Safe and Robust Planning for Uncertain Robots: A Closed-Loop State Sensitivity Approach

Amr Afifi, Tommaso Belvedere, Andrea Pupa, Paolo Robuffo Giordano, Antonio Franchi

Safe and Robust Planning for Uncertain Robots: A Closed-Loop State Sensitivity Approach

Amr Afifi, Tommaso Belvedere, Andrea Pupa, Paolo Robuffo Giordano, Antonio Franchi

***Abstract*—In this work, we detail a comprehensive framework for safe and robust planning for robots in presence of model uncertainties. Our framework is based on the recent notion of *closed-loop state sensitivity*, which is extended in this work to also include uncertainties in the initial state. The proposed framework, which considers the sensitivity of the nominal closed-loop system w.r.t. *both* model parameters and initial state mismatches, is exploited to compute *tubes* that accurately capture the worst-case effects of the considered uncertainties. In comparison to the current state-of-the-art for safe and robust planning, the proposed *closed-loop state sensitivity* framework has the important advantage of computational simplicity and minimal assumptions (and simplifications) regarding the underlying robot *closed-loop* dynamics. The approach is validated via both extensive simulations and real-world experiments. In the experiments we consider as case study a nonlinear trajectory optimization problem aimed at generating an *intrinsically robust and safe trajectory* for an aerial robot for safely performing an obstacle avoidance maneuver despite the uncertainties. Simulation and experimental results further confirm the viability and interest of the proposed approach.**

***Index Terms*—Planning under Uncertainty; Robot Safety; Constrained Motion Planning**

I. Introduction

ROBOTIC systems are increasingly becoming a part of our ecosystem. Robots are playing important roles in maintaining our infrastructure and even our homes. Therefore, designing safe robotic systems is a necessity for truly embracing this technology. A common approach for safety is to design the robot behavior taking into account safety constraints that, if violated, would cause failure of the task or even damage to the robot or the surrounding environment. The process of designing the robot behavior often relies on a model of the robot and its environment. However, due to the approximate nature of any model, the actual robot behavior might deviate from the planned one and even violate its safety constraints at runtime.

There exists a number of different approaches attempting to provide safety guarantees of a robot's behavior. Several approaches, for instance [1], propose formulations that allow obtaining formal guarantees on the robot safety, for example a constraint guaranteed to be always satisfied. Such guarantees are, however, obtained on a *specific model* of the robot which is ultimately only an approximation of the real system. Therefore, any formal safety guarantee based on the robot model can, in general, be only approximately met in any real-world scenario. One such approach is the Hamilton-Jacobi (HJ) reachability analysis [2], which requires the solution of a set of partial differential equations. In order to obtain solutions, HJ reachability approaches resort to discretizing the state-space using grids, a technique which suffers from an exponential increase in computational complexity as the dimension of the state space increases. In order to tackle this computational complexity, recent HJ approaches require a linear time varying (LTV) representation of the dynamics [3], possibly obtained using linearization, for which there is an analytic solution to the HJ PDE, [4]. Another approach presented in [5] and further applied in [6]. The approach entails computing funnels (or tubes) that capture the state's deviation from its nominal value. The approach requires solving an optimization problem with a sum of squares formulation. However, the computational complexity of the approach increases significantly with the dimension of the state-space. Another approach towards providing guarantees of robot behavior uses contraction theory [7], [8]. This approach requires a complex procedure to develop a specific controller, limiting the applicability of the approach.

An alternative class of approaches relies on the recent notion of *closed-loop state sensitivity* originally introduced in [9]. In particular, it has been shown in [9]–[12] that it is possible to exploit this notion for deriving uncertainty metrics and bounds to be optimized in trajectory generation problems, thereby allowing to produce motion plans that possess an *intrinsic* robustness against parametric uncertainty in the robot model. This approach is characterized by its computational simplicity and minimal assumptions on the robot/controller structure, indeed only requiring the differentiability of the robot dynamics and control action (which can be freely chosen). Such properties then make this approach an interesting alternative to current state-of-the-art methods in safe and robust trajectory

Manuscript received: April 15, 2024; Revised: July 26, 2024; Accepted: September 5, 2024.

This paper was recommended for publication by the Editor H. Kurniawati upon evaluation of Associate Editor and the Reviewers' comments. This work was partially funded by the ANR-20-CE33-0003 "CAMP" project and EU: AUTOASSESS project, EU Grant agreement ID: 101120732.)

Amr Afifi is with the Robotics and Mechatronics Lab, Faculty of Electrical Engineering, Mathematics & Computer Science, University of Twente, Enschede, The Netherlands. a.n.m.g.afifi@utwente.nl

Tommaso Belvedere and Paolo Robuffo Giordano are with CNRS, Université de Rennes, Inria, IRISA, Rennes, France. tommaso.belvedere@irisa.fr; prg@irisa.fr

Andrea Pupa is with the Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Italy andrea.pupa@unimore.it

Antonio Franchi is with the Robotics and Mechatronics Lab, Faculty of Electrical Engineering, Mathematics & Computer Science, University of Twente, Enschede, The Netherlands and is with Department of Computer, Control and Management Engineering, Sapienza University of Rome, Rome, Italy. a.franchi@utwente.nl , antonio.franchi@uniroma1.it

Digital Object Identifier (DOI): see top of this page.

generation for robotic systems.

Motivated by the previous considerations, the goal of this work is to present a comprehensive framework for safe and robust planning of robot motions based on the closed-loop state sensitivity. Our main contributions are:

- The extension of the *closed-loop state sensitivity* framework to also include uncertainties in the initial state besides model parameters, which (to the best of our knowledge) is a novel contribution of this work.
- The proposition of a novel and more accurate approach (compared to [11]) to compute uncertainty *tubes* on the states/inputs based on sensitivity framework.
- An extensive validation campaign involving both simulation and real-world experiments on an aerial robot. In particular, contrarily to previous works on this subject [9]–[12], we provide an *experimental validation* by considering a large spectrum of uncertainties spanning (i) poorly known physical parameters, (ii) unmodeled residual dynamics, and (iii) initial state mismatches. This, in our opinion, demonstrates the interest, viability and computational feasibility of the proposed approach in dealing with *non-trivial real-world cases*. A software package is released to the community as an example of the framework[1].
- The demonstration of the entire framework's utility to motion planing by embedding it into a trajectory optimization problem.

The paper is structured as follows. Sect. II presents the notion of the closed-loop state sensitivity. The novel tube computation is described in Sect. III. The aerial robot used as a case study is presented in Sect. IV, which is followed by the validation in Sect. V. The robust trajectory optimization is presented in Sect. VI. Lastly, the conclusions are outlined in Sect. VII.

II. Closed-loop State Sensitivity

We start by reviewing the notion of closed-loop state sensitivity originally introduced in [9]. Consider the following representation for the nonlinear closed-loop dynamics of a robotic system. The robot dynamics are described by a set of ordinary differential equations (ODEs) given by

$$\dot{\boldsymbol{q}} = \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{u}, \boldsymbol{p}), \qquad \boldsymbol{q}(0) = \boldsymbol{q}_0, \tag{1}$$

where $\boldsymbol{q} \in \mathbb{R}^{n_q}$ is the robot state, $\boldsymbol{u} \in \mathbb{R}^{n_u}$ the robot inputs, $\boldsymbol{p} \in \mathbb{R}^{n_p}$ is a vector of physical parameters such as the mass of robotic link or location of the center of mass, and $\boldsymbol{q}_0 \in \mathbb{R}^{n_q}$ represents the initial state. The robot output $\boldsymbol{y} \in \mathbb{R}^{n_y}$, e.g., the end-effector pose of a manipulator, is in general a function of the state and parameters

$$\boldsymbol{y} = \boldsymbol{f}_y(\boldsymbol{q}, \boldsymbol{p}). \tag{2}$$

The robot task is usually realized by properly designing a reference trajectory $\boldsymbol{y}^r(t)$ for the robot output and by synthesising a controller able to force the output $\boldsymbol{y}$ to follow the reference trajectory. Such a controller can be represented generically as

$$\begin{aligned} \dot{\boldsymbol{\xi}} &= \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{\xi}, \boldsymbol{p}_n) \\ \boldsymbol{u} &= \boldsymbol{h}(\boldsymbol{q}, \boldsymbol{y}^r, \boldsymbol{\xi}, \boldsymbol{p}_n) \end{aligned} \tag{3}$$

[1]https://gitlab.com/AIR-TEAM/closed-loop-state-sensitivity-clss

where $\boldsymbol{\xi} \in \mathbb{R}^{n_\xi}$ represents the controller internal states, e.g., an integral action. The controller may also require nominal values for the system parameters denoted as $\boldsymbol{p}_n$.

We define the nominal closed-loop state trajectory, $\boldsymbol{q}_n(t)$, as the solution of (1–3) using nominal values for the system physical parameters $\boldsymbol{p} = \boldsymbol{p}_n$ and initial state $\boldsymbol{q}(0) = \boldsymbol{q}_{0_n}$. In presence of uncertainties in either the parameters or initial state, $\boldsymbol{q}_n(t)$ will not match exactly the real closed-loop state trajectory $\boldsymbol{q}(t)$. The following quantities are then introduced for quantifying deviations around the system nominal behavior due to such uncertainties.

We consider the notion of closed-loop state sensitivities with respect to perturbations in the parameter $\boldsymbol{p}$ as the following matrix

$$\boldsymbol{\Pi}(t) = \left.\frac{\partial \boldsymbol{q}(t)}{\partial \boldsymbol{p}}\right|_{\boldsymbol{p}=\boldsymbol{p}_n} \qquad \boldsymbol{\Pi}(t) \in \mathbb{R}^{n_q \times n_p}. \tag{4}$$

Matrix $\boldsymbol{\Pi}$ is in general not available in closed-form, but a closed-form expression for its dynamics can be obtained as follows (see [9], [11] for all details)

$$\begin{aligned} \dot{\boldsymbol{\Pi}} &= \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}}\boldsymbol{\Pi} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\boldsymbol{\Theta} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}}, \quad \boldsymbol{\Pi}(0) = \boldsymbol{0} \in \mathbb{R}^{n_q \times n_p} \\ \boldsymbol{\Theta} &= \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}}\boldsymbol{\Pi} + \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\xi}}\boldsymbol{\Pi}_{\boldsymbol{\xi}} \\ \dot{\boldsymbol{\Pi}}_{\boldsymbol{\xi}} &= \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}}\boldsymbol{\Pi} + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}}\boldsymbol{\Pi}_{\boldsymbol{\xi}}, \quad \boldsymbol{\Pi}_{\boldsymbol{\xi}}(0) = \boldsymbol{0} \in \mathbb{R}^{n_\xi \times n_p} \end{aligned} \tag{5}$$

where $\boldsymbol{\Theta} \in \mathbb{R}^{n_u \times n_p}$ is the so-called *input sensitivity* and $\boldsymbol{\Pi}_{\boldsymbol{\xi}} \in \mathbb{R}^{n_\xi \times n_p}$ the sensitivity of the controller states. Furthermore, for any function of the state such as, e.g., the output (2), its sensitivity can be easily obtained as

$$\boldsymbol{\Pi}_{\boldsymbol{y}} = \frac{\partial \boldsymbol{f}_y}{\partial \boldsymbol{q}}\boldsymbol{\Pi} + \frac{\partial \boldsymbol{f}_y}{\partial \boldsymbol{p}}. \tag{6}$$

This is relevant when the quantity to be verified is a function of the state, e.g. a control point along the robot structure that should avoid an obstacle. An example in this sense will be provided in the evaluation of Sect. VI.

We now extend the previous formulation to include a novel quantity introduced (to the best of our knowledge) for the first time in this work: the closed-loop sensitivity w.r.t. perturbations of the initial state $\boldsymbol{q}_0$. This is defined as the following square matrix

$$\boldsymbol{\Psi}(t) = \left.\frac{\partial \boldsymbol{q}(t)}{\partial \boldsymbol{q}_0}\right|_{\boldsymbol{q}_0=\boldsymbol{q}_{0_n}}, \quad \boldsymbol{\Psi}(t) \in \mathbb{R}^{n_q \times n_q} \tag{7}$$

which, again, has no closed-form expression in general case. However, by following a procedure conceptually similar to the one yielding (5), it is possible to show that the dynamics of $\boldsymbol{\Psi}$ has expression

$$\dot{\boldsymbol{\Psi}} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}}\boldsymbol{\Psi} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\boldsymbol{\Phi}, \quad \boldsymbol{\Psi}(0) = \boldsymbol{I} \in \mathbb{R}^{n_q \times n_q} \tag{8}$$

where

$$\boldsymbol{\Phi} = \left.\frac{\partial \boldsymbol{u}(t)}{\partial \boldsymbol{q}_0}\right|_{\boldsymbol{q}_0=\boldsymbol{q}_{0_n}} \tag{9}$$

represents the controller input sensitivity w.r.t. perturbations in the initial state $\boldsymbol{q}_0$ with expression

$$\boldsymbol{\Phi} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}}\boldsymbol{\Psi} + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\xi}}\boldsymbol{\Psi}_{\boldsymbol{\xi}}. \tag{10}$$

Similarly to (5), for evaluating (10) one also needs to propagate the sensitivity of the controller internal states w.r.t. perturbations in $\boldsymbol{q}_0$

$$\boldsymbol{\Psi}_{\boldsymbol{\xi}} = \left.\frac{\partial \boldsymbol{\xi}(t)}{\partial \boldsymbol{q}_0}\right|_{\boldsymbol{q}_0=\boldsymbol{q}_{0_n}} \tag{11}$$

with the following update law

$$\dot{\boldsymbol{\Psi}}_{\boldsymbol{\xi}} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{q}}\boldsymbol{\Psi} + \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\xi}}\boldsymbol{\Psi}_{\boldsymbol{\xi}}, \quad \boldsymbol{\Psi}_{\boldsymbol{\xi}}(0) = \mathbf{0} \in \mathbb{R}^{n_\xi \times n_q}. \tag{12}$$

One can then obtain the behavior of $\boldsymbol{\Pi}(t)$ and $\boldsymbol{\Psi}(t)$ by forward integrating (5), (8) and (12) along a given future trajectory.

It is interesting to note that matrix $\boldsymbol{\Psi}$ in (7) is, essentially, a state transition matrix [13]. In the context of linear time-invariant systems, the state transition matrix is useful for computing the evolution of the state in closed-form. In the context of this work, we consider the state transition matrix for a generic nonlinear systems and in *closed-loop*, i.e., by explicitly considering the control action chosen for controlling the robot. This allows to compute the sensitivity of the closed-loop state w.r.t. mismatches in the robot initial conditions which, as it will be shown in Sect. V, proves to be an important information for producing robust motion plans. To the best of our knowledge, this is a novel application of the classical notion of state transition matrix to the robust planning problem for (nonlinear) robot/controller pairs.

III. Sensitivity-based Tube Computation

Assuming that deviations from the nominal trajectory are bounded and form a time-varying set at each instant, the evolution of this set can be represented as a *tube* centered around the nominal trajectory. This tube bounds the envelope of all perturbed trajectories. In the subsequent discussion, we illustrate how to compute this tube by leveraging the previous concepts of closed-loop state sensitivity. For clarity we will focus our illustration on computing the tubes due to sensitivity of the states w.r.t. physical parameters $\boldsymbol{p}$. However it is straightforward to extend the procedure by considering the sensitivities $\boldsymbol{\Theta}(t)$ and $\boldsymbol{\Psi}(t)$, as it is outlined at the end of the section.

Suppose that the uncertain parameters $\boldsymbol{p}$ have a maximum deviation $\Delta\boldsymbol{p}_{\max} = (\Delta p_{\max,1}, \dots, \Delta p_{\max,n_p})$ w.r.t. their nominal values $\boldsymbol{p}_n$, such that the parameters belong to the set $\mathcal{P} = \{\boldsymbol{p} \in \mathbb{R}^{n_p} : -\Delta\boldsymbol{p}_{\max} \leq \boldsymbol{p} - \boldsymbol{p}_n \leq \Delta\boldsymbol{p}_{\max}\}$. To estimate the effect that a parameter deviation has on the closed-loop trajectory, it is possible to exploit ellipsoids to define a mapping from the parameter space to the state space. Letting $\boldsymbol{\sigma}_{\boldsymbol{p}}$ be the scaled parameter deviation, such that $\Delta\boldsymbol{p} = \boldsymbol{p} - \boldsymbol{p}_n = \boldsymbol{W}^{\frac{1}{2}}\boldsymbol{\sigma}_{\boldsymbol{p}}$, with $\boldsymbol{W} = \text{diag}\{\Delta p^2_{\max,i}\}$, it is possible to map the unit sphere[2]

$$\mathcal{S}_{\boldsymbol{p}} = \left\{\boldsymbol{\sigma}_{\boldsymbol{p}} \in \mathbb{R}^{n_p} \ : \ \boldsymbol{\sigma}_{\boldsymbol{p}}^T\boldsymbol{\sigma}_{\boldsymbol{p}} \leq 1\right\} \tag{13}$$

[2]Note that this sphere represents an ellipsoid centered at $\boldsymbol{p}_n$ when mapped in the original parameter coordinates.

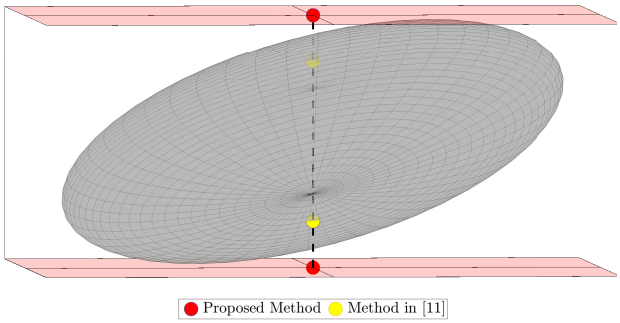


Fig. 1: Representation of the radius α along a generic direction $\boldsymbol{n}$ (dotted line) computed via the proposed formula (17) (red) and via the one in [11] (yellow). One can appreciate how the proposed (17) computes the distance to the ellipsoid tangent plane (light red) with normal $\boldsymbol{n}$, while the method proposed in [11] underestimates the actual maximum deviation. This difference will also be highlighted in the results of Sect. V.

into the corresponding ellipsoids of the space whose sensitivity matrix is available. As already mentioned, without loss of generality, we will focus on the state space, i.e. using $\boldsymbol{\Pi}(t)$. If the parameter deviation $\Delta\boldsymbol{p}$ is small enough, it is possible to approximate $\Delta\boldsymbol{q}(t) = \boldsymbol{q}(t) - \boldsymbol{q}_n(t)$ as

$$\Delta\boldsymbol{q}(t) \simeq \boldsymbol{\Pi}(t)\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{\sigma}_{\boldsymbol{p}}, \tag{14}$$

which, applied to the boundary equation of (13), yields

$$\Delta\boldsymbol{q}^T(\boldsymbol{\Pi}\boldsymbol{W}^{\frac{1}{2}})^{\dagger T}(\boldsymbol{\Pi}\boldsymbol{W}^{\frac{1}{2}})^{\dagger}\Delta\boldsymbol{q} = \Delta\boldsymbol{q}^T\left(\boldsymbol{\Pi}\boldsymbol{W}\boldsymbol{\Pi}^T\right)^{\dagger}\Delta\boldsymbol{q} \leq 1,$$

where the time dependency has been dropped for ease of notation. Letting $\boldsymbol{K}_{\boldsymbol{\Pi}} = \boldsymbol{\Pi}\boldsymbol{W}\boldsymbol{\Pi}^T$, it is possible to define the state space set:

$$\mathcal{E}_{\boldsymbol{q}} = \left\{\boldsymbol{q} \in \mathbb{R}^{n_q} \ : \ \Delta\boldsymbol{q}^T\boldsymbol{K}_{\boldsymbol{\Pi}}^{\dagger}\Delta\boldsymbol{q} \leq 1\right\}. \tag{15}$$

Equation (15) defines an ellipsoid in the state space centered at $\boldsymbol{q}_n$ with a symmetric and positive semi-definite kernel matrix $\boldsymbol{K}_{\boldsymbol{\Pi}}$. This ellipsoid allows to map the deviations from the parameter space to the state space. However, the ellipsoid axes are in general not aligned with the canonical basis. This implies that calculating the deviation of each state component q_i is not straightforward, as direct use of the ellipsoid semi-axes length is not possible. This may be even more complex if we are interested in analyzing the evolution of the system along directions other than those aligned with the canonical basis. In general, given a direction $\boldsymbol{n} \in \mathbb{R}^{n_q}$, we are interested in finding the maximum deviation that the system can exhibit along $\boldsymbol{n}$, i.e., in obtaining the radius of the tube along $\boldsymbol{n}$. More formally, we want to find the minimum scalar $\alpha > 0$ such that

$$\forall\Delta\boldsymbol{q} \in \mathcal{E}_{\boldsymbol{q}} \quad \boldsymbol{n}^T\Delta\boldsymbol{q} \leq \alpha.$$

A graphical representation of the ellipsoid and of the tube radius α along a generic direction $\boldsymbol{n}$ is depicted in Fig. 1. The radius can be found exploiting the linear approximation of the state sensitivity, i.e., equation (14). Indeed, setting $\alpha = \boldsymbol{n}^T\Delta\boldsymbol{q}$, one can obtain:

$$\alpha \simeq \boldsymbol{n}^T\boldsymbol{\Pi}\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{\sigma}_{\boldsymbol{p}}. \tag{16}$$

By pseudoinverting (16) and plugging $\Delta\boldsymbol{p}$ in (13) one obtains

$$\alpha^2(\boldsymbol{n}^T\boldsymbol{\Pi}\boldsymbol{W}^{\frac{1}{2}})^{\dagger T}(\boldsymbol{n}^T\boldsymbol{\Pi}\boldsymbol{W}^{\frac{1}{2}})^{\dagger} = 1,$$

from which it is possible to derive

$$\alpha = \sqrt{\boldsymbol{n}^T \boldsymbol{K}_{\boldsymbol{\Pi}} \boldsymbol{n}}. \tag{17}$$

We highlight that in a previous work [11] an alternative procedure (and formula) has been proposed for computing the radius α. However, w.r.t. [11], the proposed (improved) equation (17) has two main advantages: (i) (17) is a differentiable expression everywhere, thus making it more suited for gradient-based optimization and, even more importantly, (ii) (17) provides a *more accurate* evaluation of the actual tube radius. This is graphically illustrated in Fig. 1 and demonstrated in the results of Sect. V.

Given a nominal trajectory $\boldsymbol{q}_n(t)$ defined for $t \in [0,\ T]$, we can then use (17) to compute bounds on any function of the state and inputs through their closed-loop sensitivity. For instance, in the case of the state tubes, we can evaluate the sensitivity $\boldsymbol{\Pi}(t)$ exploiting (5) and applying (17) along the canonical basis of $\mathbb{R}^{n_q}$ $(\boldsymbol{e}_1, \dots, \boldsymbol{e}_{n_q})$ for obtaining the tube bounds $\boldsymbol{\alpha}_q(t) \in \mathbb{R}^{n_q}$ as

$$\alpha_{q,i}(t) = \sqrt{\boldsymbol{e}_i^T \boldsymbol{K}_{\boldsymbol{\Pi}} \boldsymbol{e}_i} \quad i = 1, \dots, n_q,\ t \in [0,\ T]. \tag{18}$$

These will represent bounds on the deviation of each component of the state vector around the nominal trajectory.

The previous computation can be straightforwardly extended to include the sensitivity w.r.t. initial conditions by simply defining a new sensitivity matrix $\boldsymbol{\Gamma}(t)$ as the concatenation of matrices $\boldsymbol{\Pi}(t)$ and $\boldsymbol{\Psi}(t)$, i.e., $\boldsymbol{\Gamma}(t) = \begin{bmatrix}\boldsymbol{\Pi}(t) & \boldsymbol{\Psi}(t)\end{bmatrix}$. With this in mind, in the rest of the paper we will use the vector

$$\boldsymbol{\gamma} = \begin{bmatrix}\boldsymbol{p}^T & \boldsymbol{q}_0^T\end{bmatrix}^T$$

to represent all the uncertain parameters (which will include both the physical parameters $\boldsymbol{p}$ and the initial conditions $\boldsymbol{q}_0$).

IV. Case Study: Fully Actuated Aerial Robot

The closed-loop state sensitivity approach can be applied to general robotic systems such as quadrotors or robotic manipulators [9]–[11]. In this work, we make use of a fully actuated hexarotor aerial robot as case study for validating the approach. In particular, we use the Tilthex design presented in [14]. This choice is motivated by the fact that parameter uncertainty plays a prominent role in the robot's actuation model. Unlike standard multi-rotors designs, the Tilthex's thrust model suffers from uncertainty along all thrust directions while for a standard aerial vehicle, like a quad-rotor, the thrust is uncertain only along the body frame's vertical axis.

In the following, we present the dynamics for the Tilthex aerial robot with the main objective of exposing the system's parameters and highlighting the generality of our approach in considering the parameters regardless of their non-linearity within the dynamics. Consider the following frames: an inertial frame, $\mathbb{F}_w$, with origin denoted by O_w. The body fixed frame, $\mathbb{F}_v$, rigidly attached to the geometric center of the aerial robot's rigid body, with origin denoted by O_v. The configuration of the robot is described by the position $\boldsymbol{p}_v \in \mathbb{R}^3$ expressed in an inertial frame $\mathbb{F}_w$ and the orientation $\boldsymbol{R}_v \in \mathbb{R}^{3\times3}$, of frame $\mathbb{F}_v$ with respect to the inertial frame $\mathbb{F}_w$. The robot velocities $\boldsymbol{\nu}_v \in \mathbb{R}^6$ are described by $\dot{\boldsymbol{p}}_v \in \mathbb{R}^3$ and $\boldsymbol{\omega}_v \in \mathbb{R}^3$ which represent, respectively, the linear velocity wrt frame $\mathbb{F}_w$ and the angular velocity of frame $\mathbb{F}_v$ wrt $\mathbb{F}_w$, expressed in frame $\mathbb{F}_v$. We define the state of the aerial robot as $\boldsymbol{q} = (\boldsymbol{p}_v, \boldsymbol{R}_v, \dot{\boldsymbol{p}}_v, \boldsymbol{\omega}_v) \in \mathcal{M} \subseteq \mathbb{R}^3 \times SO(3) \times \mathbb{R}^3 \times \mathbb{R}^3$.

The kinematics of the orientation are given by $\dot{\boldsymbol{R}}_v = \boldsymbol{R}_v[\boldsymbol{\omega}_v]_\times$. We derive the dynamic equations of motion (EoM) of the robot using the Newton-Euler formalism, (we refer the reader to [15] for full details). Consider the following inertia matrix $\boldsymbol{M} \in \mathbb{R}^{6\times6}$

$$\boldsymbol{M} = \begin{bmatrix} m_v \boldsymbol{I}_3 & -m_v[\boldsymbol{r}_c]_\times \\ m_v[\boldsymbol{r}_c]_\times & \boldsymbol{I}_v - m_v[\boldsymbol{r}_c]_\times[\boldsymbol{r}_c]_\times \end{bmatrix} \tag{19}$$

where $m_v \in \mathbb{R}^+$ represents the mass of the robot, $\boldsymbol{I}_v \in \mathbb{R}^{3\times3}$ represents the mass moment of inertia, and $\boldsymbol{r}_c \in \mathbb{R}^3$ is an offset of the center of mass with respect to the $\mathbb{F}_v$. We can compactly describe the system dynamics as

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}_v + \boldsymbol{h}(\boldsymbol{\nu}_v) = \boldsymbol{F}(\boldsymbol{q})\boldsymbol{\omega}_p^2 \tag{20}$$

with

$$\boldsymbol{h}(\boldsymbol{\nu}_v) = \begin{bmatrix} m_v g \boldsymbol{R}_v \boldsymbol{e}_3 + m_v \boldsymbol{R}_v[\boldsymbol{\omega}_v]_\times[\boldsymbol{\omega}_v]_\times \boldsymbol{r}_c \\ m_v g[\boldsymbol{r}_c]_\times \boldsymbol{R}_v^T \boldsymbol{e}_3 + [\boldsymbol{\omega}_v]_\times(\boldsymbol{I}_v - [\boldsymbol{r}_c]_\times[\boldsymbol{r}_c]_\times) \end{bmatrix}$$

and g being the gravity acceleration and $\boldsymbol{e}_3 = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$.

The aerial robot is actuated by commanding the square of the propeller spinning velocities $\boldsymbol{\omega}_p^2$. The square of each individual propeller spinning velocity $\omega_{p_i}^2$ is related to its thrust $\boldsymbol{f}_{p_i} \in \mathbb{R}^3$ and drag moment $\boldsymbol{\tau}_{p_i} \in \mathbb{R}^3$, in good approximation, quadratically as follows

$$\begin{aligned} \boldsymbol{f}_{p_i}^v &= (-1)^i k_f \operatorname{sign}(\omega_{p_i}) \omega_{p_i}^2 \boldsymbol{R}_{p_i}^v \boldsymbol{e}_3 \\ \boldsymbol{\tau}_{p_i}^v &= k_\tau \operatorname{sign}(\omega_{p_i}) \omega_{p_i}^2 \boldsymbol{R}_{p_i}^v \boldsymbol{e}_3 \end{aligned} \tag{21}$$

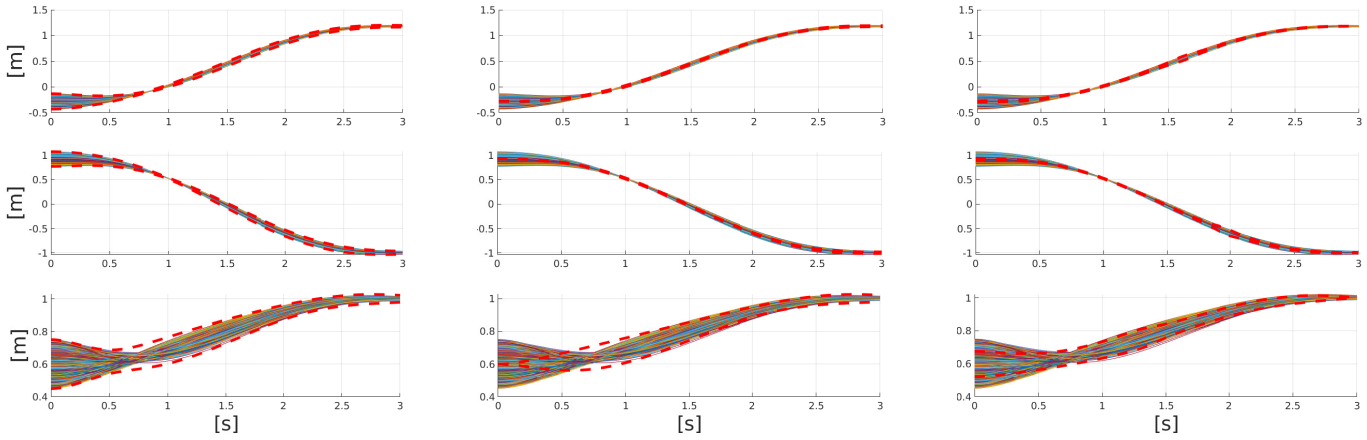
where, $i = 1, \dots, 6$ and $\boldsymbol{R}_{p_i}^v \in \mathbb{R}^{3\times3}$, denotes the orientation of the i-th propeller frame $\mathbb{F}_{p_i}$ wrt $\mathbb{F}_v$. The rotation matrix depends on the orientation of each propeller which, in the case of the Tilthex platform, is the tangential angle α_{tilt} for each of the 6 propellers. By commanding $\boldsymbol{\omega}_p^2$, the total force and moment applied at O_v expressed in $\mathbb{F}_w$ and $\mathbb{F}_v$ respectively is

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{\omega}_p^2) &= \boldsymbol{R}_v \sum_{i=1}^{6} \boldsymbol{f}_{p_i}^v = \boldsymbol{R}_v \boldsymbol{F}_1 \boldsymbol{\omega}_p^2 \\ \boldsymbol{\tau}(\boldsymbol{\omega}_p^2) &= \sum_{i=1}^{6} (\boldsymbol{p}_{p_i}^v \times \boldsymbol{f}_{p_i}^v + \boldsymbol{\tau}_{p_i}^v) = \boldsymbol{F}_2 \boldsymbol{\omega}_p^2 \end{aligned} \tag{22}$$

where, $\boldsymbol{p}_{p_i}^v \in \mathbb{R}^3$ is the position of i-th propeller expressed in $\mathbb{F}_v$. Matrices $\boldsymbol{F}_1 \in \mathbb{R}^{3\times6}$ and $\boldsymbol{F}_2 \in \mathbb{R}^{3\times6}$ include the physical and geometrical characteristics of the aerial robot and compose matrix $\boldsymbol{F}$ in eq. (20) as $\boldsymbol{F}(\boldsymbol{q}) = \begin{bmatrix}\boldsymbol{R}_v \boldsymbol{F}_1 \\ \boldsymbol{F}_2\end{bmatrix}$. Thanks to the Tilthex's full actuation, the robot can be controlled by a static feedback linearization scheme. Considering the robot dynamics in the compact form shown in eq. (20), we can compute the control action as

$$\boldsymbol{\omega}_p^2 = \boldsymbol{F}(\boldsymbol{q})^{-1}(\boldsymbol{M}\boldsymbol{\nu}_v^{\text{vi}} + \boldsymbol{h}(\nu_v)) \tag{23}$$

with $\boldsymbol{\nu}_v^{\text{vi}}$ representing a virtual input, encompassing a proportional, damping and integral action, in addition to a feed-forward acceleration. The various sensitivity matrices introduced in Sect. II will then be evaluated by considering (23) as controller for the Tilthex.



(a) (b) (c)

Fig. 2: The figure shows the results for 500 flight trajectories in simulation. Figure 2a demonstrates how the sensitivity-tubes (shown in dashed red) computed with the proposed approach precisely encapsulate the perturbed trajectories, i.e. computed considering both the state sensitivity $\mathbf{\Pi}$ and the initial state uncertainty $\mathbf{\Psi}$. Fig. 2b shows the tubes computed without considering $\mathbf{\Psi}$, which leads to lower performance. Figure 2c, shows the sensitivity-tubes computed using the approach presented in [11], which, as discussed in Sec. II and Fig. 1, underestimates the effects of the uncertainties.

V. Approach Validation

In the previous Sections we have outlined the theoretical framework behind the computation of sensitivity tubes. One main aspect of the approach is the approximation introduced in (14). In the following, we proceed to investigate how effectively the approach captures the trajectory envelope of the state. To this end, we use an empirical strategy, relying on extensive simulations and experiments in presence of non-trivial model uncertainty.

A. Numerical Validation via Extensive Simulations

Firstly, the effectiveness of our approach is tested using numerical simulations with the Tilthex. A subset of the robot physical parameters is chosen as uncertain. In particular, the chosen parameters are the robot mass m_v, the center of mass location $\boldsymbol{r}_c$, the aerodynamic parameters k_f, k_τ for each of the 6 propellers and the tilting angle for each propeller α_{tilt}, which amounts to 22 physical parameters. Moreover, we consider the initial robot position to be uncertain. This choice of the uncertain parameters (total of 25) is motivated by their real-world relevance and modelling difficulty. Non-trivial and arbitrary physical parameter and initial state perturbation bounds are chosen for the simulations, i.e., realistic bounds for the parametric uncertainty that can affect a real aerial robotic platform in practice. The chosen bounds for the perturbation to m_v, k_f, k_τ, α_{tilt} represent 10% of their nominal value, while for each component of $\boldsymbol{r}_c$ the perturbation is chosen as 0.01 m. The perturbations to the initial position are set to 0.15 m for each component. We then proceed to validate via simulations the effectiveness by which the computed sensitivity-tubes capture the state evolution in the presence of parametric and initial condition uncertainties. To this end, we performed $N_{\text{sim}} = 1500$ flight simulations, where the aerial robot is tasked to track an arbitrary full pose trajectory. Each 500 simulations constitute a desired trajectory with different boundary conditions (initial and final positions). The trajectory duration is $T = 3$ s. In each simulation run, a different perturbation $\Delta\boldsymbol{p}$ and $\Delta\boldsymbol{q}_0$ is applied, i.e., a mismatch is induced by changing the true values of the robot parameters in the simulator. The perturbations are sampled randomly from a uniform distribution within an ellipsoid with semi-axis defined by the chosen bounds on the uncertain parameters. In Fig. 2a, as an example, we show an extract of 500 simulations for a particular set of boundary conditions. It is then possible to see the ability of the sensitivity-tubes to precisely bound the envelope of the 500 flight trajectories, thereby confirming the soundness of the whole approach and, in particular, of the tube computation. We report that for all 1500 simulations, the position trajectories are completely confined within the tubes. Due to space limitations, the plots for the other boundary conditions are shown in the supplementary video. Figure 2b highlights another important aspect of the proposed framework, that is, the importance of also considering the sensitivity w.r.t. mismatches in the initial robot state. Indeed one can appreciate how, when *not considering* $\mathbf{\Psi}$, the sensitivity-tubes are no longer able to correctly bound the envelope of the perturbed trajectories, as it was instead the case in the results of Fig. 2a. In addition we also implemented the sensitivity-based tube computation presented in [11] to compare it with the novel approach as described in Sect. II. The results are presented in Fig. 2c which, compared to Fig. 2a, highlights how the tube computation in [11] could only provide a (non-negligible) *underestimation* of the deviation of the state. Finally, by using the software provided with this work, the interested reader can test the approach with arbitrary boundary conditions and parameter deviations.

1) Computational considerations: The sensitivity propagation and the computation of the tube radii are performed on a standard laptop with an Intel i7 processor and Nvidia Quadro T1000 graphics card. Matlab 2021b is used together with the autodiff library Casadi v3.5 [16], to perform the necessary computation. It takes Matlab approx. 80 ms to perform the

propagation and tube computation along a trajectory of interest. An integration timestep of 0.01 s is used. This shows the small computational burden of computing the sensitivity-based tubes. Consequently, the analysis and constraint checking for robot trajectories can be performed in fast manner and with minimal computational burden.

B. Validation via Experiments

In the following, we extend our validation to the real Tilthex robotic system. In this more practical and experimental case, knowledge of some physical parameters such as the aerodynamic coefficients or rotor dynamics can be uncertain and the bounds on the uncertainty can be difficult to define. In order to overcome this challenge, we augment the model of the Tilthex robot with an additional additive term $\boldsymbol{W}_{\text{residual}} \in \mathbb{R}^6$ as shown in (24). This additive term represents the unknown (and unmodeled) residual dynamics

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}_v + \boldsymbol{h}(\boldsymbol{\nu}_v) = \boldsymbol{F}(\boldsymbol{q})\boldsymbol{\omega}_p^2 + \boldsymbol{W}_{\text{residual}}. \tag{24}$$

In the nominal model used by the controller $\boldsymbol{W}_{\text{residual}}$ is considered to be equal to zero, thus the controller does not compensate for it. In order to compute bounds on $\boldsymbol{W}_{\text{residual}}$ we use the residual generator method in [17]. In particular, the robot is tasked to fly different trajectories multiple times and the residual wrench $\boldsymbol{W}_{\text{residual}}(t)$ is recorded. The maximum value along each of its 6 components is then extracted and used as bound on the residual dynamics. Additionally to the unknown residual dynamics, we consider a scenario where the real aerial robot's mass is not known precisely but lies within a prescribed range around the nominal value. In particular, we consider the perturbation to the mass of the vehicle in the range of $+150$ to -150 grams. This proposed perturbation is approximately 7% of the vehicle mass. The γ-ellipsoid semi-axes are then composed of the maximum bounds on the $\boldsymbol{W}_{\text{residual}}$, the bounds on the robot's mass, and bounds on the initial state mismatch.

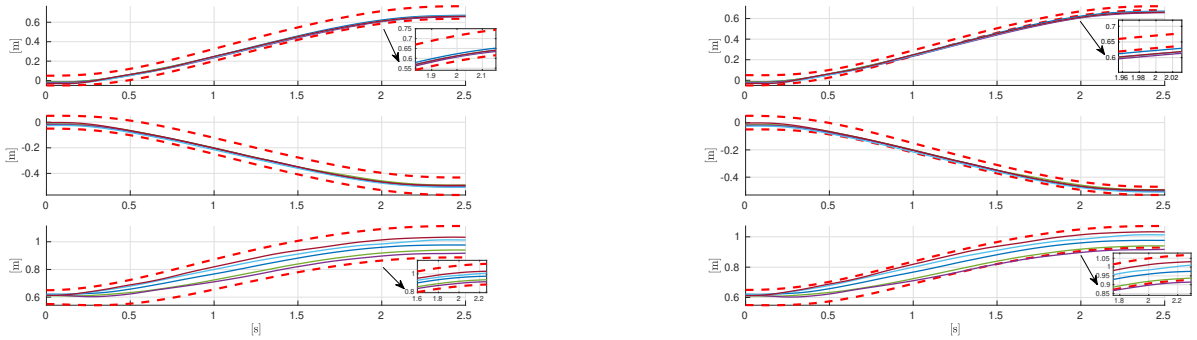
In the experiments, the aerial robot is tasked to track an arbitrary full-pose trajectory 5 times. For each flight, a different perturbation of the mass is applied within the prescribed range. In order to perform the tests in a systematic manner and avoid unintentionally introducing other sources of uncertainties, the perturbation is applied in software to the nominal value of the mass used within the controller. The desired full-pose trajectory for the aerial robot constitutes a rest-to-rest motion. The applied perturbations are -80, 80, -150 and 150 grams respectively. The results of the experiments are shown in Fig. 3a. The plots show, again, the ability of the proposed framework to compute tubes accurately capturing the uncertainty in the robots trajectories also in the real conditions. Moreover, we highlight the importance of having also considered the unknown residual dynamics $\boldsymbol{W}_{\text{residual}}$ and its identified bounds: to this end, we present in Fig. 3b the same experimental results but with the sensitivity-tubes obtained *without* considering $\boldsymbol{W}_{\text{residual}}$ as additional 'uncertain parameter'. By comparing Fig. 3b with Fig. 3a, one can then note how in this latter case the sensitivity-tubes fail in accurately capturing the deviation in some of the experimental runs, as expected.

As final remark, we wish to highlight that the possibility of decomposing and accounting for different sources of uncertainties is, in our opinion, an important asset of our framework. In a realistic scenario, our method provides the freedom to consider different sources of uncertainties: poor knowledge of some physical parameters in the robot model, or residual dynamics that is not explicitly accounted for at the modeling stage. Moreover, the ability to *also* consider uncertainties in the initial state allows the user to embed information on, for example, the position accuracy of their robot at arriving at a certain position, from which a safety critical maneuver is to be performed. This flexibility makes our proposed sensitivity-based approach an effective and viable framework for robust and safe planning in real-world cases.

VI. Robust Trajectory Optimization

We conclude with a case study example of the use of the sensitivity tubes in a trajectory optimization (TO) problem. Given the Tilthex start position and a goal end position, we seek a dynamically feasible and obstacle free robot trajectory. We are considering an experimental scenario where the robot would be required to fly close to an object. We consider one obstacle with known position $\boldsymbol{p}_{\text{obs}}$. Without loss of generality, we chose as a cost function the robot power consumption along the trajectory, computed as $P_w(t) = \sum_i^6 k_\tau \omega_{p_i}^3(t)$. Moreover, we exploit the controller structure and parameterize the reference trajectory denoted by $\boldsymbol{y}^r$ in eq. (3) by means of polynomial functions $\boldsymbol{y}^r = \boldsymbol{r}(\boldsymbol{a}, t)$, where $\boldsymbol{a}$ represents the coefficients of the polynomial of a particular order. The Tilthex robot is fully actuated, therefore one can plan a full pose reference trajectory. We denote the reference position by $\boldsymbol{p}_v^r \in \mathbb{R}^3$ and the reference orientation as $\boldsymbol{R}_v^r \in \mathbb{R}^{3\times 3}$. We parameterize $\boldsymbol{R}_v^r$ with a roll-pitch-yaw minimal representation of orientation which we denote by $\boldsymbol{\phi}_v^r$. Our choice of using such parameterization is motivated by the fact that the Tilthex actuator limitations constrain the robot admissible orientations away from the singularities of a roll-pitch-yaw representation. For our motion planning problem, we consider rest-to-rest motions from the given start pose $\boldsymbol{r}(0) = \boldsymbol{r}_0$ to the desired goal pose $\boldsymbol{r}(T) = \boldsymbol{r}_f$. Since the required rest-to-rest motion imposes 6 boundary conditions on the reference trajectory, a quintic polynomial would be sufficient to achieve the required motion. However we choose a 7th order polynomials to parameterize $\boldsymbol{r}(\boldsymbol{a}, t)$ in order provide the solver with some redundancy for optimizing the trajectory. We then formulate the following trajectory optimization (TO) problem

$$\begin{aligned}
\min_{\boldsymbol{a}} \quad & \sum_{t_0}^{t_f} P_w(t) \\
\text{s.t.} \quad & y_{p_i}(\boldsymbol{q}, \boldsymbol{\Gamma}, t) \geq 0 \quad i = 1, \dots, 6 \quad t \in [0,\, T] \\
& \overline{\boldsymbol{u}} - \boldsymbol{u}_i(t) \geq 0 \quad i = 1, \dots, 6 \quad t \in [0,\, T] \\
& \boldsymbol{u}_i(t) - \underline{\boldsymbol{u}} \geq 0 \quad i = 1, \dots, 6 \quad t \in [0,\, T] \\
& \boldsymbol{r}(0) = \boldsymbol{r}_0 \quad \boldsymbol{r}(T) = \boldsymbol{r}_f \\
& \dot{\boldsymbol{r}}(0) = \boldsymbol{0} \quad \dot{\boldsymbol{r}}(T) = \boldsymbol{0} \\
& \ddot{\boldsymbol{r}}(0) = \boldsymbol{0} \quad \ddot{\boldsymbol{r}}(T) = \boldsymbol{0}
\end{aligned} \tag{25}$$



(a) (b)

Fig. 3: This figure demonstrates the generality and usefulness of our approach in also considering uncertain residual dynamics. This is demonstrated using experimental flights and comparing the tube's performance (shown in dashed red) in estimating the effects of uncertainty *with and without* including residual dynamics uncertainty in Fig. 3a and Fig. 3b respectively.

The problem includes inequality constraints that represent the hardware limits on the control inputs $\boldsymbol{u}$ and boundary constraints on the reference trajectory. The obstacle avoidance constraint is represented as $y_{p_i}(\boldsymbol{q}, \boldsymbol{\Gamma}, t) \geq 0$. The obstacle geometry is approximated with a sphere of radius r_{obs}. We approximate the geometry of the robot by considering the position of each of the 6 different propellers that surround the aerial robot structure, denoted by $\boldsymbol{p}_{p_i}(t), i = 1, \ldots, 6$. The position of each propeller is encapsulated within a sphere of radius r_{props}, to take into account the propeller radius. The choice of approximating the robot's geometry, considering some points along the robot's structure, as opposed to, for example, encapsulating it with a sphere, is to avoid being overly conservative and allow the robot to get closer to the object. We consider a trajectory to be 'collision free' if the euclidean distance between each propeller and the obstacle, $d_{p_i} = \|\boldsymbol{p}_{p_i}(t) - \boldsymbol{p}_{\text{obs}}\|$, satisfies the following condition

$$y_{p_i}(\boldsymbol{q}, \boldsymbol{\Gamma}, t) = d_{p_i} - d_{\text{safe}} \geq 0 \quad i = 1, \ldots, 6 \quad t \in [0,\, T] \tag{26}$$

where

$$d_{\text{safe}} = r_{\text{obs}} + r_{\text{props}} + \alpha_{y_{p_i}}(\boldsymbol{q}, \boldsymbol{\Gamma}_{d_{p_i}}, t)$$

and $\alpha_{y_{p_i}}(\boldsymbol{q}, \boldsymbol{\Gamma}_{d_{p_i}}, t)$ is the radius of the tubes around the evolution given by $d_{p_i}(t) \quad i = 1, \ldots, 6$. Indeed, since $\alpha_{y_{p_i}}(\boldsymbol{q}, \boldsymbol{\Gamma}_{d_{p_i}}, t)$ is a function of the state, one can compute its sensitivity to uncertainty in parameters simply as

$$\boldsymbol{\Gamma}_{d_{p_i}}(t) = \frac{\partial d_{p_i}}{\partial \boldsymbol{q}} \boldsymbol{\Gamma}(t), \tag{27}$$

see also (6). This then allows obtaining the corresponding uncertainty tubes.

1) Experimental Results: As in Sect. V, we consider the following uncertainties: an artificially induced uncertainty in the mass of the vehicle, the unmodeled residual dynamics $\boldsymbol{W}_{\text{residual}}$ and an initial state mismatch. The aerial robot is tasked to track the optimal full-pose trajectory 5 times. For each flight, a different perturbation of the mass is applied within the prescribed range (in software as in Sect. V). For each flight we record the robot state and compute the evolution of the function $d_{p_i}(t)$. Figure 4, shows the Tilthex platform and a snapshot of the robot executing the optimized full pose trajectory. The results are presented in Fig 5a where the reader can appreciate how the sensitivity tubes for function $d_{p_i}(t)$ are well inside their bounds (thanks to the optimization) and, at the same time, the various perturbed runs are well confined inside the tubes and, therefore, do not violate any safety constraint. For the sake of comparison, we also consider a variant of the TO problem (25) which does not take into account the sensitivity-tubes. We report in Fig. 5b the results when the aerial robot tracks the reference trajectory from the *non-robust* TO. One can see that in this case the tubes do violate the constraints which results in the constraint violation for some of the actual trajectories.

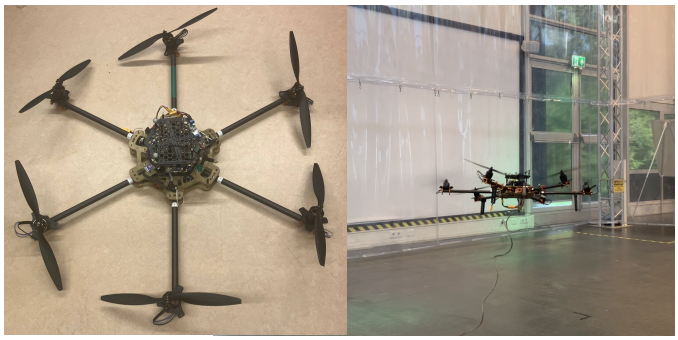
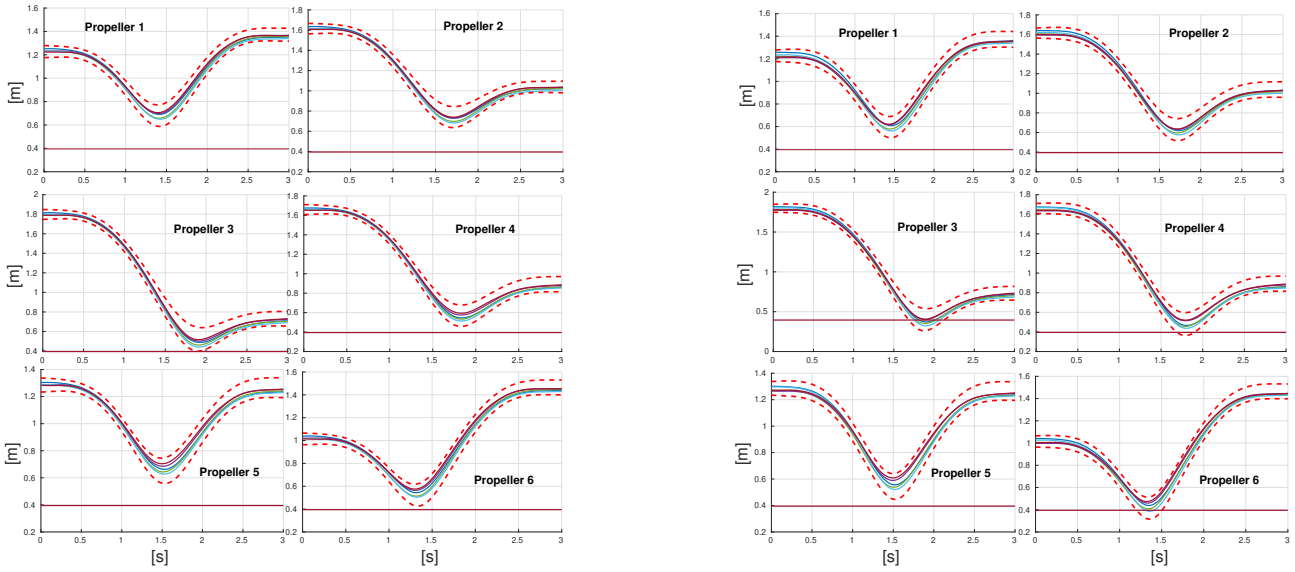


Fig. 4: The figure shows, on the left, the Tilthex platform which was used in the experiments. Note the propeller tilt angles which allow full actuation of the platform. While the image on the right shows the Tilthex executing the optimized full pose trajectory

VII. Conclusions

In this work we presented a comprehensive framework for the safe and robust planning for robots in the presence of uncertainty. Our framework relies on the recent notion of *closed-loop state sensitivity*, which we extended to include uncertainties in the initial state. We also proposed a novel (and more accurate) procedure to compute *tubes* based on the closed-loop state sensitivity. An extensive simulation and real-world testing campaign demonstrated the efficacy and computational simplicity of the approach, and its possible



(a) (b)

Fig. 5: The figure shows two cases for the evolution of $d_{p_i}(t)$ together with the obstacle avoidance constraint shown as the horizontal line. A constraint violation corresponds to $d_{p_i}(t)$ crossing that line. Figure 5a, shows the results of the robot tracking the *robust* optimized trajectory (five times with different induced parameter perturbation). It can be seen that the sensitivity-tubes, in dashed red, do not violate the constraint and that the real system trajectories are completely confined within the tube. In contrast, Fig. 5b show the robot executing the non-robust optimized trajectory. Note how some of the real system trajectories, while being completely confined within the tube, *violate* the constraint.

use in the context of nonlinear trajectory generation problems possessing an *intrinsic* robustness against uncertainties. Our approach is based on a first-order analysis and, therefore, cannot provide strong formal guarantees. Nevertheless, it is able to produce tubes that tightly capture the envelope of the perturbed trajectories while requiring minimal assumptions on the closed-loop dynamics and at a very low computational cost. Because of these features, we believe that our approach provides a concrete step forward towards the development of general and computationally simple planning algorithms for attaining safety and robustness for robotic systems in realistic conditions. Future work will focus on devising online control and planning schemes exploiting the closed-loop sensitivity framework. One interesting avenue is using our approach within model predictive control, which is notoriously sensitive to model-uncertainty.

REFERENCES

[1] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, 2014.

[2] I. Mitchell, A. Bayen, and C. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.

[3] H. Seo, D. Lee, C. Y. Son, I. Jang, C. J. Tomlin, and H. J. Kim, "Real-time robust receding horizon planning using hamilton–jacobi reachability analysis," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 90–109, 2023.

[4] J. Darbon and S. Osher, "Algorithms for overcoming the curse of dimensionality for certain hamilton–jacobi equations arising in control theory and elsewhere," *Research in the Mathematical Sciences*, vol. 3, no. 1, p. 19, 2016.

[5] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," *The International Journal of Robotics Research*, vol. 36, no. 8, pp. 947–982, 2017.

[6] Z. Manchester and S. Kuindersma, "Robust direct trajectory optimization using approximate invariant funnels," *Autonomous Robots*, vol. 43, pp. 375–387, 2019.

[7] W. Lohmiller and J.-J. E. Slotine, "On contraction analysis for non-linear systems," *Automatica*, vol. 34, no. 6, pp. 683–696, 1998.

[8] S. Singh, A. Majumdar, J.-J. Slotine, and M. Pavone, "Robust online motion planning via contraction theory and convex optimization," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5883–5890.

[9] P. Robuffo Giordano, Q. Delamare, and A. Franchi, "Trajectory generation for minimum closed-loop state sensitivity," in *2018 IEEE Int. Conf. on Robotics and Automation*, Brisbane, Australia, May 2018, pp. 286–293.

[10] A. Pupa, P. Robuffo Giordano, and C. Secchi, "Optimal Energy Tank Initialization for Minimum Sensitivity to Model Uncertainties," in *2023IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2023.

[11] P. Brault, "Robust trajectory planning algorithms for robots with parametric uncertainties," Ph.D. dissertation, Université de Rennes, 2023, available at http://rainbow-doc.irisa.fr/pdf/2023_phd_brault.pdf.

[12] A. Srour, S. Marcellini, T. Belvedere, M. Cognetti, A. Franchi, and P. Robuffo Giordano, "Experimental validation of sensitivity-aware trajectory planning for a quadrotor uav under parametric uncertainty," in *2024 Int. Conf. on Unmanned Aircraft Systems*, 2024, pp. 572–578.

[13] P. J. Antsaklis and A. N. Michel, *Linear systems*. Springer, 1997, vol. 8.

[14] M. Ryll, G. Muscio, F. Pierri, E. Cataldi, G. Antonelli, F. Caccavale, D. Bicego, and A. Franchi, "6D interaction control with aerial robots: The flying end-effector paradigm," vol. 38, no. 9, pp. 1045–1062, 2019.

[15] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

[16] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[17] S. Haddadin, A. Albu-Schaffer, A. De Luca, and G. Hirzinger, "Collision detection and reaction: A contribution to safe physical human-robot interaction," in *2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 3356–3363.