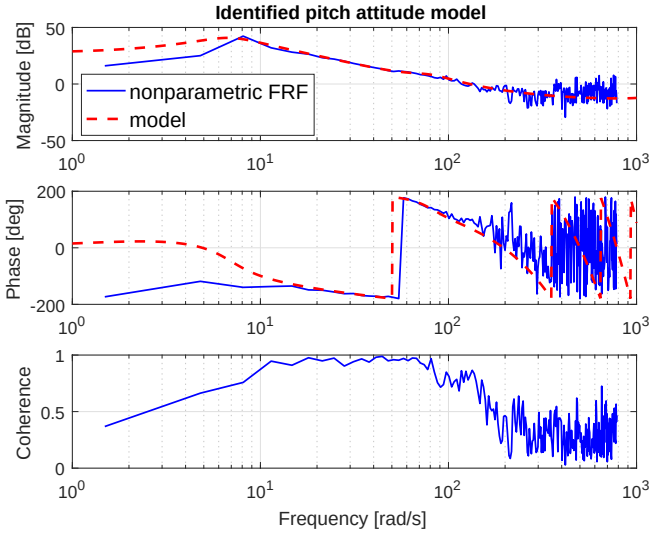

Fig. 6 Pitch attitude dynamics identified model.

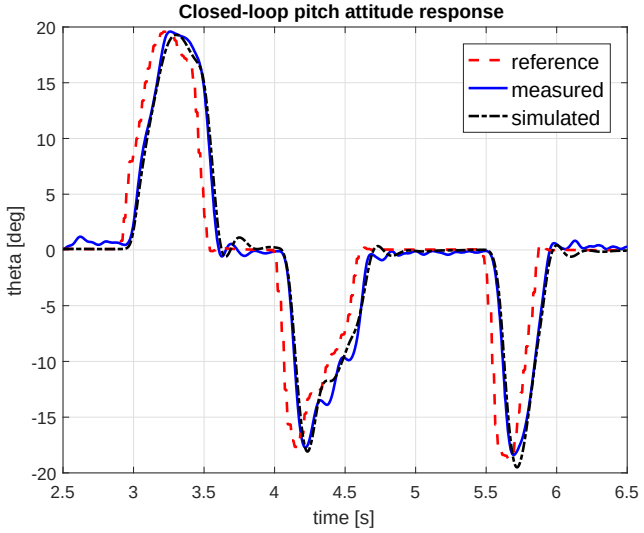

Fig. 7 Validation of identified model.

4.3 Requirements definition

The control requirements for structured H_∞ synthesis are stated in the form of weighting functions in the frequency domain. In particular, weighting functions are rational, stable, proper transfer functions. Two different requirements were taken into account:

- performance: the requirement on performance is defined as a weighting function on the attitude angle sensitivity, *i.e.*, the transfer function from the disturbance on the pitch angle d to θ (compare with equations (20) and (30) for the cascade and the geometric PI-like architectures, respectively);
- control moderation: the requirement on control moderation is defined as a weighting function on the control sensitivity, *i.e.*, the transfer function from d to M.

It was decided to characterize performance in terms of disturbance rejection [28, 14] (*i.e.*, breaking the loop in the output), rather than in terms of the response to the reference input (*i.e.*, breaking the loop in correspondence of the tracking error signal): indeed, the latter inherently depends both on the feedback and feed-forward properties of the system, while the former only depends on the feedback regulator. This approach allows to compare the performance of the two considered control architectures in terms only of the feedback properties (see also the discussion in Remark 2).

The weighting function on the sensitivity is defined as:

$$W_S(s) = k_S \frac{s + z_S}{s + p_S} \tag{37}$$

where the inverse of the frequency response magnitude $\left|W_S(j\omega)^{-1}\right|$ represents an upper bound on the sensitivity frequency response magnitude $|S(j\omega)|$; the parameters of the weighting function are chosen so as to enforce a low frequency constraint $\left|W_S(j0)^{-1}\right| = K_{LF}$, a high frequency constraint $\left|W_S(j\infty)^{-1}\right| = K_{HF}$, and a bandwidth constraint $\left|W_S(j\omega_{BW})^{-1}\right| = -3dB$. In particular, ω_{BW} represents the desired sensitivity bandwidth, also referred to as the disturbance rejection bandwidth (DRB), while K_{HF} represents a constraint on the peak of the sensitivity magnitude, also referred to as the disturbance rejection peak (DRP) [28].

Two different levels of performance were defined:

- a *high-bandwidth* requirement (H_{BW}), representing aggressive performance requirements, with a large bandwidth and allowing for a large sensitivity peak;
- a *low-bandwidth* requirement (L_{BW}), featuring a more stringent constraint on the sensitivity peak, on the other hand trading off a lower bandwidth.

The parameters of the corresponding weighting functions are reported in Table 2.

The control sensitivity weigthing function was chosen as follows:

$$W_R(s) = k_R \frac{s + z_R}{s + p_R} \tag{38}$$

Table 2 Sensitivity weighting function parameters.

	ω_{BW} [rad/s]	K_{HF} [dB]	K_{LF} [dB]	k_S	p_S	z_S
H_{BW}	9	8	-40	0.398	0.122	30.64
L_{BW}	2	3	-40	0.708	0.02445	3.454

with $k_R = 0.2$, $p_R = 23.26$ and $z_R = 10^{-4}$, in order to limit high frequency control action beyond the actuators' bandwidth.

4.4 Control law synthesis problem statement

Let ρ be the vector of tunable controller parameters, which for the cascade architecture is $\rho = \rho_C = \left[K_{P_i}, K_{I_i}, K_{D_i}, \tilde{K}_{R_i}\right]^T$ and for the geometric PI-like architecture is $\rho = \rho_G = \left[\tilde{K}_{R_i}, K_{\omega_i}, K_{I_i}, K_{\omega_{I_i}}\right]^T$, where $i = \{1,2,3\}$ indicates respectively the roll, pitch or yaw axis, and the control law tuning is carried out one axis at a time.

Let $S(s,\rho)$ be the sensitivity function and $R(s,\rho)$ the control sensitivity function, where the dependence on ρ is made explicit. Let $J_S(\rho)$ be the cost function related to the performance requirement, and let $J_R(\rho)$ be the cost function related to the control moderation requirement:

$$J_S(\rho) = \|W_S(s)S(s,\rho)\|_\infty \tag{39}$$

$$J_R(\rho) = \|W_R(s)R(s,\rho)\|_\infty . \tag{40}$$

The synthesis problem can be stated as an optimization problem:

$$\rho^* = \arg\min_\rho J_S(\rho) \tag{41}$$

$$\text{subject to} \tag{42}$$

$$J_R(\rho) \leq 1 \tag{43}$$

with ρ^* being the optimal value of the controller gain vector.

4.5 Numerical results

Three different control law gain sets were obtained for the cascade architecture:

- a *high-bandwidth* cascade control law, denoted as C_H, which is subject to the H_{BW} performance requirement defined in Section 4.3;
- a *low-bandwidth* cascade control law, denoted as C_L, subject to the L_{BW} performance requirement defined in Section 4.3;

- a *low-bandwidth* cascade control law with no derivative action in the inner loop, denoted as C_L^{D0}, which is subject to the same L_{BW} performance requirement as the C_L controller, but with the additional constraint $K_D = 0$.

Numerical optimization was carried out by means of the MATLAB `systune` routine. The gain values for the three cascade control laws are shown in Table 3, along with the achieved sensitivity bandwidth.

Table 3 Cascade control architecture gain values: pitch axis ($i = 2$).

Gain	C_H	C_L	C_L^{D0}
K_P	0.142	0.187	0.0646
K_I	0.287	0.362	0.107
K_D	0.00263	0.00344	0
$\tilde{K}_R$	12.5	3.30	3.10
ω_{BW} [rad/s]	10	2.98	2.26

Finally, gains for the geometric PI-like architecture were tuned according to the L_{BW} performance requirement and are stated in Table 4; this control law is denoted as G_L. Unlike the cascade architecture, it was not possible to obtain a solution achieving the H_{BW} requirements with the geometric PI-like control architecture, likely due to the absence of a derivative action on the angular rate.

Table 4 Geometric PI-like control architecture gain values: pitch axis ($i = 2$).

Gain	G_L
$\tilde{K}_R$	0.264
K_ω	0.0757
K_I	0.3
K_{ω_I}	0.0390
ω_{BW} [rad/s]	2.85

Figure 8 shows the sensitivity function magnitude of the system closed in loop with controllers C_L, C_L^{D0} and G_L, along with the inverse of the weighting function associated with the low performance requirement; in all the cases, the optimization routine is able to meet the constraint. Similar considerations hold true for the case of the C_H controller, not shown for brevity. The effect of the derivative action on angular rate can be appreciated in that the C_L controller achieves a significantly lower sensitivity magnitude peak with respect to the other two controllers (which are not provided with angular rate derivative action), and a larger bandwidth with respect to the required one.

Figure 9 shows the Bode magnitude plot of the complementary sensitivity function for the four controllers. It can be noticed that the C_L and C_L^{D0} controllers achieve a similar complementary sensitivity magnitude shape, while the C_H controller

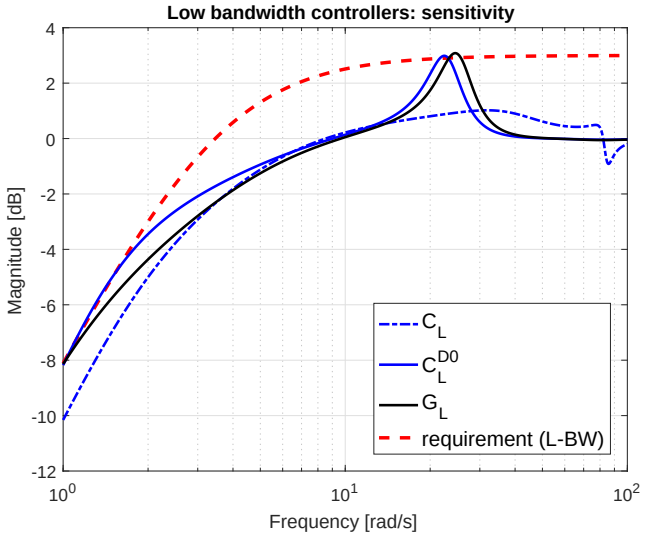

Fig. 8 Sensitivity function: low-bandwidth controllers.

achieves higher bandwidth to reference response, as expected; on the other hand, the complementary sensitivity for the G_L controller features a magnitude shape closer to the high-bandwith C_H controller rather than the cascade low-bandwith controllers C_L and C_L^{D0}, despite having been designed for low-bandwidth requirements. It is thus expected that the G_L controller achieves a fast response, though with some oscillations (due to the presence of a resonance peak). Furthermore, from the phase plot it can be noticed that G_L features the smallest phase lag among the four controllers.

The filter $F(s)$ of equation (34), described in Section 3.3, was appended upstream the geometric PI-like controller, and the part of filter $F_\theta(s)$ related to attitude angle reference was appended upstream the cascade controller; in this way, a fair comparison between the two control architectures can be carried out, in terms of response to piloted attitude angle reference. The step responses of the system closed in loop with the four controllers are shown in Figure 10, along with the response of the filter $F_\theta(s)$ to a step reference, which itself acts as a reference signal to be tracked by the four controllers.

5 Concluding remarks

A comparison of two nonlinear control architectures for UAV attitude control has been carried out by considering a cascade geometric P/PID-like architecture and a geometric PI-like controller. A systematic methodology for tuning the controller gains, based on structured H_∞, has been sketched and applied to the system closed in loop with the linearized version of the two controllers.

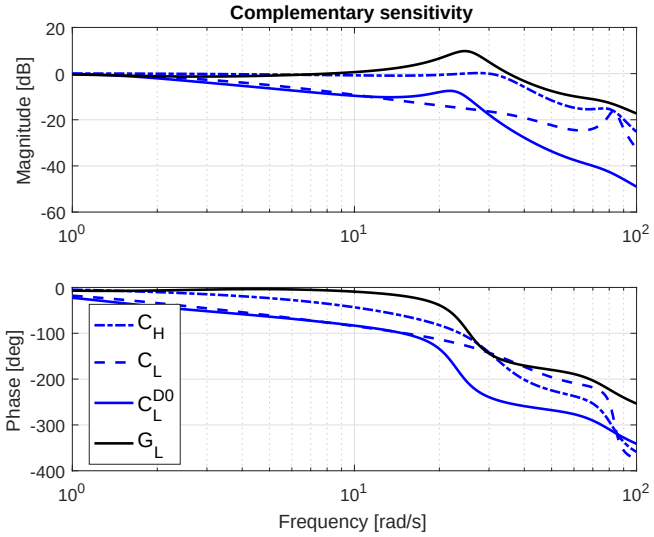

Fig. 9 Complementary sensitivity: comparison between controllers.

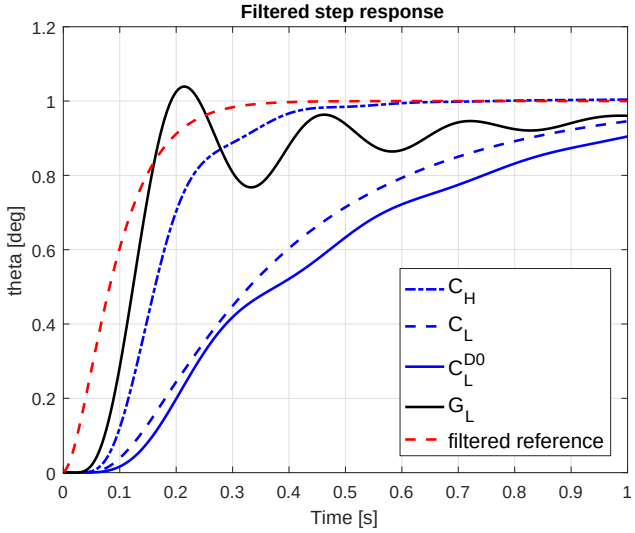

Fig. 10 Closed-loop step response to attitude angle reference.

The cascade controller was easily tuned to achieve a desired level of performance and a major advantage over the PI-like controller, in terms of architecture complexity, is that it requires only a desired attitude signal to be implemented on-board, which has two practical implications. First, when included in a hierarchical strategy for position tracking, it does not require the computation of a desired angular velocity and acceleration, which depends upon the snap of the desired position trajectory. Moreover, when manually piloted, the PI-like controller cannot exploit information about the desired angular velocity and acceleration, which significantly reduces the expected performance. A nonlinear trajectory generator filter, which could be easily implemented on-board, has been developed to mitigate such deficiency and to allow for a future experimental comparison. In this case, by inspecting the complementary sensitivity function obtained with the PI-like controller, it is expected that it should outperform the cascade architecture in terms of tracking.

Future work will be aimed to experimentally validate the preliminary conclusions drawn in this paper and to gather additional insights into the benefits and disadvantages of the two architectures.

References

1. A. Tayebi and S. McGilvray. Attitude stabilization of a four-rotor aerial robot. In *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, volume 2, pages 1216–1221 Vol.2, Dec 2004.
2. S. Zhao, W. Dong, and J. A. Farrell. Quaternion-based trajectory tracking control of VTOL-uavs using command filtered backstepping. In *Proc. American Control Conf*, pages 1018–1023, June 2013.
3. C. G. Mayhew, R. G. Sanfelice, and A. R. Teel. Quaternion-based hybrid control for robust global attitude tracking. *IEEE Transactions on Automatic Control*, 56(11):2555–2566, Nov 2011.
4. N. Chaturvedi, A. Sanyal, and N. McClamroch. Rigid-body attitude control. *IEEE Control Systems*, 31:30–51, 2011.
5. D. Mellinger, N. Michael, and V. Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. In *Experimental robotics*, pages 361–373. Springer, 2014.
6. T. Lee, M. Leok, and H. McClamroch. Geometric tracking control of a quadrotor UAV on SE(3). In *IEEE Conference on Decision and Control, Atlanta, USA*, 2010.
7. F. Goodarzi, D. Lee, and T. Lee. Geometric nonlinear PID control of a quadrotor UAV on se(3). In *Proc. European Control Conf. (ECC)*, pages 3845–3850, July 2013.
8. D. H. S. Maithripala and J. M. Berg. An intrinsic PID controller for mechanical systems on Lie groups. *Automatica*, 54(4):189–200, 2015.
9. L. Meier. Px4 Firmware. https://github.com/PX4/Firmware, 2008.
10. P. Panizza, D. Invernizzi, F. Riccardi, S. Formentin, and M. Lovera. Data-driven attitude control law design for a variable-pitch quadrotor. In *American Control Conference, Boston, USA*, 2016.
11. D. Invernizzi, P. Panizza, F. Riccardi, S. Formentin, and M. Lovera. Data-driven attitude control law of a variable-pitch quadrotor: a comparison study. *IFAC-PapersOnLine*, 49(17):236 – 241, 2016. 20th IFAC Symposium on Automatic Control in Aerospace.
12. P. Apkarian and D. Noll. Nonsmooth H_∞ synthesis. *IEEE Transactions on Automatic Control*, 51(1):71–86, 2006.
13. P. Apkarian, P. Gahinet, and C. Buhr. Multi-model, multi-objective tuning of fixed-structure controllers. In *European Control Conference (ECC)*, 2014.

14. S. Panza. *Structured flight control law design for helicopters and tiltrotors*. PhD thesis, Politecnico di Milano, Dipartimento di Scienze e Tecnologie Aerospaziali, 2018.
15. D. Navarro-Tapia, P. Simplício, A. Iannelli, and A. Marcos. Robust flare control design using structured-H_∞ synthesis: a civilian aircraft landing challenge. In *20th IFAC World Congress, Toulouse, France*, 2017.
16. D. Navarro-Tapia, A. Marcos, S. Bennani, and C. Roux. Structured H_∞ and linear parameter varying control design for the VEGA launch vehicle. In *7th European Conference for Aeronautics and Aerospace Sciences (EUCASS), Milan, Italy*, 2017.
17. J. Lesprier, J.-M. Biannic, and C. Roos. Nonlinear structured H_∞ controllers for parameter-dependent uncertain systems with application to aircraft landing. In *IEEE Conference on Control Applications, Antibes, France*, 2014.
18. R. Mahony, V. Kumar and P. Corke. Multirotor Aerial Vehicles: Modeling, Estimation and Control of Quadrotor. *IEEE Robotics & Automation Magazine*, 19(3):20–32, 2012.
19. D. Falanga, E. Mueggler, M. Faessler, and D. Scaramuzza. Aggressive quadrotor flight through narrow gaps with onboard sensing and computing using active vision. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5774–5781, May 2017.
20. M. Furci. *Mobile Robots Control and Path Planning Strategies*. PhD thesis, Alma Mater Studiorum Università di Bologna, Maggio 2016.
21. D. Invernizzi, M. Lovera, and L. Zaccarian. Geometric trajectory tracking with attitude planner for vectored-thrust VTOL UAVs. In *2018 Annual American Control Conference (ACC)*, pages 3609–3614, June 2018.
22. M.-D. Hua, T. Hamel, P. Morin, and C. Samson. Introduction to feedback control of underactuated VTOL vehicles: A review of basic control design ideas and principles. *IEEE Control Systems*, 33(1):61–75, 2013.
23. K. Rudin, M.-D. Hua, G. Ducard, and S. Bouabdallah. A robust attitude controller and its application to quadrotor helicopters. *IFAC Proceedings Volumes*, 44(1):10379 – 10384, 2011. 18th IFAC World Congress.
24. D. Brescianini, M. Hehn, and R. D'Andrea. Nonlinear quadrocopter attitude control. *Department of Mechanical and Process Engineering, ETHZ, Tech. Rep*, 2013.
25. D. Invernizzi, M. Giurato, P. Gattazzo, and M. Lovera. Full pose tracking for a tilt-arm quadrotor UAV. In *Proc. IEEE Conf. Control Technology and Applications (CCTA)*, pages 159–164, August 2018.
26. D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotor helicopters. In *IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China*, 2011.
27. G. van der Veen, J.-W. van Wingerden, M. Bergamasco, M. Lovera, and M. Verhaegen. Closed-loop subspace identification methods: an overview. *IET Control Theory and Applications*, 7(10):1339–1358, 2013.
28. T. Berger, C.M. Ivler, M.G. Berrios, M.B. Tischler, and D.G. Miller. Disturbance rejection handling qualities criteria for rotorcraft. In *72nd Annual Forum of the American Helicopter Society, West Palm Beach, USA*, 2016.