This article has been accepted for publication in IEEE Transactions on Industrial Electronics. Citation information: DOI 10.1109/TIE.2023.3245180, IEEE Transactions on Industrial Electronics.

Safe Planning and Control under Uncertainty: A Model-Free Design with One-Step Backward Data

Cong Li†, Qingchen Liu†, Jiahu Qin*, Martin Buss and Sandra Hirche

***Abstract*—Autonomous systems are desired to safely accomplish predetermined tasks with guaranteed performance despite uncertainties. This paper proposes a safe planning and performance-guaranteed control (SP-PGC) scheme to accomplish safe execution of autonomous systems suffering from uncertainties and disturbances. This is realized by investigating mutual influences between planning and control levels, either explicitly considering control-level attainable performance bounds into safe planning algorithms, or directly relating planned safe boundaries to control-level performance bounds. In particular, we first utilize one-step backward data to construct incremental systems, which are equivalent representations of the investigated autonomous systems but without using explicit model information (kinematics and/or dynamics). The formulated incremental systems transform the influence of uncertainties and disturbances into the effect of provably bounded estimation errors, caused by the difference between current and one-step backward states. Then, we introduce the concept of input-to-state stable with provable safety barrier Lyapunov function (ISS-PS-BLF) to facilitate the performance-guaranteed tracking controller design based on the incremental systems, wherein the estimation errors are rigorously analyzed through an input-to-state stable approach. Finally, either the guaranteed tracking performance bound of the ISS-PS-BLF based controller is considered in the safe planning algorithm to guide the reference trajectory generation, or the safe planned boundary is used to determine the explicit value of the control-level performance bound for safe execution under uncertainty. The efficiency of our developed SP-PGC scheme is validated through both numerical and experimental validations.**

***Index Terms*—Performance guaranteed control, input-to-state stability, one-step backward data**

I. Introduction

The safe execution of autonomous systems under uncertainty is required for safety-critical scenarios such as robot manipulators for public services and quadrotors for cave rescue [1], [2]. The common solution to the problem is to firstly build an accurate map wherein a safe smooth desired trajectory is then planned, and finally the resulting reference trajectory is supposed to be precisely followed by capable tracking controllers [3]. However, the planned collision-free trajectory does not imply the guaranteed safe execution given potential tracking controllers' inefficiency caused by model uncertainties and environmental disturbances. The deviation between the real execution trajectory and the planned trajectory might result in unsafe behaviours. In practice, the gap between planning and control levels mentioned above could be solved via replanning [4]. However, the additionally introduced computational load and complexity cause the associated solution fragile, and no theoretical guarantee of robust safety is provided. Alternatively, this work solves the problem of safe execution under uncertainty via analyzing the mutual influence between planning and control levels from a systematic perspective, rather than independently investigating in either planning or control levels [3], [4]. In particular, the planning level generates one collision-free reference trajectory lying in a safe set, and the control level robustly guarantees tracking performance even under model uncertainties and environmental disturbances. More importantly, the quantified actual control-level tracking performance bounds and the boundary of the planned safety set are interconnected with each other. Our adopted compatible safe planning algorithm and performance-guaranteed control strategy collaborate to realize the guaranteed safe execution of autonomous systems under uncertainty.

A. Related Work

The safety (collision avoidance in particular) and robustness properties have been separately investigated from planning and control perspectives. The planning level mainly solves safety concerns via numerical optimization algorithms [4] or sampling methods [5]. The corresponding numerical or experimental validations are often conducted assuming available perfect controllers that precisely track planned trajectories or paths [4], [5]. However, imperfect controllers inevitably result in discrepancies from the collision-free desired trajectory. Thereby, the practical and safe execution cannot be guaranteed. Departing

† Equivalent contribution as the common first authors.

This work was supported in part by the National Natural Science Foundation of China under Grants (61922076), and in part by Science and Technology Major Project of Anhui Province (202203a06020011), and in part by the European Research Council (ERC) Consolidator Grant "Safe data-driven control for human-centric systems (COMAN)" under grant agreement number 864686, and in part by ONE MUNICH Strategy Forum Project "Next generation Human-Centered Robotics". (Corresponding Author: Jiahu Qin)

C. Li and M. Buss are with the Chair of Automatic Control Engineering, Technical University of Munich, Theresienstr. 90, 80333, Munich, Germany, e-mail: {cong.li, mb}@tum.de.

Q. Liu is with the Department of Automation, University of Science and Technology of China, 230027, Hefei, China, e-mail: qingchen_liu@ustc.edu.cn.

J. Qin is with the Department of Automation, University of Science and Technology of China, Hefei 230027, China, and also with the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230088, China, e-mail: jhqin@ustc.edu.cn.

S. Hirche is with the Chair of Information-oriented Control, Technical University of Munich, Barer Strasse 21, 80333, Munich, Germany, e-mail: hirche@tum.de.

from the planning level, the control level attempts to use methods such as model predictive control [6] and control barrier function [7] to realize safety with preference to theoretical guarantees. However, these theory-oriented methods [6], [7] struggle with adaptability to complex scenarios such as obstacle-filled environments. The above analysis presents the weakness of the related works [4]–[7] in isolatedly addressing safety problems from one single level (either planning or control level in particular). Thus, this work adopts one compatible planning and control solution to realize the practical and safe execution. Robustness is mainly considered from the control level using various methods including H-∞ control [8], sliding mode control [9], and (partial) model-free control via parametric [10]–[14] or non-parametric methods [15]–[17]. Departing from the methods used in the works [8]–[17] mentioned above, the so-called one-step backward data is exploited in this work to embed our approach with robustness against model uncertainties and environmental disturbances in a computationally efficient way.

Except for the safety or robustness properties disjointly discussed above, the robust safety property (i.e., provable safety under uncertainty) is desired to ensure the safety solution is applicable even with respect to uncertainties. The robust safety is nontrivial and few related works focus on this topic. Existing works either assume available uncertainty bounds [18] or estimate uncertainty bounds [19] to rigorously analyse the influence of uncertainties on safety. However, autonomous systems often behave conservatively given that the utilized uncertainty bounds [18], [19] might treat collision-free areas as unsafe regions. Compared to the works mentioned above [18], [19], no prior-given or estimated disturbance bounds are required in our work to enforce the practical and safe execution under uncertainty. Thus, conservative behaviours are avoided.

Recently, barrier Lyapunov functions (BLFs) emerge as efficient tools to design stable controllers with guaranteed tracking performance [20]. This is because BLFs enjoy the properties of both Lyapunov functions and barrier functions, i.e., stabilizing autonomous systems while confining system states into specific regions. The usage of BLFs for safe control highly depends on accurate model knowledge (kinematics and/or dynamics). However, practical applications inevitably involve uncertainties. The safety check based on inaccurate model knowledge is untrustworthy. This problem is especially obvious for controlled plants such as robot manipulators that might involve both uncertain dynamics (e.g., lifting different workloads) and unknown kinematics (tools installed in end-effectors with arbitrary angles and effective lengths). There exist works utilizing parametric or non-parametric methods to estimate unknown model knowledge [10]- [17]. However, the problem concerning safety checks based on potentially inaccurate model knowledge (i.e., kinematics and/or dynamics are still in learning processes before the accurate ones are learned) has not been thoroughly analyzed. Compared to the above works, our work utilizes one-step backward data to formulate the evolution model of the uncertain controlled plant into an incremental system. This formulated incremental system offers an equivalent model-free (kinematics-free and/or dynamics-free) representation of the original controlled plant for safety check, wherein the effect of the estimation error on safety is rigorously analyzed through an input-to-state stable approach.

B. Contribution

The contributions of our work are summarized as follows.

- The compatible safe planning and performance-guaranteed control (SP-PGC) scheme is developed from a systematic perspective to accomplish the nontrivial provable safe execution under uncertainty.
- Departing from existing model-free control methods, one-step backward data is reused to realize model-free control that embeds robots with robustness against model uncertainties and environmental disturbances in a computationally efficient way.
- The common BLF is extended to input-to-state stability with provable safety BLF (ISS-PS-BLF) that contributes to design controllers capable of realizing stabilization and safety robustly.

C. Organization

This paper is organized as follows. Section II firstly presents the preliminaries and the problem formulation. Then, the incremental system developed in Section III serves as the basis for the controller design process illustrated in Section IV. The developed approach is numerically and experimentally validated in Section V and Section VI, respectively. Finally, Section VII concludes this paper.

Notations: Throughout this paper, $\mathbb{R}$, $\mathbb{R}^+$, and $\mathbb{R}_0^+$ denote the set of real, positive, and non-negative real numbers, respectively; $\mathbb{R}^n$ is the Euclidean space of n-dimensional real vector; $\mathbb{R}^{n\times m}$ is the Euclidean space of $n \times m$ real matrices; The i-th entry of a vector $x := [x_1, ..., x_n]^\top \in \mathbb{R}^n$ is denoted by x_i, and $|x| := \sqrt{\sum_{i=1}^{N} |x|^2}$ is the Euclidean norm of the vector x; The ij-th entry of a matrix $D \in \mathbb{R}^{n\times m}$ is denoted by d_{ij}, and $|D| := \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} |d_{ij}|^2}$ is the Frobenius norm of the matrix D; The pseudo inverse of the full column rank D is denoted as $D^\dagger := (D^\top D)^{-1} D^\top \in \mathbb{R}^{m\times n}$; $\mathrm{diag}(x)$ is the $n\times n$ diagonal matrix with the i-th diagonal entry equals x_i. $\mathrm{Int}(\mathbb{S})$ and $\partial\mathbb{S}$ denote the interior and boundary of the set $\mathbb{S}$; For any two real vectors $a, b \in \mathbb{R}^n$, $a \preceq (\prec)\ b$ is the component-wise comparison, i.e., $a_i \leq (<)\ b_i$, $\forall i \in \{1, \cdots, n\}$.

II. Preliminaries and Problem Formulation

A. Input-to-State Stable with Provable Safety BLF

This subsection presents the preliminaries to develop our approach by focusing on the state evolution model

$$\dot{x} = f(x) + g(x)u(x) + g(x)d, \tag{1}$$

where $x \in \mathbb{R}^n$, $u(x) : \mathbb{R}^n \to \mathbb{R}^m$ are the system state and control input, respectively. Both $f(x) : \mathbb{R}^n \to \mathbb{R}^n$ and $g(x) : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are bounded and locally Lipschitz. $d \in \mathbb{L}_\infty^m$ is a bounded disturbance with the (essential) supremum norm $|d|_\infty := \sup |d(t)|, t \geq 0$.

As stated in [21], iff the system (1) admits an input-to-state stable (ISS) Lyapunov function as Definition 1, the system (1)

is ISS as Definition 2. Therefore, ISS control can be realized by using a ISS-Lyapunov function to perform the controller design.

Definition 1 (ISS-Lyapunov Function [21]). *A smooth function $V(x): \mathbb{R}^n \to \mathbb{R}_0^+$ is an ISS-Lyapunov function for the system (1) if there exists $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathcal{K}_\infty$ such that $\forall\ x,\ d$*

$$\alpha_1(|x|) \le V(x) \le \alpha_2(|x|) \tag{2a}$$
$$\dot{V}(x,d) \le -\alpha_3(|x|) + \alpha_4(|d|). \tag{2b}$$

Definition 2 (ISS [22]). *The system (1) is ISS if there exists $\lambda \in \mathcal{KL}$ and $\gamma \in \mathcal{K}_\infty$ such that*

$$|x(t, x_0, d)| \le \lambda(|x_0|, t) + \gamma(|d|_\infty), \forall\, x_0,\ d, \forall\, t \ge 0.$$

The Definitions 1-2 inspire us to extend the original BLF [20], which is defined on an ideal accurate dynamics $\dot{x} = f(x) + g(x)u(x)$, to the uncertainty scenario in Definition 3.

Definition 3 (ISS-PS-BLF). *A smooth function $V(x) := V_1(x_1) + V_2(x_2) \in \mathbb{R}_0^+$, where $x := [x_1^\top, x_2^\top]^\top \in \mathbb{R}^{n_1+n_2}$, $x_1 \in \mathbb{R}^{n_1}$, $x_2 \in \mathbb{R}^{n_2}$, is an ISS-PS-BLF for the system (1) on the open region $\mathbb{S} := \{x_1 \in \mathbb{R}^{n_1} : -\underline{\epsilon} \prec x_1 \prec \overline{\epsilon}\}$, where $\underline{\epsilon}_i$, $\overline{\epsilon}_i \in \mathbb{R}^+$, $\forall i \in \{1, \cdots, n_1\}$, if there exist functions $\beta_i \in \mathcal{K}_\infty$, $i = 1, \cdots, 6$, such that $\forall\ x,\ d$*

$$\beta_1(|x_1|) \le V_1(x_1) \le \beta_2(|x_1|) \tag{3a}$$
$$\beta_3(|x_2|) \le V_2(x_2) \le \beta_4(|x_2|) \tag{3b}$$
$$\dot{V}(x,d) \le -\beta_5(|x|) + \beta_6(|d|) \tag{3c}$$
$$V_1(x_1) \to \infty, \quad x_1 \to \partial\mathbb{S}. \tag{3d}$$

The resulting ISS-PS-BLF formulated in Definition 3 is a valid ISS-Lyapunov function according to Definition 1 given the establishment of the inequalities (3a), (3b), (3c). Furthermore, the presented (3d) implies that a bounded ISS-PS-BLF would confine the state x_1 into the predetermined safe region $\mathbb{S}$. Thereby, our defined ISS-PS-BLF (3) provides designers with an efficient tool to realize the desired input-to-sate stabilization with provable safety.

To satisfy the requirements presented in Definition 3, combining with the result in [20], this work chooses

$$V(x) := \underbrace{\frac{1}{2}\sum_{i=1}^{n_1}\left[\frac{\overline{\epsilon}_i\underline{\epsilon}_i x_{1_i}}{(\overline{\epsilon}_i - x_{1_i})(\underline{\epsilon}_i + x_{1_i})}\right]^2}_{V_1(x_1)} + \underbrace{\frac{1}{2}x_2^\top x_2}_{V_2(x_2)}, \tag{4}$$

as the candidate ISS-PS-BLF to conduct the controller design.

B. Problem Formulation

This work attempts to realize the provable safe execution of uncertain autonomous systems in obstacle-filled environments. Our solution to this nontrivial problem is our developed SP-PGC scheme: the combination of the *performance-guaranteed control* that explicitly quantifies the control-level performance under uncertainty, and the *safe planning* where the collision-free desired trajectory is planned within the consideration of the attainable performance of the utilized controllers, see Fig. 1. This work adopts the safe planning algorithms that satisfy the requirement presented in the following assumption.

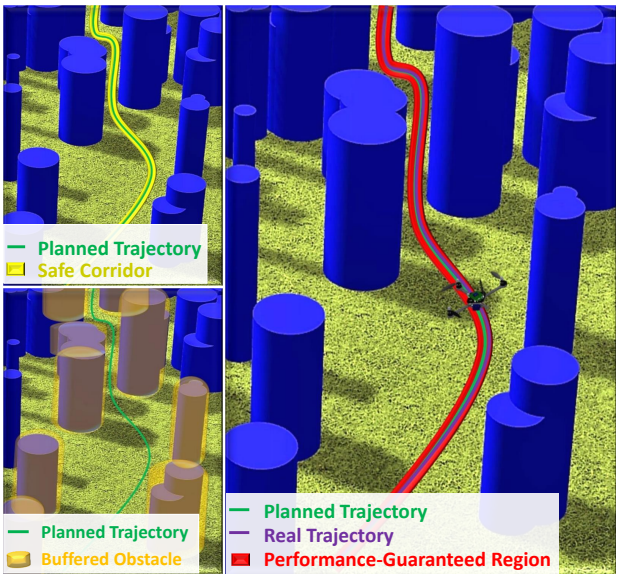


Fig. 1: Schematic of the SP-PGC scheme. The safe reference trajectory is planned within the consideration of the control-level performance bound ϵ, either establishing safe corridors with radius ϵ (the left above figure), or inflating obstacles (i.e., buffered obstacle) via size ϵ (the left below figure). The control level guarantees tracking performance even under uncertainties and disturbances ignored in the planning level (the right figure).

Assumption 1. *The planning level outputs a collision-free desired trajectory $p_d \in \mathbb{R}^m$ lying in a safe set $\mathbb{D} := \{p(t) \in \mathbb{R}^m : \underline{p}(t) \prec p(t) \prec \overline{p}(t)\}$, where $\underline{p}(t)$, $\overline{p}(t) \in \mathbb{R}^m$.*

Assumption 1 easily holds using off-the-self planning algorithms [23] conducted based on buffered obstacles [1], whose buffer size is $\epsilon \in \mathbb{R}^m$, $\epsilon_i \in \mathbb{R}^+$, $\forall i \in \{1, \cdots, m\}$ (see the left below figure in Fig. 1). The associated safe execution region is $\mathbb{D} := \{p \in \mathbb{R}^m : \underline{p} := p_d - \epsilon \prec p \prec \overline{p} := p_d + \epsilon\}$. Regarding this case, the tracking error $e_1 := p - p_d \in \mathbb{R}^m$ should satisfy $e_1 \in \mathbb{E} := \{e_1 \in \mathbb{R}^m : -\underline{\epsilon} := -\epsilon \prec e_1 \prec \epsilon := \overline{\epsilon}\}$ to achieve safety, where $\underline{\epsilon}$, $\overline{\epsilon} \in \mathbb{R}^m$ are lower and upper performance bounds of e_1. Alternatively, Assumption 1 is easily satisfied by the reachable set based algorithms [24], or corridor (funnel) based algorithms [25], [26] (see the left above figure in Fig. 1). In this case, $e_1 \in \mathbb{E} := \{e_1 \in \mathbb{R}^m : -\underline{\epsilon} := \underline{p} - p_d \prec e_1 \prec \overline{p} - p_d := \overline{\epsilon}\}$ should be guaranteed to avoid collision during practical executions [2].

Through the aforementioned analysis, we interpret the provable safe execution under uncertainty problem as a robust performance-guaranteed tracking control problem. This problem is nontrivial given that state constraints are considered under model uncertainties and environmental disturbances. We solve this nontrivial problem via our formulated incremental system in Section III and the ISS-PS-BLF facilitated controller in Section IV.

[1]This case matches the robot manipulator numerical and experimental validations displayed in Section V and Section VI separately.

[2]This case matches the quadrotor numerical simulation in Section V-B.

III. Data Informed Incremental System

This section utilizes one-step backward data to formulate the incremental system that equivalently describes the movement of the original autonomous system (1). By doing so, no explicit model knowledge (kinematics and/or dynamics) is required. The formulated incremental system serves as the basis for the controller design process presented in Section IV.

A. Development of Incremental System

In the following, we focus on the control-affine nonlinear system (1) satisfying Assumption 2 to clarify the formulation of the one-step backward data informed incremental system.

Assumption 2. *The columns $g_1, g_2, \cdots, g_m \in \mathbb{R}^n$ of the system matrix $g := [g_1, g_2, \cdots, g_m]$ are linearly independent.*

Remark 1. *Here $g(x)$ is assumed to be full column rank such that its pseudo inverse $g^\dagger$ could be expressed as a simple algebraic formula (the inverse of $g^\top(x)g(x)$ exists). This property is widely observed in many physical systems, such as the quadrotor presented in Example 1, and the robot manipulator shown in Example 2.*

Firstly, introducing a prior-chosen constant matrix $\bar{g} \in \mathbb{R}^{n\times m}$ and multiplying its pseudo inverse $\bar{g}^\dagger$ on (1), we obtain

$$\bar{g}^\dagger \dot{x} = h + u, \tag{5}$$

where $h := (\bar{g}^\dagger - g^\dagger)\dot{x} + g^\dagger f + d \in \mathbb{R}^n$ embodies the unknown knowledge of the system (1).

Accordingly, the following equation holds

$$\bar{g}^\dagger \dot{x}_0 = h_0 + u_0, \tag{6}$$

where $(\bullet)_0 := (\bullet)(t - t_s)$ denotes one-step backward data, and $t_s \in \mathbb{R}^+$ is the sampling time.

Then, we use one-step backward data in (6) to estimate h as

$$\hat{h} = h_0 = \bar{g}^\dagger \dot{x}_0 - u_0, \tag{7}$$

Finally, substituting (7) into (5), we get the incremental system:

$$\dot{x} = \dot{x}_0 + \bar{g}\Delta u + \bar{g}\xi, \tag{8}$$

where $\Delta u := u - u_0 \in \mathbb{R}^n$, and $\xi := h - \hat{h} \in \mathbb{R}^n$ is the estimation error proved to be bounded and vanishing in Lemma 1 under the properly chosen $\bar{g}$.

Remark 2. *The theoretical derivation* (5)–(8) *exploits one-step backward data to transform model uncertainties and external disturbances of* (1) *into a provably bounded estimation error ξ of* (8)*. This is beneficial to achieve provable safety under uncertainty given that the influence of the estimation error ξ on safety could be rigorously analyzed via an ISS approach as presented in Section IV. To achieve the same goal with our work, however, related works either estimate disturbance bounds explicitly using computation-intensive methods such as GP [15] or directly assume a known bound of uncertainty [18]. The utilized uncertainty bound often results in conservative behaviours.*

Remark 3. *The state observer [27] or the numerical differentiation technique [28] could be applied to get the indirectly measurable state or state derivatives that are required to construct the incremental system* (8)*. Furthermore, the inevitable measurement noise could be addressed by the robust exact differentiator technique [29].*

Through the processes (5)-(8), we get an equivalent form of (1) under Assumption 2 without using explicit model information. In the subsequent Section IV, we use the above formulated incremental system (8) and our proposed ISS-PS-BLF (4) together to design the robust tracking controller with guaranteed performance.

Before proceeding to the controller design process, we provide two explicit examples to clarify how to derive the associated incremental systems from the quadrotor dynamics and the robot manipulator kinematics and dynamics.

Example 1 (Quadrotor)**.** *The Euler-Lagrange (E-L) equation of a quadrotor follows [30]*

$$m\ddot{\zeta} + mg_c I_z = RT_B + T_d \tag{9a}$$

$$J(\eta)\ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta} = \tau_B + \tau_{Bd}, \tag{9b}$$

where $\zeta := [x, y, z]^\top \in \mathbb{R}^3$, and $\eta := [\phi, \theta, \psi]^\top \in \mathbb{R}^3$ represent the absolute linear position and Euler angles defined in the inertial frame, respectively; $m \in \mathbb{R}^+$ denotes the mass of the quadrotor; $g_c \in \mathbb{R}^+$ is the gravity constant; $I_z := [0, 0, 1]^\top$ represents a column vector; $T_B = [0, 0, T]^\top \in \mathbb{R}^3$, where $T \in \mathbb{R}$ is the thrust in the direction of the body z-axis; $\tau_B := [\tau_\phi, \tau_\theta, \tau_\psi]^\top \in \mathbb{R}^3$ denotes the torques in the direction of the corresponding body frame angles; $T_d =\in \mathbb{R}^3$ and $\tau_d \in \mathbb{R}^3$ denote the external disturbance; R, $J(\eta)$, $C(\eta, \dot{\eta}) \in \mathbb{R}^{3\times3}$ represent the rotation matrix, Jacobian matrix, and Coriolis term, respectively. We could rewrite the above translation dynamics (9a) *or the attitude dynamics* (9b) *as*

$$\dot{x}_1 = x_2 \tag{10a}$$

$$\dot{x}_2 = f + gu + gd, \tag{10b}$$

via letting $x_1 := \zeta$ or $\eta \in \mathbb{R}^3$, $x_2 := \dot{\zeta}$ or $\dot{\eta} \in \mathbb{R}^3$, $f := -g_c I_z$ or $-J^{-1}C(\eta, \dot{\eta})\dot{\eta} \in \mathbb{R}^3$, $g := R/m$ or $J^{-1} \in \mathbb{R}^{3\times3}$, $u := T_B$ or $\tau_B \in \mathbb{R}^3$, $d := R^{-1}T_d$ or $\tau_{Bd} \in \mathbb{R}^3$, respectively. Applying the theoretical derivation processes (5)–(8) *mentioned above on* (10b)*, we would get*

$$\dot{x}_1 = x_2 \tag{11a}$$

$$\dot{x}_2 = \dot{x}_{2,0} + \bar{g}\Delta u + \bar{g}\xi, \tag{11b}$$

which is an equivalent representation of (9) *but without using explicit knowledge of quadrotor dynamics.*

Example 2 (Robot Manipulator)**.** *The Cartesian-space position $p \in \mathbb{R}^m$ of the robot manipulator end-effector is expressed as*

$$p = h(q), \tag{12}$$

where $q \in \mathbb{R}^n$ is the joint-space angle vector, and $h(q) : \mathbb{R}^n \to \mathbb{R}^m$ is the differential forward kinematics. Note that $m \leq n$ holds. The end-effector velocity and acceleration $\dot{p}$, $\ddot{p} \in \mathbb{R}^m$ are related to the joint velocity and acceleration $\dot{q}$, $\ddot{q} \in \mathbb{R}^n$ as

$$\dot{p} = J(q)\dot{q} \tag{13a}$$

$$\ddot{p} = \dot{J}(q)\dot{q} + J(q)\ddot{q}, \tag{13b}$$

where $J(q) := \partial h(q)/\partial q \in \mathbb{R}^{m\times n}$ is the Jacobian matrix. Besides, the robot manipulator dynamics follows [14]

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F_v(\dot{q}) = \tau + \tau_d, \tag{14}$$

where $M(q) : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the symmetric positive definite inertia matrix; $C(q,\dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the matrix of centrifugal and Coriolis terms; $G(q) : \mathbb{R}^n \to \mathbb{R}^n$ represents the gravitational term; $F_v(\dot{q}) : \mathbb{R}^n \to \mathbb{R}^n$ denotes the viscous friction; $\tau_d \in \mathbb{R}^n$ represents the external disturbance.

Substituting (13) *into* (14) *yields*

$$M_p(q)\ddot{p} + C_p(q,\dot{q})\dot{p} + G(q) + F_v(\dot{q}) = \tau + \tau_d, \tag{15}$$

where $M_p(q) := M(q)J^\dagger(q) : \mathbb{R}^n \to \mathbb{R}^{n\times m}$, $C_p(q,\dot{q}) := C(q,\dot{q})J^\dagger(q) - M(q)J^\dagger(q)\dot{J}(q)J^\dagger(q) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times m}$. The pseudo inverse follows $J^\dagger(q) := (J^\top(q)J(q))^{-1}J^\top(q) : \mathbb{R}^n \to \mathbb{R}^{n\times m}$. Then, the integrated kinematics and dynamics form (15) *could be rewritten as the form* (11) *by denoting $x_1 := p \in \mathbb{R}^m$, $x_2 := \dot{p} \in \mathbb{R}^m$, $f := -M_p^\dagger(q)(C_p(q,\dot{q})\dot{p} + G(q) + F_v(\dot{q})) \in \mathbb{R}^m$, $g := M_p^\dagger(q) \in \mathbb{R}^{m\times n}$, $u := \tau \in \mathbb{R}^n$, and $d := \tau_d \in \mathbb{R}^n$. Through the theoretical derivation processes* (5)–(8)*, we would get one associated incremental system of the robot manipulator (in the same form as* (11)*) without using explicit information of kinematics and dynamics.*

Remark 4. *Examples 1-2 build on the assumption that singularities are always avoided during the whole execution process for the quadrotor and the robot manipulator. The systematic method to avoid singularity is beyond the scope of this article. Besides, we use the pseudo-inverse of the manipulator Jacobian in* (15) *to deal with the redundancy problem of the robot manipulator case.*

Remark 5. *Note that* (15) *in Example 2 departs from the common method [12] that attempts to write* (12)*,* (13)*, and* (14) *together to formulate an integrated kinematics and dynamics form as $\bar{M}_p(q)\ddot{p} + \bar{C}_p(q,\dot{q})\dot{p} + J(q)G(q) + J(q)F_v(\dot{q}) = J(q)\tau + J(q)\tau_d$, where $\bar{M}_p(q) := J(q)M(q)J^\dagger(q) : \mathbb{R}^n \to \mathbb{R}^{m\times m}$, $\bar{C}_p(q,\dot{q}) := J(q)C(q,\dot{q})J^\dagger(q) - J(q)M(q)J^\dagger(q)\dot{J}(q)J^\dagger(q) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{m\times m}$. Based on this form, the kinematics free control is impossible following the controller design process illustrated in Section IV. In particular, a controller in the form $\tau_p := J(q)\tau$ will be firstly designed. Then, an inversion calculation using the explicit kinematic knowledge, $\tau = J^\dagger(q)\tau_p$ in particular, is required to recover the torque applied at each joint. Our formulated* (15) *directly links the joint-space control input τ with the task-space position p without introducing the computationally intensive inverse kinematics calculation. This allows us to use one-step backward data to realize kinematics free control later and utilize ISS-PS-BLFs to encode task-space safety constraints while designing joint-space torques applied to robot manipulators.*

IV. Model-Free Performance-Guaranteed Control

This section utilizes our proposed ISS-PS-BLF (4) to develop a model-free performance-guaranteed tracking controller through a recursive controller design process. The ISS-PS-BLF provides explicit quantification of realizable tracking errors. This control-level performance quantification could feedback to the planning level to refine planned trajectories accounting for actual implementation tracking errors. The recursive controller design process based on the incremental system formulated in the previous section is illustrated as follows.

Step 1: Focusing on (11) [3], the position tracking error follows $e_1 := x_1 - p_d \in \mathbb{R}^m$. To ensure that the tracking error e_1 always lies in a predetermined performance bound, $e_1 \in \mathbb{E} := \{e_1(t) \in \mathbb{R}^m : -\underline{\epsilon} \prec e_1(t) \prec \bar{\epsilon}\}$ in particular, we firstly use the corresponding Lyapunov function $V_1(e_1)$ given in (4) to facilitate the controller design. The derivative of $V_1(e_1)$ follows

$$\dot{V}_1(e_1) = \sum_{i=1}^{m} e_{1_i} \underbrace{\frac{\bar{\epsilon}_i^3\underline{\epsilon}_i^3 + \bar{\epsilon}_i^2\underline{\epsilon}_i^2 e_{1_i}^2}{(\bar{\epsilon}_i - e_{1_i})^3(\underline{\epsilon}_i + e_{1_i})^3}}_{p_i} \dot{e}_{1_i} = e_1^\top P\dot{e}_1, \tag{16}$$

where $P := \mathrm{diag}(P_1, P_2, \cdots, P_m) \in \mathbb{R}^{m\times m}$.

Let $e_2 := x_2 - z \in \mathbb{R}^m$, where $z \in \mathbb{R}^m$ is a stabilizing term designed for stability analysis. Combining with (11a), the explicit form of $\dot{e}_1$ used in (16) follows

$$\dot{e}_1 = \dot{x}_1 - \dot{p}_d = x_2 - \dot{p}_d = e_2 + z - \dot{p}_d. \tag{17}$$

Then, we design $z := \dot{p}_d - P^{-1}L_1e_1$ to facilitate a stabilizing control policy, wherein $L_1 := \mathrm{diag}(L_{11}, L_{12}, \cdots, L_{1m}) \in \mathbb{R}^{m\times m}$, $L_{1j} \in \mathbb{R}^+$, $j = 1, \cdots, m$.

Substituting (17) into (16) yields

$$\dot{V}_1 = -e_1^\top L_1 e_1 + e_1^\top P e_2. \tag{18}$$

Step 2: According to (4), we choose the ISS-PS-BLF as $V := V_1 + V_2$, wherein the explicit form of V_2 follows

$$V_2(e_2) := \frac{1}{2}e_2^\top e_2. \tag{19}$$

Then, combining with (11b) and (16), we get

$$\begin{aligned}\dot{V} &= \dot{V}_1 + \dot{V}_2 \\ &= -e_1^\top L_1 e_1 + e_1^\top P e_2 + e_2^\top(\dot{x}_{2,0} + \bar{g}\Delta u + \bar{g}\xi - \dot{z}).\end{aligned} \tag{20}$$

Finally, we develop the incremental control input as

$$\Delta u = \bar{g}^\dagger(\dot{z} - \dot{x}_{2,0} - L_2e_2 - Pe_1), \tag{21}$$

to input-to-state stabilize the tracking errors e_1 and e_2 to a small neighbourhood around zero as proved in Theorem 1, wherein $L_2 := \mathrm{diag}(L_{21}, L_{22}, \cdots, L_{2m}) \in \mathbb{R}^{m\times m}$ is a positive definite matrix, $L_{2j} \in \mathbb{R}^+$, $j = 1, \cdots, m$. Accordingly, the control input applied at the controlled plant is recovered as

$$u = u_0 + \Delta u. \tag{22}$$

In the following, we theoretically analyze the properties of our designed performance-guaranteed control strategy (22). We firstly present the rigorous proof of the bounded estimation error in Lemma 1. Then, the proved bounded estimation error allows us to analyse the desirable provable safety under uncertainty in Theorem 1.

Lemma 1. *Given a sufficiently high sampling rate [4], there*

[3] We purposely focus on (11) rather than (8) given that the controller design process based on (11) is more difficult than the one related to (8).

[4] This is a prerequisite for estimating the unknown h in (5) by reusing one-step backward data, which could be chosen as the value that is larger than 30 times the system bandwidth [31]. Under this setting, one digital control system could be treated as one continuous system so that the gap of h in (5) between adjacent sampling periods is tiny. Thus, the estimation error ξ in (8) is sufficiently small.

exists a positive constant $\bar{\xi} \in \mathbb{R}^+$ *such that* $|\xi| \leq \bar{\xi}$.

Proof. Combining with (5), (7) and (11), we get

$$\begin{aligned}\xi = h - h_0 &= (\bar{g}^\dagger - g^\dagger)(\dot{x}_2 - \dot{x}_{2,0}) + (g_0^\dagger - g^\dagger)\dot{x}_{2,0} \\ &+ g^\dagger(f - f_0) + (g^\dagger - g_0^\dagger)f_0 + d - d_0.\end{aligned} \tag{23}$$

Besides, focusing on (11b), the following equation holds

$$\begin{aligned}\dot{x}_2 - \dot{x}_{2,0} &= f + gu + gd - f_0 - g_0u_0 - g_0d_0 \\ &= g\Delta u + (g - g_0)u_0 + f - f_0 + g(d - d_0) + (g - g_0)d_0.\end{aligned} \tag{24}$$

Then, substituting (24) into (23) reads

$$\xi = (\bar{g}^\dagger g - I_{n\times n})\Delta u + \delta_1, \tag{25}$$

where $\delta_1 := \bar{g}^\dagger(g - g_0)u_0 + \bar{g}^\dagger(f - f_0) + \bar{g}^\dagger g(d - d_0) + \bar{g}^\dagger(g - g_0)d_0 \in \mathbb{R}^n$. For representation simplicity, let $v := \dot{z} - L_2e_2 - Pe_1$. Accordingly, $v_0 := \dot{z}_0 - L_2e_{2,0} - P_0e_{1,0}$.

Then, invoking (5), (7) and (21), we get

$$\begin{aligned}\Delta u &= \bar{g}^\dagger(v - \dot{x}_{2,0}) = \bar{g}^\dagger v - h_0 - u_0 \\ &= \bar{g}^\dagger v - (\bar{g}^\dagger - g_0^\dagger)\dot{x}_{2,0} + g_0^\dagger f_0 - u_0 \\ &= \bar{g}^\dagger v - (\bar{g}^\dagger - g_0^\dagger)(f_0 + g_0u_0) + g_0^\dagger f_0 - u_0 \\ &= \bar{g}^\dagger v - \bar{g}^\dagger(f_0 + g_0u_0) \\ &= \bar{g}^\dagger(v - v_0) - \bar{g}^\dagger(\dot{x}_{2,0} - v_0).\end{aligned} \tag{26}$$

Combining (11b) with (21) yields

$$\dot{x}_2 = v + \bar{g}\xi. \tag{27}$$

Besides, according to (27), we get

$$\xi = \bar{g}^\dagger(\dot{x}_2 - v), \quad \xi_0 = \bar{g}^\dagger(\dot{x}_{2,0} - v_0). \tag{28}$$

Substituting (28) into (26) implies

$$\Delta u = \bar{g}^\dagger(v - v_0) - \xi_0. \tag{29}$$

Finally, substituting (29) into (25), we get

$$\xi = (I_{n\times n} - \bar{g}^\dagger g)\xi_0 + \delta_1 + \delta_2, \tag{30}$$

where $\delta_2 := (\bar{g}^\dagger g - I_{n\times n})\bar{g}^\dagger(v - v_0) \in \mathbb{R}^n$.

To analyse the estimation error bound better, we rewrite (30) into a discrete-time domain as

$$\xi(k) = (I_{n\times n} - \bar{g}^\dagger g(k))\xi(k-1) + \delta_1(k) + \delta_2(k). \tag{31}$$

Given a sufficiently high sampling rate, it is reasonable to assume that there exist positive constants $\bar{\delta}_1$, $\bar{\delta}_2 \in \mathbb{R}^+$ such that $|\delta_1| \leq \bar{\delta}_1$, and $|\delta_2| \leq \bar{\delta}_2$ hold. We choose the value of $\bar{g}$ to satisfy $\left|I_{n\times n} - \bar{g}^\dagger g(k)\right| \leq l < 1$, $l \in \mathbb{R}^+$. Then, the following equation holds

$$\begin{aligned}|\xi(k)| &\leq l\,|\xi(k-1)| + \bar{\delta}_1 + l\bar{\delta}_2 \\ &\leq l^2\,|\xi(k-2)| + (l+1)(\bar{\delta}_1 + l\bar{\delta}_2) \\ &\leq \cdots \leq l^k\,|\xi(0)| + \frac{\bar{\delta}_1 + l\bar{\delta}_2}{1-l} := \bar{\xi}\end{aligned} \tag{32}$$

As $k \to \infty$, $\bar{\xi} \to \frac{\bar{\delta}_1 + l\bar{\delta}_2}{1-l}$. □

Theorem 1. *Consider the system* (11) *with the controller* (22). *Given Assumption 1 for initial conditions lying in the safe set* $\mathbb{D}$*, the following properties hold:*

1) The tracking errors e_1 *and* e_2 *are input-to-state stabilizing to a small neighbourhood around zero.*

2) The Cartesian position tracking error e_1 *satisfies* $e_1 \in \mathbb{E}$.

3) The controlled plant realizes provable safe execution $p \in \mathbb{D}$ *under model uncertainties and environmental disturbances.*

Proof. Proof of 1) Substituting (21) into (20) yields

$$\begin{aligned}\dot{V} &= -e_1^\top L_1e_1 - e_2^\top L_2e_2 + e_2^\top \bar{g}\xi \\ &= -e_1^\top L_1e_1 - e_2^\top(L_2 - I_{m\times m})e_2 - (e_2^\top e_2 - e_2^\top \bar{g}\xi) \\ &= -e_1^\top L_1e_1 - e_2^\top(L_2 - I_{m\times m})e_2 \\ &\quad - \left|e_2 - \frac{1}{2}\bar{g}\xi\right|^2 + \frac{1}{4}\left|\bar{g}\xi\right|^2 \\ &\leq -e_1^\top L_1e_1 - e_2^\top(L_2 - I_{m\times m})e_2 + \frac{1}{4}\left|\bar{g}\right|^2\left|\xi\right|^2 \\ &= -e^\top Le + \frac{\left|\bar{g}\right|^2}{4}\left|\xi\right|^2 \leq -\eta_{\min}(L)\left|e\right|^2 + \frac{\left|\bar{g}\right|^2}{4}\left|\xi\right|^2 \\ &\leq -(\eta_{\min}(L) + \frac{\left|\bar{g}\right|^2}{4})\left|e\right|^2, \quad \forall\left|e\right| > \left|\xi\right|,\end{aligned} \tag{33}$$

where $e := [e_1^\top, e_2^\top]^\top \in \mathbb{R}^{2m}$, $L := \mathrm{diag}(L_1, L_2 - I_{m\times m}) \in \mathbb{R}^{2m\times 2m}$, and $\eta_{\min}(L) := \min\{\eta_{\min}(L_1), \eta_{\min}(L_2 - I_{m\times m})\}$ denotes the minimum eigenvalue of L. Note that $L_2 - I_{m\times m} > 0$ is required to make L as one positive definite matrix. This requirement provides practitioners with guidelines to choose suitable values of L_2. It is concluded that the tracking errors e_1 and e_2 are ISS with $\alpha_3(\bullet) = -\eta_{\min}(L)\left|\bullet\right|^2$, $\alpha_4(\bullet) = \frac{\left|\bar{g}\right|^2}{4}\left|\bullet\right|^2$ based on Definition 1. Then, $|e(t)| \leq \lambda(e(t_0), t) + \gamma(|\xi(t)|_\infty)$ holds according to Definition 2, i.e., the tracking error e remains in a ball with radius $\lambda(e(t_0), t) + \gamma(|\xi(t)|_\infty)$. Besides, as time t increases, the tracking error e approaches to a smaller ball of radius $\gamma(|\xi(t)|_\infty)$ given that for fixed $e(t_0)$, the $\mathcal{KL}$ function λ decreases to zero as $t \to \infty$.

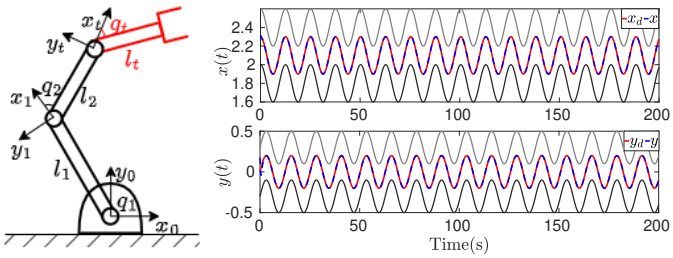
Proof of 2) The establishment of (33) implies that V is bounded. Thereby, V_1 is bounded. Given that $e_1 \to -\underline{\epsilon}$ or $e_1 \to \bar{\epsilon}$ leads to $V_1 \to \infty$ according to (3d). Thus, the bounded V_1 proves that the tracking error e_1 lies in the set $\mathbb{E}$.

Proof of 3) The actual execution position of the controlled plant is $p = p_d + e_1$. Based on the fact that $e_1 \in \mathbb{E}$, the possible trajectory lies in the set $\bar{\mathbb{D}} := \{p(t) \in \mathbb{R}^{n_1} : p_d - \underline{\epsilon} \prec p(t) \prec p_d + \bar{\epsilon}\}$. By choosing $-\underline{\epsilon} > \underline{p}(t) - p_d$ and $\bar{\epsilon} \prec \overline{p}(t) - p_d$ and combining with Assumption 1, $\underline{p}(t) \prec p_d - \underline{\epsilon}$ and $p_d + \bar{\epsilon} \prec \overline{p}(t)$ hold. Thus, it is proved that $\bar{\mathbb{D}} \in \mathbb{D}$, i.e., the actual execution trajectory $p(t)$ always lies in the safe region $\mathbb{D}$ even the controlled plant (1) suffers from model uncertainties and environmental disturbances. □

V. Numerical Simulation

A. Safe Operation of Robot Manipulator

This subsection concentrates on a 2-DoF robot manipulator Cartesian-space tracking task under varying kinematics settings to exemplify the kinematics free property of our method. The adopted 2-DoF robot manipulator serves as a benchmark to validate the effectiveness and superiority of our method. The explicit kinematic and dynamic knowledge of the robot manipulator used for simulation purposes is referred to in [10].



(a) 2-DoF robot and tool. (b) Trajectories of $x(t)$, $y(t)$ and safe boundary.

Fig. 2: The robot and the end-effector Cartesian-space position under varying kinematics settings (Task 1 case).

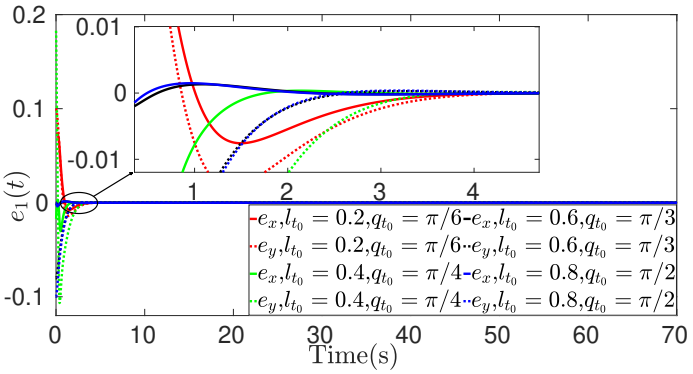


Fig. 3: The trajectories of the Cartesian-space tracking error $e_1 := [e_x, e_y]^\top$ under varying kinematics and different tasks.

The robot manipulator in one restricted environment is required to grasp diverse tools (tools in different lengths l_t, and grasping angles q_t) to complete different tracking tasks in a provable safe way, see Fig. 2a. In particular, the working space of the end-effector should be always confined to one specific region that is treated as a prohibited area for humans or other robots. Note that the information of kinematics, dynamics, and tool (i.e., values of l_t and q_t in Fig. 2a) are unavailable to practitioners to perform the controller design. To accomplish the above task, the collision-free desired trajectory $p_d \in \mathbb{R}^2$ (a circle with center $c := (c_x, c_y)$ and radius r) is firstly planned under buffered obstacles with buffer size $\epsilon = 0.3$. This buffer size then serves as the performance bound of our designed performance-guaranteed control strategy (22) to ensure that the tool end track the collision-free desired trajectory p_d with the predetermined tracking accuracy $\underline{\epsilon} = [-0.3, -0.3]^\top$, and $\bar{\epsilon} = [0.3, 0.3]^\top$.

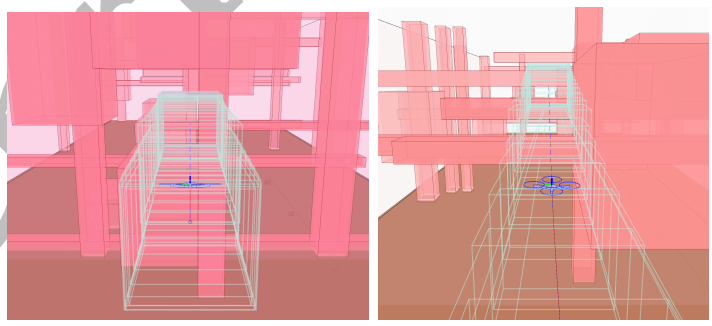
The initial conditions are set as $q(0) = [0, 0]^\top$, $\tau(0) = [0, 0]^\top$. The parameters required for the incremental control input (21) are set as: $\bar{g} = \mathrm{diag}(10, 10)$, $L_1 = \mathrm{diag}(1, 1)$, and $L_2 = \mathrm{diag}(2, 2)$. The sampling rate is 1kHz. Note that we always keep the same parameter setting to conduct the following different numerical simulations. This exemplifies the robustness of our developed method.

The robot manipulator uses different tools (different initial lengths l_{t_0}) installed with different initial angles q_{t_0} to complete the following four Cartesian-space tracking tasks. Task 1: $c_1 = (2.1, 0)$, $r_1 = 0.2$ m, $l_{t_0} = 0.2$ m, $q_{t_0} = \pi/6$; Task 2: $c_2 = (2.3, 0.1)$, $r_2 = 0.2$ m, $l_{t_0} = 0.4$ m, $q_{t_0} = \pi/4$; Task 3: $c_3 = (2.3, 0.6)$, $r_3 = 0.2$ m, $l_{t_0} = 0.6$ m, $q_{t_0} = \pi/3$; and Task 4: $c_4 = (2, 0.9)$, $r_4 = 0.2$ m, $l_{t_0} = 0.8$ m, $q_{t_0} = \pi/2$. To fully exemplify the kinematics free property of our method, we additionally consider the non-trivial varying kinematics setting here. In particular, we purposely set the tool length as $l_t = l_{t_0} - 0.0002\, t$ and the grasping angle as $q_t = q_{t_0} - 0.002\, t$ during the working process, where t denotes the current time. The above varying tool length l_t and grasping angle q_t might be caused by wear and tear or poor fixation in industrial productions. This varying kinematic setting invalidates common approaches that require the inverse kinematics calculation.

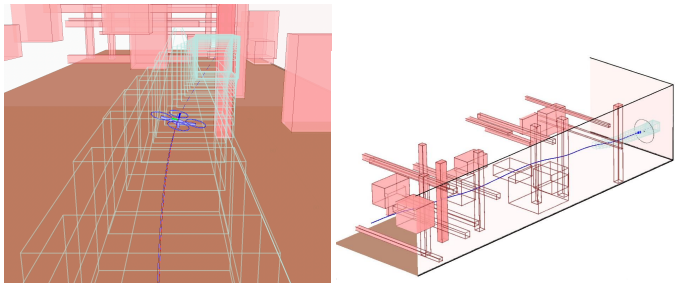
The Cartesian-space position trajectories displayed in Fig. 2b illustrate that the tool end always lies in the predetermined safe region. Furthermore, the high-accuracy tracking performance shown in Fig. 3 validates the flexibility and adaptability of our developed approach towards different tasks under the varying kinematic settings mentioned above.

B. *Safe Flight of Quadrotor*

This subsection numerically validates the generality of our proposed SP-PGC scheme under one safety-critical task of a 6-DoF quadrotor. The quadrotor is required to safely fly in an obstacle-filled environment and finally reach the target position, see Fig. 4. To realize this goal, we firstly use the reachable



(a) The flight trajectory at $t = 0.8$ s. (b) The flight trajectory at $t = 2.3$ s.



(c) The flight trajectory at $t = 6.9$ s. (d) The global view of flight trajectory.

Fig. 4: The illustration of the safe execution of quadrotor in one safety-critical environment (red line: planned trajectory; blue line: real trajectory).

set based planning algorithm [24] to generate a collision-free desired trajectory $p_d := [x_d, y_d, z_d] \in \mathbb{R}^3$ inside a tube within consideration of the distance from the quadrotor's center of mass to the rotor center $d_q = 0.27$ m. Thereby, Assumption 1 is satisfied. Then, we follow the procedures illustrated in Section

III and Section IV to design an online performance-guaranteed position tracking controller (22) to ensure that the quadrotor always flies in the planned safe tube. The incremental dynamic inversion method (a reformulation of the dynamic inversion method [32] based on the incremental system formulated in Section III) is adopted to design the attitude controller given that we have no specific performance requirements for the attitude control.

Denoting the boundary of the reachable set as $b(t) \in \mathbb{R}^3$. Given that the desired trajectory points $p_d(t)$ are on the center line of the reachable sets, the allowable control-level tracking error (performance bound) to ensure safety follows $k(t) = b(t) - p_d(t) - d_q$. Note that rather than using this varying $k(t)$ to construct ISS-PS-BLFs, we use the minimum value of $k(t)$ ($\underline{\epsilon} = [-0.05, -0.05, -0.05]^\top$, and $\bar{\epsilon} = [0.05, 0.05, 0.05]^\top$ in particular) to exemplify the realizable high-accuracy tracking performance of our designed control strategy (22). The simulation parameters for the position controller are set as: $p_0(t) = [1, 0, 5]^\top$, $\bar{g} = \mathrm{diag}(6.5, 6.5, 6.5)$, $L_1 = \mathrm{diag}(0.25, 0.25, 0.25)$, and $L_2 = \mathrm{diag}(8, 8, 8)$.

We validate the effectiveness of our approach in six randomly generated environments. The interactions between the quadrotor and the environment at specific time instants are displayed in Fig. 4. The associated videos are referred to https://youtu.be/VKlaqWJBxus. Our designed controller enables the quadrotor to always fly inside the safe tunnels (see Fig. 4a-4c) and finally reach the target position (see Fig. 4d).

VI. Experimental Validation

This section experimentally validates the robustness enhancement brought by the dynamics free property of our method via the task-space tracking task of the 3-DoF robot manipulator (see Fig. 5) in the Chair of Automatic Control Engineering (LSR), Technical University of Munich (TUM). The forward kinematics is the same as (12). The dynamics follows

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F_v(\dot{q}) = \tau + \tau_d,$$

which differs from (14) given that the robot manipulator is restricted to a horizontal plane. Thus, the gravity term is omitted. Please refer to [14] for more details of hardware. The numerical differential technique [28] is adopted to get the velocity and acceleration information in practice.

An industrial welding and cutting task is considered here. We choose $\underline{\epsilon} = [-0.01, -0.01]^\top$, $\bar{\epsilon} = [0.01, 0.01]^\top$ for our designed performance-guaranteed tracking controller (22) to drive the robot manipulator end-effector to realize precision machining. To ensure robust safety, the above determined performance bounds ($\underline{\epsilon}$ and $\bar{\epsilon}$ in particular) are considered in the planning level to inflate obstacles by a $\epsilon = 0.01$ margin before generating circle reference signals $p_d := [x_d, y_d] \in \mathbb{R}^2$ (encoding the industrial task). The remaining parameters required for the tracking controller (22) are set as $\bar{g} = \begin{bmatrix} 120 & 0 & 0 \\ 0 & 120 & 80 \end{bmatrix}$, and $L_1 = \mathrm{diag}(0.1, 0.1)$, $L_2 = \mathrm{diag}(20, 20)$. The initial position is set as $x_1(0) = [0.88, 0]^\top$, $x_2(0) = [0, 0]^\top$, $u = [0, 0, 0]^\top$. The sampling rate is 1kHz.

The Cartesian-space trajectory $x(t)$ displayed in Fig. 6a (the trajectory $y(t)$ is similar to Fig. 6a, thus omitted for simplicity)

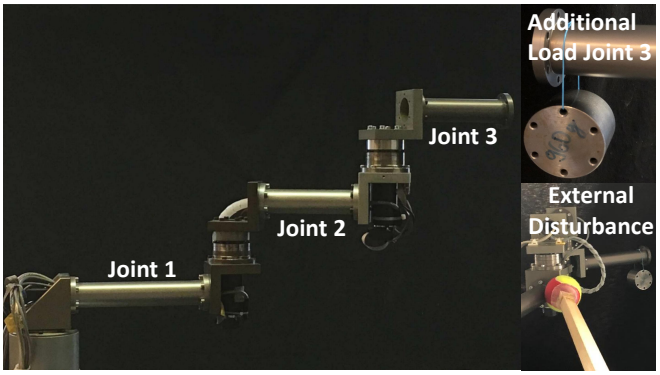
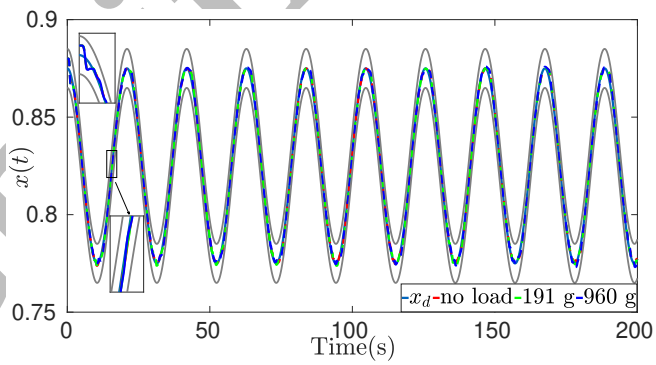
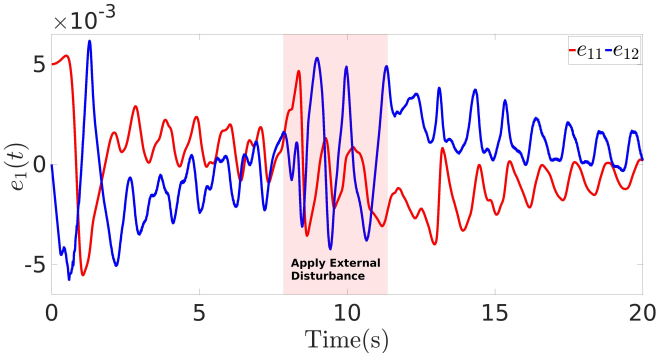


Fig. 5: The 3-DoF robot manipulator, load and external torque.

shows that the robot manipulator driven by our developed control strategy (22) efficiently tracks the desired trajectory precisely without crossing the predefined safe boundary even under different loads. To further demonstrate the robustness property of our method, we use one stick to apply additional torque to the robot manipulator. As shown in Fig.6b, the trajectories of the tracking error $e_1(t)$ firstly oscillate due to the external disturbance and then converge to a small value around zero.



(a) The trajectories of x, x_d, and safe bound under different loads.



(b) The tracking error e_1 (960 g case) under external disturbance.

Fig. 6: The experimental results of the SP-PGC scheme.

VII. Conclusion

This work realizes the safe control under uncertainty via our formulated ISS-PS-BLF and incremental system. The utilized one-step backward data reformulates kinematic and

dynamic uncertainties as well as environmental disturbances into a provably bounded estimation error, which allows us to rigorously analyze the robustness of safety via an input-to-state stable approach. The safe planning algorithm and the ISS-PS-BLF facilitated tracking controller work together to ensure that autonomous systems realize the safe execution under uncertainty with guaranteed performance. Numerical and experimental validations are conducted to show the efficiency of our proposed SP-PGC scheme. Future works aim to extend the kinematics free and dynamics free control strategy to soft robot manipulators. Besides, the influence of noisy measurements on our developed approach remains to be investigated. Furthermore, extending the proposed method to control-nonaffine system and input saturation (approximation method similar to [33]) is worth investigating.

REFERENCES

[1] N. Hudson, F. Talbot, M. Cox, J. Williams, T. Hines, A. Pitt, B. Wood, D. Frousheger, K. Lo Surdo, T. Molnar *et al.*, "Heterogeneous ground and air platforms, homogeneous sensing: Team csiro data61's approach to the darpa subterranean challenge," *Field Robotics*, vol. 2, no. 1, pp. 595–636, 2022.

[2] C. Yan, J. Qin, Q. Liu, Q. Ma, and Y. Kang, "Mapless navigation with safety-enhanced imitation learning," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 7, pp. 7073–7081, 2023.

[3] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2015.

[4] J. Tordesillas, B. T. Lopez, M. Everett, and J. P. How, "Faster: Fast and safe trajectory planner for navigation in unknown environments," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 922–938, 2021.

[5] Y. Li, Z. Littlefield, and K. E. Bekris, "Asymptotically optimal sampling-based kinodynamic planning," *The International Journal of Robotics Research*, vol. 35, no. 5, pp. 528–564, 2016.

[6] W. Gan, D. Zhu, Z. Hu, X. Shi, L. Yang, and Y. Chen, "Model predictive adaptive constraint tracking control for underwater vehicles," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 9, pp. 7829–7840, 2019.

[7] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

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

[9] S. Lian, W. Meng, Z. Lin, K. Shao, J. Zheng, H. Li, and R. Lu, "Adaptive attitude control of a quadrotor using fast nonsingular terminal sliding mode," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 2, pp. 1597–1607, 2021.

[10] H. Wang and Y. Xie, "Adaptive inverse dynamics control of robots with uncertain kinematics and dynamics," *Automatica*, vol. 45, no. 9, pp. 2114–2119, 2009.

[11] L. Cheng, Z.-G. Hou, and M. Tan, "Adaptive neural network tracking control for manipulators with uncertain kinematics, dynamics and actuator model," *Automatica*, vol. 45, no. 10, pp. 2312–2318, 2009.

[12] H. Ji, W. Shang, and S. Cong, "Adaptive synchronization control of cable-driven parallel robots with uncertain kinematics and dynamics," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 9, pp. 8444–8454, 2020.

[13] P. Shi, W. Sun, X. Yang, I. J. Rudas, and H. Gao, "Master-slave synchronous control of dual-drive gantry stage with cogging force compensation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2022.

[14] C. Li, F. Liu, Y. Wang, and M. Buss, "Concurrent learning-based adaptive control of an uncertain robot manipulator with guaranteed safety and performance," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

[15] J. Boedecker, J. T. Springenberg, J. Wülfing, and M. Riedmiller, "Approximate real-time optimal control based on sparse gaussian process models," ser. 2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL). IEEE, 2014, pp. 1–8.

[16] Z. Liu, W. Lin, X. Yu, J. J. Rodríguez-Andina, and H. Gao, "Approximation-free robust synchronization control for dual-linear-motors-driven systems with uncertainties and disturbances," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 10, pp. 10 500–10 509, 2021.

[17] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: survey, classification and perspective," *Information Sciences*, vol. 235, pp. 3–35, 2013.

[18] P. Zhao, A. Lakshmanan, K. Ackerman, A. Gahlawat, M. Pavone, and N. Hovakimyan, "Tube-certified trajectory tracking for nonlinear systems with robust control contraction metrics," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5528–5535, 2022.

[19] C. Ho, J. Patrikar, R. Bonatti, and S. Scherer, "Adaptive safety margin estimation for safe real-time replanning under time-varying disturbance," *arXiv preprint arXiv:2110.03119*, 2021.

[20] X. Jin, "Adaptive fixed-time control for mimo nonlinear systems with asymmetric output constraints using universal barrier functions," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 3046–3053, 2018.

[21] E. D. Sontag and Y. Wang, "On characterizations of the input-to-state stability property," *Systems & Control Letters*, vol. 24, no. 5, pp. 351–359, 1995.

[22] E. D. Sontag *et al.*, "Smooth stabilization implies coprime factorization," *IEEE Transactions on Automatic Control*, vol. 34, no. 4, pp. 435–443, 1989.

[23] S. M. LaValle, *Planning algorithms*. Cambridge University Press, 2006.

[24] S. Kousik, S. Vaskov, F. Bu, M. Johnson-Roberson, and R. Vasudevan, "Bridging the gap between safety and real-time performance in receding-horizon trajectory design for mobile robots," *The International Journal of Robotics Research*, vol. 39, no. 12, pp. 1419–1469, 2020.

[25] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, 2017.

[26] 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.

[27] W. Wang and Z. Gao, "A comparison study of advanced state observer design techniques," in *Proceedings of the 2003 American Control Conference, 2003.*, vol. 6. IEEE, 2003, pp. 4754–4759.

[28] T. S. Hsia, "A new technique for robust control of servo systems," *IEEE Transactions on Industrial Electronics*, vol. 36, no. 1, pp. 1–7, Feb. 1989.

[29] A. Levant, "Robust exact differentiation via sliding mode technique," *Automatica*, vol. 34, no. 3, pp. 379–384, 1998.

[30] T. Luukkonen, "Modelling and control of quadcopter," *Independent research project in applied mathematics, Espoo*, vol. 22, pp. 2–6, 2011.

[31] K. Youcef-Toumi and S.-T. Wu, "Input/output linearization using time delay control," *Journal of dynamic systems, measurement, and control*, vol. 114, no. 1, pp. 10–19, 1992.

[32] D. Enns, D. Bugajski, R. Hendrick, and G. Stein, "Dynamic inversion: an evolving methodology for flight control design," *International Journal of Control*, vol. 59, no. 1, pp. 71–91, 1994.

[33] C. Wen, J. Zhou, Z. Liu, and H. Su, "Robust adaptive control of uncertain nonlinear systems in the presence of input saturation and external disturbance," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1672–1678, 2011.