Controller and Trajectory Optimization for a Quadrotor UAV with Parametric Uncertainty

Ali Srour, Antonio Franchi, Paolo Robuffo Giordano

Controller and Trajectory Optimization for a Quadrotor UAV with Parametric Uncertainty

Ali Srour[1], Antonio Franchi[2,3,4], Paolo Robuffo Giordano[1]

***Abstract*— In this work, we exploit the recent notion of *closed-loop state sensitivity* to critically compare three typical controllers for a quadrotor UAV with the goal of evaluating the impact of controller choice, gain tuning and shape of the reference trajectory in minimizing the sensitivity of the closed-loop system against uncertainties in the model parameters. To this end, we propose a novel optimization problem that takes into account both the shape of the reference trajectory and the controller gains. We then run a large statistical campaign for comparing the performance of the three controllers which provides some interesting insight for the goal of increasing closed-loop robustness against parametric uncertainties.**

I. INTRODUCTION

The requirement for autonomous/intelligent systems such as robots to operate in unpredictable and/or uncertain real-world conditions is one of the biggest challenges in our community. In fact, whether directly or indirectly, robot decisions and control actions are based on a model of the "world", which is oftentimes only a rough representation of the physical reality. When considering motion tasks, one of the common sources of uncertainty is in the robot model parameters that can be hard to measure or also vary at runtime. Adaptive [1] or robust control [2] techniques are typical ways to deal with parametric uncertainty by either trying to estimate the parameters online, or by trading off performance for robustness vs. parametric uncertainty. Also, Model Predictive Control [3], [4] predicts system behavior and optimizes a cost function over a finite time horizon using a dynamic model of the system and its environment. However, real-time rescheduling for optimization at each time step can result in significant computational overhead. Another recent approach to deal with these issues has been proposed in [5]–[10] where several metrics based on the notion of *closed-loop state/input sensitivity* have been introduced. These metrics are able to quantify how uncertainties in the parameters of a robot model can affect the evolution of the robot states and inputs in *closed-loop* and as a function of the reference trajectory being tracked (i.e., the 'motion task'). One can then attempt to increase robustness against uncertain parameters by planning a suitable feedforward (or desired) trajectory that minimizes the sensitivity metrics, thereby generating an intrinsically-robust and control-aware motion plan. These ideas have also been extended to the problem of combining state observabilty with parametric robustness in [11], and in [12] a recent extension is proposed for obtaining the 'tubes' of perturbed trajectories given a known range of parametric deviation.

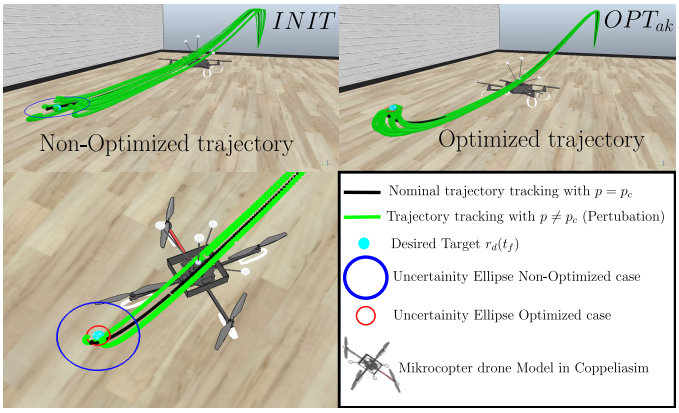

Fig. 1. Drone trajectory tracking under parametric uncertainties (green) to reach a target $\boldsymbol{s}_d(t_f)$ (light blue sphere). Non-optimized ($INIT$, top left) and optimized (OPT_{ak}, top right) trajectories are compared. The size of the uncertainty ellipses are compared in the bottom left, where the blue and red ellipses correspond to $INIT$ and OPT_{ak}, respectively. A video demonstration is available at: https://youtu.be/VBRL8XDiJ5c.

All these previous works have shown very promising results by focusing on the optimization of the reference trajectory to be tracked by the robot under suitable state and input constraints. However, one may also consider the concurrent optimization of the gains of the chosen control strategy (besides the reference trajectory) for the purpose of minimizing a sensitivity metric. In this paper we then consider this latter possibility applied, as case study, to a tracking task for a quadrotor UAV under three popular control strategies: a near-hovering (or linear) controller, a nonlinear geometric controller, and a sliding-mode controller. We present and discuss the results of a thorough statistical analysis aimed at comparing the three controllers in different conditions (non-optimized, by only optimizing for the reference trajectory, by optimizing for the control gains only and by optimizing for both the trajectory and the control gains). The aim is twofold: (i) to show how gain optimization (or tuning) can be done while taking into account a concrete robustness metric (the state sensitivity vs. parametric uncertainties) and (ii) to get statistical insights

[1]A. Srour and P. Robuffo Giordano are with CNRS, Univ Rennes, Inria, IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France. email: {ali.srour,prg}@irisa.fr

[2]A. Franchi is with the Robotics and Mechatronics Department, Electrical Engineering, Mathematics, Computer Science (EEMCS) Faculty, University of Twente, 7500 AE Enschede, The Netherlands; and with the

[3]Department of Computer, Control and Management Engineering, Sapienza University of Rome, 00185 Rome, Italy; and with the

[4]LAAS-CNRS, Université de Toulouse, 31000 Toulouse, France email: a.franchi@utwente.nl

This work was supported by the project ANR-20-CE33-0003 "CAMP" and Horizon 2020 research and innovation programme [grant agreement No. 871479] AERIAL-CORE

into the behavior, performance and robustness of the three controllers in different conditions.

The rest of the paper is structured as follows: in Sect. II we recall the main notions related to the *closed-loop state sensitivity*. Then in Sect. III we present the optimization problems considered in this work, and we detail the quadrotor model and the three tracking controller. Section IV discusses the results of the statistical analysis on the controller performance, and Sect. V concludes the paper and draws some future perspectives.

II. PRELIMINARIES

In this section we briefly summarize for the reader's convenience the main notions related to the *closed-loop state sensitivity* introduced in [5], [6] and recently extended in [12]. We consider a robot model

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},\, \boldsymbol{u},\, \boldsymbol{p}) \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^{n_q}$ is the state, $\boldsymbol{u} \in \mathbb{R}^{n_u}$ the control inputs and $\boldsymbol{p} \in \mathbb{R}^{n_p}$ a vector of (possibly uncertain) model parameters. We assume some quantity of interest $\boldsymbol{s}(\boldsymbol{x}) \in \mathbb{R}^{n_s}$ (e.g., the position of the robot center) needs to track a desired trajectory $\boldsymbol{s}_d(t,\, \boldsymbol{a})$ defined for $t \in [t_0,\, t_f]$ and function of a finite set of parameters $\boldsymbol{a} \in \mathbb{R}^{n_a}$. For this task a tracking controller is used

$$\begin{cases} \dot{\boldsymbol{\xi}} = \boldsymbol{g}\,(\boldsymbol{\xi},\, \boldsymbol{x},\, \boldsymbol{a},\, \boldsymbol{p}_c,\, \boldsymbol{k}_c,\, t) \\ \boldsymbol{u} = \boldsymbol{h}\,(\boldsymbol{\xi},\, \boldsymbol{x},\, \boldsymbol{a},\, \boldsymbol{p}_c,\, \boldsymbol{k}_c,\, t) \end{cases}, \tag{2}$$

which is evaluated at a nominal value $\boldsymbol{p}_c$ for the uncertain parameters $\boldsymbol{p}$. The vector $\boldsymbol{\xi} \in \mathbb{R}^{n_\xi}$ represents the possible internal controller states (e.g., an integral action) and $\boldsymbol{k}_c \in \mathbb{R}^{n_k}$ is the vector of control gains.

The state sensitivity for the closed-loop system (1–2) is defined as

$$\boldsymbol{\Pi}(t) = \left.\frac{\partial \boldsymbol{x}(t)}{\partial \boldsymbol{p}}\right|_{\boldsymbol{p}=\boldsymbol{p}_c} \tag{3}$$

and the input sensitivity

$$\boldsymbol{\Theta}(t) = \left.\frac{\partial \boldsymbol{u}(t)}{\partial \boldsymbol{p}}\right|_{\boldsymbol{p}=\boldsymbol{p}_c}. \tag{4}$$

Matrix $\boldsymbol{\Pi}(t)$ quantifies how variations in the parameters $\boldsymbol{p}$ around a nominal value $\boldsymbol{p}_c$ will affect the evolution of the states $\boldsymbol{x}$ (in closed-loop). Analogously, matrix $\boldsymbol{\Theta}(t)$ relates variations in $\boldsymbol{p}$ to variations in the inputs $\boldsymbol{u}$. A closed-form expression for matrixes $\boldsymbol{\Pi}(t)$ and $\boldsymbol{\Theta}(t)$ is in general not available, but in [5], [6] it is shown how these two quantities can be easily evaluated via forward integration of a differential equation along the system trajectories.

Matrixes $\boldsymbol{\Pi}(t)$ and $\boldsymbol{\Theta}(t)$ can be used for several purposes such as optimization of the reference trajectory $\boldsymbol{s}_d(t,\, \boldsymbol{a})$ for producing minimally sensitive motion plans (by minimizing some combined norm of $\boldsymbol{\Pi}(t)$ and $\boldsymbol{\Theta}(t)$ as done in [5], [6]). An extension of these ideas has been recently proposed in [12] where a suitable *weighted* norm for the state sensitivity $\boldsymbol{\Pi}(t)$ is introduced. Assume that each parameter p_i can vary in a given range δp_i centered at a nominal p_{c_i}

$$p_i \in [p_{c_i} - \delta p_i,\, p_{c_i} + \delta p_i] \tag{5}$$

and define the diagonal weight matrix $\boldsymbol{W} = \mathrm{diag}(\delta p_i^2)$. Letting $\Delta\boldsymbol{p} = \boldsymbol{p} - \boldsymbol{p}_c$, an ellipsoid in parameter space centered at $\boldsymbol{p}_c$ and with semi-axes δp_i has equation

$$\Delta\boldsymbol{p}^T \boldsymbol{W}^{-1} \Delta\boldsymbol{p} = 1. \tag{6}$$

From (6) and (3–4) one can obtain the corresponding ellipsoids in state space

$$\Delta\boldsymbol{x}^T (\boldsymbol{\Pi}\boldsymbol{W}\boldsymbol{\Pi}^T)^{-1} \Delta\boldsymbol{x} = 1 \tag{7}$$

and in input space

$$\Delta\boldsymbol{u}^T (\boldsymbol{\Theta}\boldsymbol{W}\boldsymbol{\Theta}^T)^{-1} \Delta\boldsymbol{u} = 1, \tag{8}$$

see [12]. Here $\Delta\boldsymbol{x}$ stands for $\Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}_{nom}$ where $\boldsymbol{x}_{nom}$ is the state evolution of (1–2) in the unperturbed case $\boldsymbol{p} = \boldsymbol{p}_c$, and analogously for $\Delta\boldsymbol{u} = \boldsymbol{u} - \boldsymbol{u}_{nom}$.

The state and input space ellipsoids can be used for different purposes. First of all, one can define a 'weighted sensitivity norm' by considering the eigenvalues λ_i of the kernel matrix $\boldsymbol{\Pi}\boldsymbol{W}\boldsymbol{\Pi}^T$ in (7). In particular, in [12] and in this work we consider the following matrix norm

$$\|\boldsymbol{\Pi}\|_W = \max(\lambda_i(\boldsymbol{\Pi}\boldsymbol{W}\boldsymbol{\Pi}^T)) \tag{9}$$

which represents the largest (worst-case) deviation of the states $\boldsymbol{x}$ assuming a parametric uncertainty as in (5). Furthermore, one can also exploit (7–8) for obtaining the tubes of perturbed trajectories for the individual components of the states and the inputs. By again referring to [12] for all details, for each direction of interest in the state space one can obtain the 'tube radius' $r_i(t)$ such that

$$q_{nom,i}(t) - r_i(t) \le q_i(t) \le q_{nom,i}(t) + r_i(t). \tag{10}$$

where, as usual, $q_{nom,i}(t)$ is the behavior of the state $q_i(t)$ in the unperturbed case $\boldsymbol{p} = \boldsymbol{p}_c$. Equation (10) bounds from above/below the envelope of perturbed states when the parameter uncertainty is bounded as in (5), and an analogous upper/lower bound can also be obtained for the inputs components $u_i(t)$.

III. OPTIMIZATION PROBLEM

As explained in Sect. I, in this work we are interested in comparing the performance of three typical trajectory tracking controllers for a quadrotor UAV in the context of minimally sensitive trajectory optimization. To this end, and considering the various quantities introduced in Sect. II, we consider a first trajectory optimization problem

$$\begin{aligned} \boldsymbol{a}^* = \arg\min_{\boldsymbol{a}} \|\boldsymbol{\Pi}(t_f)\|_W \\ s.t. \quad \boldsymbol{M}\boldsymbol{a} = \boldsymbol{b} \\ U_{min,i} \le u_{nom,i}(t) - r_i(t)\ \forall i\, \forall t \in [t_0,\, t_f] \\ u_{nom,i}(t) + r_i(t) \le U_{max,i}\ \forall i\, \forall t \in [t_0,\, t_f]. \end{aligned} \tag{11}$$

We seek to find the optimal value $\boldsymbol{a}^*$ for the shape parameter $\boldsymbol{a}$ of the reference trajectory $\boldsymbol{s}_d(\boldsymbol{a}, , t)$ for minimizing the weighted norm (9) at the final time t_f. The constraints consist of given initial/final conditions for $\boldsymbol{s}_d(\boldsymbol{a}, , t)$, represented by the linear constraints $\boldsymbol{M}\boldsymbol{a} = \boldsymbol{b}$, and constraints that bound the envelope of perturbed inputs within actuation limits

$U_{min,i} \leq U_{max,i}$, ensuring that the tracking of the optimized reference trajectory will be feasible for any value of the uncertain parameters $\boldsymbol{p}$ in the range (5). These constraints leverage the 'tubes' as in (10) but are evaluated for the inputs. Variants of this problem have already been considered in [5], [6], [12] but by always focusing on a single control strategy.

We also consider a second optimization problem

$$\begin{aligned} \boldsymbol{k}_c^* = \arg\min_{\boldsymbol{k}_c} \|\boldsymbol{\Pi}(t_f)\|_W \\ s.t. \quad \boldsymbol{M}\boldsymbol{a} = \boldsymbol{b} \\ K_{min,i} \leq k_{c_i} \leq K_{max,i} \\ U_{min,i} \leq u_{nom,i}(t) - r_i(t) \; \forall i \, \forall t \in [t_0,\, t_f] \\ u_{nom,i}(t) + r_i(t) \leq U_{max,i} \; \forall i \, \forall t \in [t_0,\, t_f] \end{aligned} \tag{12}$$

where we optimize w.r.t. the control gains $\boldsymbol{k}_c$ inside suitable bounds $0 < K_{min,i} \leq K_{max,i}$. We then consider a third optimization problem that optimizes both the shape parameter $\boldsymbol{a}$ and the control gains $\boldsymbol{k}_c$ subject to the same constraints as in equation (12).

$$\begin{aligned} (\boldsymbol{a}^*,\, \boldsymbol{k}_c^*) = \arg\min_{\boldsymbol{a},\, \boldsymbol{k}_c} \|\boldsymbol{\Pi}(t_f)\|_W \\ s.t. \quad \boldsymbol{M}\boldsymbol{a} = \boldsymbol{b} \\ K_{min,i} \leq k_{c_i} \leq K_{max,i} \\ U_{min,i} \leq u_{nom,i}(t) - r_i(t) \; \forall i \, \forall t \in [t_0,\, t_f] \\ u_{nom,i}(t) + r_i(t) \leq U_{max,i} \; \forall i \, \forall t \in [t_0,\, t_f] \end{aligned} \tag{13}$$

We note that in the previous problems we focus on the minimization of the state sensitivity norm *at the final time* t_f. In other words, we look for the combination of reference trajectory and/or control gains that will result in the largest tracking accuracy at the final time (for, e.g., reaching at best a target location). Of course, different choices are also possible such as minimization of (the integral) of the sensitivity norm along the whole trajectory (for increasing the average tracking accuracy during motion). Note also that additional constraints, such as obstacle avoidance, could be easily added to the optimization problem by exploiting the tubes on the states, see [13] for an example in this sense. One can also optimize the sensitivity at specific waypoints along the trajectory where, for instance, the effects of uncertainties in high speed regimes can be minimized.

A. Quadrotor Model

Let $\boldsymbol{x} = (\boldsymbol{r},\, \boldsymbol{v},\, \boldsymbol{q},\, \boldsymbol{\omega}) \in \mathbb{R}^6 \times \mathbb{S}^3 \times \mathbb{R}^3$ be the quadrotor state vector consisting of the world-frame position $\boldsymbol{r} = (r_x,\, r_y,\, r_z)$ and velocity $\boldsymbol{v} = (v_x,\, v_y,\, v_z)$, the rotation from body to world frame represented by a unit quaternion $\boldsymbol{q} = (q_w,\, q_x,\, q_y,\, q_z)$ and the body-frame angular velocity $\boldsymbol{\omega} = (\omega_x,\, \omega_y,\, \omega_z)$. The quadrotor model is

$$\begin{cases} \dot{\boldsymbol{r}} = \boldsymbol{v} \\ \dot{\boldsymbol{v}} = \dfrac{f}{m} \begin{bmatrix} 2(q_w q_y + q_x q_z) \\ 2(q_y q_z - q_w q_x) \\ 1 - 2(q_x^2 + q_y^2) \end{bmatrix} - \boldsymbol{g} \\ \dot{\boldsymbol{q}} = \dfrac{1}{2} \begin{bmatrix} 0 \\ \boldsymbol{w} \end{bmatrix} \otimes \boldsymbol{q} \\ \dot{\boldsymbol{w}} = \boldsymbol{I}^{-1}(\boldsymbol{\tau} - \boldsymbol{w} \times \boldsymbol{I}\boldsymbol{w}) \end{cases} \tag{14}$$

In this model, f and $\boldsymbol{\tau}$ denote the total thrust and body-frame moment, $\boldsymbol{g} = (0,\, 0,\, g)$ is the gravity vector in world frame, and m and $\boldsymbol{I}$ represent the mass and body-frame inertia. Although quaternions are used to represent the quadrotor orientation in model (14), the conversion of quaternions to the roll ϕ, pitch θ, and yaw ψ angles is also considered in two of the control strategies.

Let w_i be the squared velocity of the i-th propeller and define the quadrotor control input $\boldsymbol{u} = (w_1, , \ldots, , w_4)$. An allocation matrix is used to relate the inputs $\boldsymbol{u}$ (i.e., the squared propeller speeds) to the thrust/torques $(f, , \boldsymbol{\tau})$.

$$\begin{aligned} \begin{bmatrix} f \\ \boldsymbol{\tau} \end{bmatrix} = k_f \begin{bmatrix} 1 & 1 & 1 & 1 \\ -g_y & L - g_y & -g_y & -(L + g_y) \\ -(L - g_x) & g_x & L + g_x & g_x \\ k_t & -k_t & k_t & -k_t \end{bmatrix} \boldsymbol{u} = \\ = \boldsymbol{T}\boldsymbol{u} \end{aligned} \tag{15}$$

The aerodynamic coefficients of the propellers are denoted by k_f and k_t, the x and y coordinates of the barycenter in the body frame are denoted by g_x and g_y, and the length of the propeller arms is denoted by L (see [14] for a more detailed derivation). These coefficients are difficult to measure reliably and may change at runtime depending on the flight regime. Therefore, the vector of uncertain parameters considered in this work is $\boldsymbol{p} = (k_f,\, k_t,\, g_x,\, g_y)$.

B. Near-Hovering Controller (NH)

The first control strategy considered is the *near-hovering* or linear controller, commonly used for its ease of implementation and tuning. However, it only performs well and maintains stability near a hovering state with small roll/pitch angles [15]. Tracking is done using a 4D reference trajectory $\boldsymbol{s}_d(t) = (\boldsymbol{r}_d(t), \psi_d(t))$, which includes the quadrotor 3D position $\boldsymbol{r}_d(t) = (x_d(t),\, y_d(t),\, z_d(t))$ and yaw angle $\psi_d(t)$. The final control equations are

$$\begin{cases} \ddot{x}_r = \ddot{r}_{d_x} + k_{d_x}(\dot{r}_{d_x} - \dot{x}) + k_{p_x}(r_{d_x} - x) \\ \quad + k_{i_x} \displaystyle\int (r_{d_x} - x) \mathrm{d}t \\ \ddot{y}_r = \ddot{r}_{d_y} + k_{d_y}(\dot{r}_{d_y} - \dot{y}) + k_{p_y}(r_{d_y} - y) \\ \quad + k_{i_y} \displaystyle\int (r_{d_y} - y) \mathrm{d}t \\ \phi_d = -\arcsin\left(\dfrac{m}{f}\left(-\sin(\psi)\ddot{x}_r + \cos(\psi)\ddot{y}_r\right)\right) \\ \theta_d = \arcsin\left(\dfrac{m}{f\cos(\phi)}(\cos(\psi)\ddot{x}_r + \sin(\psi)\ddot{y}_r)\right) \\ f = \dfrac{m}{\cos(\phi)\cos(\theta)}\left(g + \ddot{r}_{d_z} + k_{d_z}(\dot{r}_{d_z} - \dot{z}) + k_{p_z}(r_{d_z} - z)\right. \\ \quad \left. + k_{i_z} \displaystyle\int (r_{d_z} - z) \mathrm{d}t\right) \\ \tau_x = -k_{d_\phi}\omega_x + k_{p_\phi}(\phi_d - \phi) \\ \tau_y = -k_{d_\theta}\omega_y + k_{p_\theta}(\theta_d - \theta) \\ \tau_z = -k_{d_\psi}\omega_z + k_{p_\psi}(\psi_d - \psi) \end{cases} . \tag{16}$$

Finally, the quadrotor inputs are computed as

$$\boldsymbol{u} = \boldsymbol{T}|_{\boldsymbol{p}_c}^{-1} \begin{bmatrix} f \\ \boldsymbol{\tau} \end{bmatrix} \tag{17}$$

where the the allocation matrix $\boldsymbol{T}$ from (15) is evaluated at the nominal parameters $\boldsymbol{p}_c$.

The vector of control gains $\boldsymbol{k}_c$ considered in problem (12–13) for this controller consists of all the 15 gains in (16), that is, the three gains $(k_{p_\bullet}, k_{d_\bullet}, k_{i_\bullet})$ for the x, y and z channels, and the two gains $(k_{p_\bullet}, k_{d_\bullet})$ for the ϕ, θ and ψ channels.

C. 3D Geometric Controller (LEE)

The 3D Geometric controller [16] is a popular control strategy that differs from the near-hovering controller by not relying on approximations of the quadrotor dynamics and providing almost global stability. However, it is more challenging to tune than the near-hovering controller.

Let $\boldsymbol{R}(\boldsymbol{q}) \in SO(3)$ be the rotation matrix associated to the quaternion $\boldsymbol{q}$, and let $\boldsymbol{e}_3 = (0,\, 0,\, 1)$. The idea behind this control strategy is to design a desired thrust vector $f\boldsymbol{R}_d\boldsymbol{e}_3$ for stabilizing the translational dynamics (i.e., tracking the reference $\boldsymbol{r}_d(t)$) while the torques $\boldsymbol{\tau}$ try to align the quadrotor z axis $\boldsymbol{R}\boldsymbol{e}_3$ with the desired one and also track the reference signal for the yaw angle $\psi_d(t)$. The final controller equations are (see [16])

$$\begin{cases} \boldsymbol{e}_r = \boldsymbol{r}_d - \boldsymbol{r}, \quad \boldsymbol{e}_v = \dot{\boldsymbol{r}}_d - \boldsymbol{v} \\ \boldsymbol{b}_{3_d} = \dfrac{\boldsymbol{K}_r\boldsymbol{e}_r + \boldsymbol{K}_v\boldsymbol{e}_v + \boldsymbol{K}_i \int \boldsymbol{e}_r \mathrm{d}t + mg\boldsymbol{e}_3 + m\ddot{\boldsymbol{r}}_d}{\|\boldsymbol{K}_r\boldsymbol{e}_r + \boldsymbol{K}_v\boldsymbol{e}_v + \boldsymbol{K}_i \int \boldsymbol{e}_r \mathrm{d}t + mg\boldsymbol{e}_3 + m\ddot{\boldsymbol{r}}_d\|} \\ \boldsymbol{b}_{1_d} = (\cos\psi_d,\, \sin\psi_d,\, 0), \quad \boldsymbol{b}_{2_d} = (\boldsymbol{b}_{3_d} \times \boldsymbol{b}_{1_d})/\|\boldsymbol{b}_{3_d} \times \boldsymbol{b}_{1_d}\| \\ \boldsymbol{R}_d = [\boldsymbol{b}_{2_d} \times \boldsymbol{b}_{3_d}\ \boldsymbol{b}_{2_d}\ \boldsymbol{b}_{3d}] \\ \boldsymbol{e}_R = \dfrac{1}{2}(\boldsymbol{R}_d^T\boldsymbol{R} - \boldsymbol{R}^T\boldsymbol{R}_d)^\vee, \quad \boldsymbol{e}_\omega = \boldsymbol{\omega} - \boldsymbol{R}^T\boldsymbol{R}_d\boldsymbol{\omega}_d \\ f = \left(\boldsymbol{K}_r\boldsymbol{e}_r + \boldsymbol{K}_v\boldsymbol{e}_v + \boldsymbol{K}_i \int \boldsymbol{e}_r \mathrm{d}t + mg\boldsymbol{e}_3 + m\ddot{\boldsymbol{r}}_d\right)^T \boldsymbol{R}\boldsymbol{e}_3 \\ \boldsymbol{\tau} = -\boldsymbol{K}_R\boldsymbol{e}_R - \boldsymbol{K}_\omega\boldsymbol{e}_\omega \end{cases} \tag{18}$$

where $()^\vee$ is the usual 'vee' map taking a skew-symmetric matrix into the associated 3D vector, and $\boldsymbol{K}_r$, $\boldsymbol{K}_v$, $\boldsymbol{K}_i$, $\boldsymbol{K}_R$, $\boldsymbol{K}_\omega$ are 3×3 diagonal control gain matrixes.

We note that, compared to the original [16], the moment equation (last row of (18)) neglects the compensation for the gyroscopic term as customary in actual implementations of this controller. Note also that, as in the previous near-hovering case, the actual quadrotor inputs $\boldsymbol{u}$ are retrieved by plugging in (17) the thrust/torques $(f, \boldsymbol{\tau})$ computed from (18).

The vector of control gains $\boldsymbol{k}_c$ considered in problem (12–13) for this controller consists of all the 15 gains in (18) (the diagonal entries of $\boldsymbol{K}_r$, $\boldsymbol{K}_v$, $\boldsymbol{K}_i$, $\boldsymbol{K}_R$, $\boldsymbol{K}_\omega$).

D. Sliding Mode Controller (SMC)

The last controller considered in this work is based on sliding mode, which is a control technique known to generally deal well with model uncertain parameters [17]–[19]. The controller is built by following the typical steps of the sliding mode design. We start by considering the vertical z dynamics and define the error $e_z = r_{d_z} - z$ and the sliding surface $\sigma_z = \dot{e}_z + \lambda_z e_z$ where $\lambda_z > 0$ is a control gain. The Lyapunov candidate $V_z = 1/2\ \sigma_z^2$ has time derivative along the system trajectories

$$\dot{V}_z = \sigma_z\dot{\sigma}_z = \sigma_z\left(\ddot{r}_{d_z} - \frac{f}{m}\cos(\phi)\cos(\theta) + g + \lambda_z\dot{e}_z\right). \tag{19}$$

We then choose the thrust f as

$$f = \frac{m}{(\cos(\phi)\cos(\theta))}(g + \ddot{r}_{d_z} + k_{1z}\tanh(\sigma_z) + k_{2z}\sigma_z) \tag{20}$$

where the $\tanh(\cdot)$ provides a smooth approximation of the classical $\mathrm{sign}(\cdot)$ used in sliding mode. Following analogous arguments, one can define the torque commands as

$$\begin{cases} \tau_x = k_{1\phi}\tanh(\sigma_\phi) + k_{2\phi}\sigma_\phi \\ \tau_y = k_{1\theta}\tanh(\sigma_\theta) + k_{2\theta}\sigma_\theta \\ \tau_z = k_{1\psi}\tanh(\sigma_\psi) + k_{2\psi}\sigma_\psi \end{cases} \tag{21}$$

where $\sigma_\phi = \dot{e}_\phi + \lambda_\phi e_\phi$, $\sigma_\theta = \dot{e}_\theta + \lambda_\theta e_\theta$ and $\sigma_\psi = \dot{e}_\psi + \lambda_\psi e_\psi$ are the sliding surfaces, and $e_\phi = \phi_d - \phi$, $e_\theta = \theta_d - \theta$ and $e_\psi = \psi_d - \psi$. The desired roll/pitch angles are defined as in (16) but with

$$\begin{cases} \ddot{x}_r &= \ddot{r}_{d_x} + k_{1_x}\tanh\sigma_x + k_{2_x}\sigma_x \\ \ddot{y}_r &= \ddot{r}_{d_y} + k_{1_y}\tanh\sigma_y + k_{2_y}\sigma_y \end{cases} \tag{22}$$

Here, again $\sigma_x = \dot{e}_x + \lambda_x e_x$ and $e_x = r_{d_x} - r_x$, and analogously for σ_y.

The vector of control gains $\boldsymbol{k}_c$ considered in problem (12–13) consists of all the 18 gains in (20–22), that is, the 'sliding' gains $\lambda_\bullet$ and the two gains $(k_{1_\bullet}, k_{2_\bullet})$ for the x, y, z, ϕ, θ, and ψ channels.

IV. STATISTICAL ANALYSIS

We now report the results of a statistical analysis for the three controllers of Sect. III when considering the three optimization problems (11–13). For the analysis we generate $N_{traj} = 25$ initial trajectories $\boldsymbol{s}_d(\boldsymbol{a},\, t)$ starting at the origin and coping with the initial/final state constraints and input saturations as in (11–13). These initial trajectories are rest-to-rest motions (from a hovering state to a hovering state) with a final position randomly generated inside a spherical shell of 4 to 6 m centered at the origin, and a final yaw angle randomly generated in the interval $[-\pi/2,\, \pi/2]$. The trajectories $\boldsymbol{s}_d(\boldsymbol{a},\, t)$ are implemented as piecewise Bezier curves as already done in [5], [6], [20] with the goal of achieving a smooth curvature with little snap, as in [21], [22]. The framework is implemented in Python and utilizes the COBYLA [23] nonlinear optimizer from the nlopt toolbox, along with the symbolic toolbox SymEngine to represent the system symbolically. Additionally, the Jitcode [24] framework is employed for in-time compilation of ordinary differential equations. In the given framework, the NH and LEE controllers have optimization timeframes of 2-3 minutes for each trajectory type, while the SMC controller requires 3-4 minutes per trajectory type.

For the sake of exposition, in the following we will let $INIT$ represent an initial (non-optimized) trajectory, OPT_a the corresponding optimized trajectory solution of (11), OPT_k the corresponding optimized trajectory solution of (12) and OPT_{ak} the corresponding optimized trajectory solution of (13). We thus obtain a total of $4N_{traj}$ trajectories:

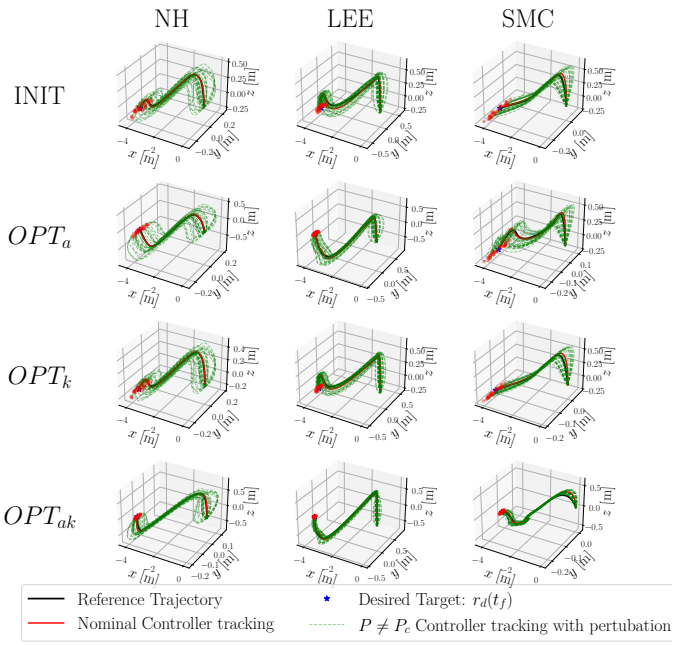

Fig. 2. Trajectory tracking results for NH, LEE, and SMC controllers are shown in the left, center, and right columns respectively, for the $INIT$, OPT_a, OPT_k, and OPT_{ak} cases. Each run has N_{sim} trajectories (green), starting from the origin and ending at the same target location, with the parameters $\boldsymbol{p}$ being randomly drawn from (6). The final positions $\boldsymbol{r}(t_f)$ of the perturbed trajectories are denoted by small red spheres at $t = t_f$, forming a point cloud that constitutes the empirical variance ellipsoid centered at $\boldsymbol{r}_d(t_f)$.

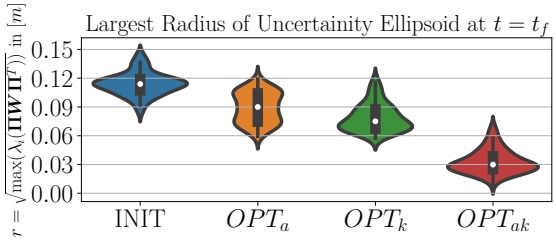

Fig. 3. Average largest radius of the uncertainty ellipsoids at $t = t_f$, calculated from the combined trajectories of all three controllers, thus reflecting the impact of the optimizations in reducing the ellipsoid size.

the N_{traj} initial ones and the $3N_{traj}$ optimized ones. Each of these $4N_{traj}$ trajectories is then provided as reference to the three quadrotor controllers in $N_{sim} = 30$ simulation runs where in each run the uncertain parameters $\boldsymbol{p}$ are uniformly sampled to lie inside volume of the ellipsoid (6). The uncertainty ranges δp_i for building matrix $\boldsymbol{W}$ are set at 15% of the nominal parameter values $\boldsymbol{p}_c$. The control gain bounds in (12–13) are set to be 50% and 200% of the initial values used for the INIT case. Therefore problems (11–13) will aim at minimizing the sensitivity of the quadrotor position $\boldsymbol{r}(t_f)$ at *the final time* t_f against variations in the parameters k_f, k_t, g_x, g_y.

For a fist assessment, in Fig. 2 we report the N_{sim} 'perturbed runs' for one INIT trajectory (1st row) and the

Average Percentage Contribution of each Parameter p_i on the sensitivity of the states q_i

Param\State	x	y	z
kf	2.46%	1.03%	98.36%
kt	0.00	0.00	0.00
gx	39.28%	58.91%	0.69%
gy	58.27%	40.06%	0.95%

TABLE I

PERCENTAGE CONTRIBUTION OF EACH PARAMETER IN $\boldsymbol{p}$ ON THE FINAL SENSITIVITY OF THE STATES $\boldsymbol{r}(t_f)$

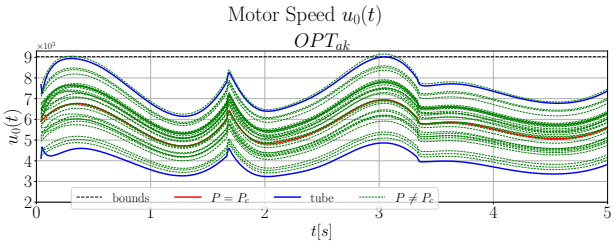

Fig. 4. (Red) behavior of one control input with the nominal parameters ($\boldsymbol{p} = \boldsymbol{p}_c$) in the OPT_{ak} case. The corresponding uncertainty tube is shown in blue, and the green lines represent the perturbed inputs when $p \neq p_c$ for N_{sim} runs. The allowable input range is depicted by (dashed black) lines.

three corresponding OPT_a (2nd row), OPT_k (3rd row) and OPT_{ak} (4th row) when considering the three controllers of the previous section on every column. The plots reports different target reach $\boldsymbol{r}(t_f)$ after each perturbed simulation (in green) forming the empirical variance ellipsoid (in red) of the final quadrotor position $\boldsymbol{r}(t_f)$ centered at the desired $\boldsymbol{r}_d(t_f)$. By referring to Fig. 2 and Fig. 3, one can note how the ellipsoids are reduced in the OPT_a, OPT_k cases and even further in the OPT_{ak} cases w.r.t. the INIT cases as expected. The largest radius of the state ellipsoid is decreased on average for the OPT_a and OPT_k from (12 cm) to (9 and 7.5 cm) respectively but further decreases in the OPT_{ak} to (3 cm). This then confirms that the sensitivity cost minimized in (11–13) captures well how variations in the parameters affect deviations in the considered states.

Furthermore, one may note that in the optimized cases OPT_a and OPT_{ak} where the shape parameter $\boldsymbol{a}$ of the reference trajectory is optimized, the optimal trajectory performs an initial maneuver for then arriving at the target location with an almost vertical straight line motion (see Fig. 1 and the attached video for more details). This trend has been empirically verified in almost all the simulation runs and is shared by all the employed controllers. A likely explanation is that a final (almost) vertical motion allows to better minimize the effects of the uncertainties in the considered parameters since the rotational dynamics is not excited during this phase, contrarily to the INIT trajectory where the quadrotor needs instead to perform a non-negligible tilting for reaching $\boldsymbol{r}_d(t_f)$ at rest. This insight can then be helpful for designing more robust hover-to-hover motions also in different conditions. This trend can also be deduced from Table. I that shows the percentage contribution of each parameter ($N_p = 4$) in $\boldsymbol{p}$ on the sensitivity of the

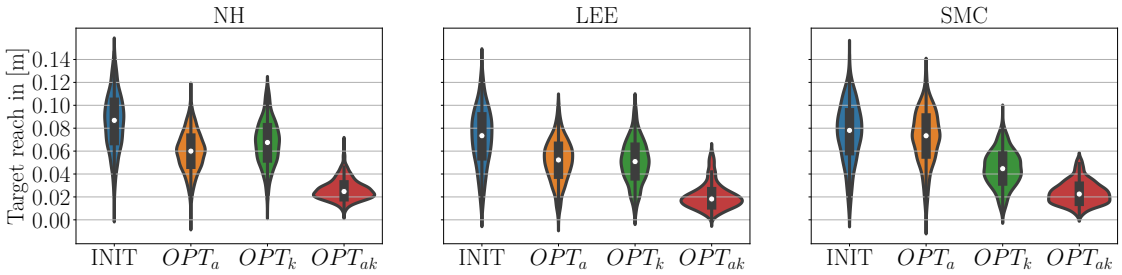

Fig. 5. Violin plots displaying the target reach (defined as $\|\boldsymbol{r}_d(t_f) - \boldsymbol{r}(t_f)\|$) in [m] for the different trajectory types using the three controllers NH, LEE and SMC. These results corresponds to the statistical campaign of trajectories ($N_{traj} = 25$) and perturbed simulations ($N_{sim} = 30$).

states $\boldsymbol{r}(t_f)$ defined by $C_{i,j}\%$ in (23) (see [25]). This analysis can effectively assess the impact of each parameter variation on the final target reach error at time t_f

$$C_{i,j}\% = \frac{\Pi_{ij}^2(t_f)\delta_{p_j}^2}{\sum_{j=1}^{N_p}\Pi_{ij}^2(t_f)\delta_{p_j}^2} \times 100 \tag{23}$$

Table. I shows that changes in the parameters g_x and g_y have the most significant impact on the sensitivity of states x and y, while state z is predominantly influenced by k_f. Consequently, when attempting to reach the final target, a vertical motion is primarily affected only by uncertainties in k_f, while the other three parameters affect the states in non-vertical motions. Furthermore, based on Table I, a 15% uncertainty in k_t does not affect the states $\boldsymbol{r}(t_f)$ and can then be disregarded.

Figure 4 reports the actuator speeds (in revolutions per minute) for the OPT_{ak} case. The red line represents the nominal actuator speed, while the green lines represent the actuation speeds in the N_{sim} perturbed runs. The dashed black lines denote the limit bounds of the actuator, and the blue line represents the input bound tubes, which ensures actuation feasibility when $\boldsymbol{p}$ varies in the ellipsoid (6).

Fig. 5 reports the so-called violin plots of the norm of the final *target reach* (defined as $\|\boldsymbol{r}_d(t_f) - \boldsymbol{r}(t_f)\|$) for the complete statistical campaign considering the three controllers and all the $4N_{traj}$ trajectories in the N_{sim} runs. The interquartile range is indicated by the thick black bar at the center, the median of the samples is shown by the white dot at the center, and the remaining portions of the distribution are indicated by the thin black line. Greater probabilities of the population adopting the specified value are represented by wider areas of the violin plot. In order to obtain a fair comparison of all controllers, we tuned their initial gains so as to have a comparable median in the INIT case, which is around 8 cm as can be seen in Fig. 5.

Observations show that all controllers demonstrate an enhanced average target reach in the OPT_a, OPT_k, and OPT_{ak} cases. In the OPT_a case, the NH and LEE controllers achieve similar results, with an average target reach of around 5.5 to 6 cm, while the SMC controller improvement is comparatively lower. This suggests that the desired trajectory shape is less critical for the SMC performance, but more relevant for the LEE and NH controllers. In the OPT_k case the NH controller average target reach is 7 cm, the LEE controller improvement is about 5 cm, and the SMC controller exhibits the most significant improvement of 4 cm. These results imply that the LEE and SMC controllers are more susceptible to controller gain tuning compared to the NH controller. Moreover, in the OPT_{ak} case, all controllers show significant improvement in both the average target reach and its variance. The median of the NH, LEE, and SMC controllers drops from about 8 cm in the $INIT$ case to around 2 cm for OPT_{ak} case (400% improvement), with a better distribution around the median.

Therefore, from this analysis we can conclude that the SMC controller is not particularly sensitive to the shape of the reference trajectory, but it is quite sensitive to the control gains which, if tuned well, can lead to good performance. The NH and LEE controllers are instead more affected by the trajectory shape in addition to the control gains. Finally, we can note that the NH and LEE controllers perform quite similarly when optimized. Our most likely explanation is that the controllers are evaluated in terms of the target reach $\|\boldsymbol{r}_d(t_f) - \boldsymbol{r}(t_f)\|$ at the final time, and not in terms of how well they are tracking the reference trajectory during motion. The LEE controller should be better at dealing with maneuvers involving, e.g., large tilting or accelerations, but it can have a comparable performance to the NH one when it comes to reach a final hovering state.

V. CONCLUSIONS

In this work we have proposed a novel optimization problem in which the effect of parametric uncertainties in a robot model (quantified by the notion of *state sensitivity*) can be minimized by acting on the reference trajectory to be tracked and/or on the control gains. A tracking task for a quadrotor UAV has been chosen as case study, by employing three popular tracking controllers widely used in the community. The results of a large statistical analysis have shown the effectiveness of the proposed joint optimization of the reference trajectory and gains in minimizing the effects of uncertainties in selected quadrotor parameters. In addition to identifying the most influential parameters on the system states, we can also gain valuable insights into controller performance and gain tuning robustness concerning parametric uncertainty.

We are now working on an experimental validation of these findings and, on a more methodological side, we are also interested in finding more principled ways to impose bounds on the control gains in problems (12-13) besides simple 'box constraints' (e.g., by explicitly considering the stability of the closed-loop via a Lyapunov function).

REFERENCES

[1] K. J. Astrom and B. Wittenmark, "Adaptive control 2nd edition," *Addison-Wesley Pub Co.*, vol. 1994, 1994.

[2] K. Zhou and J. C. Doyle, *Essentials of robust control*. Prentice hall Upper Saddle River, NJ, 1998, vol. 104.

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

[4] D. Hanover, P. Foehn, S. Sun, E. Kaufmann, and D. Scaramuzza, "Performance, precision, and payloads: Adaptive nonlinear mpc for quadrotors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 690–697, 2021.

[5] P. Robuffo Giordano, Q. Delamare, and A. Franchi, "Trajectory generation for minimum closed-loop state sensitivity," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 286–293.

[6] P. Brault, Q. Delamare, and P. Robuffo Giordano, "Robust trajectory planning with parametric uncertainties," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 095–11 101.

[7] S. Candido and S. Hutchinson, "Minimum Uncertainty Robot Path Planning using a POMDP Approach," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, December 2010, pp. 1408–1413.

[8] A. Ansari and T. Murphey, "Minimum Sensitivity Control for Planning with Parametric and Hybrid Uncertainty," *The International Journal of Robotics Research (IJRR)*, vol. 35, no. 7, pp. 823–839, October 2016.

[9] P. Salaris, M. Cognetti, R. Spica, and P. R. Giordano, "Online optimal perception-aware trajectory generation," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1307–1322, 2019.

[10] I. D. D. M. de Pierrepont, D. Carminati, M. Scanavino, E. Capello, *et al.*, "Model-in-the-loop testing of control systems and path planner algorithms for quadrotor uavs," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 1809–1818.

[11] C. Bohm, P. Brault, Q. Delamare, P. Robuffo Giordano, and S. Weiss, "Cop: Control & observability-aware planning," in *2022 IEEE Int. Conf. on Robotics and Automation*, 2022.

[12] P. Brault and P. Robuffo Giordano, "Tube-based trajectory optimization for robots with parametric uncertainty," submitted to IEEE Robotics and Automation Letters, http://rainbow-doc.irisa.fr/pdf/tube_based_traj_opt.pdf, 2023.

[13] S. Wasiela, P. Robuffo Giordano, J. Cortes, and T. Simeon, "A Sensitivity-Aware Motion Planner (SAMP) to Generate Intrinsically-Robust Trajectories," in *2023 IEEE Int. Conf. on Robotics and Automation*, 2023.

[14] G. Antonelli, E. Cataldi, F. Arrichiello, P. Robuffo Giordano, S. Chiaverini, and A. Franchi, "Adaptive trajectory tracking for quadrotor mavs in presence of parameter uncertainties and external disturbances," *IEEE Trans. on Control Systems Technology*, vol. 26, no. 1, pp. 248–254, 2018.

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

[16] T. Lee, M. Leok, and N. H. McClamroch, "Geometric Tracking Control of a Quadrotor UAV on $SE(3)$," in *49th IEEE Conference on Decision and Control (CDC)*, December 2010, pp. 5420–5425.

[17] C. Bensalah, N. M'sirdi, and A. Naamane, "Full modelling and sliding mode control for a quadrotor uav in visual servoing task," in *IMAACA2019*, 2019.

[18] M. Labbadi and M. Cherkaoui, "Robust integral terminal sliding mode control for quadrotor uav with external disturbances," *International Journal of Aerospace Engineering*, vol. 2019, 2019.

[19] A. Eltayeb, M. F. Rahmat, and M. B. Mohd Ariffanan, "Sliding mode control design for the attitude and altitude of the quadrotor uav," *International Journal on Smart Sensing and Intelligent Systems*, vol. 13, no. 1, pp. 1–13, 2020.

[20] F. Zhou, B. Song, and G. Tian, "Bézier curve based smooth path planning for mobile robot," *Journal of Information & Computational Science*, vol. 8, no. 12, pp. 2441–2450, December 2011.

[21] D. Mellinger and V. Kumar, "Minimum Snap Trajectory Generation and Control for Quadrotors," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 2520–2525.

[22] A. Loquercio, E. Kaufmann, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza, "Learning high-speed flight in the wild," *Science Robotics*, vol. 6, no. 59, p. eabg5810, 2021.

[23] A. R. Conn, K. Scheinberg, and P. L. Toint, "On the convergence of derivative-free methods for unconstrained optimization," *Approximation theory and optimization: tributes to MJD Powell*, pp. 83–108, 1997.

[24] G. Ansmann, "Efficiently and easily integrating differential equations with jitcode, jitcdde, and jitcsde," *Chaos: An interdisciplinary journal of nonlinear science*, vol. 28, no. 4, 2018.

[25] M. Hafezipour and S. Khodaygan, "An uncertainty analysis method for error reduction in end-effector of spatial robots with joint clearances and link dimension deviations," *International Journal of Computer Integrated Manufacturing*, vol. 30, no. 6, pp. 653–663, 2017.