SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

November 2024, Vol. 67, Iss. 11, 212207:1–212207:17
https://doi.org/10.1007/s11432-023-4115-1

Robust tube-based MPC with smooth computation for dexterous robot manipulation

Yu LUO[1], Tianying JI[1], Fuchun SUN[1*], Qie SIMA[1], Huaping LIU[1], Mingxuan JING[2] & Jianwei ZHANG[3]

[1]*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084*, China;*
[2]*Science & Technology on Integrated Information System Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing* 100190*, China;*
[3]*Department of Informatics, University of Hamburg, Hamburg* 20148*, Germany*

Received 6 November 2023/Revised 18 March 2024/Accepted 29 April 2024/Published online 23 October 2024

Abstract Dexterous robot manipulation has shone in complex industrial scenarios, where multiple manipulators, or fingers, cooperate to grasp and manipulate objects. When encountering multi-objective optimization with system constraints in such scenarios, model predictive control (MPC) has demonstrated exceptional performance in complex multi-robot manipulation tasks involving multi-objective optimization with system constraints. However, in such scenarios, the substantial computational load required to solve the optimal control problem (OCP) at each triggering instant can lead to significant delays between state sampling and control application, hindering real-time performance. To address these challenges, this paper introduces a novel robust tube-based smooth MPC approach for two fundamental manipulation tasks: reaching a given target and tracking a reference trajectory. By predicting the successor state as the initial condition for imminent OCP solving, we can solve the forthcoming OCP ahead of time, alleviating delay effects. Additionally, we establish an upper bound for linearizing the original nonlinear system, reducing OCP complexity and enhancing response speed. Grounded in tube-based MPC theory, the recursive feasibility and closed-loop stability amidst constraints and disturbances are ensured. Empirical validation is provided through two numerical simulations and two real-world dexterous robot manipulation tasks, which shows that the seamless control input by our methods can effectively enhance the solving efficiency and control performance when compared to conventional time-triggered MPC strategies.

Keywords dexterous robot manipulation, model predictive control, smooth computation

1 Introduction

Dexterous robot manipulation, the agile and precise handling of objects by robotic systems, is a rapidly evolving field at the intersection of robotics, artificial intelligence, and engineering [1]. In the realm of dexterous robot manipulation, control strategies play a pivotal role in enabling robots to perform intricate tasks with precision and adaptability. These strategies encompass a diverse range of methodologies which can be categorized into three main kinds: proportion integration differentiation (PID)-based control, optimization-based control [2, 3], and learning-based control [4]. Model predictive control (MPC), an optimization control strategy [5, 6], holds a significant place in optimization-based control methods, by optimizing system targets and simultaneously managing state and control input constraints [7–9].

Within the context of dexterous manipulation, MPC is notably applied in object grasping, manipulation, tracking, assembly, and obstacle avoidance scenarios [10]. It facilitates the adjustments of joint angles, forces, and torques, enabling robots to accommodate object pose uncertainties during grasping operations [11–13]. In this paper, we abstract these tasks into two fundamental manipulation tasks: position reaching and trajectory tracking, whose derivations can be extended into numerous dexterous manipulation tasks [1, 14, 15]. Specifically, position reaching requires the endpoint of the manipulator to reach a given target point while avoiding environmental obstacles, which can be generalized to tasks

* Corresponding author (email: fcsun@tsinghua.edu.cn)

such as pick and place and grab objectives [16, 17]. On the other side, trajectory tracking means that the endpoint of the manipulator tracks a given reference trajectory to complete the task, which can be extended in tasks like assembly and spraying [18, 19].

When applying MPC, it may present certain disadvantages including excessive computational complexity, high computational delay [20], and sensitivity to parameters. At each sampling instant, the controller receives the real state and computes the input to the plant. Excessively long solution times would cause the system's state to change significantly, magnifying the mismatch between the real state and the control input thus causing unpredictable oscillations [21–23]. Besides, high computational delay can lead to error accumulation over time, causing the robot to deviate from its intended trajectory or even constraint violations [24]. Addressing these issues in the context of fast dynamic systems and resource-limited platforms has spurred impressive efforts, primarily manifesting in two approaches: reducing computational complexity or diminishing optimal control problem (OCP) solving frequency [25–27]. Magni et al. [28] computed the linearization of the stabilizing nonlinear receding horizon to reduce the computational load at each sampling instant, and Rahideh and Shaheed [29] applied this approach to a twin rotor multiple-input multiple-output (MIMO) system to improve efficiency. Another effective technique, aside from model linearization, involves curtailing prediction horizons to trim OCP computation time [30, 31]. Lowering OCP solving frequency has gained attention, leading to the development of event-triggered MPC and self-triggered MPC. Li et al. [32] and Liu et al. [33] explored event-triggered MPC's continuous-time nonlinear systems application, a concept extended to modular reconfigurable robots' decentralized tracking control in [34]. Although these efforts address computational complexity, the single solving time at each update instance often falls short of real-time requirements [2, 3, 35, 36].

In light of this practical quandary, efforts have emerged to eliminate delays caused by asynchronous sampling and input, leveraging MPC's predictive capabilities. In [37], an advanced-step nonlinear MPC controller is introduced to anticipate the next nominal system states. However, this approach encountered a challenge wherein disturbances caused deviations between the next nominal and real states, leading to inaccuracies in predictions. Addressing the issue, Ref. [38] presented a dual time-scale control scheme for linear/nonlinear systems featuring external disturbances. This scheme employs a pre-compensator and an outer MPC controller operating at disparate frequencies, aiming to subdue uncertainty and ensure stability. However, open-loop optimal control performance cannot be guaranteed under this approach. Alternatively, Ref. [39] capitalized on feed-forward action characteristics, enabling current control action computation from the preceding sampling interval for LPV models with bounded disturbance. Building on future state prediction, Ref. [40] executed the MPC policy preemptively at the current instant, supported by the assurance of recursive feasibility and closed-loop stability.

Drawing inspiration from the aforementioned challenges, we introduce a novel approach: robust tube-based smooth model predictive control for dexterous robot manipulation to deal with constraints and disturbances, as illustrated in Figure 1. In a departure from prior studies, our method foresees the next real state region and then solves the OCP ahead at the present moment, grounded in nominal predictions. Leveraging nonlinear system linearization, we mitigate computational load, and harness the robustness framework inherent in tube MPC. Compared with the previous work [41], in this work the crux of our contributions encompasses three key aspects:

• **Compact predictive disturbed state set.** By deriving state deviation bounds between the real and nominal systems, we establish a more compact predictive disturbed state set for forthcoming real states when compared to the fraction function in [41]. This set serves as the foundational initial condition for the succeeding OCP at the present instant. Consequently, the ensuing OCP is optimized ahead, with optimal outcomes directly employed in the subsequent triggering instance. This anticipatory stride effectively circumvents computational delays, improving the smoothness of the MPC framework.

• **Reduced solving complexity by piece linearization.** To enhance solving speed by diminishing the nonlinear OCP solving complexity, we employ piece linearization technology. This transformative approach translates the nonlinear OCP into a linear counterpart, thereby facilitating the efficiency of the solution. Specifically, we estimate the bound of the linearization error as an additional disturbance, leveraging the assumptions of Lipschitz continuity and Lagrange remainder. This ensures the consistency of the two systems, ensuring the recursive feasibility and closed-loop stability of the original system, which is deficient in the previous work [41].

• **Performance evaluation in simulation and real-world experiments.** We corroborate our approach's efficacy through both simulation and real-world system validation. Comparative analyses against ideal MPC and time-triggered MPC methods reveal heightened response speed and superior

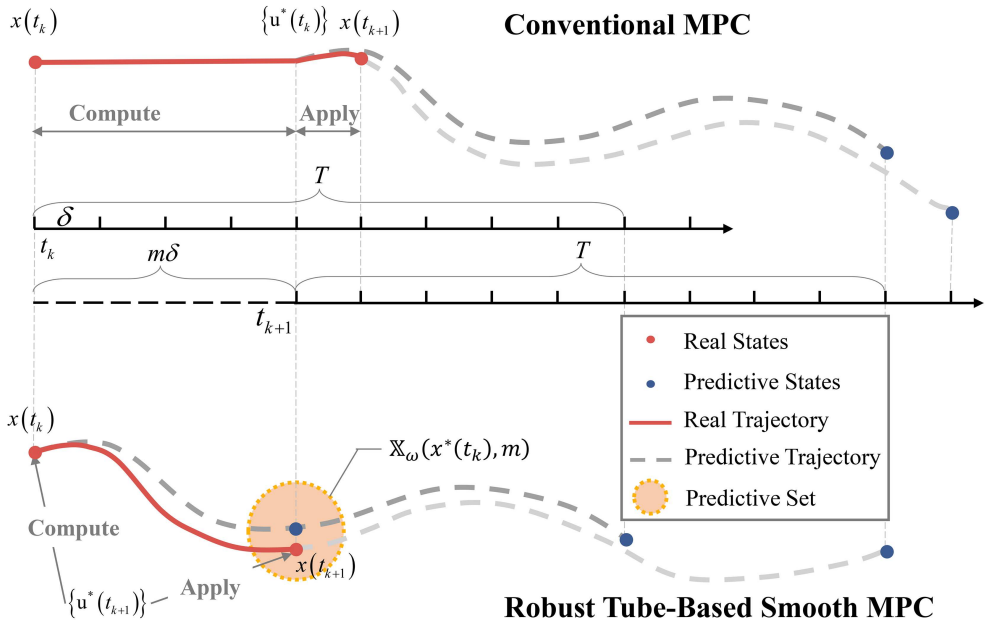

Figure 1 (Color online) Comparison between the conventional time-triggered MPC and our proposed robust tube-based smooth MPC approach. In the conventional time-triggered MPC, the computational time for solving the OCP requires several sampling intervals ($m\delta$, with $m \geqslant 1$). When yielding the optimal control input, the state would be shifted and the state deviation caused by the delay would result in suboptimality between the sampling and input stages. In contrast, our robust tube-based smooth MPC strategy predicts the next real states at t_k, denoted as $\tilde{x}(t_{k+1}|t_k)$ within the predictive disturbed state set $\mathbb{X}_\omega(x^*(t_k), m)$. Then, with this set as the initial condition for the next OCP, we can compute the optimal control input $u^*(t|t_{k+1})$ ahead. When sampling the next real system state $x(t_{k+1})$, we can seamlessly apply the computed control input without obvious delay.

optimal performance.

Notation. $\mathbb{N}$ is the natural integers and $\mathbb{R}$ is the real numbers. $\mathbb{R}^n$ means the n-dimension vector space. For a matrix M, $M \preceq 0$ denotes that the real parts of all eigenvalues of M are negative. For a vector x, $\|x\| \triangleq \sqrt{x^{\mathrm{T}}x}$ and $\|x\|_P$ with the positive definite matrix P means $\|x\|_P \triangleq \sqrt{x^{\mathrm{T}}Px}$. If a vector is shown as $x(t)$, it is a sequence $\{x(t), x(t+1), \ldots\}$. $(k+i|k)$ indicates a prediction of a variable i steps ahead from time k. We use $\bar{\cdot}$ to mean a feasible variable satisfying all constraints and $\cdot^*$ as an optimal variable obtained by solving OCP. For any set $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{n\times n}$, $\mathcal{A} \oplus \mathcal{B}$ is the Minkowski set addition, which means $\mathcal{A} \oplus \mathcal{B} \triangleq \{a+b | a \in \mathcal{A}, b \in \mathcal{B}\}$ and $\mathcal{A} \ominus \mathcal{B}$ is the Pontriagin set subtraction, which means $\mathcal{A} \ominus \mathcal{B} \triangleq \{a | \forall b \in \mathcal{B}, a+b \in \mathcal{A}\}$.

2 Problem formulation

2.1 System description

In this section, we delve into the problem formulation, focusing on a specific class of continuous nonlinear system models characterized by state and control input constraints, to represent various dexterous manipulation tasks. The system model is defined as follows:

$$\dot{x}(t) = f(x(t), u(t)) + e(t),\ x(t_0) = x_0,\ t_0 \geqslant 0, \tag{1}$$

where $x(t) \in \mathbb{R}^n$ represents the system state, $u(t) \in \mathbb{R}^m$ denotes the control input, and $f(\cdot,\cdot)$ captures the system dynamics. In our analysis, we assume that the unknown disturbances $e(t)$ are bounded, with an upper bound $\eta = \sup_{e(t)\in\mathcal{E}} \|e(t)\|$. In practical control scenarios, the system encounters mechanical, resource, or geometric limitations, thereby imposing strict constraints on both the system state and control inputs:

$$x(t) \in \mathbb{X} \subseteq \mathbb{R}^n, u(t) \in \mathbb{U} \subseteq \mathbb{R}^m. \tag{2}$$

The feasible regions $\mathbb{X}$ and $\mathbb{U}$ are considered convex and compact, encompassing the origin as an interior point. Furthermore, we introduce an assumption regarding the Lipschitz continuity and linearizability

characteristics of the nonlinear system model $f(\cdot,\cdot)$. Before embarking on the analysis of the system (1), we introduce the nominal system:

$$\dot{x}(t) = f(x(t), u(t)). \tag{3}$$

This nominal system serves as a deterministic counterpart and forms the basis for establishing the robustness of the original system.

Assumption 1 (System properties)**.** We outline key properties of the nonlinear system model function $f(x,u)$ as follows:

(i) The function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is twice continuously differentiable, with $f(0,0) = 0$.

(ii) The system is of linearizability around the origin, yielding the linearized representation:

$$\dot{x} = Ax(t) + Bu(t) + e(t), \tag{4}$$

where $A = \partial f/\partial x|(0,0)$ and $B = \partial f/\partial u|(0,0)$. Additionally, the system matrix pair (A,B) is stabilizable, supported by the existence of a feedback gain matrix K ensuring $A + BK \preceq 0$.

(iii) The system model allows linearization at each time instant t_k, yielding time-varying system matrices

$$A_{t_k} = \left.\frac{\partial f}{\partial x}\right|_{(x(t_k),u(t_k))}, \quad B_{t_k} = \left.\frac{\partial f}{\partial u}\right|_{(x(t_k),u(t_k))}. \tag{5}$$

Furthermore, within the piece control interval $[t_k, t_{k+1}]$, the Hessian matrix $H(x(t),u(t))$ is bounded by

$$\|H(x(t),u(t))\| = \left| \begin{array}{cc} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial u} \\ \\ \frac{\partial^2 f}{\partial u \partial x} & \frac{\partial^2 f}{\partial u^2} \end{array} \right|_{(x(t),u(t))} \leqslant \eta_R, \tag{6}$$

for $t \in [t_k, t_{k+1}]$.

Furthermore, we present the following lemma, elucidating the Lipschitz continuity of the system (1).

Assumption 2 (Lipschitz continuity)**.** We assume the local Lipschitz continuity property for the nonlinear system dynamics $f(x,u)$ w.r.t state x and control input u. With control inputs $u_1, u_2 \in \mathbb{U}$, and for any state pair $x_1, x_2 \in \mathbb{X}$, the system adheres to

$$\|f(x_1,u_1) - f(x_2,u_2)\| \leqslant l_1\|x_1 - x_2\| + l_2\|u_1 - u_2\|, \tag{7}$$

where l_1 and l_2 represent the Lipschitz constants characterizing the nonlinearity of the system (1).

This assumption underscores that the system's nonlinearity can be bounded by constants l_1 and l_2, paving the way for insights into state prediction strategies.

Control objective. With the established system dynamics (1) and the state and control input constraints (2), we aim to transform current state $x(t)$ to reach the expected state x^{T} or track the target state trajectory $x^{\mathrm{T}}(t)$, i.e., $\lim_{t\to\infty} \|x(t) - x^{\mathrm{T}}(t)\| = 0$ within MPC. Due to the existence of computational delay, how to ensure the one-to-one correspondence of the real system states and the optimal control input under the framework of MPC is the key improvement of this paper.

2.2 Conventional robust MPC

The primary control objective is to stabilize the system state towards the equilibrium point during the control process. To guarantee the constraints and attain optimal control performance, we initiate the process by employing conventional time-triggered robust MPC. Here, we define T as the prediction horizon. At each discrete sampling instant t_k, the conventional MPC undertakes the resolution of an OCP to determine an optimal control sequence $u^*(t|t_k)$, where $t \in [t_k, t_k + T]$. The formulation of the cost function across the prediction horizon takes the shape of

$$J(\bar{x}(t|t_k), \bar{u}(t|t_k), t_k) = \int_{t_k}^{t_k+T} L(\bar{x}(t|t_k), \bar{u}(t|t_k))\mathrm{d}t + V_f(\bar{x}(t_k + T|t_k)), \tag{8}$$

Algorithm 1 Conventional dual-mode time-triggered MPC

Require: Nominal system (3), weight matrices Q, P, and R, Lipschitz constants l_1 and l_2, radius of the robust terminal region ϵ and prediction horizon T;

```
k = 0;
while the system is not converged do
    Sample the current state x(t_k);
    if x(t_k) ∉ 𝕏_ε then
        Solve the OCP 1 and obtain the optimal input u*(t_k);
        Apply u*(t_k) until the next sampling instant;
    else
        Let u(t_k) = Kx(t_k) denote the system controller;
    end if
    Let k = k + 1;
end while
```

where $L(\bar{x}(t|t_k), \bar{u}(t|t_k)) = \|\bar{x}(t|t_k)\|_Q^2 + \|\bar{u}(t|t_k)\|_R^2$ is the stage cost function and $V_f(x(t_k+T|t_k)) = \|\bar{x}(t_k+T|t_k)\|_P^2$ is the terminal penalty cost function. Here, the matrices Q and P are positive semi-definite, while R is a positive definite matrix. Following the principles of conventional time-triggered MPC, we proceed to formulate the optimal control problem (OCP 1):

$$u^*(t_k) = \min_{\bar{u}(t|t_k)\in\mathbb{U}} J(\bar{x}(t|t_k), \bar{u}(t|t_k), t_k), \tag{9}$$

subject to

$$\bar{x}(t_k|t_k) = x(t_k), \tag{10a}$$
$$\dot{\bar{x}}(t|t_k) = f(\bar{x}(t|t_k), \bar{u}(t|t_k)), \tag{10b}$$
$$\bar{x}(t|t_k) \in \mathbb{X} \ominus \mathbb{X}_e(t), \quad \bar{u}(t|t_k) \in \mathbb{U}, \tag{10c}$$
$$\bar{x}(t_k+T|t_k) \in \mathbb{X}_\epsilon, \quad t \in [t_k, t_k+T], \tag{10d}$$

where $\mathbb{X}_e(t) = \{x : \|\bar{x}\| \leqslant t\eta(1+l_1)^t\}$ is a tightened state constraint set to improve the robustness of the system and $\mathbb{X}_\epsilon = \{x : \|\bar{x}\|_P \leqslant \epsilon, \epsilon > 0\}$ signifies the robust terminal region [42, 43].

Remark 1. Notably, despite the discrete nature of MPC implementation, we opt to sample the state at each discrete instant and address a continuous optimal control problem. The derived optimal control sequence is then applied until the subsequent sampling instant. Furthermore, given the elusive nature of disturbances, we leverage the nominal system to forecast the future behavior of the original system, as illustrated in (10b).

In the context of this assumption, when the state resides near the region of the origin, a local controller $u(t) = Kx(t)$ can be devised to maintain system stability. However, due to system constraints, this feedback controller may not be viable for the whole control process. To address this, we introduce a dual-mode control strategy. Specifically, if $\forall x \in \mathbb{X}$, we adopt MPC to obtain the optimal control input and satisfy the system constraints. Conversely, the feedback controller is employed for conciseness. We summarise the dual-mode control strategy in Algorithm 1.

To establish the theoretical underpinnings of recursive feasibility and closed-loop stability for nominal MPC, we present the following fundamental lemma supported by previous studies.

Lemma 1 (Recursive feasibility and stability for nominal MPC [43, 44])**.** For the nominal system (3), conventional MPC guarantees recursive feasibility and closed-loop stability under the following conditions:

(i) The OCP 1 possesses a feasible solution at the initial instant t_0;

(ii) A local stabilizing controller $\kappa_f(x)$ exists in the robust terminal region, satisfying $\forall x \in \mathbb{X}_\epsilon, \kappa_f(x) \in \mathbb{U}$ and leading to the condition

$$\dot{V}_f(\bar{x}(t)) \leqslant -L(\bar{x}(t), \kappa_f(\bar{x}(t))). \tag{11}$$

Additionally, for the choice of $\kappa_f(x) = Kx$, the Lyapunov equation involving weight matrices Q, R, and P holds:

$$(A+BK)^{\mathrm{T}}P + P(A+BK) \preceq -Q^*, \tag{12}$$

where $Q^* = Q + K^{\mathrm{T}}RK$.

3 Robust tube-based smooth MPC

The significant computation delay encountered during the solution of OCP 1 engenders a challenge: the system controller is compelled to retain the input value during the waiting period, potentially leading to suboptimal or even unstable system behavior. In this section, we first linearize the nominal system to reduce the complexity of solving the OCP. Then, based on the linear system, we derive the maximum state deviation between the next real state and the predictive state, to derive a prediction disturbed set. Following it up, we employ this prediction set as the initial condition of the next OCP, which can be solved by the framework of tube-based MPC. Moreover, we delve into the theoretical foundations, analyzing recursive feasibility and closed-loop stability within the framework of tube MPC, which validates the soundness and efficacy of our proposed method.

3.1 Prediction of real system states

To obtain high computational efficiency and fast response times for solving the OCP, we first investigate the linearization for the nonlinear system model (1). Within each control interval $t \in [t_k, t_{k+1}]$, we invoke the second-order Taylor polynomial expansion to the system model $f(x(t), u(t))$ as follows:

$$\begin{aligned} f(x(t),u(t)) &= f(x(t_k),u(t_k)) + A_{t_k}\big[x(t)-x(t_k)\big] + B_{t_k}\big[u(t)-u(t_k)\big] + R(x(t_k),u(t_k)) \\ &= A_{t_k}x(t) + B_{t_k}u(t) + \Omega + R(x(t_k),u(t_k)), \end{aligned} \tag{13}$$

where A_{t_k} and B_{t_k} denote the system matrices of the linear system, while Ω represents the extension error at the point $(x(t_k), u(t_k))$, defined as

$$\Omega = f(x(t_k),u(t_k)) - \left(A_{t_k}x(t_k) + B_{t_k}u(t_k)\right). \tag{14}$$

We introduce $R(x(t_k), u(t_k))$, the Lagrange remainder of the linearization error, expressed as

$$\begin{aligned} R(x(t_k),u(t_k)) = {} & \nabla^2 f[(x(t)-z(t_k)),(u(t)-u(t_k))] \\ & \cdot f[z(t_k)+\theta(z(t)-z(t_k)),u(t_k)+\theta(u(t)-u(t_k))], \end{aligned} \tag{15}$$

where $\nabla^2 f$ denotes the Hessian matrix of the pair $(x(t_k), u(t_k))$, $\theta \in (0, 1)$ is a constant, and $t \in [t_k, t_{k+1}]$. Notably, due to state and input constraints, the parameter θ can be adjusted within the piece-wise interval $[x(t_k), x(t_{k+1})] \times [u(t_k), u(t_{k+1})]$ to satisfy $R(x(t_k), u(t_k))$. By invoking the mean value theorem, we derive the bound of the Hessian matrix $H(x, u)$ for each control interval. The ensuing lemma characterizes the linearization error between the nonlinear and linear systems.

Lemma 2. Within each discrete control interval $t \in [t_k, t_{k+1}]$, the linearization error can be bounded as follows:

$$\|\Omega + R(x(t_k),u(t_k))\| \leqslant \|\Omega\| + \eta_R\Big[l_1\|x(t)-x(t_k)\| + l_2\|u(t)-u(t_k)\|\Big] \triangleq \eta_2. \tag{16}$$

Proof. Utilizing (14), Ω retains a constant value at each triggering instant. Considering the bounded nature of the Hessian matrix $\nabla^2 f(x(t), u(t))$ by η_2, and the fulfillment of the system's Lipschitz continuous property, we can take a substitution within (15) to obtain the linearization error as stipulated in (16).

This lemma effectively constrains the linearization error, facilitating tighter control of the system dynamics and providing the foundation for robustness and performance enhancement. Upon revisiting the system (1), we augment the linearization error to the existing disturbances, thereby constituting the total disturbance w_t. This total disturbance for the linearized system model can be bounded as

$$\|w_t\| = \|e(t) + \Omega + R(x(t),u(t))\| \leqslant \eta_1 + \|\Omega\| + \|R(x(t),u(t))\| \triangleq \eta. \tag{17}$$

This theoretical result enables us to represent the perturbed nonlinear system as an equivalent linear system with an extended bounded disturbance.

Besides, applying the optimal control sequence $u^*(t_k)$ to the nominal system would yield an optimal state trajectory $x^*(t_k) = \{x^*(t_k|t_k), x^*(t_k + \Delta t|t_k), \ldots, x^*(t_k + T|t_k)\}$. However, deviations between the real system and the nominal system emerge. Thus, the ensuing lemma employs the Gronwall-Bellman inequality to estimate the deviation bound, thereby offering insight into predicting the region of the upcoming real state $x(t_{k+1})$ at the present moment t_k.

Lemma 3 (Difference between real system and nominal system)**.** Let Δt be the sampling interval. With the same control input $u^*(t_k)$, the state deviation from t_k to $t_k + m\Delta t$ is bounded by

$$\|x_e(t_k + m\Delta t)\| \leqslant m\eta(l_1 + 1)^m, \quad m \in \mathbb{N}_{[0,T/\Delta t-1]}, \tag{18}$$

where η signifies the total disturbances, and $x_e(t_k + m\Delta t) \triangleq x(t_k + m\Delta t) - x^*(t_k + m\Delta t|t_k)$ denotes the error between the nominal system and the actual disturbed system.

Proof. Let us begin by introducing the function $g(x, u) = f(x, u) - x$, yielding

$$\|g(x_1, u) - g(x_2, u)\| \leqslant (l_1 + 1)\|x_1 - x_2\|. \tag{19}$$

For the real system states, we observe

$$x(t_k + (i+1)\Delta t) - x(t_k + i\Delta t) = g(x(t_k + i\Delta t), u^*(t_k + i\Delta t|t_k)) + w(t_k + i\Delta t). \tag{20}$$

Applying this recursion formula from $i = 0$ to $i = m - 1$, we deduce

$$x(t_k + m\Delta t) = x(t_k) + \sum_{i=0}^{m-1} g(x(t_k + i\Delta t), u^*(t_k + i\Delta t|t_k)) + w(t_k + i\Delta t). \tag{21}$$

Similarly, the nominal system with the optimal control input follows the same formulation:

$$x^*(t_k + m\Delta t|t_k) = x^*(t_k|t_k) + \sum_{i=0}^{m-1} g(x^*(t_k + i\Delta t|t_k), u^*(t_k + i\Delta t|t_k)). \tag{22}$$

Thus, the deviation of $x(t_k + m\Delta t)$ and $x^*(t_k + m\Delta t|t_k)$ can be bounded as

$$\begin{aligned}
&\|x(t_k + m\Delta t) - x^*(t_k + m\Delta t|t_k)\| \\
&= \left\| \sum_{i=0}^{m-1} \left[g(x(t_k + i\Delta t), u^*(t_k + i\Delta t|t_k)) - g(x^*(t_k + i\Delta t|t_k), u^*(t_k + i\Delta t|t_k)) \right] + \sum_{i=0}^{m-1} w(t_k + i\Delta t) \right\| \\
&\leqslant m\eta + (l_1 + 1) \sum_{i=0}^{m-1} \|x(t_k + i\Delta t) - x^*(t_k + i\Delta t|t_k)\|.
\end{aligned} \tag{23}$$

By applying Gronwall-Bellman inequality, it holds that

$$\|x(t_k + m\Delta t) - x^*(t_k + m\Delta t|t_k)\| \leqslant m\eta + \sum_{i=0}^{m-1} (l_1 + 1)m\eta \prod_{j=i+1}^{m-1} (l_1 + 1) = m\eta(l_1 + 1)^m. \tag{24}$$

Remark 2. Compared with previous work [41] where the difference is formulated as

$$\|x_e(t_k + m\Delta t)\| \leqslant \frac{\bar{\lambda}(A)^m - 1}{\bar{\lambda}(A) - 1}\eta, \quad m \in \mathbb{N}_{[0,T/\Delta t-1]}, \tag{25}$$

our results consider the Lipschitz constant to replace the local linearation with system matrices (A, B) used in [41], which provides a compact estimation of the state difference, and thus reduces the performance degradation.

Based on this bound, in the practical control process, we assume that the upper bound of computational time for solving the OCP is $m\Delta t$, where $m \in \mathbb{N}$ is a constant. By the triangle inequality, we can predict the future real system states within a disturbed set as

$$\|x(t_k + m\Delta t)\| \leqslant \|x^*(t_k + m\Delta t|t_k)\| + m\eta(1 + l_1)^m. \tag{26}$$

Utilizing the concept of a disturbed invariant set, we can express (26) as

$$\|x(t_k + m\Delta t)\| \in \mathbb{X}_\omega(x^*(t_k), m), \tag{27}$$

where $\mathbb{X}_\omega(x^*(t_k), m)$ denotes the predictive disturbed state set. Given nominal states $x^*(t_k)$, this set encompasses the future optimal system states $x^*(t_k + m\Delta t)$ along with a radius of $m\eta(1 + l_1)^m$. The aim is to ensure that the real states $x(t_k + m)$ remain within the predictive disturbed state set $\mathbb{X}_\omega(x^*(t_k), m)$.

3.2 Ahead optimal control for smoothness

In the previous subsection, we employed linearization to reduce optimization complexity and introduced the concept of a predictive disturbed state set. Building on this predictive state set, we now propose a strategy for achieving smooth control input by formulating an ahead optimal control problem.

As discussed before, assuming that the maximum computation time for the OCP 1 is m times the system sampling interval Δt, we define the time to the next sampling instant as

$$t_{k+1} - t_k = m\Delta t, \quad m \in \mathbb{N}_{\geqslant 1}, \quad m \leqslant T/\Delta t, \tag{28}$$

where m is a constant determined through trial computation. Thus, for each control interval $t \in [t_k, t_{k+1}]$, the optimal control $u^*(t_k)$ can be applied from $u^*(t_k|t_k)$ to $u^*(t_k + m\Delta t|t_k)$. Leveraging Lemma 2, we incorporate the linearization error into the system dynamics. Consequently, the linear system model within the interval $t \in [t_k, t_{k+1}]$ takes the form:

$$\dot{x}(t) = A_{t_k} x(t) + B_{t_k} u(t) + w_t. \tag{29}$$

Before we build this optimization problem, we first introduce tube-MPC. Tube-based MPC is a branch of MPC, where the controller forces the system state to stay within a tube around a sensible central trajectory [42, 45, 46]. In this framework, we determine the central state trajectory by solving an optimal control problem; then we can employ a feedback controller to keep the real state trajectory approach to the central one, similar to a tube with a central line. We proceed to design a controller, which combines a linear feedback term and a nominal optimal input term:

$$u(t|t_k) = v(t|t_k) + K\left[x(t) - x^*(t|t_k)\right], \tag{30}$$

where $v(t_k)$ represents the control decision variable for the linear system, and K denotes the state feedback gain computed using the Riccati equation (12), accounting for the presence of bounded disturbances. With the prediction set as the initial condition, we can regard it as the initial range of the tube to construct the optimal control problem and guarantee robustness. Building upon the linearization of the nonlinear systems and the predictive disturbed state set, we introduce the concept of the ahead optimal control problem 2 (OCP 2). This problem is formulated as follows:

$$v^*(t_{k+1}) = \min_{\bar{v}(t_{k+1}) \in \mathbb{U}} J(\bar{x}(t_{k+1}), \bar{v}(t_{k+1}), t_{k+1}|t_k), \tag{31}$$

subject to

$$x(t_{k+1}|t_k) \in \mathbb{X}_\omega(x^*(t_k), m), \tag{32a}$$
$$\dot{\bar{x}}(t|t_{k+2}) = A_{t_{k+1}} \bar{x}(t|t_{k+1}) + B_{t_{k+1}} \bar{u}(t|t_{k+1}), \tag{32b}$$
$$\bar{u}(t|t_{k+1}) = \bar{v}(t|t_{k+1}) + K[x^*(t|t_{k+1}) - \bar{x}(t|t_{k+1})], \tag{32c}$$
$$\bar{x}(t|t_{k+1}) \in \mathbb{X} \ominus \mathbb{X}_e(t), \quad \bar{u}(t|t_{k+1}) \in \mathbb{U} \ominus K\bar{x}(t|t_{k+1}), \tag{32d}$$
$$\bar{x}(t_{k+1} + T|t_{k+1}) \in \mathbb{X}_\epsilon, \quad t \in [t_{k+1}, t_{k+1} + T]. \tag{32e}$$

In these equations, $J(\bar{x}(t_{k+1}), \bar{v}(t_{k+1}), t_{k+1}|t_k)$, $\mathbb{X}_e(t)$, and $\mathbb{X}_\epsilon$ retain their definitions from OCP 1. Solving OCP 2 yields the optimal control sequence $v^*(t_k)$, with the initial m elements of this sequence applied to the system. At each sampling instant, when observing the system state, we can compute the predictive disturbed state set to predict the upcoming state, simultaneously applying the last optimal control input to the system. Then, when yielding the set, we can solve the OCP 2 to obtain the optimal control input for the next real state, which forms a "cross-cutting pace" between state sampling and control input computation. By iterating this process, the system achieves effective stabilization in practice. Leveraging predictive states and ahead optimization, we summarize our algorithm in Algorithm 2.

3.3 Theoretical analysis

Despite the advancements introduced by the predictive state and ahead optimization, the repeated control process, utilization of multi-step optimal control sequences, and the presence of unknown disturbances can introduce challenges to the feasibility and stability of the proposed control strategy. To ensure the successful implementation of robust tube-based smooth MPC, we establish the following theorem.

Algorithm 2 Robust tube-based smooth MPC

Require: Initialize the parameters m, l of system (1) and set the weight matrices Q and R. By (12), compute the terminal state feedback gain K and the weight matrix P. Then, define the terminal set to satisfy (32e). Find the optimal solution $v^*(t_0)$ by OCP 2 with the initial state $x(t_0)$.

$k = 0$;
while the system is not converged **do**
  Sample the current state $x(t_k)$;
  \\In a piece-wise control interval.
  for $t = t_k + \delta, \ldots, t_k + m\delta$ **do**
    if $x(t_k) \notin \mathbb{X}_\epsilon$ **then**
      Apply $u^*(t|t_{k-1})$ by (30) to the real system;
      \\Compute the prediction disturbance state set.
      Predict the successive state $x^*(t_k|t_{k-1})$ by the nominal linear model (29);
      \\Compute the optimal central input.
      Compute $v^*(t_k)$ by the OCP 2 and store it until the next sampling instant;
    else
      Apply $u(t_k) = Kx(t_k)$ as the system controller;
    end if
  end for
  Let $k = k + 1$;
end while

Theorem 1. For the nonlinear systems governed by (1), and under the assumptions outlined in Assumptions 1 and 2, the robust tube-based smooth MPC guarantees both feasibility and stability if the following conditions are met:

(i) The local state feedback gain K_{t_k} adheres to the inequality

$$A_{t_k} + B_{t_k} K_{t_k} \preceq 0. \tag{33}$$

(ii) The upper bound η of the total disturbances and the computational time interval m satisfy

$$m\eta(1 + l_1)^m \leqslant \sum_{i=0}^{\infty} \|(A + BK)^i\|\eta. \tag{34}$$

Proof. To establish the feasibility and stability of the proposed robust tube-based smooth MPC, we address three fundamental aspects of the control strategy as follows:

(i) Multi-step control sequence utilization. Drawing from the foundational principles of tube MPC for constrained linear systems with disturbances [47], we find that the feasibility and stability inherent to time-triggered strategies can apply to the proposed framework. Hence, we extend this insight to our smooth MPC.

(ii) Initial condition adaptation for the OCP. As discussed earlier, the shift in initial conditions in each iteration and the predictive disturbed state set align well with the principles of tube-based MPC, preserving its feasibility and stability.

(iii) System linearization. Similarly, we apply the concept of disturbance invariant sets $\mathbb{X}_\gamma$ to the proposed smooth MPC. The parameter γ is defined as

$$\gamma = \sum_{i=0}^{\infty} \|(A + BK)^i\|\eta. \tag{35}$$

When $\mathbb{X}_\omega(x^*(t_k), m) \subset \mathbb{X}_\gamma$ is satisfied, the foundation for the tube MPC framework remains robust.

The combination of these aspects ensures recursive feasibility. The fulfillment of the multi-step control usage condition

$$m\eta(1 + l_1)^m \leqslant \sum_{i=0}^{\infty} \|(A + BK)^i\|\eta \tag{36}$$

further guarantees that the state constraint remains valid:

$$\|x(t_k + m)\| \in \|x^*(t_k + m|t_k)\| \oplus \mathbb{X}_\omega(m) \subset \|x^*(t_k + m\Delta t|t_k)\| \oplus \mathbb{X}\gamma \subset \mathbb{X} \ominus \mathbb{X}_e(m). \tag{37}$$

Thus, the conditions for recursive feasibility are proven.

Table 1 Comparison of smooth MPC with time-triggered MPC

	Item	Time-triggered MPC	Smooth MPC (ours)
Similarity	Objective function	Quadratic cost	
	Robustness	Tightened state constraint set	
Difference	System model	Nonlinear system $f(x,u)$	Linear system $Ax+Bu$
	Initial condition	Real state $x(t_k)$	Prediction Set $x(t_{k+1}\|t_k)\in\mathbb{X}_\omega(x^*(t_k),m)$
	Control input	$u^*(t\|t_k), t\in[t_k,t_k+\Delta t]$	$u^*(t\|t_k), t\in[t_k,t_k+m\Delta t]$
	Computational delay	$m\Delta t$	No obvious delay
	Time cost	Long	Short
	Performance	Oscillating	Stable

Finally, with the state feedback gain K_{t_k} satisfying

$$A_{t_k}+B_{t_k}K_{t_k}\preceq 0, \tag{38}$$

we ensure that the real system states are encompassed within the disturbance invariant set of the nominal states. By leveraging insights from conventional tube MPC analysis [42,45], we conclude that the practical stability of our smooth MPC approach is assured.

Further, Table 1 gives a comparison between our proposed method and the conventional time-triggered MPC. Based on the comparison, our smooth MPC adopts linearized system dynamics and the prediction set as the constraint when solving the optimal control problem at each control interval, which can really help the response speed and yield better control performance.

4 Experiment results

In this section, we present a comprehensive series of experiments, combining numerical simulations and practical implementation on a physical robot platform. The objective of these experiments is to provide empirical evidence of the effectiveness of our proposed algorithm, showcasing its superiority in terms of control performance and responsiveness compared to established MPC controllers.

Our evaluation includes a comparative analysis of the following control strategies:

• **Ideal MPC.** This controller represents the theoretical best-case scenario, exhibiting optimal control performance with no computation delay at each sampling instant.

• **Time-triggered MPC.** As a commonly employed control strategy in real systems, this controller introduces the worst-case delay for each control period, reflecting real-world scenarios.

• **Robust tube-based smooth MPC.** Our novel approach utilizes piece-wise linearization and state prediction for enhanced control performance and smoothness.

Numerical simulation experiments are conducted using a three-axis robot arm, focusing on two typical manipulation tasks: reaching a specified point and tracking a composite trajectory. These simulations aim to demonstrate the algorithm's effectiveness across various scenarios and control objectives. Moreover, we extend our evaluation to a physical robot manipulator, implementing the same tasks in a real-world setting. This practical experimentation further reinforces the algorithm's viability and applicability to real-world robotic systems. The combination of numerical simulations and physical experiments provides a comprehensive validation of the proposed algorithm's capabilities.

4.1 Numerical simulations

4.1.1 *Experimental setting*

We begin our investigation with a three-joint robot manipulator operating within the x-y plane. The manipulator features three revolute joints and employs three angular velocity controls for plane motion.

The kinematic dynamics of the manipulator's endpoint is described by the following system equations:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta_1} \\ \dot{\theta_2} \\ \dot{\theta_3} \end{bmatrix} = \begin{bmatrix} -L_1\sin(\theta_1) & -L_2\sin(\theta_2) & -L_3\sin(\theta_3) \\ L_1\cos(\theta_1) & L_2\cos(\theta_2) & L_3\cos(\theta_3) \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \end{bmatrix}, \tag{39}$$

where $z(t) \triangleq (p(t), \theta(t))$ represents the system states, and $u(t) \triangleq (\omega_1(t), \omega_2(t), \omega_3(t))$ denotes the control input. The coordinates of the endpoint are denoted by $p(t) = (x(t), y(t))$, and $\theta(t) = (\theta_1(t), \theta_2(t), \theta_3(t))$ signifies the joint angles. The lengths of the manipulator's three links are represented by L_1, L_2, and L_3, respectively. Angular velocities are captured as $(\omega_1(t), \omega_2(t), \omega_3(t))$.

Incorporated into this nonlinear system is an additional disturbance $e(t) = (e_1(t), \ldots, e_5(t)) \in \mathbb{R}^5$, bounded by $\|e(t)\| \leqslant \eta_1$, where $\eta_1 > 0$ is a known constant. Furthermore, the real control process operates within mechanical limitations and is subject to strict constraints on both control inputs and states:

$$z(t) \in \mathbb{Z} \subseteq \mathbb{R}^5 \triangleq \{\theta_i : \underline{\theta_i} \leqslant \|\theta_i\| \leqslant \overline{\theta_i}\}, \tag{40}$$

$$u(t) \in \mathbb{U} \subseteq \mathbb{R}^3 \triangleq \{\omega_i : \|\omega_i\| \leqslant \overline{\omega_i}\}, \tag{41}$$

where $\underline{\theta_i}$, $\overline{\theta_i}$, and $\overline{\omega_i}$ are three known positive constants.

Firstly, we compute the linear system matrices at each sampling instant according to Assumption 1,

$$A = \left[\begin{array}{c|cc} \mathbf{0}_{2\times 2} & -\omega_1 L_1\cos\theta_1 & -\omega_2 L_2\cos\theta_2 & -\omega_3 L_3\cos\theta_3 \\ & -\omega_1 L_1\sin\theta_1 & -\omega_2 L_2\sin\theta_2 & -\omega_3 L_3\sin\theta_3 \\ \hline \mathbf{0}_{3\times 2} & \multicolumn{3}{c}{\mathbf{0}_{3\times 3}} \end{array}\right], B = \left[\begin{array}{ccc} -L_1\sin\theta_1 & -L_2\sin\theta_2 & -L_3\sin\theta_3 \\ L_1\cos\theta_1 & L_2\cos\theta_2 & L_3\cos\theta_3 \\ \hline \multicolumn{3}{c}{\boldsymbol{I}_{3\times 3}} \end{array}\right], \tag{42}$$

and then verify the Lipschitz continuity by Assumption 2, where the constants $l_1 = \max\{L_1, L_2, L_3\}$ and $l_2 = 1$. The specific system parameters and initial conditions for our experiments are chosen as follows:

$$\begin{cases} L_1 = L_2 = \sqrt{5},\ L_3 = \sqrt{10}, \\ \theta_1 = \dfrac{\pi}{2} + \arcsin\dfrac{2}{\sqrt{5}},\ \theta_2 = \dfrac{\pi}{2} + \arcsin\dfrac{1}{\sqrt{5}}, \\ \theta_3 = \arcsin\dfrac{1}{\sqrt{10}}, \\ p = (-1, 4). \end{cases} \tag{43}$$

In this system, the joint pose is defined as the angle from the positive x-axis to the finger phalanx in an anti-clockwise direction. The angular velocity of the finger phalanx is considered positive for anti-clockwise rotation and negative for clockwise rotation. The constraints for both states and inputs are defined as

$$\begin{cases} \dfrac{\pi}{2} \leqslant \theta_1 \leqslant \pi, 0 \leqslant \theta_2 \leqslant \pi, 0 \leqslant \theta_3 \leqslant \dfrac{\pi}{2}, \\ -\dfrac{\pi}{16} \leqslant \omega_1, \omega_2, \omega_3 \leqslant \dfrac{\pi}{16}. \end{cases} \tag{44}$$

Finally, we verify the bounded Hessian matrix as $\det(H(x(t), u(t))) \leqslant \max\{L_1, L_2, L_3\}$ and set the upper bound of the total disturbances as $\eta = 0.2$, following the conditions laid out in Theorem 1.

In the position reaching task, we set the expected state as $x^{\mathrm{T}} = (2, 6, \text{-}, \text{-}, \text{-})$, where "-" means the free dimension. This task requires the endpoint of the robot manipulator to reach the point simultaneously minimizing the cost function and keeping the constraints hold. In the trajectory tracking task, we design a reference trajectory as

$$\begin{cases} x^{\mathrm{T}}(t) = 2 + 2\cos\left(-\pi - \dfrac{\pi}{4.8}t\right), & t \in [0, 2.4], \\ y^{\mathrm{T}}(t) = 4 + 2\sin\left(-\pi - \dfrac{\pi}{4.8}t\right), & t \in [0, 2.4], \\ x^{\mathrm{T}}(t) = 2 + (t - 2.4)/1.6, & t \in [2.4, 4], \\ y^{\mathrm{T}}(t) = 6 + (t - 2.4)/1.6, & t \in [2.4, 4], \end{cases} \tag{45}$$

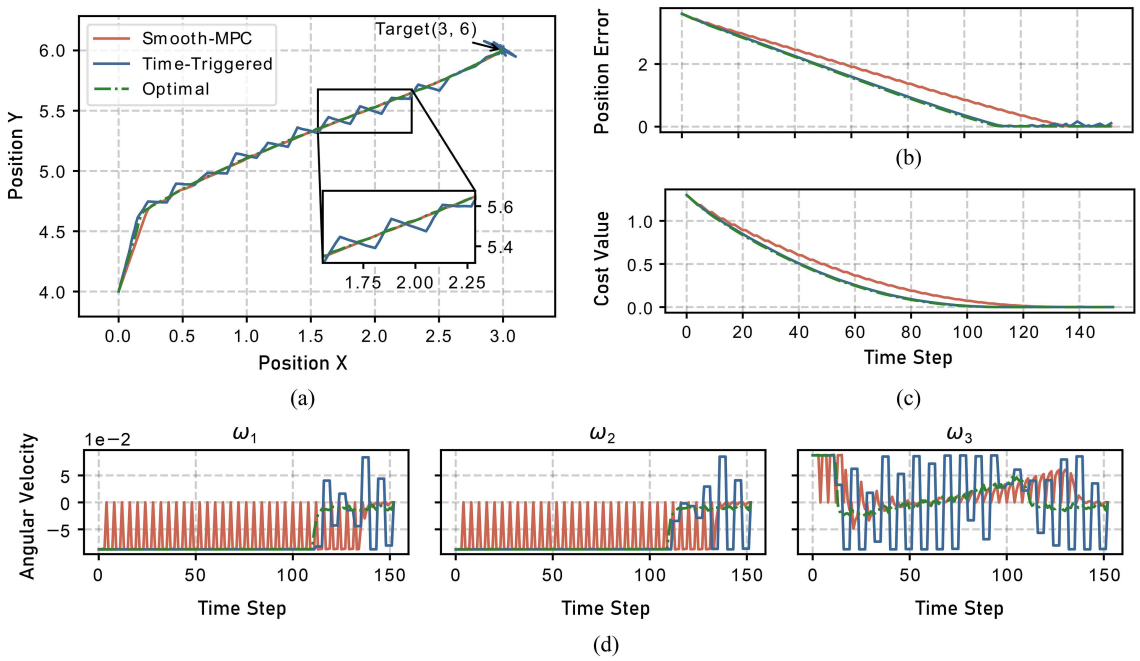

Figure 2 (Color online) Comparison of three control methods over the position tracking task, which contains the endpoint trajectory, position error, cost function value, and control input. Specifically, the position error measures the error between the current endpoint of the robot manipulator and the target point, i.e., $d = \sqrt{(x(t) - x^{\mathrm{T}}(t))^2 + (y(t) - y^{\mathrm{T}}(t))^2}$. The cost value demonstrates the various costs in the whole control process, i.e., $J = \sum_{t=0}^{T} \|x(t) - x^{\mathrm{T}}(t)\|_Q^2 + \|u(t)\|_R^2$. (a) State trajectory; (b) position error; (c) cost value in OCP; (d) angular velocity of three joints.

to form a combination of a quarter-circle and a straight line. In this task, the endpoint of the robot manipulator needs to track this reference trajectory with the minimization of cost function and the guarantee of system constraints.

For all MPC controllers in our experiments, we employ a common sampling time $\delta = 0.1$ s along with a prediction horizon of $T = 3$ s. To shape the controller's behavior, we define the weight matrices as $Q = 0.1\boldsymbol{I}^{5\times5}$ and $R = 0.01\boldsymbol{I}^{3\times3}$, where $\boldsymbol{I}$ represents the identity matrix. Based on numerous trial runs, we ascertain that the computation time for solving the nonlinear OCP 1 is consistently recorded as $m = 4\delta = 0.4$ s, while the computation time for solving the linear OCP 2 is $\Delta t = 0.05$ s. The simulation examples are implemented using the YALMIP toolbox on Matlab 2020a with intel® core™ i9-9900 CPU.

4.1.2 *Experimental results*

Position reaching. In this task, we evaluate the performance of the proposed algorithms in a position-reaching task, where the manipulator is tasked with smoothly moving an object from the initial position $(0, 4)$ to the target position $(2, 6)$. We compare the outcomes of the optimal, time-triggered, and smooth MPC controllers for this task.

As depicted in Figure 2, the trajectories of the endpoints are shown for each controller. Figure 2 reveals that our method can yield smoother endpoint trajectories than the time-triggered one, approaching the optimal case. Besides, even after reaching the target, due to the presence of external disturbances, our method has better robustness with smaller oscillations of endpoints than the time-triggered one.

Trajectory tracking. In this task, the endpoint tracks a specific trajectory with multiple splices, consisting of a quarter circle and an oblique line. We use this task to emulate complex manipulation processes with given spline curves. From Figure 3, our approach shows better tracking performance and robustness than the time-triggered one. Specifically, the position errors derived by our methods are significantly lower than the time-triggered ones. And, our control inputs have a relatively low amplitude, which can mitigate the effects of input jumps.

Comparing both tasks, it is evident that the proposed method outperforms the time-triggered MPC in achieving smooth movement and better tracking precision for the robot manipulator's endpoint. Despite potential oscillations in the three joints due to system linearization, their magnitudes are smaller compared to the time-triggered approach, particularly noticeable in the third joint, indicating superior

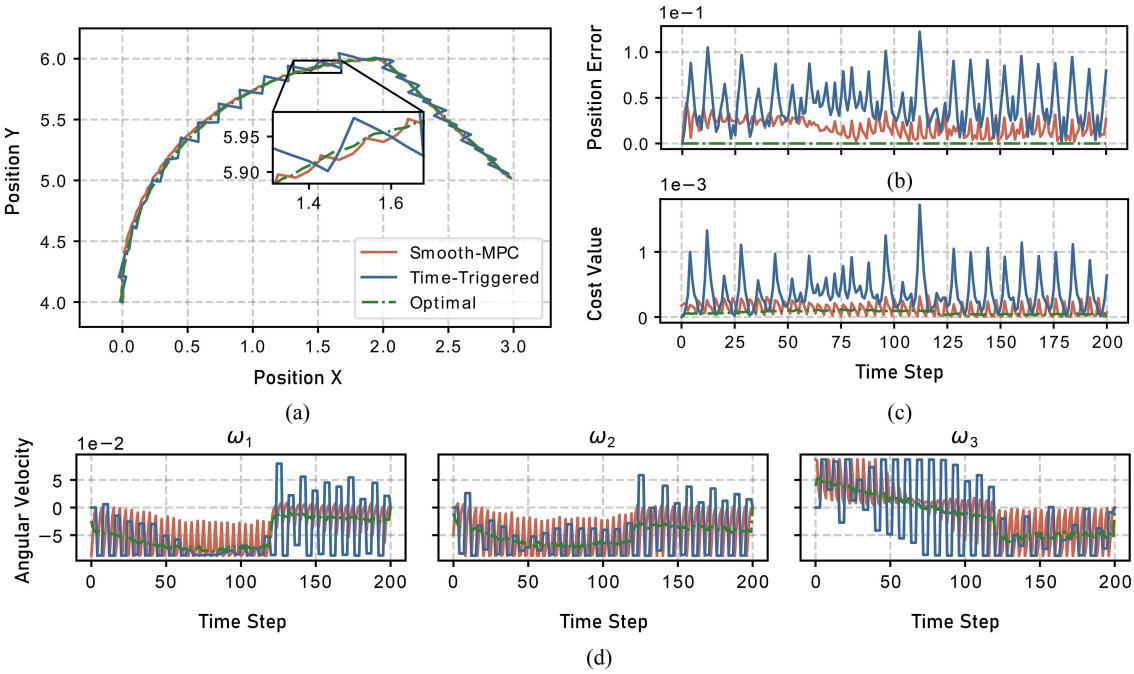

Figure 3 (Color online) Comparison of three control methods over the trajectory tracking task, which contains the endpoint trajectory, position error, cost value, and control input, referring to the definition in Figure 2. (a) State trajectory; (b) position error; (c) cost value in OCP; (d) angular velocity of three joints.

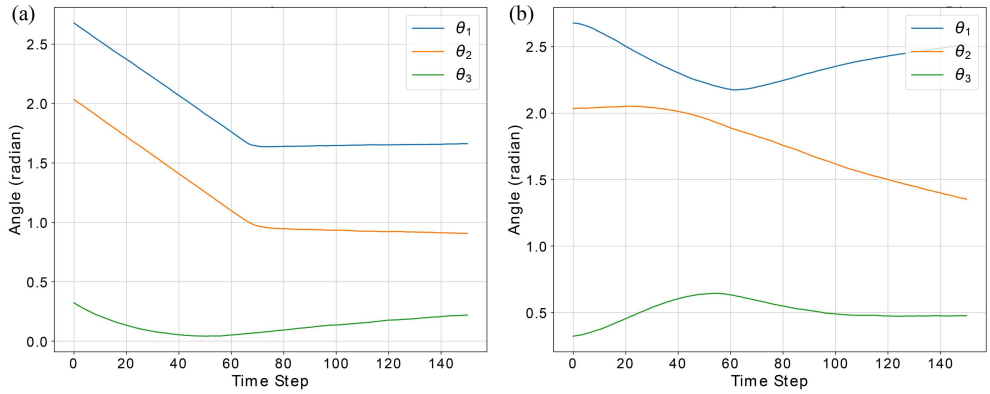

Figure 4 (Color online) Variance of joint angles of the whole control process in the two tasks by smooth MPC. (a) Target reaching task; (b) trajectory tracking task.

Table 2 Summary of comparison for computation time

	Solving time for single OCP (s)	Whole control process (s)	Percentage (%)
Time-triggered MPC	2.8	840	100
Optimal MPC w/o delay	2.5	750	89
Ours	0.3	90	12

smoothness control. Notably, despite the linear error, Lemma 3 establishes the prediction set boundary, and Theorem 1 ensures system stability, guaranteeing controllability. Furthermore, our method prioritizes smooth control across the entire state space, rather than focusing on specific dimensions, ensuring smoothness across all manipulator joint movements, as illustrated in Figure 4.

Response time. To illustrate the real-time performance of our algorithm, we report the computation time of solving the OCP once and the whole control process under three methods, as shown in Table 2. The results show that our proposed method can significantly reduce the solution time, ensuring that the OCP can be solved within two consecutive control cycles, allowing the next control to be seamlessly input. Thus, our proposed method demonstrates efficient computational performance while guaranteeing

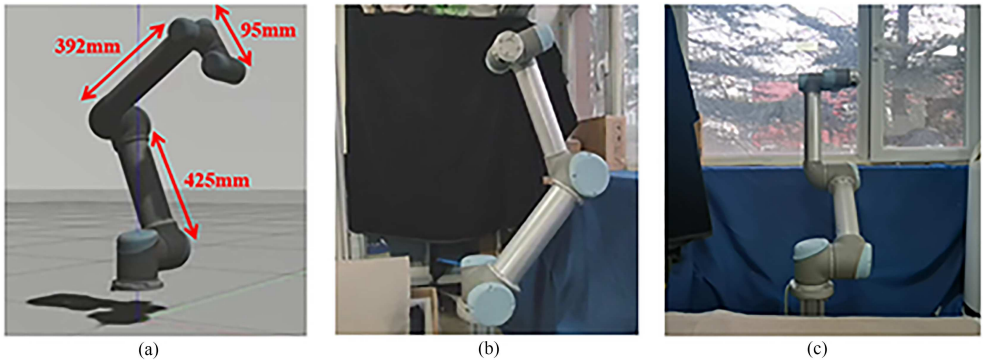

Figure 5 (Color online) (a) Model of the manipulator in the ROS simulator along with its geometries; (b) lateral and (c) frontal views of the manipulator in the real environment.

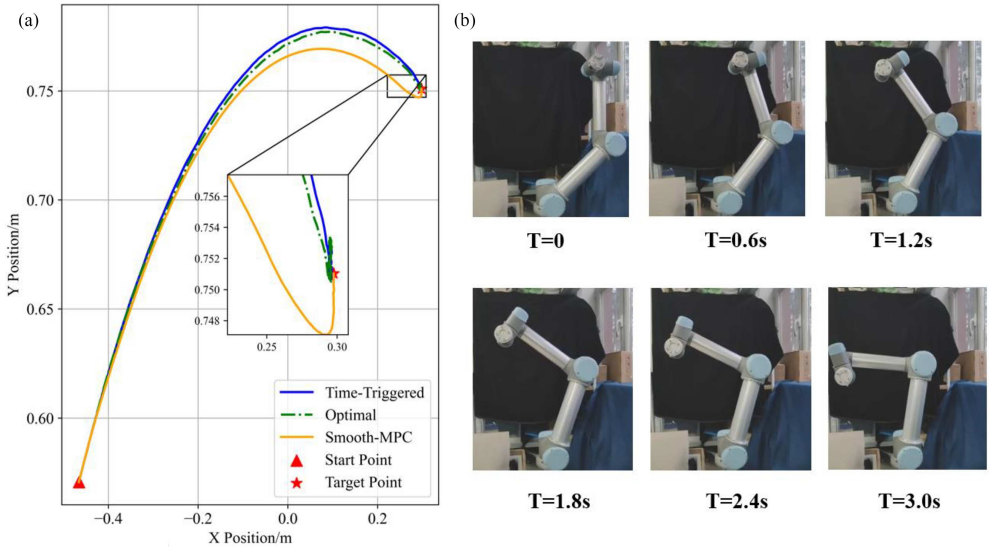

Figure 6 (Color online) Trajectories of the endpoint in the position reaching task with different methods and visualization of the scenario. (a) Trajectories of the endpoint by different methods; (b) sequential lateral perspectives of the manipulator.

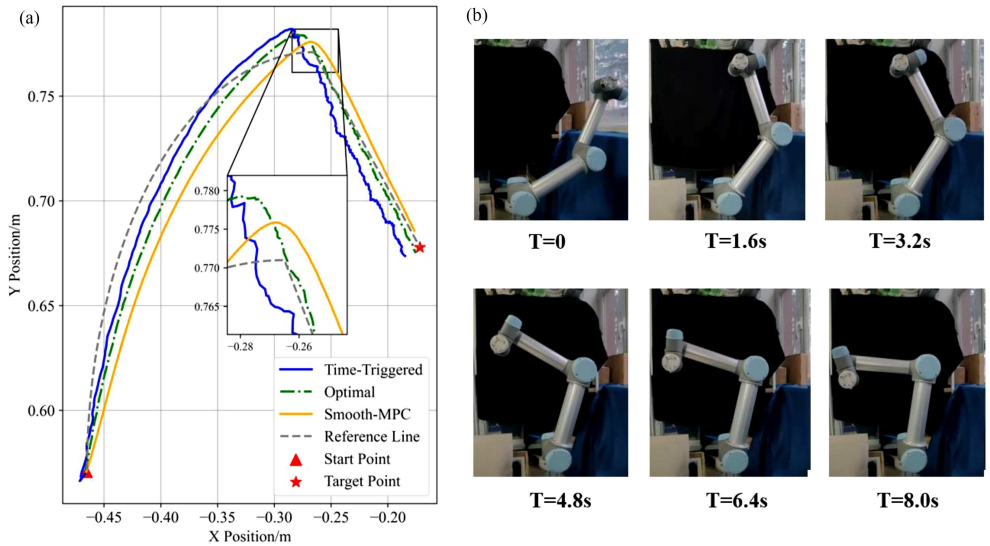

Figure 7 (Color online) Trajectories of the endpoint in the trajectory tracking task with different methods and visualization of the scenario. (a) Trajectories of the endpoint by different methods; (b) sequential lateral perspectives of the manipulator.

manipulation accuracy and control performance.

4.2 Physical experiments

4.2.1 *Experimental setting*

In recent years, the community has witnessed several MPC models transfer onto real robot platforms for validation [36, 48], predominantly focusing on manipulator scenarios. In this study, we employ a UR5 6-DOF manipulator as our physical platform, depicted in Figure 5. To maintain consistency with the three-joint manipulator setting in numerical simulations, we lock the second, fourth, and sixth joints from the base during experiments. Angle velocity control is utilized to optimize the position of the manipulator endpoint.

4.2.2 *Experimental results*

Similar to numerical simulations, we compare our proposed control strategy with three baseline MPC methods in a real-world setting shown in Figures 6 and 7. In every test episode, the manipulator conducts a series of steps to achieve the target position with a joint angel velocity combination computed by different MPC methods. The optimized trajectories are presented as experimental results to compare the performance of different methods. During the validation, we record the signal of every joint of our manipulator platform and transform the collected joint data into the trajectory of the endpoint. The results of the task moving to the target position are shown in Figure 6. Our proposed method outperforms the baselines in both aspects of convergence and robustness.

It is worth noting that, in the above experimental results, the "Optimal" method is merely the theoretically calculated optimal solution. In the physical experiments presented in this paper, the "Optimal" method is implemented by dynamically computing the next move at each moment based on the current motion state. Besides, we also conduct the trajectory following task in a physical setting; the results are shown in Figure 7. In the trajectory tracking task, our proposed smooth MPC method demonstrates the capability to follow the predefined trajectory with relatively reduced oscillations accurately.

5 Conclusion

This paper has presented a robust tube-based smooth MPC approach designed to address the computational delay issues in dexterous robot manipulation with external disturbance. This method effectively tackles computational delays by estimating the linearization error as a bounded disturbance and utilizing a predictive disturbed state set through Lipschitz continuity. The results demonstrate enhanced control performance and response speed compared to conventional MPC methods through two fundamental manipulation tasks, both in simulation and real-world scenarios. Since the smooth computation is a basic mechanism for dealing with the delay of solving OCP, our future work can combine this mechanism with event-triggered mechanism and self-triggered mechanism [49]. Therefore, while reducing the solving frequency, it alleviates the delay problem during single solving and promotes the application of MPC in real-world systems.

Acknowledgements This work was supported by "New Generation Artificial Intelligence" Key Field Research and Development Plan of Guangdong Province (Grant No. 2021B0101410002), National Science and Technology Major Project of the Ministry of Science and Technology of China (Grant No. 2018AAA0102900), and National Key Research and Development Program of China (Grant No. 2021ZD0114505).

References

1 Billard A, Kragic D. Trends and challenges in robot manipulation. Science, 2019, 364: eaat8414
2 Wang D, Pan Q, Shi Y, et al. Efficient nonlinear model predictive control for quadrotor trajectory tracking: algorithms and experiment. IEEE Trans Cybern, 2021, 51: 5057–5068
3 Ding Y, Pandala A, Li C, et al. Representation-free model predictive control for dynamic motions in quadrupeds. IEEE Trans Robot, 2021, 37: 1154–1171
4 Omer M, Ahmed R, Rosman B, et al. Model predictive-actor critic reinforcement learning for dexterous manipulation. In: Proceedings of International Conference on Computer, Control, Electrical, and Electronics Engineering, 2021. 1–6
5 Mayne D Q. Model predictive control: recent developments and future promise. Automatica, 2014, 50: 2967–2986
6 Samad T, Bauer M, Bortoff S, et al. Industry engagement with control research: perspective and messages. Annu Rev Control, 2020, 49: 1–14
7 Vazquez S, Rodriguez J, Rivera M, et al. Model predictive control for power converters and drives: advances and trends. IEEE Trans Ind Electron, 2016, 64: 935–947

8 Brunner M, Bodie K, Kamel M, et al. Trajectory tracking nonlinear model predictive control for an overactuated MAV. In: Proceedings of IEEE International Conference on Robotics and Automation, 2020. 5342–5348

9 Hannigan E, Song B, Khandate G, et al. Automatic snake gait generation using model predictive control. In: Proceedings of IEEE International Conference on Robotics and Automation, 2020. 5101–5107

10 Ozawa R, Tahara K. Grasp and dexterous manipulation of multi-fingered robotic hands: a review from a control view point. Adv Robot, 2017, 31: 1030–1050

11 Mayne D Q, Rawlings J B, Rao C V, et al. Constrained model predictive control: stability and optimality. Automatica, 2000, 36: 789–814

12 Sun Z, Dai L, Xia Y, et al. Event-based model predictive tracking control of nonholonomic systems with coupled input constraint and bounded disturbances. IEEE Trans Automat Contr, 2018, 63: 608–615

13 Köhler J, Müller M A, Allgöwer F. A nonlinear tracking model predictive control scheme for dynamic target signals. Automatica, 2020, 118: 109030

14 Suomalainen M, Karayiannidis Y, Kyrki V. A survey of robot manipulation in contact. Robot Autonom Syst, 2022, 156: 104224

15 Jauhri S, Peters J, Chalvatzaki G. Robot learning of mobile manipulation with reachability behavior priors. IEEE Robot Autom Lett, 2022, 7: 8399–8406

16 Ruehl S W, Hermann A, Xue Z, et al. Graspability: a description of work surfaces for planning of robot manipulation sequences. In: Proceedings of IEEE International Conference on Robotics and Automation, 2011. 496–502

17 He K, Newbury R, Tran T, et al. Visibility maximization controller for robotic manipulation. IEEE Robot Autom Lett, 2022, 7: 8479–8486

18 Visioli A, Legnani G. On the trajectory tracking control of industrial SCARA robot manipulators. IEEE Trans Ind Electron, 2002, 49: 224–232

19 Carron A, Arcari E, Wermelinger M, et al. Data-driven model predictive control for trajectory tracking with a robotic arm. IEEE Robot Autom Lett, 2019, 4: 3758–3765

20 Kim D, Carlo J D, Katz B, et al. Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control. 2019. ArXiv:1909.06586

21 Merry R J E, Kessels D J, Heemels W P M H, et al. Delay-varying repetitive control with application to a walking piezo actuator. Automatica, 2011, 47: 1737–1743

22 Romero A, Sun S, Foehn P, et al. Model predictive contouring control for time-optimal quadrotor flight. IEEE Trans Robot, 2022, 38: 3340–3356

23 Mohammadzadeh A, Ghaemi S. Robust synchronization of uncertain fractional-order chaotic systems with time-varying delay. Nonlinear Dyn, 2018, 93: 1809–1821

24 Schwenzer M, Ay M, Bergs T, et al. Review on model predictive control: an engineering perspective. Int J Adv Manuf Technol, 2021, 117: 1327–1349

25 Rawlings J B, Meadows E S, Muske K R. Nonlinear model predictive control: a tutorial and survey. IFAC Proc Vol, 1994, 27: 185–197

26 Korda M, Mezić I. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. Automatica, 2018, 93: 149–160

27 Rahideh A, Shaheed M H, Huijberts H J. Stable adaptive model predictive control for nonlinear systems. In: Proceedings of American Control Conference, 2008. 1673–1678

28 Magni L, Nicolao G D, Magnani L, et al. A stabilizing model-based predictive control algorithm for nonlinear systems. Automatica, 2001, 37: 1351–1362

29 Rahideh A, Shaheed M H. Stable model predictive control for a nonlinear system. J Franklin Inst, 2011, 348: 1983–2004

30 Griffith D W, Biegler L T, Patwardhan S C. Robustly stable adaptive horizon nonlinear model predictive control. J Process Control, 2018, 70: 109–122

31 Li P, Kang Y, Zhao Y B, et al. Networked dual-mode adaptive horizon MPC for constrained nonlinear systems. IEEE Trans Syst Man Cybern Syst, 2021, 51: 7435–7449

32 Li H, Shi Y. Event-triggered robust model predictive control of continuous-time nonlinear systems. Automatica, 2014, 50: 1507–1513

33 Liu C, Gao J, Li H, et al. Aperiodic robust model predictive control for constrained continuous-time nonlinear systems: an event-triggered approach. IEEE Trans Cybern, 2018, 48: 1397–1405

34 Zhao B, Liu D. Event-triggered decentralized tracking control of modular reconfigurable robots through adaptive dynamic programming. IEEE Trans Ind Electron, 2020, 67: 3054–3064

35 Fei J, Liu L. Real-time nonlinear model predictive control of active power filter using self-feedback recurrent fuzzy neural network estimator. IEEE Trans Ind Electron, 2022, 69: 8366–8376

36 Kleff S, Meduri A, Budhiraja R, et al. High-frequency nonlinear model predictive control of a manipulator. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2021. 7330–7336

37 Zavala V M, Biegler L T. The advanced-step NMPC controller: optimality, stability and robustness. Automatica, 2009, 45:

86–93

38 Su Y, Tan K K, Lee T H. Computation delay compensation for real time implementation of robust model predictive control. J Process Control, 2013, 23: 1342–1349

39 Hu J, Ding B. One-step ahead robust MPC for LPV model with bounded disturbance. Eur J Control, 2020, 52: 59–66

40 Lan J, Zhao D. Robust model predictive control for nonlinear parameter varying systems without computational delay. Intl J Robust Nonlinear, 2021, 31: 8273–8294

41 Luo Y, Sima Q, Ji T, et al. Smooth computation without input delay: robust tube-based model predictive control for robot manipulator planning. In: Proceedings of IEEE International Conference on Robotics and Automation, 2024

42 Sun Z, Dai L, Liu K, et al. Robust MPC for tracking constrained unicycle robots with additive disturbances. Automatica, 2018, 90: 172–184

43 Luo Y, Xia Y, Sun Z. Robust event-triggered model predictive control for constrained linear continuous system. Intl J Robust Nonlinear, 2019, 29: 1216–1229

44 Chen H, Allgöwer F. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. Automatica, 1998, 34: 1205–1217

45 Sieber J, Bennani S, Zeilinger M N. A system level approach to tube-based model predictive control. IEEE Control Syst Lett, 2021, 6: 776–781

46 Zhang X, Pan W, Scattolini R, et al. Robust tube-based model predictive control with Koopman operators. Automatica, 2022, 137: 110114

47 Langson W, Chryssochoos I, Raković S V, et al. Robust model predictive control using tubes. Automatica, 2004, 40: 125–133

48 Bhardwaj M, Sundaralingam B, Mousavian A, et al. Storm: an integrated framework for fast joint-space model-predictive control for reactive manipulation. In: Proceedings of Conference on Robot Learning, 2022. 750–759

49 Heemels W P, Johansson K H, Tabuada P. An introduction to event-triggered and self-triggered control. In: Proceedings of the 51st IEEE Conference on Decision and Control, 2012. 3270–3285