


## RESEARCH ARTICLE

# Optimal trajectory generation and robust flatness-based tracking control of quadrotors

Amine Abadi<sup>1,2</sup>  | Adnen El Amraoui<sup>3</sup> | Hassen Mekki<sup>1</sup> | Nacim Ramdani<sup>2</sup>

<sup>1</sup>Networks Objects Control and Communication Systems Laboratory, National Engineering School of Sousse, University of Sousse, Sousse, Tunisia

<sup>2</sup>PRISME EA 4229, INSA-CVL, University of Orléans, Orléans, France

<sup>3</sup>LGI2A EA 3926, University of Artois, Béthune, France

**Correspondence**

Amine Abadi, Networks Objects Control and Communication Systems Laboratory, National Engineering School of Sousse, University of Sousse, 4054 Sousse, Tunisia; or PRISME EA 4229, INSA-CVL, University of Orléans, F45072 Orléans, France.  
Email: amine.abadi@etu.univ-orleans.fr

**Summary**

This work proposes an optimal trajectory generation and a robust flatness-based tracking controller design to create a new performance guidance module for the quadrotor in dense indoor environments. The properties of the differential flatness, the B-spline, and the direct collocation method are exploited to convert the constrained optimization problem into a nonlinear programming one, which can be easily resolved by a classic solver. After that, the obtained optimal reference trajectory is applied to the dynamic quadrotor model and two different flatness-based controllers, namely, one based on feedback linearization and one based on feedforward linearization, are developed and compared to ensure the trajectory tracking despite the existence of disturbances and parametric uncertainties. Numerical simulation is executed to evaluate the proposed optimal trajectory generation approach and the robust tracking strategies. It turns out that the controller based on feedforward linearization outperforms the feedback linearization one in robustness and permits obtaining a performance guidance law for an uncertain quadrotor system.

**KEYWORDS**

B-spline, differential flatness, quadrotor, robust control, trajectory optimization

## 1 | INTRODUCTION

Over the last few decades, the quadrotor has become a mostly frequent drone in the literature due to its simple structure, economic cost, and ability to perform stationary flights. The quadrotor was primarily used in the military field such as sentinels and for the evaluation of sensitive areas. Recently, the field of application of these flying machines has become very varied. They can be used in industrial, civil, and emergency situations. In spite of these advantages, the quadrotor has a strongly nonlinear model with coupling multivariables. Thus, the optimal trajectory generation and the tracking problems of the quadrotor have become very complex. In the literature, a lot of numerical methods<sup>1</sup> have been utilized to solve the optimal trajectory generation problems for the quadrotor, which are classified into two kinds: indirect and direct.

Indirect methods consist in solving the problem of the two-point limit value arising from the Pontryagin's minimum principle. This method requires specific analytical calculations for each optimization trajectory example. Since the quadrotor has a nonlinear model, indirect methods have been used only for a few examples, ie, the one being examined by Hehn et al.<sup>2</sup> Direct methods are developed while avoiding the disadvantages of the two-point limit value resolution, by converting the path optimization problem into a nonlinear programming (NLP) problem, which can be solved using an NLP solver such as sparse nonlinear optimizer (SNOPT).<sup>3</sup> The principle of the direct method is to approximate the

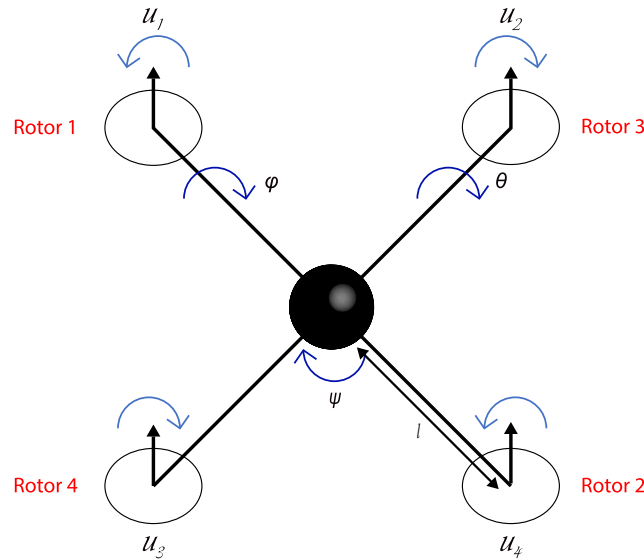
state and control variable by utilizing curvilinear functions such as B-splines,<sup>4</sup> Bezier,<sup>5</sup> the pseudospectral,<sup>6</sup> and Hermit polynomials.<sup>7</sup> Accordingly, the parameters of these curves represent the decision variable of the optimization problem. Among the applications of the direct method for the quadrotor, Wang et al<sup>8</sup> used the pseudospectral method to generate a time-optimal trajectory reference, and Kahale et al<sup>9</sup> utilized the direct collocation method to solve the same problem. The disadvantage of these two methods was the large number of variables utilized during the optimization problem, which complicated the resolution task for solvers. To solve this problem, another approach was created to solve the optimization trajectory problem of the quadrotor based on differential flatness,<sup>10</sup> where the problem dimensions were decreased to have a more computational performance. Moreover, the flatness property was extensively used for the planning and tracking of quadrotor trajectories. Aguilar-Ibáñez et al<sup>11</sup> proposed flatness and exact-tracking-error linearization to solve the trajectory tracking problem. Chamseddine et al<sup>12</sup> developed a trajectory planning/replanning strategy based on Bezier polynomials and a flatness property for the quadrotor to drive the system from an initial position to a final one as fast as possible under actuator faults and without hitting system constraints. Fernández-Caballero et al<sup>13</sup> utilized the flatness approach and the generalized proportional integral to ensure the exponentially stable behavior of the controlled quadrotor position and orientation despite the existence of noisy measurements and disturbances. Limaverde Filho et al<sup>14</sup> combined the flatness and predictive-control strategies to ensure an online trajectory tracking for the quadrotor. Lu et al<sup>15</sup> put forward an online optimization process based on the differential flatness to generate desired references and a backstepping controller to track the obtained optimal trajectory.

Although those flatness control algorithms could improve the trajectory tracking performance, all of them were designed based on a nonlinear quadrotor model that did not integrate parameter uncertainties and external disturbances, which was not real in practical applications. Thus, a robust control is necessary to be combined with the flatness control so as to design a robust flatness controller that ensures a good tracking for the quadrotor under parametric uncertainties and external disturbances.

Until now, in the literature, there have existed few approaches studying the robustness problem of flat systems. Hagenmeyer and Delaleau<sup>16,17</sup> suggested an approach to analyze the robustness of the tracking error based on the differential flatness whose stability result was studied by Kelemen. In addition, Lavinge et al<sup>18</sup> developed a modeling-based method to deal with parametric uncertainties. First, they modeled the quasi-linearized system as a disturbed linear plant composed by models belonging to a compact set of linearized plants and a model of uncertainties. Second, a linear controller based on the  $H_\infty$  method and the  $\mu$ -synthesis analysis was designed to take the uncertainties into account. In the same context, another modeling approach was presented by Zerar et al,<sup>19</sup> where a nonlinear uncertain differentially flat system was transformed into a linear parameter varying plant by a first-order linearization of the error model along the nominal trajectory. After that, a gain-scheduling H-infinity controller was developed to resolve the robust tracking trajectory problem. This approach was useful in a lot of practical applications, but when the perturbations are large, the first-order-linearization-based approach could not guarantee stability. Anritter et al<sup>20</sup> combined the flatness and interval method to improve the robustness of the tracking controller. Therefore, the interval methods were utilized to calculate the maximal admissible uncertainties in the plant, whose controlled system could be guaranteed to stay within a particularized neighborhood of the reference trajectory. Most of the methods cited for studying the robustness of flat systems have considered the uncertainties in the dynamic model, and not the uncertainties caused by invertible transformations between the flat output states and the original system states. Hence, Ryu and Agrawal<sup>21</sup> proposed a robust flatness control based on the Lyapunov redesign for a mobile robot to track the trajectory reference in the presence of slips. Chen et al<sup>22</sup> combined the flatness approach and the sliding control to create a robust tracking strategy for the flat system despite the existence of disturbances with unknown bounds.

In this paper, the contribution consists in the creation of a robust and performant guidance module for the quadrotor in dense indoor environments. Thus, two objectives are necessary. Firstly, a smooth and feasible trajectory has to be generated to avoid static obstacles. In fact, an optimization problem will be created where the obstacles are defined as constraints and the jerk as a cost function. Therefore, the optimization problem will be complicated because of the number of constraints. Then, an approach based on the theory of differential flatness, the B-spline function and the collocation algorithm are utilized to solve the optimization problem more easily. Secondly, a robust flatness-based controller is designed for the quadrotor system to track the obtained optimal reference trajectories in spite of the existence of parametric uncertainties and disturbances.

This document is organized as follows. In Section 2, the quadrotor model is presented. In Section 3, the optimization trajectory problem is defined. In Section 4, the algorithm of the optimal trajectory generation is defined as well. Section 5 is devoted to the design of the tracking controller for a quadrotor without considering uncertainties. In Section 6, we present the robust flatness-based tracking control for an uncertain quadrotor, and Section 7 deals with the simulation results.



**FIGURE 1** Quadrotor aircraft scheme [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

## 2 | QUADROTOR MODEL

The quadrotor (Figure 1) is an aircraft with four engines installed on a cross usually made of carbon fiber. The front and rear engines rotate clockwise while the right and left engines rotate in the opposite direction. A lot of work has been done to model the quadrotor. In this paper, we consider the commonly used quadrotor model obtained via the Lagrange approach<sup>23</sup> as follows:

$$\ddot{x}(t) = \frac{u_1}{M} (\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi) \quad (1)$$

$$\ddot{y}(t) = \frac{u_1}{M} (\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) \quad (2)$$

$$\ddot{z}(t) = \frac{u_1}{M} (\cos \theta \cos \phi) - g \quad (3)$$

$$\ddot{\theta}(t) = \dot{\phi} \dot{\psi} \left( \frac{I_z - I_x}{I_y} \right) + \frac{l}{I_y} u_2 \quad (4)$$

$$\ddot{\phi}(t) = \dot{\theta} \dot{\psi} \left( \frac{I_z - I_y}{I_x} \right) + \frac{l}{I_x} u_3 \quad (5)$$

$$\ddot{\psi}(t) = \dot{\phi} \dot{\theta} \left( \frac{I_y - I_x}{I_z} \right) + \frac{l}{I_z} u_4, \quad (6)$$

where the coordinates of the quadrotor center is defined by  $x$ ,  $y$ , and  $z$ ;  $M$  represents the mass;  $g$  is the acceleration;  $l$  is the distance from the center of gravity to each rotor; and  $\theta$ ,  $\phi$ , and  $\psi$  are the pitch, roll, and yaw Euler angles. The moments of inertia along the directions  $x$ ,  $y$ , and  $z$  are defined by  $I_x$ ,  $I_y$ , and  $I_z$  directions, respectively. Moreover,  $u_1$ ,  $u_2$ ,  $u_3$ , and  $u_4$  are the controlled inputs. The quadrotor model stands a relatively complex model to deal with. Thereby, we suppose that when the quadrotor flies toward the target,  $\psi = 0$ , so, the quadrotor model (1)-(6) can be rewritten as follows:

$$\ddot{x}(t) = \frac{u_1}{M} \sin \theta \cos \phi \quad (7)$$

$$\ddot{y}(t) = \frac{-u_1}{M} \sin \phi \quad (8)$$

$$\ddot{z}(t) = \frac{u_1}{M} \cos \theta \cos \phi - g \quad (9)$$

$$\ddot{\theta}(t) = \frac{l}{I_y} u_2 \quad (10)$$

$$\ddot{\phi}(t) = \frac{l}{I_x} u_3, \quad (11)$$

where the state system of the quadrotor is  $X = [x, \dot{x}, y, \dot{y}, z, \dot{z}, \theta, \dot{\theta}, \phi, \dot{\phi}]^T$ , and the control input is  $U = [u_1, u_2, u_3]^T$ .

### 3 | OPTIMAL TRAJECTORY GENERATION PROBLEM

The target is to determine a smooth desired trajectory, which allows the quadrotor to avoid obstacles in a closed environment. This problem is considered as an optimal control problem defined by the state constraints, the obstacle avoidance constraints and the cost function.

#### 3.1 | Constraint state

The boundary conditions of our optimization problem are

$$X(t_0) = [x_0, \dot{x}_0, y_0, \dot{y}_0, z_0, \dot{z}_0, \theta_0, p_0, \phi_0, q_0]^T \quad (12)$$

$$X(t_f) = [x_f, \dot{x}_f, y_f, \dot{y}_f, z_f, \dot{z}_f, \theta_f, p_f, \phi_f, q_f]^T. \quad (13)$$

To improve the performance of the guidance module of the quadrotor, specific constraints are integrated in the model. These restrictions are represented in bounds on state variables as follows:

$$x_{\min} \leq x \leq x_{\max} \quad (14)$$

$$y_{\min} \leq y \leq y_{\max} \quad (15)$$

$$z_{\min} \leq z \leq z_{\max} \quad (16)$$

$$\theta_{\min} \leq \theta \leq \theta_{\max} \quad (17)$$

$$\phi_{\min} \leq \phi \leq \phi_{\max}, \quad (18)$$

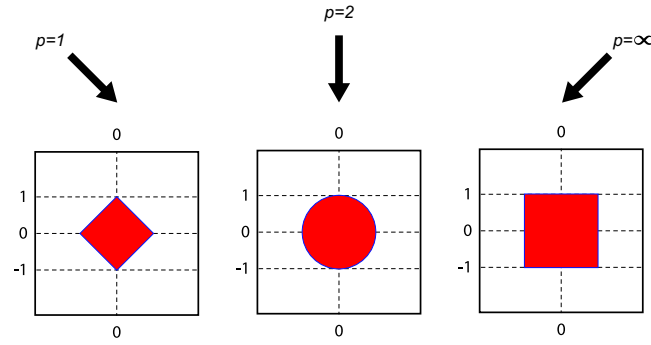
where  $x_{\min}$ ,  $x_{\max}$ ,  $y_{\min}$ ,  $y_{\max}$ ,  $z_{\min}$ ,  $z_{\max}$ ,  $\theta_{\min}$ ,  $\theta_{\max}$ ,  $\phi_{\min}$ , and  $\phi_{\max}$  represent the limit constraint states that permit the quadrotor to move in a specific defined environment.

#### 3.2 | Obstacle constraint

In the literature, there exists a lot of solutions to resolve the avoidance obstacle problem. Among these solutions is the definition of the obstacle avoidance as an inequality constraint for the optimization problem. Consequently, the obstacle model must be known or approximated from the beginning. Most work in the literature has algebraically defined the obstacles by utilizing the p-norm<sup>24</sup> to design generic shapes such as spheres and ellipses. Equation  $ob(x, y, z)$  shows the general form of the obstacle modeling used in this section as follows:

$$ob(x, y, z) = \left( \frac{x - x_{ob}}{a} \right)^p + \left( \frac{y - y_{ob}}{b} \right)^p + \left( \frac{z - z_{ob}}{c} \right)^p - 1, \quad (19)$$

where  $x, y, z$  are the quadrotor position;  $x_{ob}$ ,  $y_{ob}$ , and  $z_{ob}$  represent the center of each obstacle;  $a$  is the distance from the center to the obstacle boundary in the x-direction;  $b$  is the distance from the center to the obstacle boundary in the y-direction; and  $c$  is the distance from the center to the obstacle boundary in the z-direction. In Gatzke<sup>25</sup> research article, we can find a lot of examples explaining the utilization of the equation obstacle model (19). Figure 2 demonstrates the influence of  $p$  in the construction of the forms of obstacles. When choosing  $p = \infty$ , the optimization problem will become



**FIGURE 2** Unit  $p$  [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

computationally unresolved. For this reason,  $p$  is limited to 10 to facilitate the optimization problem. The constraint, which means that the quadrotor can avoid the obstacle, is defined as  $ob(x, y, z) > 0$ . This constraint does not take into account the size of the quadrotor because it is considered as a point that is not really in a practice flight. Hence, the constraint means that a quadrotor obstacle avoidance in a real case is defined as follows:

$$\left(\frac{x - x_{ob}}{a}\right)^p + \left(\frac{y - y_{ob}}{b}\right)^p + \left(\frac{z - z_{ob}}{c}\right)^p - 1 > R_s. \quad (20)$$

Radius  $R_s$  integrated in the obstacle representation allows additional safety around the obstacle zone. The integration of the obstacle constraint, defined by Equation (20), in the problem of optimization enables choosing a reference trajectory to avoid the collision with static obstacles. When the obstacles are dynamic, an online optimization problem will be formulated, which gradually takes into account the new position of obstacles detected by the obstacle sensor.

### 3.3 | Cost

Choosing a good cost function is important for the definition of the optimal problem. This choice depends usually on our needs, so it can directly influence the task performance ability of the quadrotor. In the literature, a lot of work has yielded desired robot trajectories such as Flash and Hogan<sup>26</sup> and Kyriakopoulos and Saridis.<sup>27</sup> Therefore, based on those results, researches generating a smooth trajectory consist in minimizing the jerk, which is defined as follows:

$$J = \int_{t_0}^{t_f} (\ddot{x}^2 + \ddot{y}^2 + \ddot{z}^2) dt. \quad (21)$$

## 4 | OPTIMAL TRAJECTORY GENERATION APPROACH

In the direct collocation method, a big number of optimization variables and constraints are generated because of the rich discretization on trajectory optimization problems. These optimization variables and constraints result in a long solution time and are linked to the time node and the number of state and control variables. Thus, we should find an approach to reduce these variables so as to decrease the solution time without any decline in solution accuracy.

In this section, an approach is presented to rapidly solve optimal trajectory problems. This approach is based on the simultaneous use of the differential flatness to decrease the dimensional space of optimal problems, the B-spline curves to approximate trajectories and the collocation method to transform an optimal problem in an NLP one.

### 4.1 | Optimal problem in flat output space

The quadrotor has a very high dimension of the state space. Consequently, the computational loads increase, hence, complicating the problem of optimization. At this stage, the differential flatness, introduced by Fliess et al.,<sup>28</sup> reduces the dimension of the optimization problem defining the flat variables. For a differentially flat system, the following nonlinear system:

$$\dot{x} = f(x, u), x \in \mathbb{R}^n, u \in \mathbb{R}^m \quad (22)$$

is differentially flat if we can find outputs such that

$$F = \xi(x, u, \dot{u}, \dots, u^{(r-1)}) \tag{23}$$

$$x = \gamma(F, \dot{F}, \ddot{F}, \dots, F^{(\alpha)}) \tag{24}$$

$$u = \gamma(F, \dot{F}, \ddot{F}, \dots, F^{(\alpha+1)}), \tag{25}$$

where  $\alpha$  and  $r$  are finite multiindices, and  $\xi$  and  $\gamma$  are smooth vector functions of the output vector  $F$ .

The first step of the approach is to transform the optimal trajectory problem to a flat output space. Thereafter, let us consider the following optimal trajectory problem, which consists in finding the state-input pair  $x(t), u(t)$  that minimizes the following performance criterion:

$$J = \int_{t_0}^{t_f} \Gamma(x(t), u(t))dt. \tag{26}$$

This criterion is subject to the following system equations:

$$\dot{x} = f(x, u), \tag{27}$$

the following boundary conditions:

$$X(t_0) = X_0, X(t_f) = X_f, \tag{28}$$

and the following trajectory constraints:

$$w_{\min} \leq \lambda(x(t), u(t)) \leq w_{\max}, \tag{29}$$

where  $\lambda(x(t), u(t))$  is composed of state and obstacle constraints defined in Section 3, and  $w_{\min}$  and  $w_{\max}$  are the lower and upper bounds of constraints. When using the flatness properties (24) and (25), the cost, the initial and final conditions, and the state and obstacle constraints will be expressed as a function of the flat outputs and their derivatives. In as much as the flat outputs indicate a minimal representation of the system behavior, the number of decision variables for the transformed optimization problem becomes highly smaller. Added to that, the flatness eliminates the step of differential equations step to obtain a geometric programming problem, which is computationally tractable and quickly solvable. Thus, the transformation of the optimization problem (26)-(29) into a flat optimal problem consists in finding the flat outputs and their derivatives, which minimizes the cost as follows:

$$J = \int_{t_0}^{t_f} \Gamma(F, \dot{F}, \ddot{F}, \dots, F^{(\alpha+1)})dt. \tag{30}$$

Furthermore, the boundary conditions are satisfied as follows:

$$\bar{F}_i(t_0) = \bar{F}_{i0}, \bar{F}_i(t_f) = \bar{F}_{if}. \tag{31}$$

Besides, the trajectory constraints are satisfied as follows:

$$w_{\min} \leq \lambda(\bar{F}_i) \leq w_{\max} \tag{32}$$

with  $\bar{F}_i = [F_i, \dot{F}_i, \ddot{F}_i, \dots, F_i^{(r)}]$ ,  $i = 1..n$ , where  $n$  is the number of the flat output and  $\lambda(\bar{F}_i)$  is the trajectory constraint.

### 4.2 | Approximation of flat variables

To get feasible trajectories, we need to set all the flat outputs as appropriate functions. In addition, to select a good function of the flat output with a reasonable number of parameters, it is necessary to define the level of continuity  $C^k$  and to estimate the effect of each parameter to quickly find a curve corresponding to the right solution.

Several curves can be used to specify the flat output (Fourier series, polynomials, ...). Actually, it is possible to express the flat output in terms of a finite-dimensional B-spline. The B-spline formalism<sup>29</sup> proposes a suitable solution for defining piecewise polynomial trajectories and offering the best elasticity due to the small number of parameters. Hence, the B-spline curves are defined in the following way:

$$F_i(t) = \sum_{k=1}^{N_c} B_{k,r}(t)\alpha_{ik}, i = 1, \dots, n, \tag{33}$$

where  $\alpha_{ik}$  is the control parameters and  $N_c$  represents the number of control parameters. Indeed,  $N_c = N * (r - s) + s$ , where  $N$  is the number of intervals,  $r$  is the degree of the polynomial pieces,  $s$  is the smoothness condition at breakpoints, and  $B_{k,r}(t)$  are the basis functions. After the approximation of the flat output, the control points  $\alpha_{ik}$  become the decision variables for the flat optimal problem (30)-(32). The choice of the B-spline parameters will greatly influence the shape of the curve and the number of control points used to solve the trajectory optimization problem. Several factors have to be taken into account when determining the B-spline parameters. Some of them are listed as follows:

- Order of system
- Number of constraints
- Severity of constraints

Generally, increasing the order and the smoothness allows having more flexibility for the curve but it leads also to more control points, which complicates the optimization problem. Therefore, the choice of the B-spline parameter is a compromise between obtaining flexible trajectories with a high continuity and using a low number of parameters to easily resolve the optimization problem.

### 4.3 | Transferring flat optimal control to NLP problem

Before using the collocation points to discretize the optimal control problem, the physical time-interval domain  $t \in [t_0, t_f]$  needs to be scaled into domain  $\tau \in [-1, 1]$  by utilizing the Chebyshev Pseudospectral Method<sup>30</sup> as follows:

$$\tau = \frac{2t - (t_f + t_0)}{t_f - t_0}. \quad (34)$$

The collocation point can be determined in the interval  $[-1, 1]$  using the Chebyshev-Gaussian-Labato as follows:

$$t_k = \cos(\pi k/N); k = 0, \dots, N. \quad (35)$$

These points are the extreme of the  $N$ th-order Chebyshev polynomial. The  $i$ th-order Chebyshev polynomial is expressed by

$$T_i(t) = \cos(i \cos^{-1}(t)). \quad (36)$$

According to the evaluation rules from the study of Milam,<sup>31</sup> the B-splines have the form shown in Equation (37) when evaluated at the points  $\tau_j$ , called the collocation points, as follows:

$$\begin{bmatrix} F_i(\tau_0) \\ \dot{F}_i(\tau_0) \\ \vdots \\ F_i^{r_i}(\tau_0) \\ \vdots \\ F_i(\tau_N) \\ \dot{F}_i(\tau_N) \\ \vdots \\ F_i^{r_i}(\tau_N) \end{bmatrix} = B_{ci} \begin{bmatrix} \alpha_{i1} \\ \vdots \\ \alpha_{iNc} \end{bmatrix}, \quad (37)$$

where  $i = 1, \dots, n$ ,  $n$  is the number of flat outputs and  $B_{ci}$  is the collocation matrix based on the B-spline basis functions for each output. The constraints will be forced to the collocation points  $\tau_j$ . Increasing the number of collocation points will raise the accuracy of the optimal solution but this leads to obtaining a strong dimension of the optimization problem, which complicates the resolution task. Thus, the number of the collocation points is chosen such that the constraints will be satisfied at all time instants, and the calculation time to resolve the optimization problem will be kept in a reasonable range. In addition, according to Louembet et al,<sup>32</sup> we must at least choose the collocation points, so that the constraints will be checked at a frequency of one point per second. Finally, the obtained NLP problem is defined as follows:

$$\min_{\bar{\alpha}} J(\bar{\alpha}) \begin{cases} \alpha_{i1} = F_{i0}, \alpha_{iNc} = F_{if} \\ w_{\min} \leq G_1(\bar{\alpha}) \leq w_{\max} \\ G_2(\bar{\alpha}) = 0 \end{cases} \quad (38)$$

with:

$$\bar{\alpha} = (\alpha_{11}, \dots, \alpha_{1Nc}, \alpha_{21}, \dots, \alpha_{2Nc}, \dots, \alpha_{n1}, \dots, \alpha_{nNc}),$$

where  $G_1(\bar{\alpha})$  and  $G_2(\bar{\alpha})$  are the trajectory constraints as a function of control parameters. Finally, the last step is to use the SNOPT solver to resolve the NLP problem (38). The nonlinear programming SNOPT solver, developed by Gill et al,<sup>33</sup> is a general-purpose system for a large-scale nonlinearly constrained optimization that utilizes a sparse Sequential Quadratic Programming method.

## 5 | FLATNESS-BASED TRACKING CONTROL

A flatness controller is created to allow the quadrotor system to follow the obtained optimal reference trajectories. It can be shown that the quadrotor model (7)-(11) is a differently flat system, whose flat outputs are given by  $F_1 = x$ ,  $F_2 = y$ ,  $F_3 = z$ . The flatness property has proven its power in optimal planning. In addition, it enables designing an endogenous feedback linearization and a diffeomorphism to obtain a Brunovsky's canonical form, whose state vector is defined by the flat outputs. Consequently, the optimal trajectory planning and the tracking problems can be defined as an equivalent flat output domain. According to the quadrotors model (7)-(11), the input torques  $u_2$  and  $u_3$  are written as a function of the fourth-order time derivatives of the flat outputs  $x, y$ , and  $z$ . Due to the parameterization of the control input  $u_1$  that depends on the second-order time derivatives of  $x, y$ , and  $z$ , it is impossible to create a static feedback linearization, hence, the need for a second-order dynamic extension of  $u_1$  to create an invertible relation between the original states and the flat outputs as well as their derivatives. To solve this problem more easily, Delaleau and Rudolph<sup>34</sup> suggested tracking control for a flat system based on a quasi-static state feedback. This approach has been applied successfully for chemical reactors<sup>35</sup> and gantry cranes.<sup>36</sup> The main advantage of using the quasi-static state feedback in the design of flatness-based tracking control consists in the fact that there are no extra dynamics integrated into the control loop as in dynamic feedback linearization. As the latter, it also permits obtaining linear tracking error dynamics for nonlinear flat systems. The idea of a quasi-static state feedback is based on the framework of generalized state representations defined as follows:

$$\dot{\hat{x}}_i = A_i(\hat{x}, \dot{u}, \ddot{u}, \dots, u^{(\epsilon_i)}) \quad i = 1, \dots, n, \quad (39)$$

where the set  $\hat{x} = (x_1, \dots, x_n)$  is called a (generalized) state.

According to Rudolph and Delaleau,<sup>37</sup> the control input  $u$  and the closed-loop input  $v$  are related by a quasi-static feedback of the (generalized) state  $x$ , if

- the feedback law is defined as follows:

$$\hat{x}_i = \Xi_i(\hat{x}, \dot{v}, \ddot{v}, \dots, v^{(\zeta_i)}) \quad i = 1, \dots, n \quad (40)$$

- this feedback is invertible:

$$v_i = \Xi_i(\hat{x}, \dot{u}, \ddot{u}, \dots, u^{(\zeta_i)}) \quad i = 1, \dots, n \quad (41)$$

-  $\hat{x}$  is a (generalized) state of the closed-loop system with a new input  $v$ .

In the following, based on the quasi-static feedback properties, two different flatness-based tracking control designs are proposed for the quadrotor.

### 5.1 | Exact feedback linearization

Tuple  $\hat{x} = [x, \dot{x}, \ddot{x}, \overset{\dots}{x}, y, \dot{y}, \ddot{y}, \overset{\dots}{y}, z, \dot{z}]^T$  represents a generalized Brunovsky state for the quadrotor system. Therefore, the new input  $v = [v_1, v_2, v_3]^T = [\overset{\dots}{x}, \overset{\dots}{y}, \overset{\dots}{z}]^T$  defines a quasi-static state feedback. However, the control inputs  $u_1$ ,  $u_2$ , and  $u_3$  must be expressed by functions, which depend on the known state  $x$ , the new input  $v$ , and the time derivatives of  $v$  only. Based on the quadrotor flat outputs  $F = [x, y, z] = [F_1, F_2, F_3]^T$ , it is possible to express all the state and control inputs as a function of the flat outputs  $F$  and their derivatives, and as function of  $v$  and their derivatives as follows:

$$\theta = \arctan \left( \frac{\ddot{F}_1}{g + v_3} \right) \quad (42)$$

$$\phi = \arcsin \left( \frac{-\ddot{F}_2}{\sqrt{\ddot{F}_1^2 + \ddot{F}_2^2 + (g + v_3)^2}} \right) \quad (43)$$



$$u_1 = M \sqrt{\ddot{F}_1^2 + \ddot{F}_2^2 + (g + v_3)^2} \quad (44)$$

$$u_2 = \frac{l}{I_y} \left( \frac{v_1}{g + v_3} - \frac{\ddot{F}_1 \dot{v}_3}{(g + v_3)^2} - 2 \frac{\ddot{\ddot{F}}_1 \dot{v}_3}{(g + v_3)^2} + 2 \frac{\ddot{F}_1 (\dot{v}_3)^2}{(g + v_3)^3} \right) \quad (45)$$

$$u_3 = \frac{Ml}{I_x} \left( \frac{(\dot{u}_1 F_2 - u_1 \dot{v}_2) \left( u_1 \sqrt{u_1^2 - M^2 \ddot{F}_2^2} \right) - (\dot{u}_1 \ddot{F}_2 - u_1 \ddot{\ddot{F}}_2) \left( \dot{u}_1 \sqrt{u_1^2 - M^2 \ddot{F}_2^2} \right)}{u_1^2 (u_1^2 - M^2 \ddot{F}_2^2)} \right) + \frac{Ml}{I_x} \left( \frac{\left( u_1 \left( u_1^2 (u_1^2 - M^2 \ddot{F}_2^2)^{1/2} \left( u_1 \dot{u}_1 - M^2 \ddot{F}_2 \ddot{\ddot{F}}_2 \right) \right) \right)}{u_1^2 (u_1^2 - M^2 \ddot{F}_2^2)} \right), \quad (46)$$

where  $\dot{u}_1$  and  $\ddot{u}_1$  are defined as follows:

$$\dot{u}_1 = M \frac{(\ddot{F}_1 \ddot{\ddot{F}}_1 + \ddot{F}_2 \ddot{\ddot{F}}_2 + \dot{v}_3 (g + v_3))}{\sqrt{\ddot{F}_1^2 + \ddot{F}_2^2 + (g + v_3)^2}} \quad (47)$$

$$\ddot{u}_1 = M \left( \frac{(\ddot{\ddot{F}}_1)^2 + \ddot{F}_1 \dot{v}_1 + (\ddot{\ddot{F}}_2)^2 + \ddot{F}_2 \dot{v}_2 + (\dot{v}_3)^2 (v_3 + g) \dot{v}_3}{\sqrt{(\ddot{F}_1)^2 + (\ddot{F}_2)^2 + (v_3 + g)^2}} - \frac{(\ddot{F}_1 \ddot{\ddot{F}}_1 + \ddot{F}_2 \ddot{\ddot{F}}_2 + (v_3 + g) \dot{v}_3)^2}{((\ddot{F}_1)^2 + (\ddot{F}_2)^2 + (v_3 + g)^2)^{3/2}} \right). \quad (48)$$

Consider  $F_{1d}$ ,  $F_{2d}$ , and  $F_{3d}$  to be the reference trajectories for the flat outputs  $F_1$ ,  $F_2$ , and  $F_3$ , respectively. Since our control problem requires obtaining accurate tracking of the reference, then the control input  $v_1$ ,  $v_2$ , and  $v_3$  are defined as follows:

$$v_1 = \ddot{\ddot{F}}_{1d} - K_{14} e_1^{(3)} - K_{13} \dot{e}_1 - K_{12} \dot{e}_1 - K_{11} e_1 \quad (49)$$

$$v_2 = \ddot{\ddot{F}}_{2d} - K_{24} e_2^{(3)} - K_{23} \dot{e}_2 - K_{22} \dot{e}_2 - K_{21} e_2 \quad (50)$$

$$v_3 = \ddot{F}_{3d} - K_{32} \dot{e}_3 - K_{31} e_3 \quad (51)$$

with  $e_1 = F_1 - F_{1d}$ ,  $e_2 = F_2 - F_{2d}$ , and  $e_3 = F_3 - F_{3d}$ . In addition to  $v_1$ ,  $v_2$ , and  $v_3$ , the first and the second time derivatives of  $v_3$  are also required. The differentiation of (51) yields

$$\dot{v}_3 = \ddot{\ddot{F}}_{3d} - K_{32} \dot{e}_3 - K_{31} \dot{e}_3 \quad (52)$$

$$\ddot{v}_3 = \ddot{\ddot{\ddot{F}}}_{3d} - K_{32} \ddot{e}_3 - K_{31} \ddot{e}_3. \quad (53)$$

The dynamics of the closed-loop tracking error turns into the following form:

$$\dot{e} = He \quad (54)$$

$$\text{with } e = [e_1, \dot{e}_1, \ddot{e}_1, \ddot{\ddot{e}}_1, e_2, \dot{e}_2, \ddot{e}_2, \ddot{\ddot{e}}_2, e_3, \dot{e}_3], H = \begin{bmatrix} H_1 & 0_{44} & 0_{42} \\ 0_{44} & H_2 & 0_{42} \\ 0_{24} & 0_{24} & H_3 \end{bmatrix}, H_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ K_{11} & K_{12} & K_{13} & K_{14} \end{bmatrix}, H_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ K_{21} & K_{22} & K_{23} & K_{24} \end{bmatrix}, \text{ and}$$

$H_3 = \begin{bmatrix} 0 & 1 \\ K_{31} & K_{32} \end{bmatrix}$ . The zero matrix of size  $n * m$  is denoted by  $0_{nm}$ . The feedback gains  $K_{11}$ ,  $K_{12}$ ,  $K_{13}$ ,  $K_{14}$ ,  $K_{21}$ ,  $K_{22}$ ,  $K_{23}$ ,  $K_{24}$ ,  $K_{31}$ , and  $K_{32}$  can be chosen so that the characteristic polynomial associated to each flat output tracking error will be Hurwitz.

## 5.2 | Exact feedforward linearization

Although the exact feedback linearization based on flatness shows good results, utilizing flatness only in this context is too limited. Hence, the notion of an exact linearization method based on flatness was created by Hagenmeyer and Delaleau<sup>38</sup> to ensure more robustness compared to the exact feedback linearization for a nonlinear flat system. Therefore,

according to the aforementioned authors,<sup>38</sup> when the desired trajectory  $F_d$  is known, we can compute the desired  $x_d$  and the feedforward control  $u_d$  as follows:

$$x_d = \gamma \left( F_d, \dot{F}_d, \ddot{F}_d, \dots, F_d^{(\alpha)} \right) \tag{55}$$

$$u_d = \gamma \left( F_d, \dot{F}_d, \ddot{F}_d, \dots, F_d^{(\alpha+1)} \right), \tag{56}$$

The control input (56) is known as an exact feedforward linearization because it gives a Brunovsky representation during all the times if the initial conditions of the desired trajectory and the system are coherent. Moreover, feedback control is generally included in the feedforward control (56) to enhance the tracking performance. In this case, two-degrees-of-freedom tracking control is developed, which includes two parts: a feedforward part,  $u_d = \gamma(F_d, \dot{F}_d, \ddot{F}_d, \dots, F_d^{(\alpha+1)})$  and a feedback part  $v$  that represents simple linear control able to stabilize the obtained linearized system. Based on the reference trajectory and the quasi-static feedback defined in the previous section, we can define the flatness feedforward linearization control applied to the quadrotor as follows:

$$u_{1d} = M \sqrt{\ddot{F}_{1d}^2 + \ddot{F}_{2d}^2 + (g + v_3)^2} \tag{57}$$

$$u_{2d} = \frac{l}{I_y} \left( \frac{v_1}{g + v_3} - \frac{\ddot{F}_{1d}\dot{v}_3}{(g + v_3)^2} - 2\frac{\ddot{F}_{1d}\dot{v}_3}{(g + v_3)^2} + 2\frac{\ddot{F}_{1d}(\dot{v}_3)^2}{(g + v_3)^3} \right) \tag{58}$$

$$u_{3d} = \frac{Ml}{I_x} \left( \frac{(\ddot{u}_{1d}F_{2d} - u_{1d}v_2) \left( u_{1d} \sqrt{u_{1d}^2 - M^2\ddot{F}_{2d}^2} \right) - (\dot{u}_{1d}\ddot{F}_{2d} - u_{1d}\ddot{F}_{2d}) \left( \dot{u}_{1d} \sqrt{u_{1d}^2 - M^2\ddot{F}_{2d}^2} \right)}{u_{1d}^2 (u_{1d}^2 - M^2\ddot{F}_{2d}^2)} \right) + \frac{Ml}{I_x} \left( \frac{\left( u_{1d} \left( u_{1d}^2 (u_{1d}^2 - M^2\ddot{F}_{2d}^2) \right)^{1/2} \left( u_{1d}\dot{u}_{1d} - M^2\ddot{F}_{2d}\ddot{F}_{2d} \right) \right)}{u_{1d}^2 (u_{1d}^2 - M^2\ddot{F}_{2d}^2)} \right) \tag{59}$$

with:

$$\dot{u}_{1d} = M \frac{(\ddot{F}_{1d}\ddot{F}_{1d} + \ddot{F}_{2d}\ddot{F}_{2d} + \dot{v}_3(g + v_3))}{\sqrt{\ddot{F}_{1d}^2 + \ddot{F}_{2d}^2 + (g + v_3)^2}} \tag{60}$$

$$\ddot{u}_{1d} = M \left( \frac{(\ddot{F}_{1d})^2 + \ddot{F}_{1d}v_1 + (\ddot{F}_{2d})^2 + \ddot{F}_{2d}v_2 + (\dot{v}_3)^2(v_3 + g)\dot{v}_3}{\sqrt{(\ddot{F}_{1d})^2 + (\ddot{F}_{2d})^2 + (v_3 + g)^2}} - \frac{(\ddot{F}_{1d}\ddot{F}_{1d} + \ddot{F}_{2d}\ddot{F}_{2d} + (v_3 + g)\dot{v}_3)^2}{((\ddot{F}_{1d})^2 + (\ddot{F}_{2d})^2 + (v_3 + g)^2)^{\frac{3}{2}}} \right), \tag{61}$$

where the correction terms  $v_1, v_2, v_3, \dot{v}_3,$  and  $\ddot{v}_3$  are defined by Equations (49)-(53).

The exact feedforward linearization differs from the exact feedback linearization in the way that in feedback linearization (44)-(46) state  $F$  is utilized, but in the feedforward linearization (57)-(59) the desired trajectory  $F_d$  is applied.

According to Hagenmeyer and Delaleau,<sup>38</sup> when using the flatness feedforward linearization control (57)-(59), the corresponding tracking error system can be denoted as follows:

$$\dot{e} = H_d e + Y, \tag{62}$$

where  $e = [e_1, \dot{e}_1, \ddot{e}_1, \ddot{e}_1, e_2, \dot{e}_2, \ddot{e}_2, \ddot{e}_2, e_3, \dot{e}_3], H_d = \begin{bmatrix} H_{d1} & 0_{44} & 0_{42} \\ 0_{44} & H_{d2} & 0_{42} \\ 0_{24} & 0_{24} & H_{d3} \end{bmatrix}, H_{d1} = H_{d2} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, H_{d3} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},$

$$Y = [0 \ 0 \ 0 \ Y_1(e, \hat{F}_d) \ 0 \ 0 \ 0 \ Y_2(e, \hat{F}_d) \ 0 \ Y_3(e, \hat{F}_d)]^T, \hat{F}_d = [F_{1d}, \dot{F}_{1d}, \ddot{F}_{1d}, \ddot{F}_{1d}, F_{2d}, \dot{F}_{2d}, \ddot{F}_{2d}, \ddot{F}_{2d}, F_{3d}, \dot{F}_{3d}]^T.$$

The robust stability of the tracking error (62) can be analyzed by making use of a result introduced by Kelemen.<sup>39</sup>

The two flatness-based tracking control laws defined by Equations (44)-(46) and (57)-(59), combined with the feedbacks

(49)-(53), ensure a good tracking of the optimal desired reference provided that the quadrotor model does not contain parametric uncertainties and external disturbances. In the next section, we study the design of the robust flatness-based control for an uncertain quadrotor.

## 6 | ROBUST FLATNESS CONTROLLER OF UNCERTAIN QUADROTOR

In this section, we first depict the effect of uncertainties on the dynamics of the closed-loop tracking error. Second, we present the robust control strategy.

### 6.1 | Effect of uncertainties

Whatever the method used to obtain a mathematical model of a physical system, there is always a compromise between the simplicity of the model and its ability to describe all the phenomena that characterize it. These differences between the model and the system are often modeled by error quantities (input error, output error, model error) because real systems are mostly nonlinear, poorly defined, or subject to external disturbances. For this reason, the quadrotor model defined in (7)-(11) does not really describe the unpredictable changes in the flight environment, and consequently, the performance of the guidance law can be degraded. Hence, a new quadrotor model that takes into account parametric uncertainties and external disturbances can be defined as follows:

$$\ddot{x}(t) = \frac{u_1}{M} \sin \theta \cos \phi + \frac{d_x}{M} \quad (63)$$

$$\ddot{y}(t) = \frac{-u_1}{M} \sin \phi + \frac{d_y}{M} \quad (64)$$

$$\ddot{z}(t) = \frac{u_1}{M} \cos \theta \cos \phi - g + \frac{d_z}{M} \quad (65)$$

$$\ddot{\theta}(t) = \frac{l}{I_y} u_2 + \frac{\Delta_l}{I_y} u_2 + \frac{l}{I_y} d_\theta \quad (66)$$

$$\ddot{\phi}(t) = \frac{l}{I_x} u_3 + \frac{\Delta_l}{I_x} u_3 + \frac{l}{I_x} d_\phi, \quad (67)$$

where  $\Delta_l$  represents the parameter uncertainties, and  $d_x, d_y, d_z, d_\theta,$  and  $d_\phi$  are external disturbances. It is assumed that the variation in  $\Delta_l$  is limited to 50% of its nominal value  $l$ . Therefore,  $\Delta_l$  is bounded as follows:

$$\|\Delta_l\| \leq \alpha l, 0 \leq \alpha \leq 0.5. \quad (68)$$

The disturbance and their derived are bounded as follows:

$$\|d\| \leq \tau_1, \|\dot{d}\| \leq \tau_2, \|\ddot{d}\| \leq \tau_3, \quad (69)$$

where  $d = [d_x, d_y, d_z, d_\theta, d_\phi]^T$  and  $\tau_i, i = 1 \dots 3$  are known values as a function of the state variables. When considering the uncertainties in the quadrotor model (63)-(67), the tracking error dynamics will be expressed as follows:

$$\dot{e} = He + J\rho, \quad (70)$$

where  $\rho$  and  $J$  are two matrices defined as follows:

$$\rho = Ev + G, J = [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1]^T, \quad (71)$$

where  $v = [v_1, v_2, v_3]^T$ ,  $E$  and  $G$  are two matrices described as a function of the state system, the uncertain parameter  $\Delta_l$ , and the external disturbances  $d$  and their derivatives. According to Equation (70), the error dynamics do not converge to zero due to the existence of  $\rho$ . As a result, the actual trajectory diverges from the reference. In the next section, we investigate the application of a robust control strategy presented for a robot mobile<sup>21</sup> to an uncertain quadrotor system.

## 6.2 | Robust control

To compensate the effect of uncertainties, an additional term  $v_r$  is added to the auxiliary control law  $v$ . This gives a new input  $\bar{v}$  with robustness compensation

$$\bar{v} = v + v_r, \quad (72)$$

where  $v$  is given by Equations (49)-(51). With this new controller, the tracking error dynamics will be defined as follows:

$$\dot{e} = He + J(\rho + v_r). \quad (73)$$

The construction of  $v_r$  is based on the Lyapunov redesign, which is chosen as follows:

$$v_r = \begin{cases} -\sigma(e, t) \frac{J^T P e}{\|J^T P e\|} & \text{for } \|J^T P e\| \neq 0 \\ 0 & \text{for } \|J^T P e\| = 0, \end{cases} \quad (74)$$

where  $P$  is the unique positive-definite symmetric solution of the Lyapunov equation

$$H^T P + PH + Q = 0, \quad (75)$$

where  $Q$  represents the symmetric positive-definite matrix, and  $v_r$  and  $\rho$  are bounded as follows:

$$\|v_r\| \leq \sigma, \|\rho\| \leq \sigma. \quad (76)$$

Let us define the Lyapunov function candidate as follows:

$$V = e^T P e. \quad (77)$$

Its derivative is given by

$$\dot{V} = \dot{e}^T P e + e^T P \dot{e}. \quad (78)$$

Substituting  $\dot{e}$  by Equation (73), the derivative of the Lyapunov function is defined as follows:

$$\dot{V} = e^T (H^T P + PH) e + 2e^T P J (v_r + \rho) \quad (79)$$

$$= e^T (-Q) e + 2\xi^T (v_r + \rho) \quad (80)$$

with  $\xi = J^T P e$ .

If  $\xi = 0$ ,  $\dot{V} = e^T (-Q) e$ .

If  $\xi \neq 0$ , the second term of Lyapunov function turns out to be as follows:

$$\begin{aligned} \xi^T \left( -\sigma \frac{\xi}{\|\xi\|} + \rho \right) &= -\sigma \frac{\xi^T \xi}{\|\xi\|} + \xi^T \rho \\ &\leq -\sigma \|\xi\| + \|\xi\| \|\rho\| \\ &\leq \|\xi\| (\|\rho\| - \sigma). \end{aligned} \quad (81)$$

Since  $\|\rho\| \leq \sigma$ , then  $\dot{V} \leq 0$ .

Next, we can calculate the function  $\sigma(e, t)$  as follows:

$$\begin{aligned} \|\rho\| &= \|E\bar{v} + G\| \\ &\leq \|E\| \|\bar{v}\| + \|G\| \\ &\leq \|E\| \|v\| + \|E\| \|v_r\| + \|G\| \\ &\leq \omega_1 \|v\| + \omega_1 \sigma + \omega_2 \end{aligned} \quad (82)$$

with  $\|E\| \leq \omega_1$  and  $\|G\| \leq \omega_2$ .

According to inequality (82) and the inequality defined by  $\|\rho\| \leq \sigma$ , we can deduce that

$$\sigma = \frac{1}{1 - \omega_1} (\omega_1 \|v\| + \omega_2). \quad (83)$$

We can obtain  $\omega_1$  and  $\omega_2$  by algebra calculation as follows:

$$\begin{aligned}\|E\| = \|N\| &= \sqrt{\lambda_{\max}(N^T N)} \\ &= \frac{\sqrt{\Delta_l^2}}{\|l\|} \\ &= \frac{\|\Delta L\|}{\|l\|}.\end{aligned}\quad (84)$$

From Equation (68), we can deduce that

$$\|N\| \leq \alpha. \quad (85)$$

Matrix  $G$  is written as a function of the uncertain parameters  $\Delta L$ , disturbance  $d$  and its derivatives. On examining the structure of particular terms that compose  $G$ , utilizing assumptions (68) and (69), it is feasible to get function  $\beta$  of the known values and the state variables as follows:

$$\|G\| \leq \beta. \quad (86)$$

According to Equations (85) and (86), we can deduce that

$$\rho = \frac{1}{1 - \alpha}(\alpha\|v\| + \beta). \quad (87)$$

The control law defined by (74) provokes the chattering phenomenon of the control inputs due to the existence of discontinuous terms. Thus, to avoid this problem, we introduce the continuous control as follows:

$$v_r = \begin{cases} -\sigma(e, t) \frac{\xi}{\|\xi\|} & \text{for } \|\xi\| \geq \epsilon \\ -\sigma(e, t) \frac{\xi}{\epsilon} & \text{for } \|\xi\| < \epsilon, \end{cases} \quad (88)$$

where  $\xi = e^T P J$ , and  $\epsilon$  is a small positive number chosen to reduce chattering in the control.

Substituting  $v_r$ , defined by Equation (88), into Equation (72), we obtain the robust correction term  $\bar{v}$ . Hence, the nominal control  $v$  is replaced by  $\bar{v}$  in the two flatness-based tracking controllers  $u_1$ ,  $u_2$ , and  $u_3$  as well as  $u_{1d}$ ,  $u_{2d}$ , and  $u_{3d}$  to obtain two robust flatness-based tracking controllers. The first is the Robust Feedback Control (RFB) defined by Equations (89)-(91) and the second is the Robust Feedforward Control (RFF) defined by Equations (92)-(94). These latter controllers are defined as follows:

$$u_{\text{RFB1}} = M \sqrt{\ddot{F}_1^2 + \ddot{F}_2^2 + (g + \bar{v}_3)^2} \quad (89)$$

$$u_{\text{RFB2}} = \frac{l}{I_y} \left( \frac{\bar{v}_1}{g + \bar{v}_3} - \frac{\ddot{F}_1 \ddot{v}_3}{(g + \bar{v}_3)^2} - 2 \frac{\ddot{\ddot{F}}_1 \dot{\ddot{v}}_3}{(g + \bar{v}_3)^2} + 2 \frac{\ddot{F}_1 (\dot{\bar{v}}_3)^2}{(g + \bar{v}_3)^3} \right) \quad (90)$$

$$\begin{aligned}u_{\text{RFB3}} &= \frac{Ml}{I_x} \left( \frac{(\ddot{u}_{\text{RFB1}} F_2 - u_{\text{RFB1}} \ddot{v}_2) \left( u_{\text{RFB1}} \sqrt{u_{\text{RFB1}}^2 - M^2 \ddot{F}_2^2} \right) - (\dot{u}_{\text{RFB1}} \ddot{F}_2 - u_{\text{RFB1}} \ddot{\ddot{F}}_2) \left( \dot{u}_{\text{RFB1}} \sqrt{(u_{\text{RFB1}}^2 - M^2 \ddot{F}_2^2)} \right)}{u_{\text{RFB1}}^2 (u_{\text{RFB1}}^2 - M^2 \ddot{F}_2^2)} \right) \\ &+ \frac{Ml}{I_x} \left( \frac{\left( u_{\text{RFB1}} \left( u_{\text{RFB1}}^2 (u_{\text{RFB1}}^2 - M^2 \ddot{F}_2^2)^{1/2} \left( u_{\text{RFB1}} \dot{u}_{\text{RFB1}} - M^2 \ddot{F}_2 \ddot{\ddot{F}}_2 \right) \right) \right)}{u_{\text{RFB1}}^2 (u_{\text{RFB1}}^2 - M^2 \ddot{F}_2^2)} \right)\end{aligned} \quad (91)$$

$$u_{\text{RFF1}} = M \sqrt{\ddot{F}_{1d}^2 + \ddot{F}_{2d}^2 + (g + \bar{v}_3)^2} \quad (92)$$

$$u_{\text{RFF2}} = \frac{l}{I_y} \left( \frac{\bar{v}_1}{g + \bar{v}_3} - \frac{\ddot{F}_{1d} \ddot{v}_3}{(g + \bar{v}_3)^2} - 2 \frac{\ddot{\ddot{F}}_{1d} \dot{\ddot{v}}_3}{(g + \bar{v}_3)^2} + 2 \frac{\ddot{F}_{1d} (\dot{\bar{v}}_3)^2}{(g + \bar{v}_3)^3} \right) \quad (93)$$

$$u_{\text{RFF3}} = \frac{Ml}{I_x} \left( \frac{(\ddot{u}_{\text{RFF1}} F_{2d} - u_{\text{RFF1}} \bar{v}_2) (u_{\text{RFF1}} \sqrt{u_{\text{RFF1}}^2 - M^2 \ddot{F}_{2d}^2}) - (\dot{u}_{\text{RFF1}} \ddot{F}_{2d} - u_{\text{RFF1}} \ddot{\ddot{F}}_{2d}) (\dot{u}_{\text{RFF1}} \sqrt{u_{\text{RFF1}}^2 - M^2 \ddot{F}_{2d}^2})}{u_{\text{RFF1}}^2 (u_{\text{RFF1}}^2 - M^2 \ddot{F}_{2d}^2)} \right) + \frac{Ml}{I_x} \left( \frac{(u_{\text{RFF1}} (u_{\text{RFF1}}^2 (u_{1d}^2 - M^2 \ddot{F}_{2d}^2)^{1/2} (u_{\text{RFF1}} \dot{u}_{\text{RFF1}} - M^2 \ddot{F}_{2d} \ddot{\ddot{F}}_{2d})))}{u_{\text{RFF1}}^2 (u_{\text{RFF1}}^2 - M^2 \ddot{F}_{2d}^2)} \right) \tag{94}$$

with

$$\bar{v}_1 = \ddot{\ddot{F}}_{1d} - K_{14} e_1^{(3)} - K_{13} \ddot{e}_1 - K_{12} \dot{e}_1 - K_{11} e_1 + v_{r1} \tag{95}$$

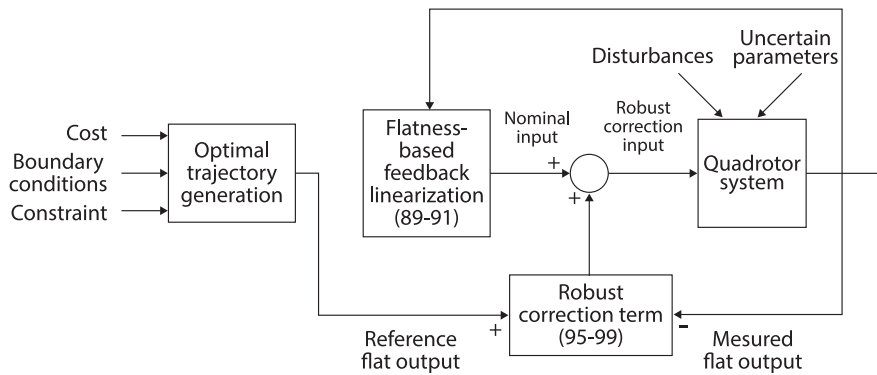
$$\bar{v}_2 = \ddot{\ddot{F}}_{2d} - K_{24} e_2^{(3)} - K_{23} \ddot{e}_2 - K_{22} \dot{e}_2 - K_{21} e_2 + v_{r2} \tag{96}$$

$$\bar{v}_3 = \ddot{\ddot{F}}_{3d} - K_{32} \ddot{e}_3 - K_{31} \dot{e}_3 + v_{r3} \tag{97}$$

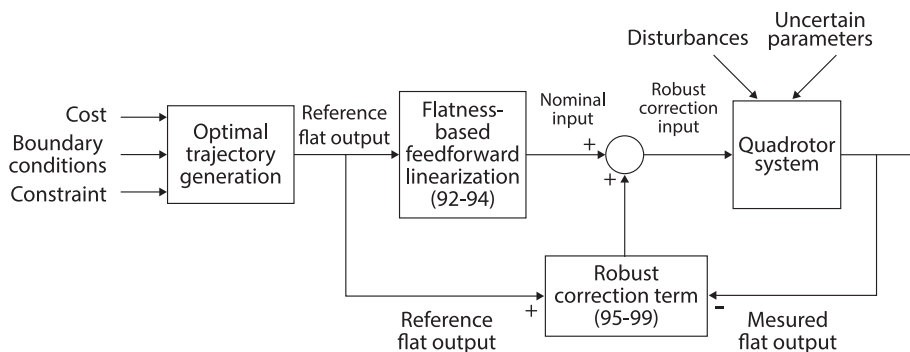
$$\dot{\bar{v}}_3 = \ddot{\ddot{F}}_{3d} - K_{32} \ddot{e}_3 - K_{31} \dot{e}_3 + \dot{v}_{r3} \tag{98}$$

$$\ddot{\bar{v}}_3 = \ddot{\ddot{F}}_{3d} - K_{32} \ddot{\ddot{e}}_3 - K_{31} \ddot{\dot{e}}_3 + \ddot{v}_{r3}. \tag{99}$$

Figure 3 and Figure 4 show the schematic diagram of the two robust flatness-based tracking controllers applied to the uncertain quadrotor.



**FIGURE 3** Schematic diagram of robust feedback linearization control applied to quadrotor



**FIGURE 4** Schematic diagram of robust feedforward linearization control applied to quadrotor

## 7 | SIMULATION RESULTS

In this section, we present the simulation results to demonstrate the efficiency of the proposed optimal trajectory generation method explained in Section 4 and the robustness of the tracking controller detailed in Section 6. Then, we consider the AR-Drone quadrotor whose parameters are given by:  $I_x = I_y = 0.002 \text{ Kg}\cdot\text{m}^2$ ,  $M = 0.5 \text{ Kg}$ ,  $l = 0.2 \text{ m}$ ,  $g = 9.81 \text{ m}\cdot\text{s}^{-2}$ .

### 7.1 | Simulation 1

A trajectory should be generated to enable the quadrotor to move from an initial state  $X(0) = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$  to a final state  $X(5) = [5, 0, 11, 0, 0, 0, 0, 0, 0, 0]^T$  in a room full of obstacles. The obtained trajectory must be minimal in jerk and be able to avoid static obstacles. In addition, the constraints of states must be respected as follows:

$$0 \leq x \leq 10 \text{ m} \quad (100)$$

$$0 \leq y \leq 15 \text{ m} \quad (101)$$

$$0 \leq z \leq 10 \text{ m} \quad (102)$$

$$-90^\circ \leq \theta \leq 90^\circ \quad (103)$$

$$-90^\circ \leq \phi \leq 90^\circ \quad (104)$$

Three cylinders and three blocks are considered as static obstacles in a closed environment. The approximation model of the rectangular block is defined as follows:

$$ob_1(x, y, z) = \left(\frac{x-2}{2}\right)^{10} + (y-7)^{10} + \left(\frac{z-5}{5}\right)^{10} - 1 \quad (105)$$

$$ob_2(x, y, z) = \left(\frac{x-8}{2}\right)^{10} + (y-7)^{10} + \left(\frac{z-5}{5}\right)^{10} - 1 \quad (106)$$

$$ob_3(x, y, z) = \left(\frac{x-7}{2}\right)^{10} + (y-3)^{10} + \left(\frac{z-5}{5}\right)^{10} - 1. \quad (107)$$

The approximation model of the cylinder is defined as follows:

$$ob_4(x, y, z) = \left(\frac{x-2}{0.9}\right)^2 + \left(\frac{y-3}{0.9}\right)^2 + \left(\frac{z-5}{5}\right)^{10} - 1 \quad (108)$$

$$ob_5(x, y, z) = \left(\frac{x-5}{0.9}\right)^2 + \left(\frac{y-9}{0.9}\right)^2 + \left(\frac{z-5}{5}\right)^{10} - 1 \quad (109)$$

$$ob_6(x, y, z) = \left(\frac{x-3}{0.9}\right)^2 + \left(\frac{y-10}{0.9}\right)^2 + \left(\frac{z-5}{5}\right)^{10} - 1. \quad (110)$$

Radius  $R = 0.1 \text{ m}$  is for safety around the obstacle zone. According to Equations (105)-(110) and (20), the obstacle constraints integrated into the optimization problem are defined as follows:

$$\left(\frac{x-2}{2}\right)^{10} + (y-7)^{10} + \left(\frac{z-5}{5}\right)^{10} - 1 > 0.1 \quad (111)$$

$$\left(\frac{x-8}{2}\right)^{10} + (y-7)^{10} + \left(\frac{z-5}{5}\right)^{10} - 1 > 0.1 \quad (112)$$

$$\left(\frac{x-7}{2}\right)^{10} + (y-3)^{10} + \left(\frac{z-5}{5}\right)^{10} - 1 > 0.1 \quad (113)$$

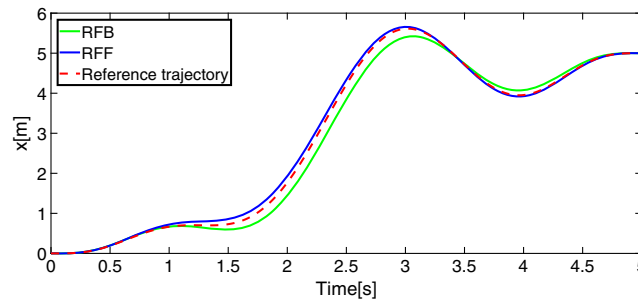
$$\left(\frac{x-2}{0.9}\right)^2 + \left(\frac{y-3}{0.9}\right)^2 + \left(\frac{z-5}{5}\right)^{10} - 1 > 0.1 \quad (114)$$

$$\left(\frac{x-5}{0.9}\right)^2 + \left(\frac{y-9}{0.9}\right)^2 + \left(\frac{z-5}{5}\right)^{10} - 1 > 0.1 \tag{115}$$

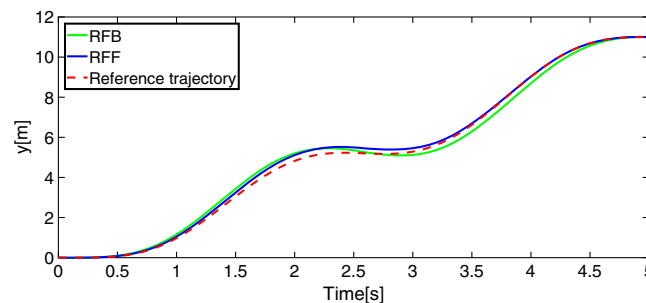
$$\left(\frac{x-3}{0.9}\right)^2 + \left(\frac{y-10}{0.9}\right)^2 + \left(\frac{z-5}{5}\right)^{10} - 1 > 0.1. \tag{116}$$

After formulating the optimization problem in terms of flat output, we choose for each flat output of  $F_1$ ,  $F_2$ , and  $F_3$ , the curve B-spline defined by the number of intervals  $n = 1$ , the smoothness condition  $s = 11$ , and the order of the polynomial  $r = 12$ . The number of collocation points is chosen to be 25. The results obtained in this paper are from the nonlinear programming package (SNOPT). The SNOPT is implemented using feasibility tolerances of  $1 \cdot 10^{-5}$ . One benefit of the direct collocation method utilized in this paper is the convenience in making an initial guess so that with a large range of initial guesses, the optimization process will converge to the optimal solution, following some iteration. When utilizing the optimal trajectory generation approach based on flatness, as well as on the collocation method and the B-spline function, to resolve the optimization trajectory problem for the quadrotor, the number of optimization variables is defined by: the numbers of a flat output \* the number of the control parameters =  $3 \cdot 12 = 36$ . On the other hand, when using the collocation method or the pseudospectral method to resolve the same trajectory optimization problem for the quadrotor, the number of optimization variables is defined by: the number of state and control variables for the quadrotor system \* the number of collocation points =  $(10+3) \cdot 25 = 325$ . Thus, based on this significant reduction in the variable optimization number, the proposed method, compared to other conventional methods, allows obtaining a faster solution to trajectory optimization problems.

To check the efficiency of the RFB and RFF control, we choose  $\alpha = 0.4$ , which means that the quadrotor undergoes a parametric variation of 40% in nominal  $l$ . Furthermore, the external disturbance acting on the quadrotor is chosen as:  $dx = dy = dz = \cos(2t) + 1$ . All the other external disturbances are set to zero. Indeed,  $\epsilon$  is chosen to be 0.1 to reduce chattering. The gains of the feedback controller are chosen to be  $K_{11} = K_{21} = 625, K_{12} = K_{22} = 500, K_{13} = K_{23} = 150, K_{14} = K_{24} = 20, K_{31} = 25,$  and  $K_{32} = 10$ . Matrix  $Q$  in the Lyapunov equation is selected to be an identity. Figures 5 to 13 depict the desired and actual trajectories of the quadrotor under the application of RFB and RFF control to the quadrotor system subjected to uncertain parameters and external disturbances.

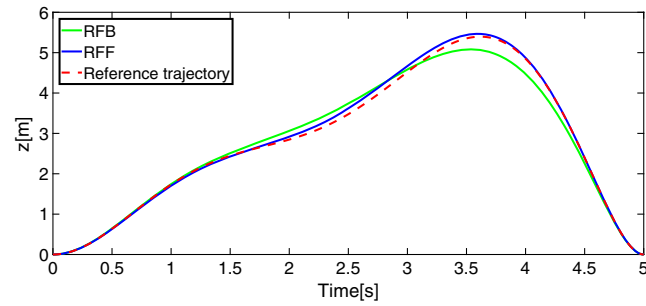


**FIGURE 5** Tracking results for position x. RFB, robust feedback control; RFF, robust feedforward control [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

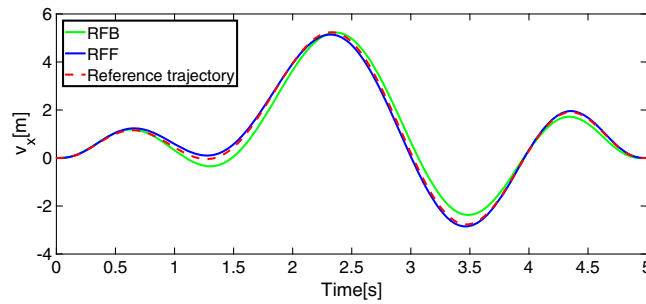


**FIGURE 6** Tracking results for position y. RFB, robust feedback control; RFF, robust feedforward control [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

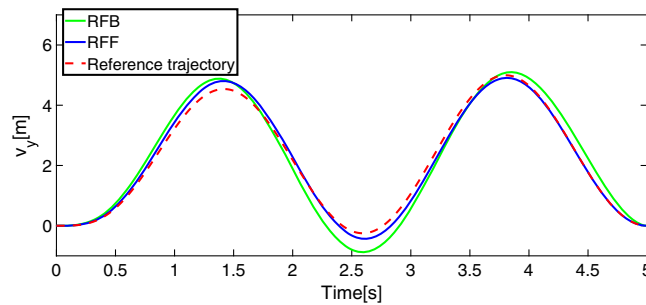




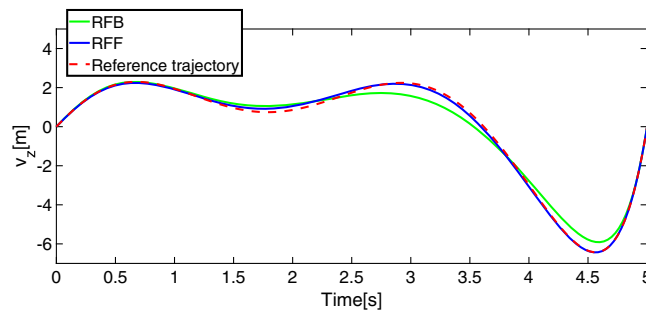
**FIGURE 7** Tracking results for position  $z$ . RFB, robust feedback control; RFF, robust feedforward control [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]



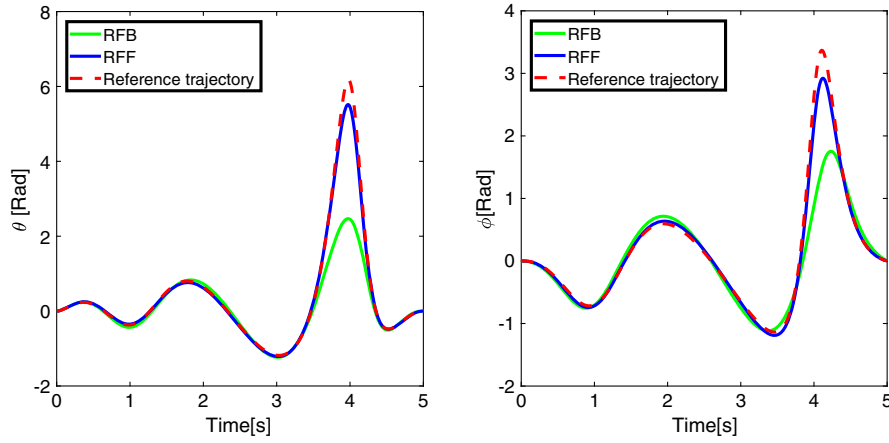
**FIGURE 8** Tracking results for velocity  $x$ . RFB, robust feedback control; RFF, robust feedforward control [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]



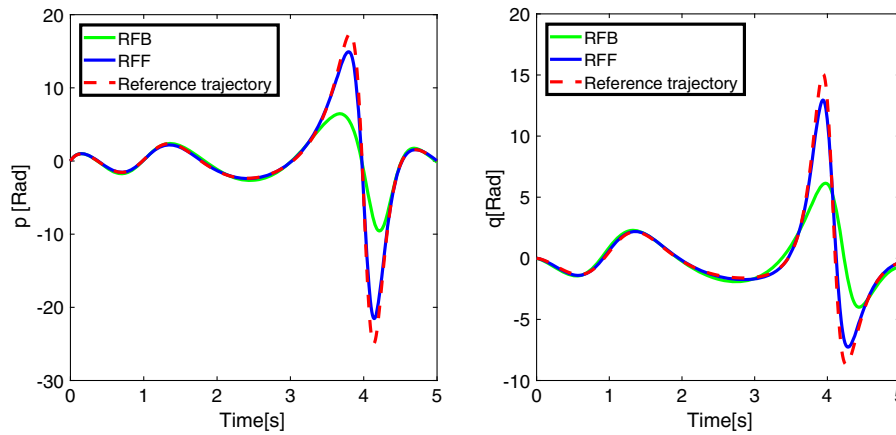
**FIGURE 9** Tracking results for velocity  $y$ . RFB, robust feedback control; RFF, robust feedforward control [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]



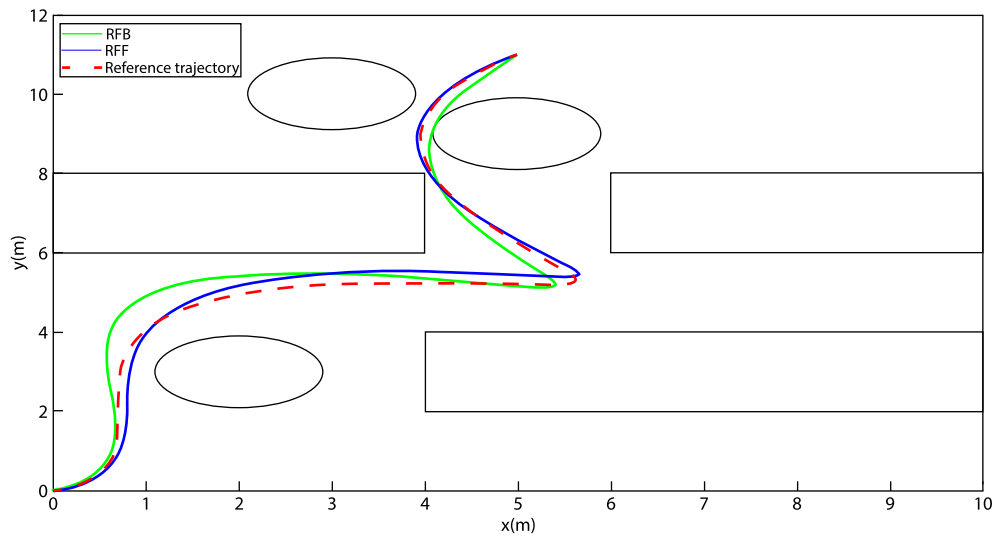
**FIGURE 10** Tracking results for velocity  $z$ . RFB, robust feedback control; RFF, robust feedforward control [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]



**FIGURE 11** Results for attitude tracking of quadrotor. RFB, robust feedback control; RFF, robust feedforward control [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]



**FIGURE 12** Results for angular velocity tracking of quadrotor. RFB, robust feedback control; RFF, robust feedforward control [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]



**FIGURE 13** Results for 2D tracking of quadrotor. RFB, robust feedback control; RFF, robust feedforward control [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

From Figures 5 to 13, it is firstly seen that the obtained desired trajectory satisfies all constraints defined in the flight mission. Secondly, it can be observed that both robust flatness-based controllers permit obtaining a good trajectory tracking performance for the quadrotor. However, the tracking of the optimal reference trajectory is more accurate using the RFF control rather than utilizing the RFB control law.

## 7.2 | Simulation 2

To better show the effectiveness of the optimal trajectory generation method and the robust flatness-based tracking control, we perform other simulation. Thus, we consider the same optimization trajectory problem defined in simulation 1 while choosing two other costs. The first cost to be considered is the time defined as follows:

$$J = \int_{t_0}^{t_f} dt. \quad (117)$$

The second cost is the control effort defined in Chamseddine et al<sup>12</sup> as follows:

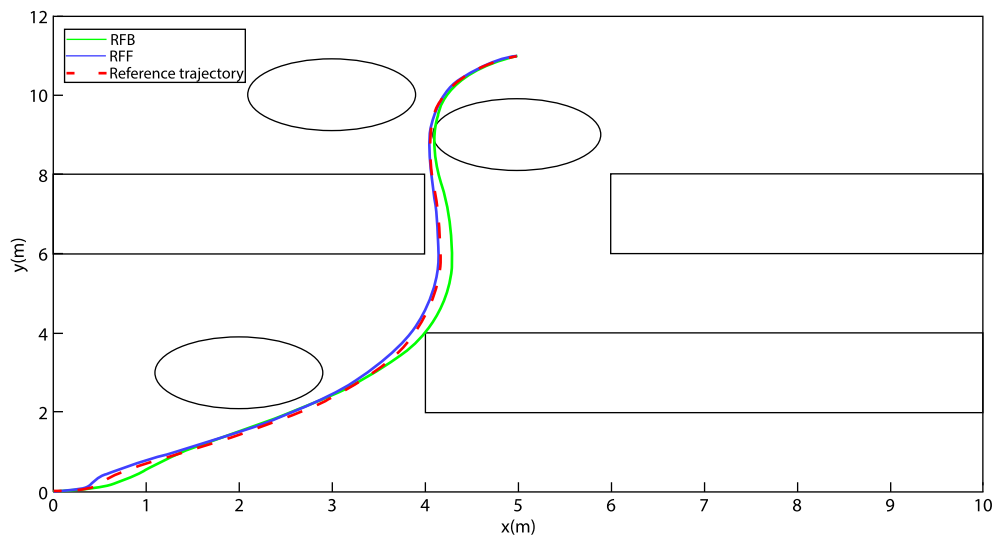
$$J = \int_{t_0}^{t_f} (4u_{1d}^2 + u_{2d}^2 + u_{3d}^2) dt. \quad (118)$$

The same B-spline parameter and SNOPT solver considered in simulation 1 are used to approximate the flat outputs and to resolve the optimization problem. Worst-case scenarios are carried out to show more the performance of RFF and RFB. According to Mallavalli and Fekih,<sup>40</sup> to assess the performance under worst-case scenarios, we subject the quadrotor to parametric variations of 50% in  $l$ , to measurement noise and to a periodically changing wind defined as follows:

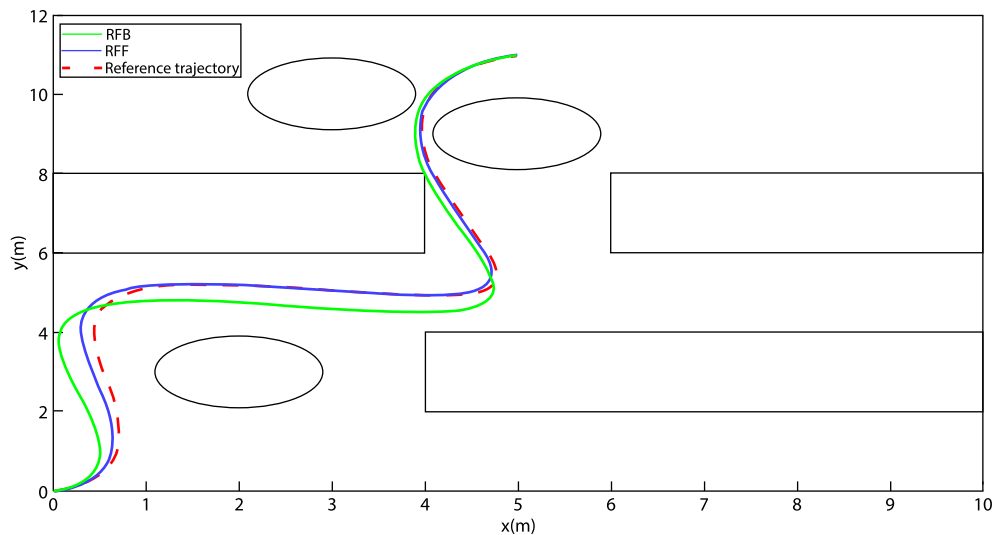
$$d_x = d_y = d_z = 0.01(10 + 5 \sin(2\pi t))\text{m/sec} \quad (119)$$

$$d_\theta = d_\phi = 0.1(10 + 5 \sin(2\pi t))\text{rad/sec} \quad (120)$$

The gains of the feedback controller are chosen to be  $K_{11} = K_{21} = 10^4$ ,  $K_{12} = K_{22} = 4.10^3$ ,  $K_{13} = K_{23} = 500$ ,  $K_{14} = K_{24} = 40$ ,  $K_{31} = 100$ , and  $K_{32} = 20$ . Figures 14 and 15 depict the results of the optimal trajectory generation method and the tracking performance when considering the time and the control effort as a cost, respectively, and when applying RFB and RFF control to the quadrotor system (63)-(67) under the worst-case parametric scenarios. The required time to complete the trajectory, shown in Figure 14, is  $t_f = 4.5$ [sec]. From Figures 14 and 15, it can be seen that the proposed optimal trajectory generation method permits resolving many trajectory optimization problems in different scenarios. In addition, it can be observed that the RFF control allows a powerful tracking of the optimal trajectory for the



**FIGURE 14** Tracking result of minimum time trajectory for the quadrotor. RFB, robust feedback control; RFF, robust feedforward control [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]



**FIGURE 15** Tracking result of minimum control effort trajectory for the quadrotor. RFB, robust feedback control; RFF, robust feedforward control [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

quadrotor than RFF control, as noted also in simulation 1. This difference is due to the fact that in the RFF control, the state is injected only in the feedback term, but in the RFB control, the state is integrated into the open-loop control part for canceling respective nonlinearities. Consequently, modeling errors and measurement noise reduce the robustness tracking performance compared to exact feedforward linearization. Hence, according to the simulation results, we can deduce that the RFF control, combined with the optimal trajectory generation approach, based on the flatness property, the collocation method, and the B-spline function, permits obtaining a performing guidance law for an uncertain quadrotor system.

## 8 | CONCLUSIONS

The problem of generating and robustly tracking an optimal trajectory for an autonomous quadrotor flying vehicle has been studied in this paper. The simulation results have demonstrated that the trajectory generation algorithm based on the flatness and the direct collocation is efficient. Taking the minimum jerk as a cost makes it possible to have smooth and natural trajectories that can greatly decrease the error tracking trajectories and reduce the difficulty of designing control laws. Transforming the problem of obstacle avoidance as an optimization problem facilitates obtaining a trajectory that avoids static obstacles. In the second part, a RFF is shown to outperform a RFB and prove its effectiveness for tracking a created optimal trajectory in spite of uncertainties and disturbances. Our future work will take into account the existence of disturbances with unknown bounds in a quadrotor system. Therefore, an adaptive flatness sliding controller will be designed to improve the tracking performance.

## ORCID

Amine Abadi  <https://orcid.org/0000-0001-7063-172X>

## REFERENCES

1. Rao AV. A survey of numerical methods for optimal control. *Adv Astronaut Sci.* 2009;135(1):497-528.
2. Hehn M, Ritz R, D'Andrea R. Performance benchmarking of quadrotor systems using time-optimal control. *Autonomous Robots.* 2012;33(1-2):69-88.
3. Gill PE, Murray W, Saunders MA, et al. User's guide for SNOPT 7: software for large-scale nonlinear programming. *SIAM Rev.* 2015;47:99-131.

4. Jamilnia R, Naghash A. Optimal guidance based on receding horizon control and online trajectory optimization. *J Aerosp Eng*. 2011;26(4):786-793.
5. Esmaelzadeh R. On-board near optimal flight trajectory generation using differential flatness. *Int J Comput Appl*. 2014;89(3):12-18.
6. Guo X, Zhu M. Direct trajectory optimization based on a mapped Chebyshev pseudospectral method. *Chin J Aeronaut*. 2013;26(2):401-412.
7. Dickmanns ED, Well KH. Approximate solution of optimal control problems using third order hermite polynomial functions. In: *Optimization Techniques IFIP Technical Conference*. Berlin, Germany: Springer-Verlag Berlin Heidelberg; 1975:158-166.
8. Wang D, Zhang W, Shan J. Optimal trajectory planning for a quadrotor via a Gauss pseudo-spectrum method. Paper presented at: 2013 9th International Conference on Natural Computation (ICNC); 2013; Shenyang, China.
9. Kahale E, Castillo P, Bestaoui Y. Minimum time reference trajectory generation for an autonomous quadrotor. Paper presented at: 2014 International Conference on Unmanned Aircraft Systems (ICUAS); 2014; Orlando, FL.
10. Yu J, Cai Z, Wang Y. Optimal trajectory generation of a quadrotor based on the differential flatness. Paper presented at: 2016 Chinese Control and Decision Conference (CCDC); 2016; Yinchuan, China.
11. Aguilar-Ibáñez C, Sira-Ramírez H, Suárez-Castañón MS, Martínez-Navarro E, Moreno-Armendariz MA. The trajectory tracking problem for an unmanned four-rotor system: flatness-based approach. *Int J Control*. 2012;85(1):69-77.
12. Chamseddine A, Zhang Y, Rabbath CA, Join C, Theilliol D. Flatness-based trajectory planning/replanning for a quadrotor unmanned aerial vehicle. *IEEE Trans Aerosp Electron Syst*. 2012;48(4):2832-2848.
13. Fernández-Caballero A, María Belmonte L, Morales R, Andres Somolinos J. Generalized proportional integral control for an unmanned quadrotor system. *Int J Adv Robot Syst*. 2015;12(7):85.
14. Limaverde Filho JA, Lourenço TS, Fortaleza E, Murilo A, Lopes RV. Trajectory tracking for a quadrotor system: a flatness-based nonlinear predictive control approach. Paper presented at: 2016 IEEE Conference on Control Applications (CCA); 2016; Buenos Aires, Argentina.
15. Lu H, Liu C, Coombes M, Guo L, Chen WH. Online optimisation-based backstepping control design with application to quadrotor. *IET Control Theory Appl*. 2016;10(14):1601-1611.
16. Hagenmeyer V, Delaleau E. Robustness analysis of exact feedforward linearization based on differential flatness. *Automatica*. 2003;39(11):1941-1946.
17. Hagenmeyer V, Delaleau E. Robustness analysis with respect to exogenous perturbations for flatness-based exact feedforward linearization. *IEEE Trans Autom Control*. 2010;55(3):727-731.
18. Lavigne L, Cazaurang F, Bergeon B. Modelling of disturbed flat system for robust control design. In: *Nonlinear Control Systems 2001: (NOLCOS 2001): A Proceedings Volume from the 5th IFAC Symposium, St. Petersburg, Russia, 4-6 July 2001, Volume 1*. Oxford, UK: International Federation of Automatic Control; 2001:1012-1015.
19. Zerar M, Cazaurang F, Zolghadri A. Robust tracking of nonlinear MIMO uncertain flat systems. Paper presented at: 2004 IEEE International Conference on Systems, Man and Cybernetics; 2004; Hague, The Netherlands.
20. Anritter F, Kletting M, Hofer EP. Robust analysis of flatness based control using interval methods. *Int J Control*. 2007;80(5):816-823.
21. Ryu JC, Agrawal SK. Differential flatness-based robust control of mobile robots in the presence of slip. *Int J Robot Res*. 2011;30(4):463-475.
22. Chen L, Jia Y, Du J. Robust control of flat systems using sliding mode approach. Paper presented at: 2016 American Control Conference (ACC); 2016; Boston, MA.
23. Xu R, Ozguner U. Sliding mode control of a quadrotor helicopter. In: *Proceedings of the 45th IEEE Conference on Decision and Control*; 2006; San Diego, CA.
24. Harada M, Nagata H, Simond J, Bollino K. Optimal trajectory generation and tracking control of a single coaxial rotor UAV. Paper presented at: AIAA Guidance, Navigation, and Control (GNC) Conference; 2013; Boston, MA.
25. Gatzke BT. *Trajectory Optimization for Helicopter Unmanned Aerial Vehicles (UAVs)* [thesis]. Monterey, CA: Naval Postgraduate School; 2010.
26. Flash T, Hogan N. The coordination of arm movements: an experimentally confirmed mathematical model. *J Neurosci*. 1985;5:1688-1703.
27. Kyriakopoulos KJ, Saridis GN. Minimum jerk path generation. In: *Proceedings of 1988 IEEE International Conference on Robotics and Automation*; 1988; Philadelphia, PA.
28. Fliess M, Lévine J, Martin P, Rouchon P. A lie-backlund approach to equivalence and flatness of nonlinear systems. *IEEE Trans Autom Control*. 1999;44(5):922-937.
29. De Boor C. *A Practical Guide to Splines*. New York, NY: Springer-Verlag; 1978. *Applied Mathematical Sciences*; vol. 27.
30. Garg D, Patterson M, Hager WW, Rao AV, Benson DA, Huntington GT. A unified framework for the numerical solution of optimal control problems using pseudospectral methods. *Automatica*. 2010;46(11):1843-1851.
31. Milam MB. *Real-Time Optimal Trajectory Generation for Constrained Dynamical Systems* [PhD thesis]. Pasadena, CA: California Institute of Technology; 2003.
32. Louembet C, Cazaurang F, Zolghadri A, Charbonnel C, Pittet C. Path planning for satellite slew manoeuvres: a combined flatness and collocation-based approach. *IET Control Theory Appl*. 2009;3(4):481-491.
33. Gill PE, Murray W, Saunders MA. SNOPT: an SQP algorithm for large-scale constrained optimization. *SIAM Rev*. 2005;47(1):99-131.
34. Delaleau E, Rudolph J. Control of flat systems by quasi-static feedback of generalized states. *Int J Control*. 1998;71(5):745-765.
35. Rudolph J. Flatness-based control by quasi-static feedback illustrated on a cascade of two chemical reactors. *Int J Control*. 2000;73(2):115-131.
36. Kolar B, Rams H, Schlacher K. Time-optimal flatness based control of a gantry crane. *Control Eng Pract*. 2017;60:18-27.

37. Rudolph J, Delaleau E. Some examples and remarks on quasi-static feedback of generalized states. *Automatica*. 1998;34(8):993-999.
38. Hagenmeyer V, Delaleau E. Exact feedforward linearization based on differential flatness. *Int J Control*. 2003;76(6):537-556.
39. Kelemen M. A stability property. *IEEE Trans Autom Control*. 1986;31(8):766-768.
40. Mallavalli S, Fekih A. A fault tolerant tracking control for a quadrotor UAV subject to simultaneous actuator faults and exogenous disturbances. *Int J Control*. 2018:1-14.

**How to cite this article:** Abadi A, El Amraoui A, Mekki H, Ramdani N. Optimal trajectory generation and robust flatness-based tracking control of quadrotors. *Optim Control Appl Meth*. 2019;40:728-749. <https://doi.org/10.1002/oca.2508>