This is a repository copy of *Real-Time Dynamic Obstacle Avoidance for Robot Manipulators Based on Cascaded Nonlinear MPC With Artificial Potential Field*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/205260/

Version: Accepted Version

Real-Time Dynamic Obstacle Avoidance for Robot Manipulators based on Cascaded Nonlinear MPC with Artificial Potential Field

Tianqi Zhu, *Student Member, IEEE*, Jianliang Mao, *Member, IEEE*, Linyan Han,
Chuanlin Zhang, *Senior Member, IEEE*, and Jun Yang, *Fellow, IEEE*

Abstract—**Nowadays, the realization of obstacle avoidance for robot manipulators are generally based on offline path planning, which may be insufficient for real-time dynamic obstacle avoidance scenarios. In order to achieve safe and smooth avoidance of dynamic obstacles, a low-latency motion planning algorithm which takes into account the dynamic planning is of practical significance. Towards this end, this paper proposes a cascaded nonlinear model predictive control (MPC) assigned with a two-stage optimization problem. Specially, the high-level MPC combines artificial potential field (APF) as a motion planner to generate foresight smooth trajectories. The low-level MPC acts as a trajectory tracker and a safety protector, taking along hard constraints to avoid collisions and singularities. In addition, a super-twisting observer (STO) is deployed to enhance the motion estimation accuracy of dynamic obstacles. Experimental results show that the proposed approach is beneficial to achieve safe and smooth dynamic obstacle avoidance in real-world scenarios.**

Index Terms—**dynamic obstacle avoidance, model predictive control, artificial potential field, robot manipulators, super-twisting observer.**

I. INTRODUCTION

With the rapid development of robotic science, robot manipulators are not limited to be applied in structured factory environments, but are becoming more and more popular in people's daily lives, such as medical surgery [1], dressing assistance [2] and playing table tennis [3]. If robots interact with various uncertain environments, one of the key technologies is

Tianqi Zhu and Jianliang Mao are with the College of Automation Engineering, Shanghai University of Electric Power, Shanghai 200090, China, and also with the Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing, 210096, China (e-mail: zhutianqi@mail.shiep.edu.cn; jl_mao@shiep.edu.cn).

Chuanlin Zhang is with the College of Automation Engineering, Shanghai University of Electric Power, Shanghai 200090, China (e-mail: clzhang@shiep.edu.cn).

Linyan Han is with the School of Mechanical Engineering, University of Leeds, Leeds LS2 9JT, UK (e-mail: l.y.han@leeds.ac.uk).

Jun Yang is with the College of Aeronautical and Automotive Engineering, Loughborough University, Loughborough LE11 3TU, UK (e-mail: j.yang3@lboro.ac.uk).

how to achieve dynamic obstacle avoidance in complex task scenarios.

It is well acknowledged that motion planning plays a vital role in achieving obstacle avoidance, which can be roughly divided into the following categories: sampling-based methods [4], grid-based methods [5], artificial potential filed (APF) based methods [6] and numerical optimization methods [7]. In general, the sampling-based methods such as rapidly exploring random tree (RRT) [8] and probabilistic roadmap (PRM) [9] are widely used due to their efficiency and low memory [10]. However, since these methods have certain randomness in path planning, applying them directly to dynamic obstacle avoidance may lead to the motion oscillation of the robot. For grid-based methods like A-star [11], it is difficult to perform real-time planning tasks in dynamic scenarios as the time cost grows exponentially as the dimension of the state space increases. The advantage of using APF methods is that they assign a virtual repulsive potential to each known obstacle and an attractive potential to the desired goal configuration, enabling the robot to navigate towards the goal whilst reactively avoiding obstacles. In addition, APF methods offer flexibility and adaptability by accommodating different shapes and sizes of robots through the adjustment of potential fields from a virtual point-shaped particle model. However, one limitation of this approach is the possibility of getting trapped in a local minima due to inappropriate parameter setup [12], [13]. Moreover, numerical optimization methods usually transform the dynamic constraints of motion planning into an optimal control problem. As a numerical optimization control strategy, model predictive control (MPC) has become one of the most powerful methods for multi-variable control systems since it provides an effective and efficient methodology to handle complex and constrained problems [14], [15]. Therefore, MPC-based dynamic obstacle avoidance algorithms have gradually been widely studied in unmanned aerial vehicles and autonomous ground vehicles in recent years [16], [17].

Although MPC is widely used in motion planning of intelligent systems with uncertain environments, the hard constrained optimization may lead to infeasible solution in practice when the constraints are violated [18]. Additionally, the determination of the prediction horizon is also crucial for the effect of motion planning. Taking the obstacle avoidance for robot manipulators for as an example, a short prediction horizon may lead to a lag in response to obstacles, resulting in

the jerk motion of a robot, while a long prediction horizon may be difficult to guarantee the real-time performance of dynamic obstacle avoidance.

To address these issues, some insightful works design cascaded architectures to address target requirements at different levels, rather than rely on an independent MPC for motion planning and control have been reported [19]–[21]. Specifically, the high-level controller employs reinforcement learning-based intelligent control for legged mobile manipulators with a relatively larger sampling time in [19]. Although this method has achieved success in specific environments, the establishment of the simulated environment and the training of the neural networks take a lot of time. In addition, the design of the reward mechanism also largely depends on human experience. Conversely, the construction of the high-level controller based on MPC is relatively convenient [20], [21]. However, despite the introduction of high-level MPC enlarges the predictive range, its larger discretization time may amplify the difference between the nominal model and the real-world model, resulting in the planned trajectory that possibly violates the prescribed constraints. Therefore, it is not a superior choice to directly set hard constraints on obstacle avoidance at the high level, but it is more inclined to set soft constraints such as adding an alternative penalty function to the cost function.

Additionally, the accurate measurement of dynamic obstacles is also crucial for successful obstacle avoidance. Since the motion estimation of obstacles is introduced to predict a future short-term motion behavior, it is essential to develop an observer/estimator that is able to realize the velocity estimation of obstacles. As a nonlinear state observer, the super-twisting observer (STO) can not only realize fast state estimation within a finite time, but also reduce the chattering problem produced by the traditional non-continuous sliding mode approach [22], which has been extensively explored in servo drives [23], [24] and robotic systems [25].

Motivated by the above analysis, this paper proposes a cascaded nonlinear MPC with APF for real-time dynamic obstacle avoidance tasks. Specially, the high-level MPC is constructed as a motion planner to extend the predictive range and improve the predictive ability on obstacles, whereas the low-level MPC acts as a trajectory tracker and safety protector with features of collision-free and singularity avoidance. In this regard, the proposed motion planning scheme essentially degenerates to a two-stage optimization problem. Furthermore, a STO is deployed to precisely observe the movement of obstacles and the estimates are thereafter incorporated into the optimization framework. Finally, the real-world experiments are conducted on a six-axis robot manipulator to verify the effectiveness of the proposed approach. The main contributions of this paper can be summarized as follows:

1) By associating the APF-based penalty term in the objective function, a cascaded nonlinear MPC framework is put forward to simultaneously resolve the foresight and safety issues in dynamic obstacle avoidance.
2) Compared with the existing cascaded MPC based on hard constraints, the proposed method improves the sensitivity
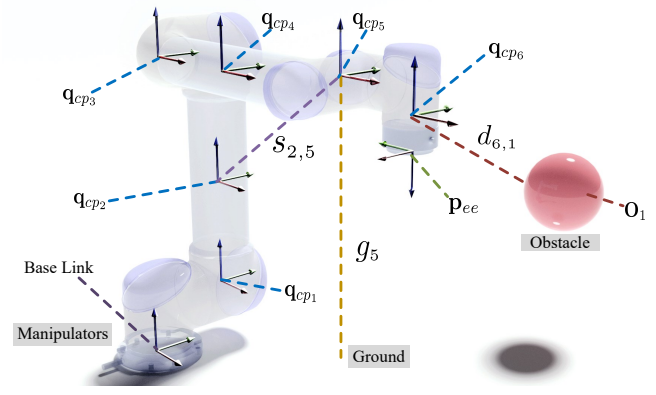

Fig. 1. Selections of critical points in UR5.

to obstacles within the prediction range and circumvents the high-level constraint violations due to obstacles.
3) The proposed scheme takes joint velocities as the input to the robot low-level controller instead of poses in Cartesian space, thus avoiding the singularity problem in conventional trajectory planning.

The paper is organized as follows. In Section II, the models of robot kinematics and dynamic obstacles are discussed. Then, the proposed cascaded nonlinear MPC for obstacle avoidance is introduced in detail in Section III. In Section IV, experiments on a real 6-axis robot manipulator are performed to illustrate the capabilities of the proposed approach. Finally, conclusions are given in Section VI.

II. MODEL DESCRIPTION AND CONTROL OBJECTIVE

A. Robot Kinematics

The core of dynamic obstacle avoidance is to ensure that the body of the robot manipulator always keeps a certain distance from all obstacles. Meanwhile, the end-effector goes as close as possible to the target pose. In this paper, obstacle avoidance is carried out by selecting the critical points on each link of the robot manipulator and ensuring that the critical points meet the safety conditions. An example of the creation of critical points for the whole body of the Universal Robots 5 (UR5) is shown in Fig. 1. Specifically, the i-th critical point position $\mathbf{q}_{cp_i} \in \mathbb{R}^3, i = 1, 2, \cdots, n$ and the end-effector pose $\mathbf{p}_{ee} \in \mathbb{R}^7$ can be calculated with forward kinematics as follows:

$$\mathbf{q}_{cp_i} = f_{cp_i}(\boldsymbol{\theta}) \tag{1}$$
$$\mathbf{p}_{ee} = f_{ee}(\boldsymbol{\theta}) \tag{2}$$

where $\boldsymbol{\theta} \in \mathbb{R}^6$ is the joint position, $f_{cpi}(\cdot)$ and $f_{ee}(\cdot)$ represent the forward kinematics mapping relative to the i-th critical point and end-effector, respectively.

Furthermore, the Cartesian velocities of the critical points and end-effector can be derived through the Jacobian matrix, which are expressed by

$$\dot{\mathbf{q}}_{cp_i} = \mathbf{J}_{cp_i}(\boldsymbol{\theta})\mathbf{u} \tag{3}$$
$$\dot{\mathbf{p}}_{ee} = \mathbf{J}_{ee}(\boldsymbol{\theta})\mathbf{u} \tag{4}$$

where $\mathbf{J}_{cp_i}(\cdot)$ and $\mathbf{J}_{ee}(\cdot)$ denote the Jacobian matrices relative to the i-th critical point and end-effector, respectively, $\mathbf{u} \in \mathbb{R}^6$
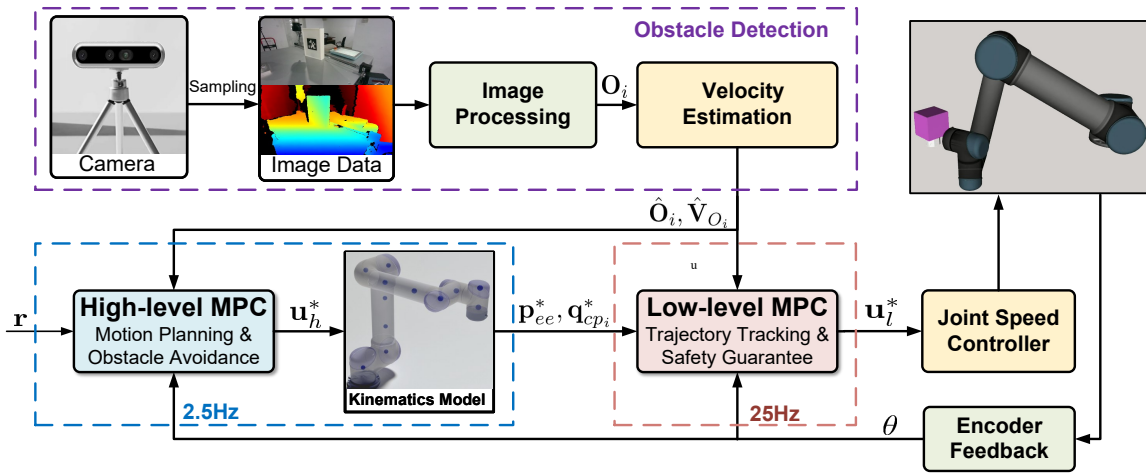
Fig. 2. Framework of the cascaded nonlinear MPC system. $\mathbf{r}$ denotes the reference Cartesian pose, $\mathbf{u}_h^*$ represents the first entry in the optimal control sequence of the high-level MPC, $\mathbf{p}^*_{ee}$ and $\mathbf{q}^*_{cp_i}$ represent the optimal end-effector pose and the i-th optimal critical point position for low-level optimization, respectively, $\mathbf{u}_l^*$ represents the first entry in the optimal control sequence of the low-level MPC, $\mathbf{O}_i$ denotes the spatial position of the i-th obstacle, $\hat{\mathbf{V}}_{O_i}$ represents the velocity estimation of the i-th obstacle, and $\boldsymbol{\theta}$ is the feedback joint position.

is the system input, indicating the joint velocity of the robot manipulator. Discretizing (3) and (4) by Euler method with sampling period T yields

$$\mathbf{q}_{cp_i}(k+1) = \mathbf{q}_{cp_i}(k) + T\mathbf{J}_{cp_i}(\boldsymbol{\theta}(k))\mathbf{u}(k) \tag{5}$$

$$\mathbf{p}_{ee}(k+1) = \mathbf{p}_{ee}(k) + T\mathbf{J}_{ee}(\boldsymbol{\theta}(k))\mathbf{u}(k) \tag{6}$$

B. Obstacle Definition

Considering converting a dynamic obstacle avoidance task into an optimization problem with dynamic constraints, the original obstacle physical model needs to be simplified by treating it as a square or sphere with simple physical characteristics. In this paper, we assume that the obstacles are sphere, then the spatial position of the i-th obstacle is defined as $\mathbf{O}_i = [x_{o_i}, y_{o_i}, z_{o_i}]^\top$.

In addition to the position, the obstacle velocity also needs to be derived since the obstacles are moving. Denote the i-th obstacle velocity as $\mathbf{V}_{O_i} \in \mathbb{R}^3$. With the sampling period T, the relationship between the velocity and position of the i-th obstacle at k sampling instant can be expressed as

$$\mathbf{O}_i(k+1) = \mathbf{O}_i(k) + T\mathbf{V}_{O_i}(k) \tag{7}$$

C. Control Objective

The control task involved in dynamic obstacle avoidance for robot manipulators is a complex process. It primarily requires designing an effective strategy to enable the robot to move smoothly and accurately towards a specified target, despite the presence of dynamic obstacles along its path. To this end, it is crucial to synthesize advanced motion planning and control algorithms that not only prioritize optimal trajectory planning but also offer real-time responsiveness for efficient control.

III. MOTION PLANNING AND CONTROL DESIGN

In this section, a novel cascaded nonlinear MPC framework combined APF is proposed for real-time dynamic obstacle avoidance, as shown in Fig. 2. The specific design of each module is given step by step.

A. STO-Based Velocity Estimation of Obstacles

For the motion estimation of obstacles, a STO is constructed to realize the observation of the obstacle velocity based on the obstacle position obtained by computer vision system. To this end, the motion equation of the i-th obstacle can be further indicated by

$$\mathbf{O}_i(k+1) = \mathbf{O}_i(k) + T\mathbf{V}_{O_i}(k)$$
$$\mathbf{V}_{O_i}(k+1) = \mathbf{V}_{O_i}(k) + T\boldsymbol{\xi}_{ia}(k) \tag{8}$$

where $\boldsymbol{\xi}_{ia}$ denotes the acceleration of the i-th obstacle.

For system (8), the STO is designed as:

$$\hat{\mathbf{O}}_i(k+1) = \hat{\mathbf{O}}_i(k) + T\left(\hat{\mathbf{V}}_{O_i}(k) + \mathbf{Z}_{iv}(k)\right)$$
$$\hat{\mathbf{V}}_{O_i}(k+1) = \hat{\mathbf{V}}_{O_i}(k) + T\mathbf{Z}_{ia}(k) \tag{9}$$

where $\hat{\mathbf{O}}_i$ and $\hat{\mathbf{V}}_{O_i}$ are the state estimation results of $\mathbf{O}_i$ and $\mathbf{V}_{O_i}$, respectively, and $\mathbf{Z}_{iv}$ and $\mathbf{Z}_{ia}$ are the correction terms to guarantee finite-time stability, which is constructed by

$$\mathbf{Z}_{iv}(k) = \mathbf{L_1}\left|\mathbf{O}_i(k) - \hat{\mathbf{O}}_i(k)\right|^{\frac{1}{2}}\text{sign}\left(\mathbf{O}_i(k) - \hat{\mathbf{O}}_i(k)\right)$$
$$\mathbf{Z}_{ia}(k) = \mathbf{L_2}\,\text{sign}\left(\mathbf{O}_i(k) - \hat{\mathbf{O}}_i(k)\right) \tag{10}$$

where $\mathbf{L_1}, \mathbf{L_2} \in \mathbb{R}^{3\times 3}$ are the parameters matrices and $\text{sign}(\cdot)$ denotes the signum function.

Define the the system observation errors as follows:

$$\mathbf{e}_{o_i}(k) = \hat{\mathbf{O}}_i(k) - \mathbf{O}_i(k),$$
$$\mathbf{e}_{v_i}(k) = \hat{\mathbf{V}}_{O_i}(k) - \mathbf{V}_i(k) \tag{11}$$

According to (8) and (9), the error dynamics can be derived as

$$\mathbf{e}_{o_i}(k+1) = \mathbf{e}_{o_i}(k) + T\left(\mathbf{e}_{v_i}(k) - \mathbf{L_1}|\mathbf{e}_{o_i}(k)|^{\frac{1}{2}}\text{sign}(\mathbf{e}_{o_i}(k))\right)$$
$$\mathbf{e}_{v_i}(k+1) = \mathbf{e}_{v_i}(k) + T\left(-\mathbf{L_2}\,\text{sign}(\mathbf{e}_{o_i}(k)) - \boldsymbol{\xi}_{ia}(k)\right) \tag{12}$$

Following Theorem 1 presented in [26], by properly selecting $\mathbf{L_1}, \mathbf{L_2}$, one can arrive at the conclusion that the error dynamic system (12) is finite-time stable.
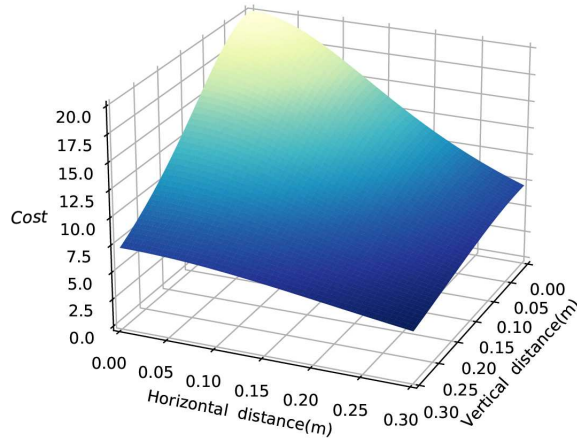
Fig. 3. Cost of repulsive function for the horizontal and vertical distances between a critical point and an obstacle.

B. Motion Planning: High-level MPC

The high-level MPC employs APF to incorporate obstacle information into the cost function, which improves the overall prediction range of the system and increases the predictive foresight to obstacles.

1) Artificial potential field function: To create the potential field, we use the repulsive force to indicate the penalty cost about the distance to obstacles, i.e., let the optimization of MPC tend to be close to the desired pose and away from obstacles with repulsive forces. Based on the world coordinate system, the horizontal distance refers to the straight-line distance in the X-Y plane, while the vertical distance is measured along the Z-axis. A visual image of the repulsive function for the horizontal and vertical distances between a critical point and an obstacle is depicted in Fig. 3.

Denote the distance between the i-th critical point and the j-th obstacle as $d_{i,j}$. For the k sampling instant, we have

$$d_{i,j}(k) = ||\mathbf{q}_{cp_i}(k) - \mathbf{O}_j(k)||_2 \tag{13}$$

By combining $d_{i,j}(k)$, an alternative repulsive function is constructed as

$$P_{i,j}(k) = \begin{cases} \frac{A_o}{d_{i,j}(k)^2 + 0.05}, & d_{i,j}(k) \leq d_{lim} \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

where A_o denotes the weight of the repulsive potential field and d_{lim} represents the farthest distance of the repulsive forces.

Remark 1: The selection of a small constant value 0.05 in (14) serves the main purpose of preventing singularity problems during the optimization and minimizing the impact of significant cost changes resulting from minor distance variations in the APF function.

For simplicity of subsequent expressions, the sum of the total repulsive forces at k sampling instant is expressed by referring to the new variable P_o as

$$P_o(k) = \sum_{i=1}^{N_{cp}} \sum_{j=1}^{N_o} P_{i,j}(k) \tag{15}$$

where N_{cp} represents the total number of critical points and N_o represents the total number of obstacles.

Remark 2: It is worth noting that if only a conventional hard constraints-based MPC is developed for motion planning, there mainly exist three shortcomings: a) The planned trajectory tends to stick close to constraints belt, which may cause the low-level controller to violate constraints; b) Hard constraints are not foresighted and only reactive to obstacles at moments when constraints are to be violated within the predicted range, possibly resulting in jitter phenomenon of the robot; c) A larger discretization time may amplify the difference between the nominal model and the real-world model, which may cause the planned trajectory that possibly violates the prescribed constraints.

2) Constraint settings: For robot manipulators, the first constraints to be considered is that they conform to the actual physical characteristics, i.e., the joint position and joint velocity are physically limited. Regarding the joint position, we have

$$\theta_{i,\min} \leq \theta_i \leq \theta_{i,\max} \tag{16}$$

where $\theta_{i,\min}$ and $\theta_{i,\max}$ represent the minimum and maximum joint positions for the i-th joint, respectively. Similarly, the joint velocity needs to be limited by

$$u_{i,\min} \leq u_i \leq u_{i,\max} \tag{17}$$

where $u_{i,\min}$ and $u_{i,\max}$ represent the minimum and maximum joint velocities for the i-th joint, respectively.

The second issue to be considered for robot manipulators are self-collision. The core of the self-collision avoidance is that each joints are always kept at a safe distance from each other. Mathematically, it can be converted into minimum distance between critical points as

$$s_{i,j} \geq s_{i,j,\min} \tag{18}$$

where $s_{i,j}$ and $s_{i,j,\min}$ represent the distance and minimum constraint distance between the i-th critical point and the j-th critical point, respectively.

Remark 3: To reduce the computational load during optimization, in certain circumstances, this distance can be further converted into constraints on specific joint positions. However, it should be noted that such an approach could potentially reduce the feasible solutions due to the constraints imposed on the joint range. The corresponding joint constraints can be designed as follows:

$$\theta_{i,self,\min} \leq \theta_i \leq \theta_{i,self,\max} \tag{19}$$

where $\theta_{i,self,\min}$ and $\theta_{i,self,\max}$ represent the minimum and maximum joint positions for the i-th joint to avoid self-collision, respectively.

Additionally, it is also necessary to take account of collisions between the robot manipulator and the base ground. For this issue, it can be achieved by constraining the height of critical points from the ground within a certain range, which can be expressed as:

$$g_i \geq g_{i,\min} \tag{20}$$

where g_i and $g_{i,\min}$ respectively represent the height and minimum safe height between the i-th critical point and the ground.

3) MPC for motion planning and obstacle avoidance: In order to achieve the goal of motion planning by reaching the reference Cartesian pose $\mathbf{r} \in \mathbb{R}^7$ and implementing obstacle avoidance, the end-effector pose $\mathbf{p}_{ee,k}$ and critical point positions $\mathbf{q}_{cp_i,k}$ are introduced. According to (5) and (6), one needs the current desired input $\mathbf{u}_k$ to compute the above expected values. To this end, by defining $\mathbf{u}_k = \mathbf{u}_h(k)$ where $\mathbf{u}_h(k)$ is the decision variable, the high-level stage optimization problem in conjunction with APF and constraint settings can be formulated by

$$\mathcal{U}_h^* = \underset{\mathcal{U}_h}{\mathbf{argmin}} \, J(\mathcal{U}_h) \tag{21}$$

with

$$J(\mathcal{U}_h) = \sum_{j=1}^{N_{hp}} \|\mathbf{p}_{ee}(k+j) - \mathbf{r}(k+j)\|_{\mathbf{Q}_h}^2$$
$$+ \sum_{j=0}^{N_{hc}-1} \|\mathbf{u}_h(k+j)\|_{\mathbf{R}_h}^2 + \sum_{j=1}^{N_{hp}} P_o(k+j) \tag{22}$$

subject to

$$\mathbf{q}_{cp_i}(k+1) = \mathbf{q}_{cp_i}(k) + T_h \mathbf{J}_{cp_i}(\boldsymbol{\theta}(k))\mathbf{u}_h(k), \tag{23a}$$
$$\mathbf{p}_{ee}(k+1) = \mathbf{p}_{ee}(k) + T_h \mathbf{J}_{ee}(\boldsymbol{\theta}(k))\mathbf{u}_h(k), \tag{23b}$$
$$\mathbf{O}_j(k+1) = \mathbf{O}_j(k) + T_h \hat{\mathbf{V}}_{O_j}(k), \tag{23c}$$
$$d_{i,j}(k) = \|\mathbf{q}_{cp_i}(k) - \mathbf{O}_j(k)\|_2, \tag{23d}$$
$$\theta_{\iota,\min} \leq \theta_\iota(k) \leq \theta_{\iota,\max}, \tag{23e}$$
$$u_{\iota,\min} \leq u_{h,\iota}(k) \leq u_{\iota,\max}, \tag{23f}$$
$$s_{i,\kappa}(k) \geq s_{i,\kappa,\min}, \tag{23g}$$
$$g_i(k) \geq g_{i,\min}, \tag{23h}$$
$$i = 1, 2, \cdots, N_{cp}, \quad j = 1, 2, \cdots, N_o, \tag{23i}$$
$$\iota = 1, 2, \cdots, 6, \quad \kappa = 1, 2, \cdots, N_{cp} \tag{23j}$$

where $\mathbf{Q}_h$ and $\mathbf{R}_h$ are weighting matrices on the states and inputs, respectively, N_{hp} is the prediction horizon, N_{hc} is the control horizon, $\mathcal{U}_h = [\mathbf{u}_h(k)^\top, \mathbf{u}_h(k+1)^\top, \cdots, \mathbf{u}_h(k+N_{hc}-1)]^\top$ is the control sequence, and T_h is the high-level sampling period.

It can be observed from (22) that the smaller the cost function J means that the optimal control sequence $\mathcal{U}_h^*$ is likely to be closer to the desired pose and farther away from obstacles. Furthermore, by taking the first action of the optimal control sequence, the optimal end-effector pose $\mathbf{p}_{ee,k}^*$ and the i-th optimal critical point position $\mathbf{q}_{cp_i,k}^*$ can be obtained. Subsequently, the resulting optimized values will be served as the low-level MPC input references.

C. Safety Guarantee: Low-level MPC

For the low-level layer, we deploy MPC to ensure that the robot manipulator can smoothly track the optimized trajectory and guarantee sufficient safety without collisions and singularities. We utilize only hard constraints without the inclusion of APF functions for the optimization, resulted from the following reasons: a) Due to the nonlinearity of the APF function, the calculation burden is usually large when performing the receding optimization, which reduces the control bandwidth; b) Soft constraints such as APF are not sufficient to guarantee that the robot manipulator consistently maintains a specific safety distance from obstacles.

1) Constraint settings: First of all, the constraints (16), (17), (18) and (20) on the joint positions, joint velocities, self-collision and ground-collision are taken into consideration. To meet the demands on dynamic obstacle avoidance, the following hard constraints are further required to be guaranteed:

$$d_{i,j} \geq d_{i,j,\min} \tag{24}$$

where $d_{i,j,\min}$ represents the minimum distance between the i-th critical point and the j-th obstacle.

In addition, singularity avoidance can also be considered into the proposed optimization framework, which can be realized by introducing a manipulability ellipsoid $M(\boldsymbol{\theta})$ with the constraint as

$$M(\boldsymbol{\theta}) = \sqrt{\det\left(\mathbf{J}_{ee}(\boldsymbol{\theta})\mathbf{J}_{ee}{}'(\boldsymbol{\theta})\right)} \geq \gamma > 0 \tag{25}$$

where γ is a lower bound of manipulability. It is found that as long as $M(\boldsymbol{\theta})$ is always greater than zero, the robot manipulator will not go through singular configurations that lead to sudden peaks of joint velocities or accelerations.

2) MPC for trajectory tracking and safety guarantee: Since the high-level MPC keeps the current planned path until the next sampling instant, the low-level MPC needs to implement trajectory tracking perfectly with avoidance of collisions and singularities. For this purpose, the optimization problem in the low-level layer is formulated as:

$$\mathcal{U}_l^* = \underset{\mathcal{U}_l}{\mathbf{argmin}} \, J(\mathcal{U}_l) \tag{26}$$

with

$$J(\mathcal{U}_l) = \sum_{j=1}^{N_{lq}} \sum_{i=1}^{N_{cp}} \|\mathbf{q}_{cp_i}(k+j) - \mathbf{q}^*{}_{cp_i,k}\|_{\mathbf{Q}_{lq}}^2$$
$$+ \sum_{j=1}^{N_{lp}} \|\mathbf{p}_{ee}(k+j) - \mathbf{p}^*{}_{ee,k}\|_{\mathbf{Q}_{lp}}^2 \tag{27}$$
$$+ \sum_{j=0}^{N_{lc}-1} \|\mathbf{u}(k+j)\|_{\mathbf{R}_l}^2$$

subject to

$$\mathbf{q}_{cp_i}(k+1) = \mathbf{q}_{cp_i}(k) + T_l \mathbf{J}_{cp_i}(\boldsymbol{\theta}(k))\mathbf{u}(k), \tag{28a}$$
$$\mathbf{p}_{ee}(k+1) = \mathbf{p}_{ee}(k) + T_l \mathbf{J}_{ee}(\boldsymbol{\theta}(k))\mathbf{u}(k), \tag{28b}$$
$$\mathbf{O}_j(k+1) = \mathbf{O}_j(k) + T_l \hat{\mathbf{V}}_{O_j}(k), \tag{28c}$$
$$d_{i,j}(k) = \|\mathbf{q}_{cp_i}(k) - \mathbf{O}_j(k)\|_2, \tag{28d}$$
$$\theta_{\iota,\min} \leq \theta_\iota(k) \leq \theta_{\iota,\max}, \tag{28e}$$
$$u_{\iota,\min} \leq u_{h,\iota}(k) \leq u_{\iota,\max}, \tag{28f}$$
$$s_{i,\kappa}(k) \geq s_{i,\kappa,\min}, \tag{28g}$$
$$g_i(k) \geq g_{i,\min}, \tag{28h}$$
$$M(\boldsymbol{\theta}(k)) \geq \gamma, \tag{28i}$$
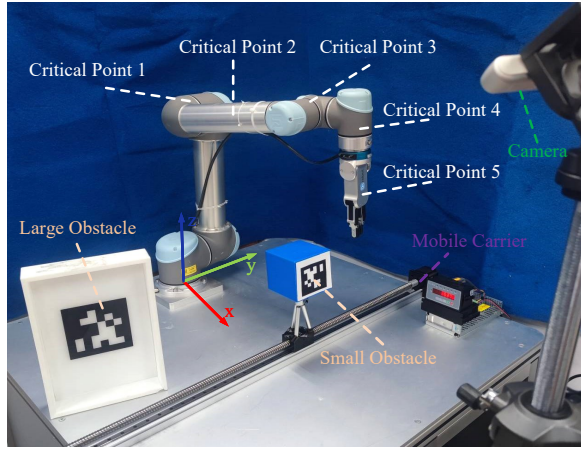
Fig. 4. Experimental setup.
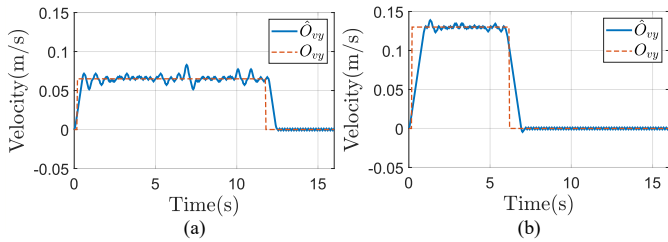

Fig. 6. Frame-by-frame plots under HC-MPC.

Fig. 5. Velocity estimation results for different obstacles: (a) slow moving obstacle and (b) fast moving obstacle.

$$d_{i,j}(k) \geq d_{i,j,\min}, \tag{28j}$$

$$i = 1, 2, \cdots, N_{cp}, \quad j = 1, 2, \cdots, N_o, \tag{28k}$$

$$\iota = 1, 2, \cdots, 6, \quad \kappa = 1, 2, \cdots, N_{cp} \tag{28l}$$

where $\mathbf{Q}_{lq}$, $\mathbf{Q}_{lp}$ and $\mathbf{R}_l$ are weighting matrices, N_{lp} is the prediction horizon and N_{lc} is the control horizon, $\mathcal{U}_l = [\mathbf{u}(k)^\top, \mathbf{u}(k+1)^\top, \cdots, \mathbf{u}(k+N_{hc}-1)]^\top$ is the control sequence, and T_l is the low-level sampling period.

Similarly, by taking the first action of the optimal control sequence $\mathcal{U}_l^*$, the desired joint velocities can be derived for the dynamic controller of the robot manipulator.

IV. EXPERIMENTAL VALIDATION

In this section, to illustrate the effectiveness of the proposed cascaded nonlinear MPC strategy, comparative experiments are performed on a UR5 robot. Detailed results in real world experiments can also be referred in our video on the website.[1]

A. Experimental Setup

The experimental setup consisted of a 6-axis robot manipulator and two obstacles, as shown in Fig. 4. In order to reduce the burden of visual recognition, both of these obstacles were marked with AprilTags [27]. For the sake of explanation, we refer to a rectangular prism with dimensions of 20 cm in length, 30 cm in height, and 6 cm in width as the large obstacle, while a cube with an edge length of 10 cm as
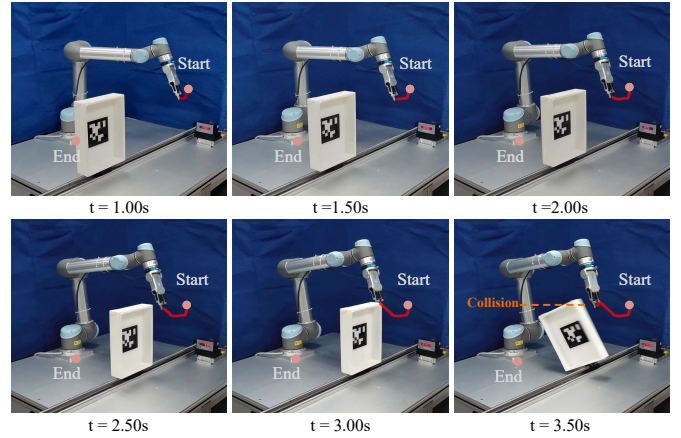
[1] https://www.bilibili.com/video/BV1TG4y1o7Z7.

TABLE I
INITIAL AND DESIRED POSES OF THE END-EFFECTOR

	Cartesian poses of end-effector						
	x (m)	y (m)	z (m)	q_w	q_x	q_y	q_z
Initial	0.40	0.40	0.45	0.8404	-0.1443	0.5075	0.1236
Desired	0.50	-0.40	0.50	0.7068	0.0003	0.7074	0.0003

the small obstacle. The real-time image was captured by a Realsense camera. The software structure was based on robot operating system (ROS) and MoveIt! [28]. ACADO toolkit was employed to solve the two-stage optimization problem [29]. Image processing of the obstacle was realized by VISP library [30]. The software ran on a Lenovo laptop with an i7-10750H CPU and 32 GB RAM.

We used the basic link of the manipulator as the origin of the world coordinate system. The initial and desired poses of the end-effector were shown in Table I. The initial position of the large obstacle relative to the world coordinate system was $[0.58, -0.49, 0.31]^\top$ m, and the small obstacle was $[0.58, -0.49, 0.25]^\top$ m. To thoroughly evaluate the test with obstacles moving at different speeds, we designed a motor-driven mechanism to control the obstacle movement speeds at two distinct levels: 6.5 cm/s and 13 cm/s. For the sake of simplicity in subsequent descriptions, we refer to a speed of 6.5 cm/s as the slow speed and a speed of 13 cm/s as the fast speed. The velocities for both the slow and fast speeds, along the Y-axis, can be represented as follows:

$$O_{vy,slow} = \begin{cases} 0.065 \text{ m/s} & 0\text{s} \leq t \leq 12\text{s} \\ 0 & t > 12\text{s} \end{cases}$$

$$O_{vy,fast} = \begin{cases} 0.13 \text{ m/s} & 0\text{s} \leq t \leq 6\text{s} \\ 0 & t > 6\text{s} \end{cases}$$

B. Comparative Controller Settings

To better demonstrate the effectiveness of the proposed APF-based cascaded MPC (APF-Double-MPC), two kinds of control strategies with different architectures were designed
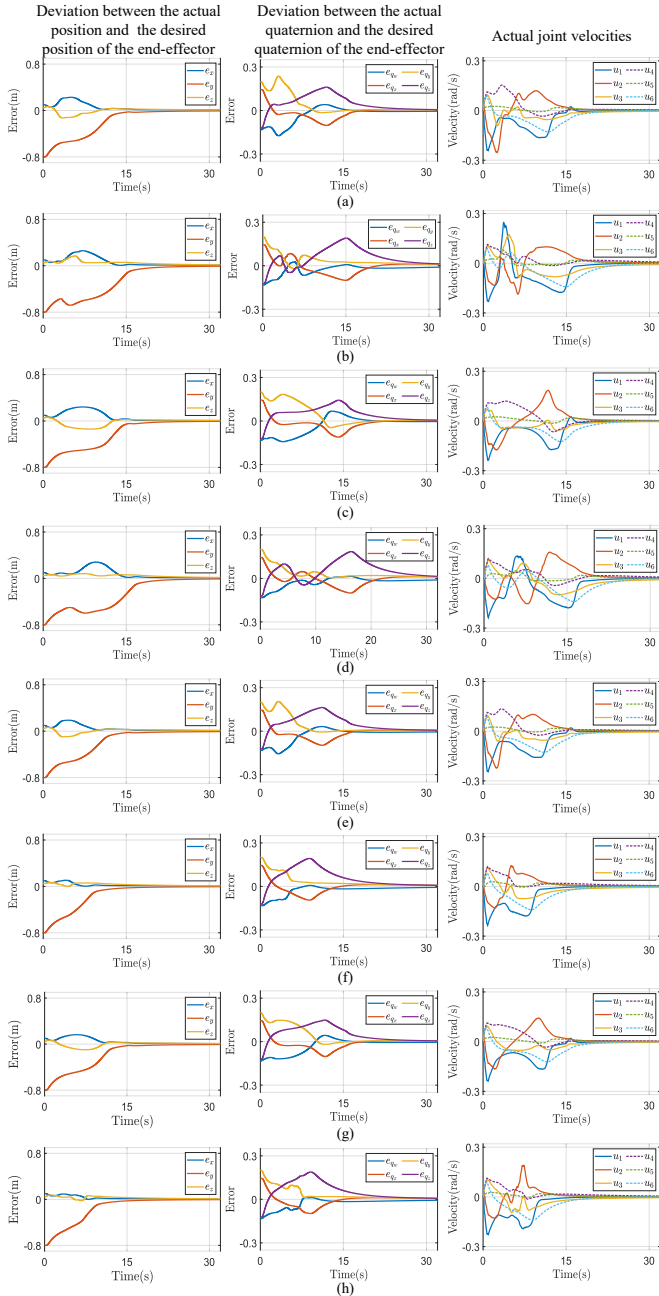
Fig. 7. Error and control input curves of APF-Double-MPC and HC-Double-MPC for obstacles with different sizes and different speeds: (a) APF-Double-MPC with the large fast obstacle, (b) HC-Double-MPC with the large fast obstacle, (c) APF-Double-MPC with the large slow obstacle, (d) HC-Double-MPC with the large slow obstacle, (e) APF-Double-MPC with the small fast obstacle, (f) HC-Double-MPC with the small fast obstacle, (g) APF-Double-MPC with the small slow obstacle, and (h) HC-Double-MPC with the small slow obstacle.
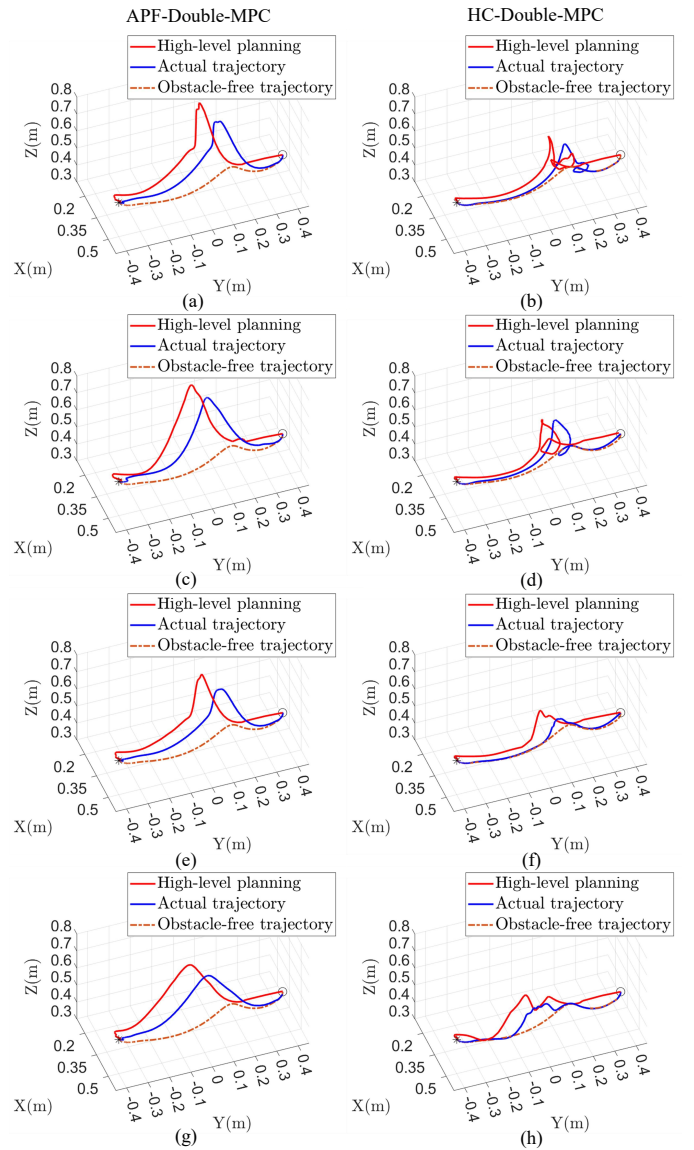

Fig. 8. Actual and obstacle-free trajectories of the robot manipulator's end-effector during the high-level planning: (a) APF-Double-MPC with the large fast obstacle, (b) HC-Double-MPC with the large fast obstacle, (c) APF-Double-MPC with the large slow obstacle, (d) HC-Double-MPC with the large slow obstacle, (e) APF-Double-MPC with the small fast obstacle, (f) HC-Double-MPC with the small fast obstacle, (g) APF-Double-MPC with the small slow obstacle, and (h) HC-Double-MPC with the small slow obstacle.

for performance comparison, i.e., the independent MPC with hard constraints (HC-MPC) in [17] and the cascaded MPC with hard constraints (HC-Double-MPC) in [21].

For the configured device, we chose 5 critical points, as shown in Fig. 4, namely, $N_{cp} = 5$, $N_o = 1$. To prevent self-collision within the specified range of joint motion, the position of the fourth joint was restricted between -2 rad and 2 rad, the position of the fifth joint was limited between -2 rad and 0 rad, and the remaining joints were constrained between -2.5 rad and 2.5 rad. Moreover, the joint velocities were limited between -0.6 rad/s and 0.6 rad/s, and the minimum distance between each critical point and the ground $g_{i,\min}$ was set as 10 cm. To guarantee the efficient obstacle avoidance, the value of d_{lim} should be set according to the respective obstacle sizes. For the large obstacle, d_{lim} was set as 25 cm, while for the small obstacle, d_{lim} was set as 15 cm.

In order to ensure the fairness of experimental comparisons as much as possible, for the high-level layer of APF-Double-MPC and HC-Double-MPC, the sampling period T_h was set as 0.4s. Due to the large sampling period already present at the higher level, HC-Double-MPC, relying on hard constraints,
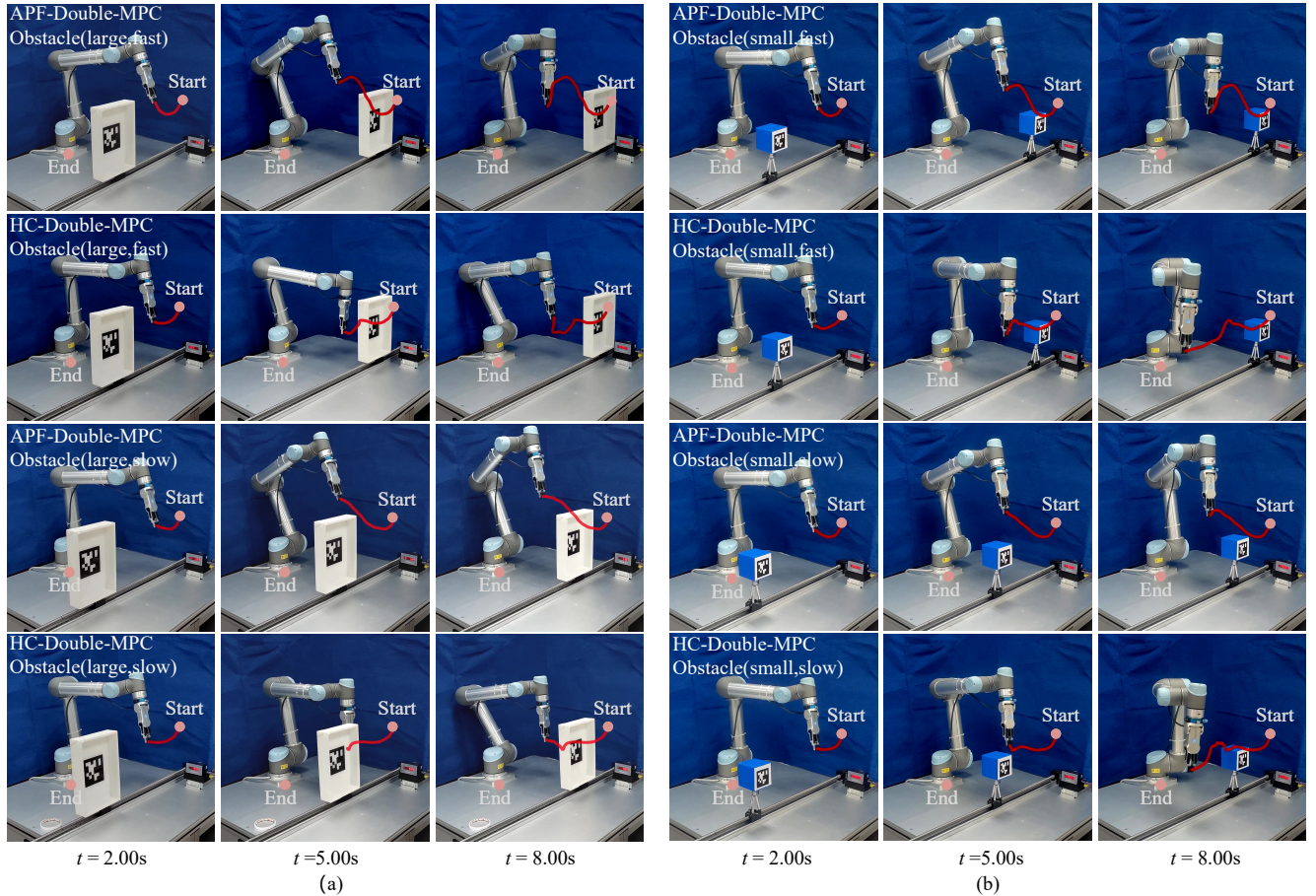
Fig. 9. Frame-by-frame plots under APF-Double-MPC and HC-Double-MPC: (a) results under the large obstacle and (b) results under the small obstacle.

is highly susceptible to violating them when the prediction horizon is excessively prolonged. So the HC-Double-MPC and APF-Double-MPC prediction horizon N_{hp} was set as 3 and the control horizon N_{hc} was set as 1. For the weighting matrices $\mathbf{Q}_h$ and $\mathbf{R}_h$, the elements on the diagonal were selected as 10 and 1 respectively. For the APF function, as the weight parameter A_o increases, the robot manipulator tends to maintain a greater distance from obstacles. However, a higher gradient in the APF may introduce oscillations in the planned trajectory. Therefore, we set A_o as 1 in the proposed approach.

For the parameters selection of STO, to achieve an accurate velocity estimation of the obstacle, the diagonal matrix $\mathbf{L_1}$ was set as $0.3\mathbf{I}_{3\times3}$ and $\mathbf{L_2}$ was set as $0.24\mathbf{I}_{3\times3}$.

C. Experiments on Dynamic Obstacle Avoidance

First of all, the effect of STO obstacle velocity estimation is shown in Fig. 5. It can be observed that the estimation of the obstacle velocity can be basically completed within 1s.

In order to illustrate the advantages of the designed motion planning and control schemes on dynamic obstacle avoidance, five aspects in terms of task reachability, motion smoothness, constraint violations, predictive foresight and computational time consumption are compared.

1) Task reachability: As shown in Fig. 7, the first column of the images shows the curve of position error changing

with time, the second column shows the curve of posture error changing with time, and the third column shows the curve of joint velocity changing with time. For each case with varying obstacle sizes and speeds, the tracking errors of APF-Double-MPC and HC-Double-MPC eventually converge to zero. However, as shown in Fig. 6, HC-MPC has crashed with the obstacle during the planning task, which is due to that the obstacle avoidance behavior is too aggressive. Since the prediction range of the independent MPC is short, it is very easy to cause the failure of the task, therefore, we will not further compare HC-MPC in the follow-up content.

2) Motion smoothness: As can be seen from the frame-by-frame images in Fig. 9, the actual end-effector trajectory under APF-Double-MPC control is noticeably smoother than that of HC-Double-MPC. Additionally, the planning images in Fig. 8 reveal that the high-level planning of APF-Double-MPC is significantly smoother than HC-Double-MPC and does not exhibit abrupt changes in the planned trajectory. Moreover, the velocity curves in Fig. 7 also corroborate this observation, as the joint velocity changes in APF-Double-MPC are noticeably less abrupt compared to those in HC-Double-MPC.

3) Constraint violations: We introduce the index Λ to measure the constraint violations as follows:

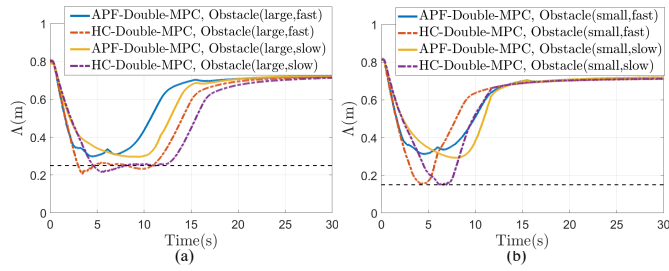$$\Lambda(k) = \min_{i,j} d_{i,j}(k) \tag{29}$$

Fig. 10. Minimum distance between the obstacle and critical points under APF-Double-MPC and HC-Double-MPC: (a) results under the large obstacle and (b) results under the small obstacle.
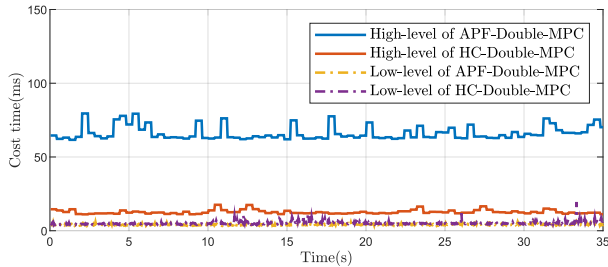

Fig. 11. Computational time consumption of APF-Double-MPC and Hc-Double-MPC.

Fig. 10 displays the closest distance between the obstacles and the robot manipulator during the obstacle avoidance process, with the black dashed lines representing the corresponding constraint bands. Regardless of whether encountering a large or small obstacle, the HC-Double-MPC violates the constraint conditions multiple times during obstacle avoidance. In contrast, APF-Double-MPC, guided by the high-level APF, effectively avoids getting too close to the obstacles.

4) Predictive foresight: The predictive foresight of the algorithms can be observed from the planned trajectories and the minimum distance to the obstacle. As seen in Fig. 8, during the early stages of an approaching obstacle, APF-Double-MPC demonstrates a clear trend of adjusting its trajectory to avoid the obstacle, while HC-Double-MPC essentially coincides with the original obstacle-free trajectory, only altering its trajectory when it is on the verge of violating the constraints. In Fig. 10, the minimum distance to the obstacle also indicates that when the obstacle gets closer (around 4 seconds), the slope of the APF-Double-MPC curve exhibits a more gradual trend, whereas HC-Double-MPC maintains its original trajectory until it approaches the constraint band.

5) Computational time consumption: The computational time consumption for APF-double-MPC and HC-double-MPC as illustrated in Fig. 11. The computation time for both low-level controllers is almost identical, with a maximum time of 17.584 ms. For high-level planning, since APF-double-MPC introduces a highly nonlinear term in the cost function, it results in a significantly longer computation time than HC-double-MPC. But it is notable that its maximum computation time is 79.514 ms, which also satisfies the planning requirement to be completed within 400 ms.

V. Conclusion

In this paper, a novel cascaded nonlinear MPC framework for dynamic obstacle avoidance has been proposed. The purpose of this solution is to improve the safety of robot manipulators in motion planning and control. For the velocity estimation of the dynamic obstacle, a STO has been firstly deployed to improve the estimated accuracy. With the construction of a two-stage MPC optimization problem, the optimized joint velocities are derived to guarantee the collision-free and singularity-free motion planning. The performance of the proposed strategy is experimentally verified on a six-axis robot manipulator in terms of task reachability, motion smoothness, predictive foresight, constraint violations, and computational time consumption. The reported results show that the proposed method has more distinguishable features for dynamic obstacle avoidance in comparison with the conventional cascaded MPC. Our future work would focus on the adaptability of our proposed strategy in different uncertain environments.

References

[1] G. Fang, M. C. Chow, J. D. Ho, Z. He, K. Wang, T. Ng, J. K. Tsoi, P.-L. Chan, H.-C. Chang, D. T.-M. Chan *et al.*, "Soft robotic manipulator for intraoperative mri-guided transoral laser microsurgery," *Science Robotics*, vol. 6, no. 57, 2021, Art. no. eabg5575.

[2] F. Zhang and Y. Demiris, "Learning garment manipulation policies toward robot-assisted dressing," *Science robotics*, vol. 7, no. 65, 2022, Art. no. eabm6010.

[3] Y. Huang, D. Büchler, O. Koç, B. Schölkopf, and J. Peters, "Jointly learning trajectory generation and hitting point prediction in robot table tennis," in *2016 International Conference on Humanoid Robots (Humanoids)*, pp. 650–655, 2016.

[4] H. Ma, C. Li, J. Liu, J. Wang, and M. Q.-H. Meng, "Enhance connectivity of promising regions for sampling-based path planning," *IEEE Transactions on Automation Science and Engineering*, DOI 10.1109/TASE.2022.3191519, 2022.

[5] G. Tanzmeister, D. Wollherr, and M. Buss, "Grid-based multi-road-course estimation using motion planning," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 1924–1935, 2016.

[6] Y. Huang, H. Ding, Y. Zhang, H. Wang, D. Cao, N. Xu, and C. Hu, "A motion planning and tracking framework for autonomous vehicles based on artificial potential field elaborated resistance network approach," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 2, pp. 1376–1386, 2019.

[7] J. Ma, X. Li, W. Liang, and K. K. Tan, "Parameter space optimization towards constrained controller design with application to tray indexing," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 7, pp. 5575–5585, 2020.

[8] B. Hu, Z. Cao, and M. Zhou, "An efficient RRT-Based framework for planning short and smooth wheeled robot motion under kinodynamic constraints," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 4, pp. 3292–3302, 2021.

[9] K. Jang, J. Baek, S. Park, and J. Park, "Motion planning for closed-chain constraints based on probabilistic roadmap with improved connectivity," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 4, pp. 2035–2043, 2022.

[10] J. Qi, H. Yang, and H. Sun, "MOD-RRT*: A sampling-based algorithm for robot path planning in dynamic environment," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 8, pp. 7244–7251, 2021.

[11] L. Kästner, X. Zhao, T. Buiyan, J. Li, Z. Shen, J. Lambrecht, and C. Marx, "Connecting deep-reinforcement-learning-based obstacle avoidance with conventional global planners using waypoint generators," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1213–1220, 2021.

[12] S. Haddadin, H. Urbanek, S. Parusel, D. Burschka, J. Roßmann, A. Albu-Schäffer, and G. Hirzinger, "Real-time reactive motion generation based on variable attractor dynamics and shaped velocities," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3109–3116, 2010.

[13] A. De Luca, A. Albu-Schäffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the dlr-iii lightweight manipulator arm," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1623–1630, 2006.

[14] J. Yang, W. X. Zheng, S. Li, B. Wu, and M. Cheng, "Design of a prediction-accuracy-enhanced continuous-time MPC for disturbed systems via a disturbance observer," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 9, pp. 5807–5816, 2015.

[15] K. Zhang and Y. Shi, "Adaptive model predictive control for a class of constrained linear systems with parametric uncertainties," *Automatica*, vol. 117, p. 108974, 2020.

[16] Y. Shi and K. Zhang, "Advanced model predictive control framework for autonomous intelligent mechatronic systems: A tutorial overview and perspectives," *Annual Reviews in Control*, vol. 52, pp. 170–196, 2021.

[17] B. Lindqvist, S. S. Mansouri, A.-a. Agha-mohammadi, and G. Nikolakopoulos, "Nonlinear MPC for collision avoidance and control of uavs with dynamic obstacles," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6001–6008, 2020.

[18] Z. Zuo, X. Yang, Z. Li, Y. Wang, Q. Han, L. Wang, and X. Luo, "MPC-Based cooperative control strategy of path planning and trajectory tracking for intelligent vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 3, pp. 513–522, 2021.

[19] Y. Ma, F. Farshidian, T. Miki, J. Lee, and M. Hutter, "Combining learning-based locomotion policy with model-based manipulation for legged mobile manipulators," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2377–2384, 2022.

[20] S. Chai, L. Wang, and E. Rogers, "A cascade MPC control structure for a PMSM with speed ripple minimization," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 8, pp. 2978–2987, 2013.

[21] J. Ding, L. Han, L. Ge, Y. Liu, and J. Pang, "Robust locomotion exploiting multiple balance strategies: An observer-based cascaded model predictive control approach," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 4, pp. 2089–2097, 2022.

[22] A. Chalanga, S. Kamal, L. M. Fridman, B. Bandyopadhyay, and J. A. Moreno, "Implementation of super-twisting control: Super-twisting and higher order sliding-mode observer-based approaches," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 6, pp. 3677–3685, 2016.

[23] T. Zhang, Z. Xu, J. Li, H. Zhang, and C. Gerada, "A third-order super-twisting extended state observer for dynamic performance enhancement of sensorless ipmsm drives," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 7, pp. 5948–5958, 2019.

[24] D. Xu, B. Ding, B. Jiang, W. Yang, and P. Shi, "Nonsingular fast terminal sliding mode control for permanent magnet linear synchronous motor via high-order super-twisting observer," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 3, pp. 1651–1659, 2022.

[25] Y. Kali, M. Saad, and K. Benjelloun, "Optimal super-twisting algorithm with time delay estimation for robot manipulators based on feedback linearization," *Robotics and Autonomous Systems*, vol. 108, pp. 87–99, 2018.

[26] I. Salgado, I. Chairez, B. Bandyopadhyay, L. Fridman, and O. Camacho, "Discrete-time non-linear state observer based on a super twisting-like algorithm," *IET Control Theory & Applications*, vol. 8, no. 10, pp. 803–812, 2014.

[27] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, pp. 3400–3407, 2011.

[28] M. Görner, R. Haschke, H. Ritter, and J. Zhang, "Moveit! task constructor for task-level motion planning," in *2019 International Conference on Robotics and Automation*, pp. 190–196, 2019.

[29] B. Houska, H. J. Ferreau, and M. Diehl, "ACADO toolkit-an open-source framework for automatic control and dynamic optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.

[30] É. Marchand, F. Spindler, and F. Chaumette, "ViSP for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics & Automation Magazine*, vol. 12, no. 4, pp. 40–52, 2005.

Tianqi Zhu received the B.Sc. degree in automation from the Qingdao University of Science and Technology, QingDao, China, in 2020. He is currently working toward the master degree with the School of Automation, Shanghai University of Electric Power, ShangHai, China. His research interests include motion planning, model predictive control and visual servoing.

Jianliang Mao (S'15-M'19) received the B.Sc., M.Sc., and Ph.D. degrees from the School of Automation, Southeast University, Nanjing, China, in 2011, 2014, and 2018, respectively. From 2018 to 2021, he worked at the Research and Development Institute, Estun Automation Co., Ltd. He was selected into "Jiangsu Province entrepreneurship and innovation plan" in 2019. He is currently with the College of Automation Engineering, Shanghai University of Electric Power. His research interests include model predictive control, sliding mode control, vision based interactive control and their applications to electric drives and robot manipulators.

Linyan Han received the M.Sc. degree in Monitoring Technology and Automation Installation from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2017, and Ph.D. degree in Control Science and Engineering from the School of Automation, Southeast University, Nanjing, China, in 2022. She is currently a postdoctoral researcher at the School of Mechanical Engineering, University of Leeds. Her interests include force control, nonlinear control theory and their applications to robotic systems.

Chuanlin Zhang (M'14-SM'19) received the B.Sc. degree in mathematics and the Ph.D. degree in control theory and control engineering from the School of Automation, Southeast University, Nanjing, China, in 2008 and 2014, respectively. He was a Visiting Ph.D. Student with the Department of Electrical and Computer Engineering, University of Texas at San Antonio, USA, from 2011 to 2012; a Visiting Scholar with the Energy Research Institute, Nanyang Technological University, Singapore, from 2016 to 2017; a visiting scholar with Advanced Robotics Center, National University of Singapore, from 2017 to 2018. Since 2014, he has been with the College of Automation Engineering, Shanghai University of Electric Power, Shanghai, where he is currently a Professor. His research interests include nonlinear system control theory and applications for power systems.

Jun Yang (M'11-SM'18-F'22) received the B.Sc. degree in automation from the Department of Automatic Control, Northeastern University, Shenyang, China, in 2006, and the Ph.D. degree in control theory and control engineering from the School of Automation, Southeast University, Nanjing, China in 2011. He joined the Department of Aeronautical and Automotive Engineering at Loughborough University from 2020 as a Senior Lecturer. His research interests include disturbance estimation and compensation, advanced control theory and its application to flight control systems and motion control systems.