Received: 14 December 2021 | Revised: 30 June 2022 | Accepted: 10 August 2022

DOI: 10.1002/asjc.2946

RESEARCH ARTICLE

WILEY

A model predictive obstacle avoidance method based on dynamic motion primitives and a Kalman filter

Qinwen Li[1,2,3] | **Zhiqian Wang**[1,2] | **Wenrui Wang**[1,2,3] | **Zhiyang Liu**[3] | **Yiwen Chen**[3] | **Xianyao Ng**[3] | **Marcelo H. Ang Jr.**[3]

[1]Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China

[2]University of Chinese Academy of Sciences, Beijing 100049, China

[3]Advanced Robotics Centre, Department of Mechanical Engineering, National University of Singapore, Singapore 117608, Singapore

Correspondence
Qinwen Li, Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China; University of Chinese Academy of Sciences, Beijing 100049, China.
Email: liqinwen93@163.com

Present Address
Qinwen Li, 3888 Dongnanhu Rd, Erdao District, Changchun, Jilin, China.

Funding information
China Scholarship Council, Grant/Award Number: 202004910793

Summary

A dynamic motion primitive (DMP) is a robust framework that generates obstacle avoidance trajectories by introducing perturbative terms. The perturbative term is usually constructed with an artificial potential field (APF) method. Dynamic obstacle avoidance is rarely considered with this approach; furthermore, even when dynamic obstacles are considered, only the velocity and position information of the current state are incorporated into the obstacle avoidance framework. However, if the position of an obstacle changes suddenly, a robot may be placed in a dangerous position close to the obstacle, resulting in large obstacle avoidance accelerations, sharp trajectories, or even obstacle avoidance failure. Therefore, we present a model predictive obstacle avoidance method based on dynamic motion primitives and a Kalman filter. This method has three main components: Dynamic motion primitives are used to generate the desired trajectory and introduce perturbations to achieve obstacle avoidance; the Kalman filter method is adopted to estimate the future positions of the obstacles; and model predictive control is employed to optimize the repulsive force generated by the APF while minimizing the defined cost function, thus guaranteeing the safety and flexibility of the method. We validate the presented method with 2D and 3D obstacle avoidance simulations. The method is also verified with a real robot: the-Kinova MOVO. The simulation and experimental results show that the proposed method not only avoids dynamic obstacles but also tracks the desired trajectory more smoothly and precisely.

KEYWORDS

artificial potential field (APF), dynamic motion primitive (DMP), dynamic obstacle avoidance, kalman filter, model predictive control (MPC)

1 | INTRODUCTION

Robots are widely deployed in various aspects of life and industrial production, and they must operate in both static and dynamic environments. Thus, to guarantee the safety of the workflow in robot applications, robots must generate collision-free trajectories in dynamic environments.

Two different types of strategies have been used to achieve dynamic obstacle avoidance for robots: global strategies and local strategies. Global strategies typically use search processes or path planning to explore collision-free trajectories [1, 2], such as rapidly exploring random trees (RRTs) and constrained motion planning networks. However, global strategies are often computationally expensive and time consuming. Local strategies usually accomplish obstacle avoidance by combining closed-loop control with artificial potential field (APF) methods [3, 4]. While local strategies perform rapid computations, the obtained trajectories are usually suboptimal. Therefore, some optimization methods have been developed to improve the obstacle avoidance performance.

An instantaneous optimal control (IOC) planner coupled with a symplectic penalty iteration has been proposed to address the kinodynamic planning problem of differential algebraic equation (DAE) systems [5]. In this framework, penalty techniques are used to obtain a suboptimal path in a complex dynamic environment in which the target or obstacles are moving [6]. A new machine learning algorithm for learning optimal feedback control policies has also been developed for guiding robots to goals in the presence of obstacles [7]. This algorithm can resist local minima, handle moving obstacles and reduce computational costs. Furthermore, an optimal trajectory planning technique for robot manipulators has been presented [8] in which a novel ranking technique was designed for the penalty function and the bounds on joint velocities, accelerations, jerks, and force/torques were taken into account. In our work, dynamic motion primitives (DMPs) are combined with artificial potential fields (APFs) as a versatile framework (DMP-APF) for addressing this issue.

An additional perturbative term derived from the APF can be used to modify a trajectory generated with a DMP to achieve obstacle avoidance. Different types of APFs are constructed to apply in various obstacle avoidance scenarios. For point-like obstacles, static and dynamic potential fields are used to address static and moving obstacles, respectively [9, 10]. Superquadric potential functions have also been adopted to model obstacles and achieve volumetric obstacle avoidance [11]. However, when DMP frameworks are applied, a DMP-APF method may lack the ability to return to the primitive motion after the obstacles are avoided [3]. Furthermore, most DMP-APF methods focus on only the immediate perception information and do not consider potential collisions in the near future. However, if the position of the obstacle changes suddenly, it may be difficult to safely avoid fast-moving objects [12]. Thus, a model predictive obstacle avoidance method based on dynamic motion primitives and a Kalman filter (DMP-KMPC) is presented to address these issues.

The use of model predictive control to optimize additional DMP disturbance terms is another strategy for optimizing the obstacle avoidance performance that is explored in this paper. On the one hand, the obstacle avoidance constraint can be realized, and better performance can be obtained by designing a model predictive control (MPC) cost function [13]; on the other hand, the calculation efficiency can be optimized by adjusting the MPC prediction horizon. To obtain information about the obstacles in the prediction horizon, a Kalman filtering method is used to estimate the state of the obstacles, including their velocity and position. Volumetric obstacle avoidance based on superquadric potential functions [11] is employed to construct the obstacle regions of the dynamic obstacles. Moreover, we use the APF method to compute an initial optimization value at each step in the prediction horizon. Only the size and direction of the perturbative term need to be adjusted based on the initial value, as opposed to determining an optimal input in an indeterminate value domain. The proposed model obstacle avoidance method optimizes the obstacle avoidance effect by adjusting the repulsive force field within the prediction horizon. The main contributions of this paper are summarized as follows:

1. Under the MPC framework, the generated collision-free trajectory can be optimized online according to the position and velocity information of dynamic obstacles in the prediction horizon. The obstacle information can be estimated by the Kalman filter.
2. By designing the cost function of the MPC framework, collision-free trajectories can be obtained, and the tracking performance can be guaranteed. Furthermore, the DMP can return to the primitive motion after avoiding obstacles in this manner.
3. The results of simulation experiments demonstrate the capabilities of the DMP-KMPC method in generating safe trajectories. Compared with other DMP-APF methods, this approach has a better obstacle avoidance ability and improved trajectory tracking performance, especially when faced with dynamic obstacles.

The remainder of this paper is organized as follows. In Section 2, we survey related works on dynamic obstacle avoidance methods and MPC-based obstacle avoidance methods. We discuss the details of the presented DMP-KMPC method in Section 3. In Section 4, the results and analyses of the simulations and experiments are provided, and the effectiveness and superiority of the DMP-KMPC method are verified. We present our conclusions in Section 5.

2 | RELATED WORKS

Collision avoidance with dynamic obstacles is a well-studied problem, and many methods have been proposed to address this issue. In APF methods, the dynamic potential field is usually defined to avoid moving obstacles

[3, 9]. The positions and velocities of the obstacles are adopted to formulate the repulsive potential and generate collision-free trajectories. The velocity and position information are collected by sensors. However, the position and velocity of the obstacles may change rapidly in the near future. As a result, obstacle avoidance methods may lose their effect, and obstacle avoidance may fail once the obstacles are no longer quasistatic (i.e., slow relative motion) or move quickly. An APF-based dynamic obstacle avoidance method has been combined with event cameras to overcome this issue [12]. Event cameras can be used to distinguish static and dynamic obstacles, allowing the robot to avoid dynamic obstacles safely and in a timely manner. However, this approach is difficult to generalize because specific equipment (event cameras) must be used. Furthermore, global strategies are typically applied in path planning problems in known or static environments [14]. When the environment varies, the planned trajectories may no longer be feasible, and motion replanning is necessary, potentially resulting in significant computational costs. Even if the states of the obstacles throughout the whole process are predicted and incorporated into global strategic planning, the calculation becomes expensive [15]. Moreover, the accuracy of the predicted state is not guaranteed.

MPC-based obstacle avoidance methods have been widely applied in dynamic environments. A state interception method based on a convex MPC formulation was used to generate a collision-free trajectory for a quadrotor [16]. The MPC formulation allows vehicles to make a more informed and therefore safer decisions, thus enabling autonomous vehicles to avoid dynamic obstacles [17]. These MPC-based methods are mostly used for autonomous vehicle and unmanned aerial vehicle (UAV). This approach maximizes the amount of information that an autonomous vehicle can gain along its trajectory [18]. As a result, robots can maximize the amount of information gathered along their trajectories, allowing them to perform safer and timelier obstacle avoidance actions [19]. Obstacle information is typically used to formulate the penalty function or regarded as constraints in an MPC framework [13, 20,21]. However, MPC-based methods are rarely extended to obstacle avoidance for robotic arms [22]. Because robotic arms generally have more than six joints and, thus, more control inputs, this approach is unfavorable for MPC online optimization solutions.

In our work, MPC was innovatively combined with a DMP obstacle avoidance framework to optimize the additional perturbative term. The perturbative term is generated with a superquadric potential, allowing obstacles with more complex shapes to be introduced. The term generated by an APF is regarded as a repulsive force; thus, we need to optimize only three inputs if the robot operates in three-dimensional space. We define a cost function that includes a penalty function and a trajectory optimization term, and obstacle avoidance and trajectory tracking optimization are realized by minimizing the cost function. In the prediction horizon, the state of the robot can be calculated with a DMP, while the states of dynamic obstacles are estimated by a Kalman filter [23]. To improve the computational efficiency and ensure the accuracy of the estimated obstacle states, we select a relatively small prediction horizon.

3 | ARCHITECTURE OF THE PROPOSED OBSTACLE AVOIDANCE METHOD

Figure 1 shows a schematic of the proposed DMP-KMPC obstacle avoidance method. This section elaborates on this framework and explains the role of each component (including the DMP, Kalman filter, and MPC) in our approach. First, the DMP is used to learn the nonlinear force term according to a demonstration, and obstacle

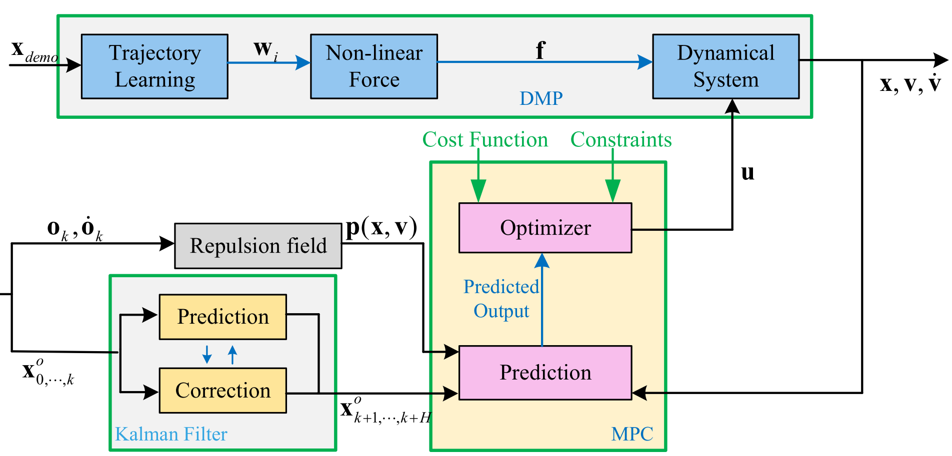

FIGURE 1 The schematic of the DMP-KMPC method [Color figure can be viewed at wileyonlinelibrary.com]

avoidance is achieved by adding a repulsive term to the DMP. We use the weight coefficient $\boldsymbol{\omega}_i$ of the Gaussian basis function to modulate the forcing term $\mathbf{f}$. The value of $\boldsymbol{\omega}_i$ is obtained through weighted linear regression to ensure that the actual execution trajectory $\mathbf{x}$ is essentially consistent with the demonstration trajectory $\mathbf{x}_{demo}$ without the influence of the repulsive term [24]. Then, a Kalman filter is used to estimate the position and velocity of the obstacles in the next steps. Finally, the MPC component optimizes the additional perturbation term of the DMP component by using the estimated state of the obstacle. Here, the prediction horizon of the MPC component is the same as the estimation horizon of the Kalman filter. To achieve better obstacle avoidance and tracking performance, the defined cost function is minimized in the prediction horizon while satisfying the MPC constraints.

3.1 | Obstacle avoidance based on the DMP method

A DMP uses a self-stabilizing second-order system to construct an attraction point model so that the final state of the system can be changed by changing the attraction point [24]. In n-dimensional space, the attractor equations of DMPs can be defined as [25]

$$\begin{aligned} \tau\dot{\mathbf{v}} &= \mathbf{K}(\mathbf{g}-\mathbf{x}) - \mathbf{D}\mathbf{v} + diag(\mathbf{g}-\mathbf{x}_0)\mathbf{f}, \\ \tau\dot{\mathbf{x}} &= \mathbf{v}, \end{aligned} \quad , \tag{1}$$

where $\mathbf{K}$ and $\mathbf{D} \in \mathbb{R}^{n\times n}$ are positive constant diagonal matrices, $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{v} \in \mathbb{R}^n$ are the position and velocity of the system, $\mathbf{x}_0 \in \mathbb{R}^n$ and $\mathbf{g} \in \mathbb{R}^n$ are the starting position and goal position, τ is a time scaling constant, and $diag(\mathbf{g}-\mathbf{x}_0) \in \mathbb{R}^{n\times n}$ is a diagonal matrix with elements equal to $\mathbf{g}-\mathbf{x}_0$ [26]. $\mathbf{f} \in \mathbb{R}^n$ is a nonlinear force term, and arbitrarily complex motion can be generated by adjusting $\mathbf{f}$. $\mathbf{f}$ is denoted as

$$\mathbf{f}(s) = \frac{\sum_{i=1}^{N} \boldsymbol{\omega}_i \varphi_i(s)}{\sum_{i=1}^{N} \varphi_i(s)} s, \tag{2}$$

where $\boldsymbol{\omega}_i \in \mathbb{R}^n$ is the weight of N basis functions $\varphi_i(s)$ and $\varphi_i(s)$ is a Gaussian function with its center located at c_i, and a width of h_i. Thus, $\varphi_i(s)$ can be written as

$$\varphi_i(s) = \exp(-h_i(s-c_i)). \tag{3}$$

The nonlinear force term $\mathbf{f}(s)$ does not depend directly on time; instead, it depends on the phase variable s, which is a time-independent canonical variable that changes monotonically from 1 to 0 during movement and can be expressed as

$$\tau\dot{s} = -\alpha s, \tag{4}$$

where α is a constant. In the trajectory learning stage, the speed and acceleration of each time step t are calculated according to the recorded motion trajectory; then, the value of the function $\mathbf{f}(t)$ is obtained with Equation (1). Next, Equation (4) is integrated, and $s(t)$ is evaluated. Finally, the weight $\boldsymbol{\omega}_i$ can be calculated using weighted linear regression when these time arrays are substituted into Equations (2) and (3). In the execution stage, by setting the starting position $\mathbf{x}_0$, using the learned weight $\boldsymbol{\omega}_i$, and solving Equation (1), a set of trajectories that is similar in shape to the demonstration trajectory and converge between $\mathbf{x}_0$ and $\mathbf{g}$ can be obtained.

However, to ensure the safety of the system, it is necessary to avoid any obstacles that may appear in the execution trajectory. Based on this consideration, a repulsive potential field is established around the obstacles, and the DMP method is combined with this potential field. We add a perturbation term $\mathbf{p}(\mathbf{x},\mathbf{v})$ to the attraction system Equation (1):

$$\begin{aligned} \tau\dot{\mathbf{v}} &= \mathbf{K}(\mathbf{g}-\mathbf{x}) - \mathbf{D}\mathbf{v} + diag(\mathbf{g}-\mathbf{x}_0)\mathbf{f} + \lambda\mathbf{p}(\mathbf{x},\mathbf{v}), \\ \tau\dot{\mathbf{x}} &= \mathbf{v}, \end{aligned} \quad , \tag{5}$$

where $\mathbf{p}(\mathbf{x},\mathbf{v})$ is the negative gradient of the potential field [27] and λ is a constant that indicates the strength of the entire field [10, 28]. The potential field depends on the relative position and velocity of the end effector with respect to the obstacle. The shape of the generated trajectory can be changed by adjusting the strength λ of this potential field. This field can be divided into a static potential field that considers only the distance between the current position and the obstacle and a dynamic potential field that also considers the position and velocity of the end effector [29]. Different potential field methods are used for point or point cloud obstacles and volume obstacles. We use the volumetric obstacle avoidance method with the proposed dynamic potential function [11] to calculate the initial value of the repulsive force, that is, $\mathbf{p}(\mathbf{x},\mathbf{v})$, in Equation (5).

For volume obstacles, to obtain isopotential contours that follow the object shape near the surface, an object may be surrounded with a superquadric potential:

$$C(\mathbf{x}) = \left(\left(\frac{x_1}{f_1(\mathbf{x})}\right)^{2n} + \left(\frac{x_2}{f_2(\mathbf{x})}\right)^{2n} \right)^{\frac{2m}{2n}} + \left(\frac{x_3}{f_3(\mathbf{x})}\right)^{2m} - 1. \tag{6}$$

The shape of any obstacle can be modeled by tuning the exponential parameters m and n and the scale functions f_1, f_2, and f_3. Then, we can define the dynamic potential

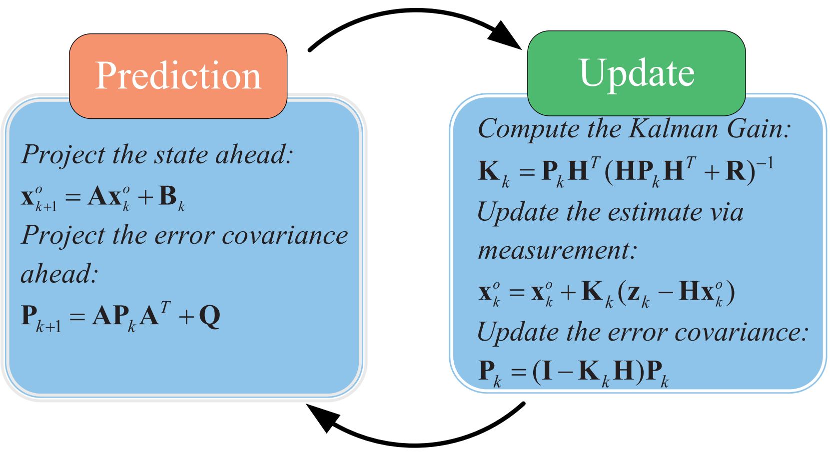

FIGURE 2 A diagram of the Kalman filter [Color figure can be viewed at wileyonlinelibrary.com]

function as follows:

$$\mathbf{U}(\mathbf{x},\mathbf{v}) = \begin{cases} (-\cos\theta)^{\beta}\frac{\|\mathbf{v}-\dot{\mathbf{o}}\|}{C^{\eta}(\mathbf{x})} & \theta \in (\frac{\pi}{2}, \pi] \\ 0 & \theta \in \left[0, \frac{\pi}{2}\right] \end{cases}, \tag{7}$$

where β and η are constant gains and $\mathbf{v}$ and $\dot{\mathbf{o}}$ are the velocities of the end effector and obstacle, respectively. For convex obstacles, the angle θ between the relative velocity $\mathbf{v}-\dot{\mathbf{o}}$ and the direction between the end effector's position $\mathbf{x}$ and the obstacle is defined as follows:

$$cos\theta = \frac{\langle \nabla_{\mathbf{x}}\mathbf{C}(\mathbf{x}), (\mathbf{v}-\dot{\mathbf{o}})\rangle}{\|\nabla_{\mathbf{x}}\mathbf{C}(\mathbf{x})\| \, \|\mathbf{v}-\dot{\mathbf{o}}\|}, \tag{8}$$

where $\nabla_{\mathbf{x}}\mathbf{C}(\mathbf{x})$ is the gradient of the isopotential $\mathbf{C}(\mathbf{x})$. Therefore, we can obtain the initial value of the repulsive force as $\mathbf{p}(\mathbf{x},\mathbf{v}) = -\nabla\mathbf{U}(\mathbf{x},\mathbf{v})$.

Moreover, $\lambda\mathbf{p}(\mathbf{x},\mathbf{v})$, a repulsive force, is an additional DMP term that is used to achieve obstacle avoidance. This term needs to be optimized to better avoid obstacles and improve the overall trajectory tracking performance. Therefore, we develop and applied an MPC algorithm to optimize the repulsive force generated by the APF. However, to evaluate the obstacle avoidance performance in the MPC prediction horizon, the states of the obstacles must be obtained. Thus, we revisit the Kalman filter algorithm to estimate the state of obstacles in the next section.

3.2 | Kalman filter for obstacle state estimation

For dynamic obstacles, we need to estimate the state of the obstacle in the next few moments, including the position and velocity information of the obstacle, to ensure that robots can respond to the future obstacle position in a timely manner. We construct the state vector of the obstacles as

$$\mathbf{x}^o = \left[\,\mathbf{o}\;\dot{\mathbf{o}}\,\right], \tag{9}$$

where $\mathbf{o}$ and $\dot{\mathbf{o}}$ are the position and velocity of the dynamic obstacle, respectively. Then, the Kalman filter model of obstacle dynamic trajectory prediction in discrete time form can be defined as follows [30]:

$$\begin{aligned} \mathbf{x}^o_{k+1} &= \mathbf{A}\mathbf{x}^o_k + \mathbf{B}\mathbf{w}_k, \\ \mathbf{z}_k &= \mathbf{H}\mathbf{x}^o_k + \delta_k, \end{aligned} \tag{10}$$

where the subscript $k+1$ represents the time step $k+1$. Equation (10) is the state equation and observation equation of the Kalman filter. $\mathbf{A}$ is the state transition matrix, which describes the obstacle movement rules. $\mathbf{B}$ is the control matrix, which describes the additional control of the control vector $\mathbf{w}_k$ on the movement of the obstacle. $\mathbf{z}_k$ represents the observation vector, which describes the observation value of the obstacle at time k. $\mathbf{H}$ and δ_k are the observation matrix and observation noise, respectively.

As a recursive estimation, the Kalman filter can be divided into two stages: prediction and update. These stages are shown in Figure 2, where $\mathbf{P}$ is the state error estimation covariance matrix, which describes the transmission of model errors throughout each generation and is updated during each iteration. $\mathbf{Q}$ is the process noise covariance matrix, which describes the uncertainty introduced by the prediction model. $\mathbf{K}$ is the Kalman gain, and $\mathbf{R}$ is the measurement noise matrix, which describes the uncertainty introduced by the measurement process. In the prediction phase, the filter estimates the state at the next moment according to the current state of the obstacle. In the update phase, the filter updates the Kalman gain, estimated value and error covariance of the estimated value according to observations of the current state to obtain a more accurate estimate. Through the prediction and update process shown in Figure 2, the state estimates of the obstacle at future times are obtained through continuous iterations.

3.3 | MPC for obstacle avoidance

As previously mentioned, the repulsive force $\lambda\mathbf{p}(\mathbf{x},\mathbf{v})$ generated by an APF is usually regarded as an additional term to accomplish obstacle avoidance. However, $\lambda\mathbf{p}(\mathbf{x},\mathbf{v})$ is applied as an input $\mathbf{u}$ to the MPC method in this section. The objective of the MPC method is to evaluate the performance and obtain the optimal input that satisfies the constraints. Collision avoidance is not only translated to control constraints but also designed as a cost function in MPC [22].

The DMP transformation system can easily be extended to higher dimensional space. Therefore, the DMP system can be reformulated as a state-space equation with the repulsive force serving as an input [31]. We can introduce the state vector

$$\boldsymbol{\eta} = \left[\mathbf{x}\ \mathbf{v}\right]^T. \tag{11}$$

Given the state vector formulated in Equation (11) and defining the repulsive term $\lambda\mathbf{p}(\mathbf{x},\mathbf{v})$ as an input $\mathbf{u}$, a nonlinear system that includes the DMP dynamics and the parameter dynamics can be formulated as follows:

$$\begin{aligned}\dot{\boldsymbol{\eta}} = \mathbf{g}\left(\boldsymbol{\eta}_k,\mathbf{u}_k\right) &= \begin{bmatrix} \frac{\mathbf{v}}{\tau} \\ \frac{\mathbf{K}(\mathbf{g}-\mathbf{x})-\mathbf{D}\mathbf{v}+(\mathbf{g}-\mathbf{x}_0)\mathbf{f}+\mathbf{u}}{\tau}\end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0} & \frac{\mathbf{I}}{\tau} \\ -\frac{\mathbf{K}}{\tau} & -\frac{\mathbf{D}}{\tau}\end{bmatrix}\boldsymbol{\eta}_k + \begin{bmatrix}\mathbf{0}\\ \mathbf{I}\end{bmatrix}\mathbf{u} + \begin{bmatrix}\mathbf{0} \\ \frac{\mathbf{K}\mathbf{g}+(\mathbf{g}-\mathbf{x}_0)\mathbf{f}}{\tau}\end{bmatrix}\end{aligned}, \tag{12}$$

where $\left[\,\mathbf{0}\ \left(\mathbf{K}\mathbf{g}+\left(\mathbf{g}-\mathbf{x}_0\right)\mathbf{f}\right)/\tau\,\right]^T$ is a constant term in the system. This term is fixed at each moment if the starting and goal positions remain constant. Furthermore, we do not need to consider the effect of this term because trajectory learning is accomplished, and the shape of the trajectory is determined. We adjust only the input $\mathbf{u}$ to achieve obstacle avoidance. In contrast to previous work, the input $\mathbf{u}$ is adjusted based on the repulsive force generated with the APF rather than exploring an optimal value in a large domain. For example, if the method is applied in n-dimensional space, $\mathbf{u}$ can be rewritten as

$$\mathbf{u} = \begin{bmatrix}\lambda^1 & & \\ & \ddots & \\ & & \lambda^n\end{bmatrix}\mathbf{p}\left(\mathbf{x},\mathbf{v}\right), \tag{13}$$

where $\lambda^1, \ldots, \lambda^n$ are regarded as the scale factors in each direction. Thus, both the size and the direction of the repulsive force can be optimized.

Most MPC methods are implemented in discrete time domain. For the time interval T, Equation (11) can be discretized as [20]

$$\boldsymbol{\eta}_{k+1} = \boldsymbol{\eta}_k + T\mathbf{g}\left(\boldsymbol{\eta}_k,\mathbf{u}_k\right) \stackrel{\Delta}{=} \mathbf{g}^{dt}\left(\boldsymbol{\eta}_k,\mathbf{u}_k\right). \tag{14}$$

The purpose of MPC is to compute the optimal control sequence $\left[\mathbf{u}_k^*\right]_{k=0}^{H-1}$, where H is the prediction horizon (prediction time steps). Given the initial state $\mathbf{x}_0$ of the prediction horizon, the optimal control sequence $\left[\mathbf{u}_k^*\right]_{k=0}^{H-1}$ in this horizon can be obtained according to the receding horizon principle, which can be solved as follows [17]:

$$\min J\left(\left[\boldsymbol{\eta}_k\right]_{k=0}^{H},\ \left[\mathbf{u}_k\right]_{k=0}^{H-1}\right), \tag{15}$$

subject to

$$\begin{aligned}\boldsymbol{\eta}_{k+1} &= \mathbf{g}^{dt}\left(\boldsymbol{\eta}_k,\mathbf{u}_k\right)\\ \lambda_{\min} \le \lambda_k^n &\le \lambda_{\max}\\ \|\mathbf{x}_k-\mathbf{o}_k\| &> \sigma\\ \forall k &\in \{0,\ldots,H\},\end{aligned} \tag{16}$$

where $J\left(\left[\boldsymbol{\eta}_k\right]_{k=0}^{H},\left[\mathbf{u}_k\right]_{k=0}^{H-1}\right)$ is the cost function in the prediction horizon. We can compute the optimal control sequence $\left[\mathbf{u}_k^*\right]_{k=0}^{H-1}$ by minimizing the cost J while satisfying the constraint in Equation (16). The constraint $\lambda_{\min} \le \lambda_k^n \le \lambda_{\max}$ determines the range of the input $\mathbf{u}_k$. Essentially, according to Equation (13), only $\lambda^1, \ldots, \lambda^n$ must be computed during this process. By using the position and velocity of the dynamic obstacle estimated by the Kalman filter, $\mathbf{p}(\mathbf{x},\mathbf{v})$ can be updated according to the system state. $\|\mathbf{x}_k-\mathbf{o}_k\| > \sigma$ ensures that no collisions occur; here, $\sigma > 0$ is a very small constant. To achieve better obstacle avoidance performance, it is necessary to add a penalty function to the cost function; thus, the cost function can be constructed as [32]

$$J = \phi\left(\tilde{\xi}_H\right) + \sum_{k=0}^{H-1}\left(L\left(\tilde{\xi}_k,\mathbf{u}_k,\Delta\mathbf{u}_k\right) + P_k^{obs}\right), \tag{17}$$

$$\phi\left(\tilde{\xi}_H\right) = \frac{1}{2}\tilde{\xi}_H^T\mathbf{P}^M\tilde{\xi}_H, \tag{18}$$

$$L\left(\tilde{\xi}_k,\mathbf{u}_k,\Delta\mathbf{u}_k\right) = \frac{1}{2}\tilde{\xi}_k^T\mathbf{Q}^M\tilde{\xi}_k + \frac{1}{2}\mathbf{u}_k^T\mathbf{R}^M\mathbf{u}_k + \frac{1}{2}\Delta\mathbf{u}_k^T\mathbf{T}^M\Delta\mathbf{u}_k, \tag{19}$$

where $\tilde{\xi}_k = \mathbf{x}_k^{des} - \mathbf{x}_k$ is the deviation between the actual position and the desired position at time step k, $\Delta\mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_{k-1}$ represents the difference between two adjacent inputs, and $\mathbf{P}^M$, $\mathbf{Q}^M$, $\mathbf{R}^M$, and $\mathbf{T}^M$ are constant positive definite weighting matrices. The cost function shown in Equation (17) can be considered a performance index for tracking a desired trajectory while minimizing the input effort and avoiding collisions with obstacles. Equation (18) and the first term in Equation (19) penalize the difference between the output trajectory and the desired trajectory along the horizon. The remaining terms in Equation (19) penalize the control input and the input deviation to reduce energy consumption and ensure a smooth input. P_k^{obs} in Equation (17) is the obstacle avoidance penalty in the prediction horizon. By constructing the cost function in this manner, the DMP-KMPC method can converge to the original motion quickly, despite the influence of various obstacles.

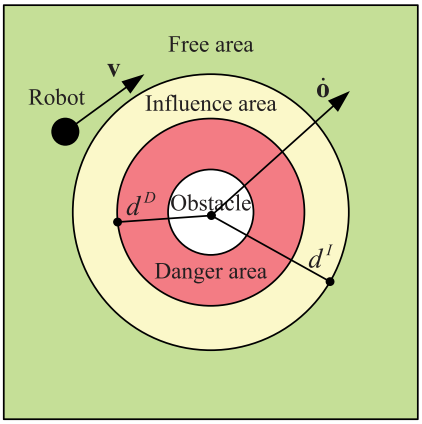

FIGURE 3 The definition of areas around the obstacle [Color figure can be viewed at wileyonlinelibrary.com]

An additional cost term P_k^{obs} is used to penalize the distance between the obstacles and the robot [33]. This term is usually defined as [32]

$$P_k^{obs} = \frac{\lambda^{obs} l_k^{obs}}{\|\mathbf{x}_k - \mathbf{o}_k\| + \varepsilon}, \tag{20}$$

where $\mathbf{x}_k$ and $\mathbf{o}_k$ are the positions of the robot and obstacle at time step k in the prediction horizon. As mentioned in the previous section, the position $\mathbf{o}_k$ of the dynamic obstacle is estimated with the Kalman filter method, and λ^{obs} and ε are constants, where ε ensures that the denominator is not equal to zero. l_k^{obs} considers the range of the APF and is defined as follows:

$$\begin{aligned} \delta_k &= d_{\max} - \|\mathbf{x}_k - \mathbf{o}_k\| \\ l_k^{obs} &= \begin{cases} 0, & \delta_k < 0 \\ 1, & \delta_k \geq 0 \end{cases} \end{aligned} . \tag{21}$$

$d_{\max}$ is the largest distance affected by the potential field of the dynamic obstacle. However, this simple definition for the obstacle penalty function cannot reflect the influence of obstacles on the motion of the robot in different situations. Therefore, we define different areas related to the distance between the robot and obstacle, as shown in Figure 3. Therefore, according to the definition $d_k = \|\mathbf{x}_k - \mathbf{o}_k\|$, Equation (20) can be modified as

$$\begin{aligned} P_k^{obs} &= \frac{\lambda^{obs}}{d_k + \varepsilon} \\ \lambda^{obs} &= \begin{cases} 0, & d_k > d^I \\ \lambda_1^{obs}, & d^D < d_k \leq d^I \\ \lambda_2^{obs}, & d_k \leq d^D \end{cases} \end{aligned} . \tag{22}$$

Here, different areas correspond to distinct values of λ^{obs}. λ_1^{obs} and λ_2^{obs} usually need to be set according to the specific situation to achieve better obstacle avoidance performance. Thus, we can change the values of λ_1^{obs} and λ_2^{obs} and adjust the range of the obstacle areas to obtain the optimal input in the prediction horizon and better obstacle avoidance performance.

Algorithm 1 A model predictive obstacle avoidance method based on dynamic motion primitives and a Kalman filter

Require:
The obstacle information: $\mathbf{x}^o$; demo trajectory: $\mathbf{x}_{des}$; start position: $\mathbf{x}_0$; goal position: $\mathbf{g}$; learned nonlinear force $\mathbf{f}$; prediction horizon: H; and scale factors for the repulsive force in each direction $\lambda^1, \cdots, \lambda^n \in [\lambda_{\min}, \lambda_{\max}]$

Ensure: $\mathbf{x}$

while $i \leq M$ **do**
  Substitute $\mathbf{x}_0$, $\mathbf{g}$, $\mathbf{f}$, τ, $\mathbf{K}$ and $\mathbf{D}$ to construct the state equation according to Equation (12)
  Obtain the current state $\mathbf{x}_i$, and previous obstacle state $\mathbf{x}_{0:i}^o$
  Estimate the state $\mathbf{x}_{i:i+H}^o$ of the dynamic obstacle in the prediction horizon H with the Kalman filter
  for $k = 0$ to $H - 1$ **do**
    Calculate the repulsive force $\mathbf{p}(\mathbf{x}_{i+k}, \mathbf{v}_{i+k})$ according to the estimated obstacle state $\mathbf{x}_{i+k}^o$ with Equations (6) and (8)
    Calculate the input $\mathbf{u}_{i+k} = diag(\lambda_{i+k}^1, \cdots, \lambda_{i+k}^n)\ \mathbf{p}(\mathbf{x}_{i+k}, \mathbf{v}_{i+k})$
    Substitute $\mathbf{u}_{i+k}$ into the state Equation (14) to calculate the next state $\mathbf{x}_{i+k+1}$
    Calculate the cost function J with Equation (17)
  end for
  Solve $\lambda^1, \cdots, \lambda^n$ by minimizing the cost function J to obtain the optimized input $\mathbf{u}_{i:i+H-1}^*$
  Substitute $\mathbf{u}_i^*$ into the state Equation (14) to calculate the next state $\mathbf{x}_{i+1}$
  i=i+1
end while

In summary, the DMP-KMPC method is based on an MPC optimization framework and realizes obstacle avoidance and improves the overall trajectory and tracking performance by optimizing additional DMP obstacle avoidance terms. The repulsive force generated by the APF is multiplied by an optimization coefficient and used as the input to the MPC system. The optimization of the repulsive force actually optimizes the coefficient multiplier matrix $diag(\lambda_1, \ldots, \lambda_n)$, which not only adjusts the size of the force but also changes the direction. In the prediction domain, the repulsive force is calculated by using the obstacle states estimated by the Kalman filter, and a penalty term related to obstacle avoidance is included in the cost function. By minimizing the defined cost function, the optimized input, namely, the optimized repulsive term, can be obtained. The presented method is summarized in Algorithm 1.

4 | SIMULATIONS AND EXPERIMENTS

To evaluate the proposed obstacle avoidance method, simulations in which a robot moved in two and three dimensions were carried out in Python. The physical experiments were verified in Gazebo with a real robot based on the Kinova MOVO robot experimental platform.

In contrast to other obstacle avoidance methods, the DMP-KMPC method focuses on improving the obstacle avoidance ability and the trajectory performance, which are enhanced by MPC and the Kalman filter. This method also allows the DMP to return to the original motion quickly after an obstacle is avoided. Therefore, the simulations and experiments do not compare the obstacle avoidance performance in different potential fields, which has been fully discussed in Zhang et al. [3] and Ginesi et al. [11]. The potential field employed in this work is a DMP-APF superquadratic potential field. In the next sections, the obstacle avoidance ability and trajectory tracking performance of the DMP-APF and DMP-KMPC methods under the same conditions are compared.

4.1 | Simulations

In this section, a robot performs the desired trajectory in 2D and 3D space while avoiding moving obstacles. Here, only the collision volume of the obstacle is considered, while the robot volume is ignored. The moving obstacles are circles and spheres in 2D and 3D space, respectively.

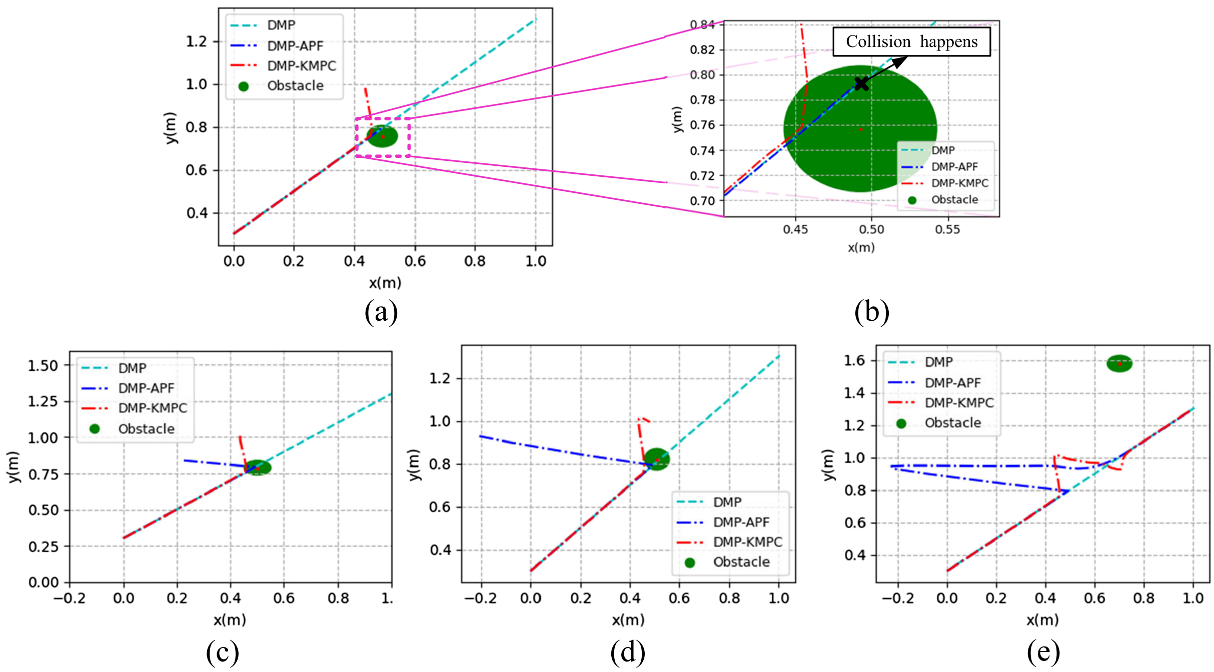

FIGURE 4 Comparison of obstacle avoidance trajectories in 2D space, where (a) and (c)–(e) are the positions of the obstacles and obstacle avoidance trajectories at different times and (b) is an enlarged view of the local details shown in (a) [Color figure can be viewed at wileyonlinelibrary.com]

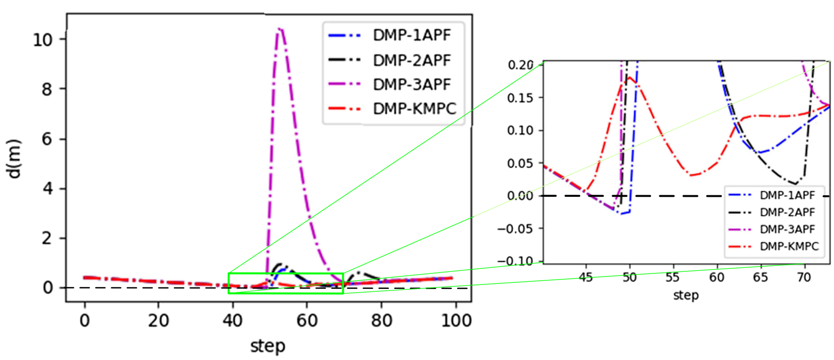

FIGURE 5 The distance to the dynamic obstacle at various time steps [Color figure can be viewed at wileyonlinelibrary.com]

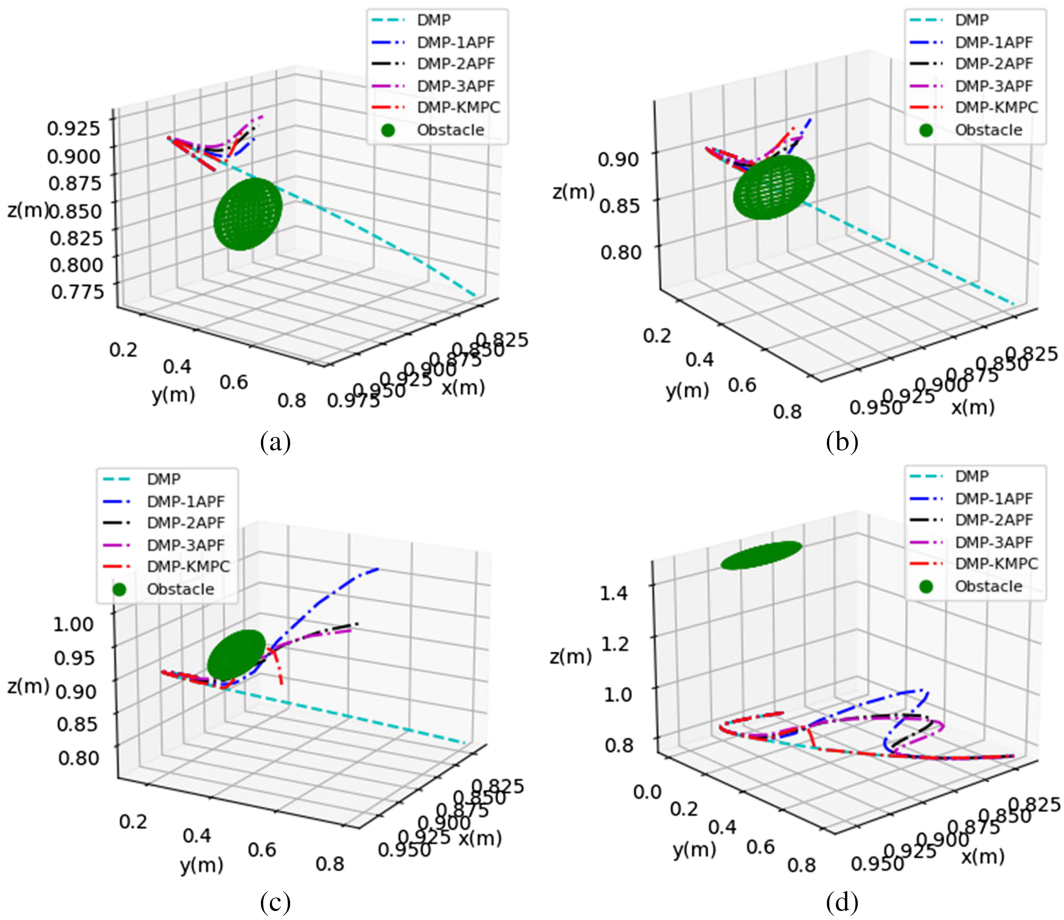

FIGURE 6 Comparison of obstacle avoidance trajectories in 3D space, where (a)–(d) are the positions of the obstacles and obstacle avoidance trajectories at different time steps [Color figure can be viewed at wileyonlinelibrary.com]

The next states (including the positions and velocities) of the obstacles are estimated by using the Kalman filter method based on the current observation information. The original trajectory of the robot is generated with a DMP, and the repulsive force generated by the APF is added as the acceleration term to change the trajectory shape and achieve obstacle avoidance. The repulsive force is optimized using MPC in the DMP-KMPC method. The MPC cost function is computed and updated according to the estimated obstacle states and robot states. By minimizing the cost function, the optimal input (repulsive force) can be obtained [34].

In the simulation in 2D space, the parameters of the DMP system are set as follows: $\mathbf{K} = 1050\mathbf{I}$, $\mathbf{D} = 2\sqrt{1050}\mathbf{I}$ and $\tau = 1$; here, $\mathbf{I}$ is the 2-D identity matrix. The start and goal positions of the desired trajectory are $\begin{bmatrix} 0 & 0.3 \end{bmatrix}^T m$ and $\begin{bmatrix} 1 & 1.3 \end{bmatrix}^T m$, respectively. The robot moves in a straight line from the starting position to the goal position. The dynamic obstacle is set as a circle with a radius of $r = 0.05m$, and the trajectory of the center of the circle is a line that starts at $\begin{bmatrix} 0.3 & 0 \end{bmatrix}^T m$. The state of the moving obstacle is estimated by using the Kalman filter method. For the

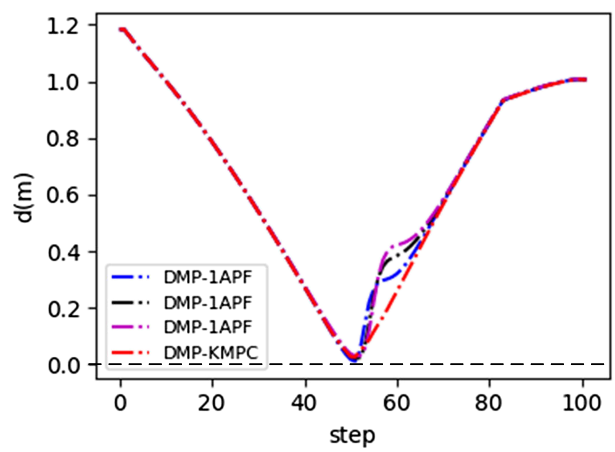

FIGURE 7 The distance to the dynamic obstacle at various time steps in 3D space [Color figure can be viewed at wileyonlinelibrary.com]

MPC part, the prediction horizon is set to $H = 5$, and the step size in the state equation, namely, Equation (14), is set to $T = 0.01s$ (this step size is applicable to the whole simulation whether it is 2D or 3D). In the cost function, we set $d^I = 0.3m$ and $d^D = 0.1m$. The range of the scale factors for the repulsive force in each direction is set as $\lambda^1, \ldots, \lambda^n \in [-1, 1]$. To compare the performance of the DMP-APF and DMP-KMPC methods, the strength of the potential field in the DMP-APF method is set as $\lambda = 1$. Figure 4 shows a comparison of the DMP-KMPC and DMP-APF methods in 2D space. The green circle in Figure 4 represents the obstacle, and the red point represents the center of the circle. We show some of dynamic obstacle avoidance moments in Figure 4a,c–e. Figure 4b shows an enlarged view of the partial details shown in Figure 4a, demonstrating a situation in which the DMP-APF method fails to avoid obstacles, while the DMP-KMPC method successfully avoids collisions. The DMP-KMPC method can avoid obstacles better than the DMP-APF method. To further verify this result, Figure 5 compares the distance to the obstacle between the DMP-APF method with different potential field strengths and the DMP-KMPC method. In Figure 5, the DMP-1APF, DMP-2APF, and DMP-3APF curves correspond to potential field strengths of $\lambda = 1$, $\lambda = 2$, and $\lambda = 3$, respectively. The robot clearly collides with the obstacle when the DMP-APF method with the potential field strength settings shown in Figure 5 is used. Although the obstacle avoidance ability of the DMP-APF method can be improved by increasing the potential field strength, this increased potential field strength is disadvantageous for tracking the desired trajectory. By using the proposed method, the repulsive force can be optimized by considering the possibility of collisions in the next few steps. Additionally, with the DMP-APF method, when obstacles are gradually approaching, the system produces sharp mutations to avoid collisions, as shown by the blue line in Figure 4d. This situation causes the generated trajectory to deviate greatly from the original trajectory. However, under the action of the defined obstacle avoidance cost function, the DMP-KMPC method can avoid such a situation, and the red line is noticeably smoother than the blue line. Therefore, the proposed method generated a smoother trajectory with less tracking errors while avoiding obstacles.

In the 3D space simulation, the DMP has the same parameter settings as in the 2D space simulation, with I expanded to a three-dimensional identification matrix. The trajectory starts at $\left[0.9\ 0.15\ 0.86\right] m$ and ends at the goal position $\left[0.82\ 0.8\ 0.76\right] m$. The demonstration trajectory is an irregular trajectory (the cyan curve in Figure 6). The moving obstacle is a sphere with a radius of $0.03m$. The center of the obstacle starts at $\left[1\ 1\ 0\right] m$ and moves at a certain speed, as shown by the green ball in

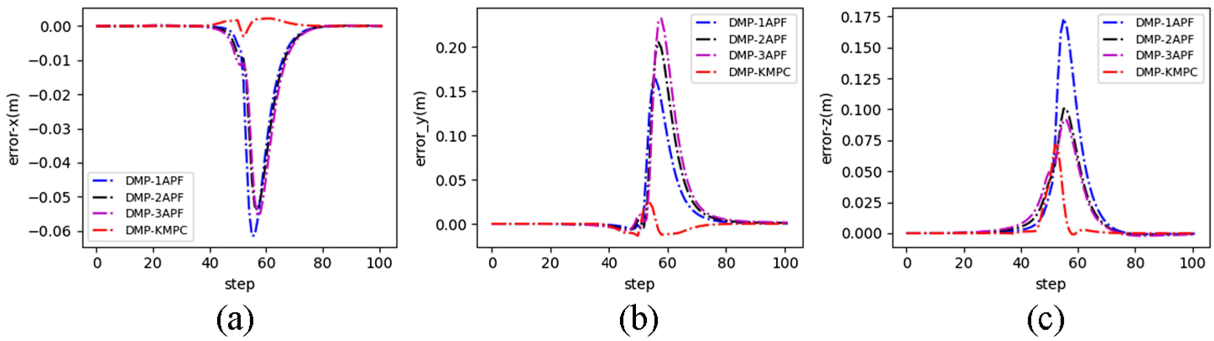

FIGURE 8 The deviation between the desired trajectory and the actual trajectory generated by the two methods in three directions [Color figure can be viewed at wileyonlinelibrary.com]

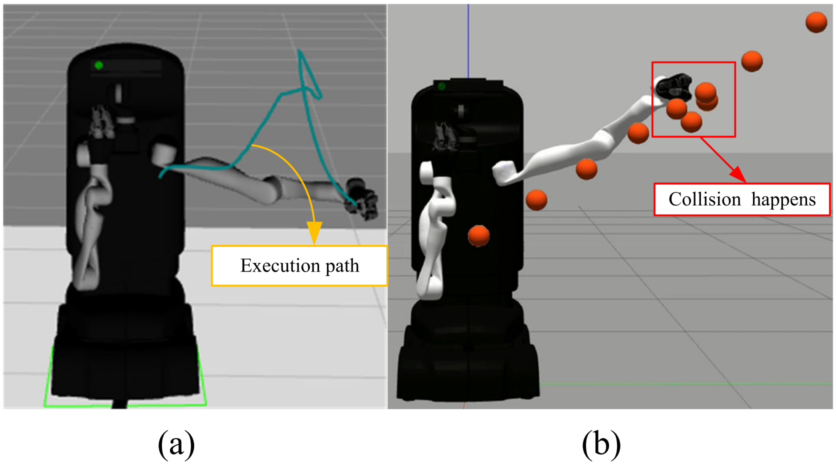

FIGURE 9 The execution trajectory recorded in RViz and the obstacle avoidance results in Gazebo with the DMP-APF method [Color figure can be viewed at wileyonlinelibrary.com]

Figure 6. The same parameters are set in the MPC part, including H, T, d^I, and d^D. We set $[\lambda_{\min}, \lambda_{\max}] = [-3, 3]$ in the DMP-KMPC method and set the strength of the potential as $\lambda = 1$, $\lambda = 2$ and $\lambda = 3$ in the DMP-APF method. As a result, it is guaranteed that the force produced by the DMP-KMPC method is not less than that produced by the DMP-APF method at the same position. Figure 6 shows the moments when the two methods avoid dynamic obstacles in 3D space, demonstrating a situation in which the DMP-KMPC and DMP-APF methods both successfully avoid obstacles. The distance between the robot and the moving obstacle is shown in Figure 7, and the minimum distance of all the trajectories is greater than 0. Although both methods avoid obstacles, the generated trajectories are completely different. Furthermore, the proposed method has better tracking performance and ensures that the DMP converges back to the original trajectory quickly. Figure 8 shows the deviation of the trajectory generated by the dynamic obstacle avoidance methods and the original DMP trajectory along the x, y, and z axes. The DMP-KMPC method has smaller trajectory deviations and a smoother trajectory. Moreover, by comparing the deviation curve, it can be seen that the two methods differ in the deviation direction, as shown in Figures 6d and 8a,b. Therefore, the proposed MPC-based method not only adjusts the obstacle avoidance offset but also changes the obstacle avoidance direction. By adjusting the direction of the repulsive force, more favorable motions can be achieved without reducing the trajectory tracking performance.

Overall, when compared with the simulation results of the DMP-APF method, the proposed method has a stronger obstacle avoidance ability and a better trajectory tracking performance in both 2D and 3D space. Our comparison considers two situations: failure and success in avoiding dynamic obstacles. As shown in Figures 4 and 5, when the DMP-APF method fails to avoid dynamic obstacles, the DMP-KMPC method successfully avoids dynamic obstacles. When both methods avoid obstacles, the proposed method is smoother and deviates less from the original trajectory, as shown in Figures 6 to 8. In addition, the proposed DMP-KMPC method optimizes the magnitude and direction of the repulsive force. Moreover, the robot can explore the most advantageous obstacle avoidance action by choosing the optimal input sequences among those in the prediction horizon.

4.2 | Experiments

In this section, we use physical experiments to verify that the proposed method has a stronger obstacle avoidance ability and a better tracking performance than the DMP-APF method. The settings of the experiments are similar to those of the simulations. However, in contrast to the simulations, if a collision occurs between an obstacle and a real robot, the robot may be damaged. Our collision experiments are carried out in Gazebo, and the obstacle used in these experiments is a virtual sphere instead of a real sphere. Furthermore, since the DMP is used only to learn the trajectory of the end-effector of the real robot in our work, this section conducts experiments with end-effector collisions only. The applied experimental platform is a Kinova MOVO robot, which is a dual-arm mobile manipulator. The arms of this robot include two 7-DOF manipulators.

In Gazebo and the real world, one arm of the Kinova MOVO robot is required to track the same task path as the 3D simulation while avoiding dynamic obstacles. The obstacle is a sphere with a radius of $r = 0.03m$ in both Gazebo and the real experiment, and the obstacle starts to move along a straight line beginning at position $\begin{bmatrix} 1 & -1 & 0 \end{bmatrix}^T$m. The states of the dynamic obstacle are set to default values for easy observation and collection, and

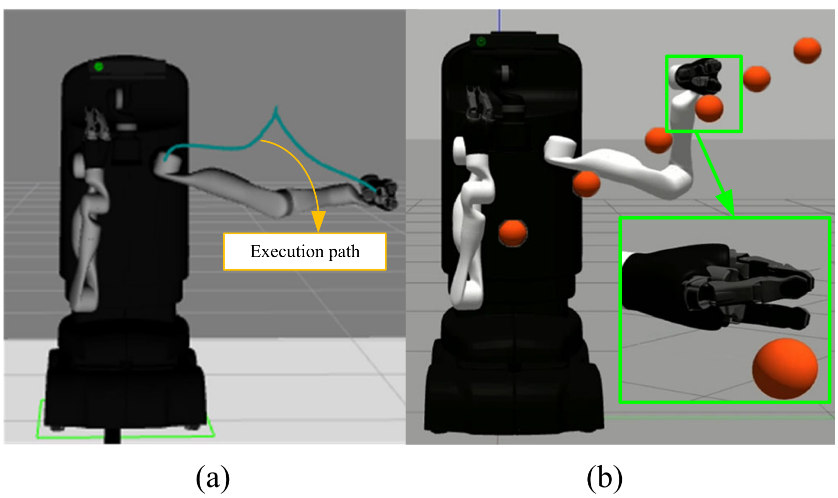

FIGURE 10 The execution trajectory recorded in RViz and the obstacle avoidance results in Gazebo with the DMP-KMPC method [Color figure can be viewed at wileyonlinelibrary.com]

the effects of sensors that collect visual information are not discussed in this work. In contrast to the simulations, the volume of the robot's end-effector and the volume of the obstacle must be considered in the physical experiment. Therefore, a situation in which a collision does not occur in the simulation might include a collision in the physical experiment shown in Figure 9b. Because there is no gravity setting for the obstacle in Gazebo and the obstacle has a constant velocity, the obstacle continues to move in the same direction after a collision. The actual execution trajectory generated with the DMP-APF method can be found in RViz, as shown in Figure 9a, and the potential field strength of the DMP-APF method is set as $\lambda = 3$. In the DMP-APF method, although the robot's end-effector is equivalent to a sphere with a radius of $0.02m$ and the influence range of the potential field is enlarged by $0.02m$, a collision still occurs in the Gazobo experiments, as shown in Figure 9b. As shown by the results in Figure 10, by setting $[\lambda_{\min}, \lambda_{\max}] = [-3, 3]$ in the DMP-KMPC method, the obstacle can be avoided in Gazebo, as shown in Figure 10b, and the actual execution trajectory is shown in Figure 10a. Therefore, the proposed method is further verified by the Gazebo results, demonstrating that the proposed method has a stronger ability to avoid obstacles than previous methods.

For situations in which both methods can avoid obstacles, we focus on comparing the tracking performance. Here, the Kinova MOVO robot performs the same task path as before in the real world with the DMP-APF and DMP-KMPC obstacle avoidance methods. The tracking performance of the DMP-KMPC method is compared with that of DMP-APF methods with different potential field strength settings in the simulation. The strength of the potential field in the two methods is set to the same level to compare the tracking performance in the physical experiment. For a case in which both methods can

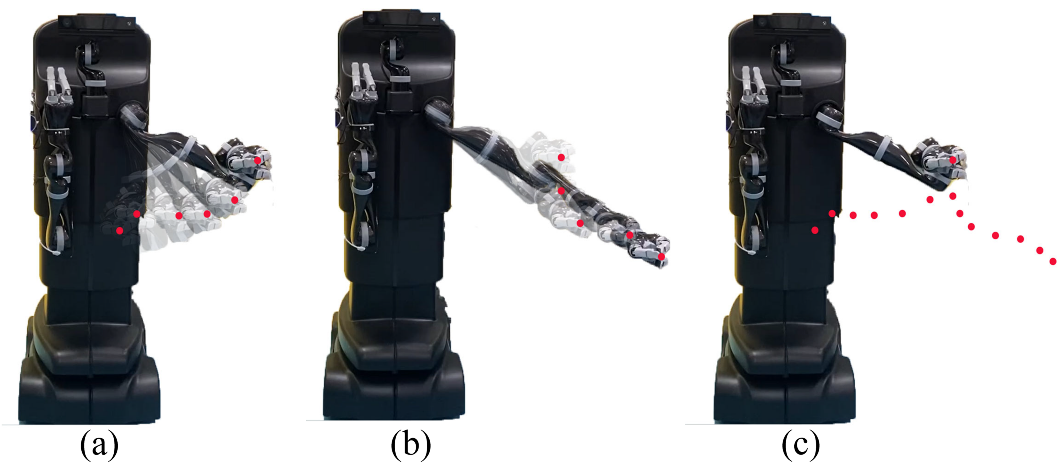

FIGURE 11 The execution trajectory of the physical robot with the DMP-KMPC method [Color figure can be viewed at wileyonlinelibrary.com]

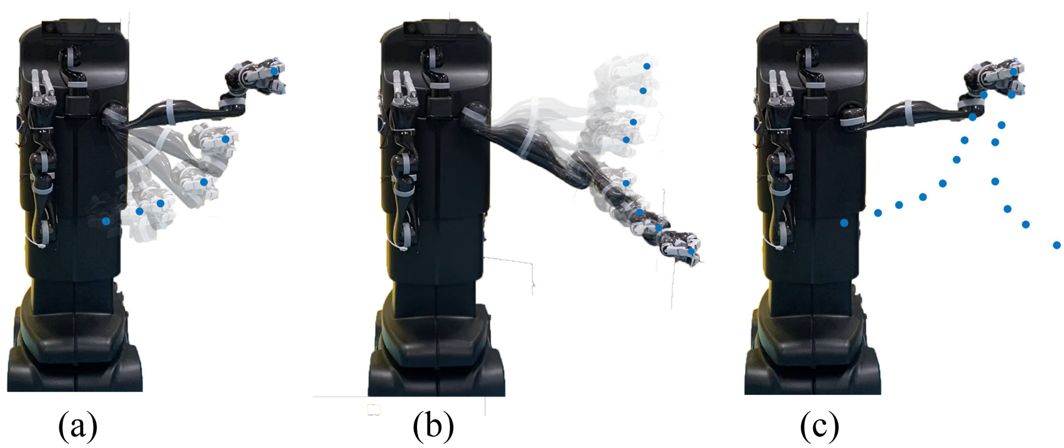

FIGURE 12 The execution trajectory of the physical robot with the DMP-APF method [Color figure can be viewed at wileyonlinelibrary.com]

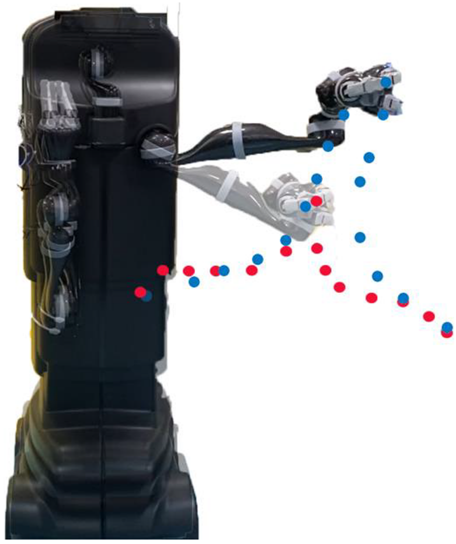

FIGURE 13 A comparison of the trajectories generated by the DMP-KMPC and DMP-APF methods in physical experiments [Color figure can be viewed at wileyonlinelibrary.com]

avoid obstacles, the actual execution trajectories of the robot generated by the DMP-KMPC and DMP-APF methods are shown in Figures 11 and 12. The red curve has a smaller peak than the blue curve in Figure 13. In the prediction horizon, the proposed method achieves better obstacle avoidance action planning, in contrast to the DMP-APF method, which adjusts the trajectory according to the acceleration generated by the fixed potential strength. In this case, the robot can optimize the obstacle avoidance action to achieve a better tracking performance.

The above results prove that the DMP-KMPC method has the same obstacle avoidance effect and advantages in real experiments as in the simulations. Regardless of whether the potential field strength is set to the same or different levels, the proposed method exhibits better obstacle avoidance and improved trajectory tracking performance. However, only end-effector collisions were considered in this section. For a DMP, it is easy to extend the training trajectory from the end effector to the joints of the robot. Similarly, the proposed method can be extended to achieve joint obstacle avoidance. Some of the experimental videos can be seen online (at https://youtu.be/bBdTuT5Mams).

5 | CONCLUSIONS

This paper proposes an obstacle avoidance method for robot manipulators that consists of a DMP, a Kalman filter, and an MPC component. In this work, the DMP acts as a framework for generating motion, the Kalman filter estimates the state of the obstacle and constructs the cost function, and the MPC component optimizes the additional DMP term to achieve better obstacle avoidance and trajectory tracking performance. By minimizing the defined cost function, the magnitude and direction of the repulsive force generated by the APF can both be optimized. As a result, the robot can select better actions, achieving obstacle avoidance and returning to the primitive motion after overcoming obstacles, resulting in a reduced tracking error. In conclusion, the use of an obstacle avoidance prediction approach combined with the Kalman filtering method and the DMP framework ensures a better obstacle avoidance effect in the prediction horizon, improving the overall obstacle avoidance performance throughout the whole movement process. The simulation and experimental results show the effectiveness, smoothness, and superiority of the proposed method.

ACKNOWLEDGMENT

The authors are grateful to the China Scholarship Council (202004910793).

CONFLICT OF INTEREST

The authors declare no potential conflict of interest.

AUTHOR CONTRIBUTIONS

Qinwen Li: conceptualization, methodology, software, validation, formal analysis, writing - original draft, writing-review and editing. **Zhiqian Wang:** writing-review and editing. **Wenrui Wang:** formal analysis, writing-review and editing. **Zhiyang Liu:** software, data curation. **Yiwen Chen:** visualization, software **Xianyao Ng:** visualization, software. **Marcelo H.:** supervision, project administration, resources, writing-review and editing.

ORCID

Qinwen Li https://orcid.org/0000-0003-2056-7437

REFERENCES

1. L. Zhang, Z. Lin, J. Wang, and B. He, *Rapidly-exploring Random Trees multi-robot map exploration under optimization framework*, Robot. Auton. Syst. **131** (2020), 103565. https://doi.org/10.1016/j.robot.2020.103565
2. A. H. Qureshi, J. Dong, A. Baig, and M. C. Yip, *Constrained motion planning networks x*, IEEE Trans. Robot. **34** (2021), no. 1, 1–19. https://doi.org/10.1109/TRO.2021.3096070
3. W. Zhang, H. Cheng, L. Hao, X. Li, M. Liu, and X. Gao, *An obstacle avoidance algorithm for robot manipulators based on decision-making force*, Robot. Comput.- Integr. Manuf. **71** (2021), 102114. https://doi.org/10.1016/j.rcim.2020.102114
4. W. Wang, M. Zhu, X. Wang, S. He, J. He, and Z. Xu, *An improved artificial potential field method of trajectory planning and obstacle avoidance for redundant manipulators*, Int. J. Adv. Robot. Syst. **9** (2018), 1–13. https://doi.org/10.1177/1729881418799562
5. F. Li, H. Peng, H. Yang, and Z. Kan, *A symplectic kinodynamic planning method for cable-driven tensegrity manipulators in a*

dynamic environment, Nonlinear Dynamics **106** (2021), no. 4, 2919–2941.
6. H. Peng, F. Li, J. Liu, and Z. Ju, *A symplectic instantaneous optimal control for robot trajectory tracking with differential-algebraic equation models*, IEEE Trans. Ind. Electron. **67** (2019), no. 5, 3819–3829.
7. P. Ruvolo, T. Wu, and J. R. Movellan, *Control by gradient collocation: applications to optimal obstacle avoidance and minimum torque control*, Proceedings of 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 1173–1179.
8. H. Xu, J. Zhuang, and Z. Zhu, *Global time-energy optimal planning of robot trajectories*, Proceedings of 2009 International Conference on Mechatronics and Automation, Vol. **2009**, 2009, pp. 4034–4039.
9. H. Hoffmann, P. Pastor, D. Park, and S. Schaal, *Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance*, 2009 IEEE International Conference on Robotics and Automation, 2009, pp. 2587–2592. https://doi.org/10.1109/ROBOT.2009.5152423
10. D. Park, H. Hoffmann, P. Pastor, and S. Schaal, *Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields*, Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots, 2008, pp. 91–98. https://doi.org/10.1109/ICHR.2008.4755937
11. M. Ginesi, D. Meli, A. Calanca, D. Dall'Alba, N. Sansonetto, and P. Fiorini, *Dynamic movement primitives: volumetric obstacle avoidance*, 2019 19th International Conference on Advanced Robotics (ICAR), 2019, pp. 234–239. https://doi.org/10.1109/ICAR46387.2019.8981552
12. D. Falanga, K. Kleber, and D. Scaramuzza, *Dynamic obstacle avoidance for quadrotors with event cameras*, SCI. Robot. **5** (2020), no. 40. https://doi.org/10.1126/scirobotics.aaz9712
13. M. Liu, H. Wu, J. Wang, and C. Wang, *Non-minimal state space model predictive control using Laguerre functions for reference tracking*, Asian J. Contr. **24** (2022), no. 1, 205–216. https://doi.org/10.1002/asjc.2441
14. A. Reiter, A. Müller, and H. Gattringer, *On higher order inverse kinematics methods in time-optimal trajectory planning for kinematically redundant manipulators*, IEEE T. Ind. Inform. **14** (2018), no. 4, 1681–1690. https://doi.org/10.1109/TII.2018.2792002
15. H. Zhang, H. Jin, Z. Liu, Y. Liu, Y. Zhu, and J. Zhao, *Real-time kinematic control for redundant manipulators in a time-varying environment: multiple-dynamic obstacle avoidance and fast tracking of a moving object*, IEEE T. Ind. Inform. **16** (2019), no. 1, 2841. https://doi.org/10.1109/TII.2019.2917392
16. M. W. Mueller and R. D'Andrea, *A model predictive controller for quadrocopter state interception*, European Control Conference (ECC), IEEE, 2013, pp. 11383–1389. https://doi.org/10.23919/ECC.2013.6669415
17. H. Andersen, J. Alonso-Mora, Y. H. Eng, D. Rus, and M. H. Ang, *Trajectory optimization and situational analysis framework for autonomous overtaking with visibility maximization*, IEEE T. Int. Veh. **5** (2019), no. 1, 7–20. https://doi.org/10.1109/TIV.2019.2955361
18. T. Nägeli, J. Alonso-Mora, A. Domahidi, D. Rus, and O. Hilliges, *Real-time motion planning for aerial videography with dynamic obstacle avoidance and viewpoint optimization*, IEEE Robot. Autom. Let. **2** (2017), no. 3, 1696–1703. https://doi.org/10.1109/LRA.2017.2665693
19. J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, *Combined speed and steering control in high-speed autonomous ground vehicles for obstacle avoidance using model predictive control*, IEEE T. Veh. Technol. **66** (2017), no. 10, 8746–8763. https://doi.org/10.1109/TVT.2017.2707076
20. J. M. Park, D. W. Kim, Y. S. Yoon, H. J. Kim, and K. S. Yi, *Obstacle avoidance of autonomous vehicles based on model predictive control*, P. I. Mech. Eng. D-J. Aut. **223** (2009), no. 12, 1499–1516. https://doi.org/10.1243/09544070JAUTO1149
21. Y. Yoon, J. Shin, H. J. Kim, Y. Park, and S. Sastry, *Model-predictive active steering and obstacle avoidance for autonomous ground vehicles*, Control Eng. Pract. **17** (2009), no. 7, 741–750. https://doi.org/10.1016/j.conengprac.2008.12.001
22. M. Wang, J. Luo, and U. Walter, *A non-linear model predictive controller with obstacle avoidance for a space robot*, Adv. Space. Res. **57** (2016), no. 8, 1737–1746. https://doi.org/10.1016/j.asr.2015.06.012
23. J. Guo, S. Zhang, J. Xu, and S. Zhou, *Kalman prediction based VFH of dynamic obstacle avoidance for intelligent vehicles*, 2010 International Conference on Computer Application and System Modeling (ICCASM), IEEE, 2010, pp. v3–6. https://doi.org/10.1109/ICCASM.2010.5620252
24. A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, *Dynamical movement primitives: learning attractor models for motor behaviors*, Neural. Comput. **25** (2013), no. 2, 328–373. https://doi.org/10.1162/NECO-a-00393
25. L. Koutras and Z. Doulgeri, *Dynamic movement primitives for moving goals with temporal scaling adaptation*, IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020, pp. 144–150. https://doi.org/10.1109/ICRA40945.20209196765
26. F. Stulp, E. Oztop, P. Pastor, M. Beetz, and S. Schaal, *Compact models of motor primitive variations for predictable reaching and obstacle avoidance*, Proceedings of 2009 9th IEEE-RAS International Conference on Humanoid Robots. IEEE, 2009, pp. 589–595. https://doi.org/10.1109/ICHR.2009.5379551
27. G. Liu, H. He, G. Tian, J. Zhang, and Z. Ji, *Online collision avoidance for human-robot collaborative interaction concerning safety and efficiency*, Proceedings of 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 2020, pp. 1667–1672. https://doi.org/10.1109/AIM43001.2020.9158647
28. L. Han, P. Kang, and Y. Chen, *Trajectory optimization and force control with modified dynamic movement primitives under curved surface constraints*, Proceedings of 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), IEEE, 2019, pp. 1065–1070. https://doi.org/10.1109/ROBIO49542.2019.8961446
29. D. H. Park, H. Hoffmann, P. Pastor, and S. Schaal, *Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields*, 2008-8th IEEE-RAS International Conference on Humanoid Robots. IEEE, 2008, pp. 91–98. https://doi.org/10.1109/ICHR.2008.4755937
30. Y. Huang, Y. Zhang, Z. Wu, N. Li, and J. Chambers, *A novel adaptive Kalman filter with inaccurate process and measurement noise covariance matrices*, IEEE T. Automat. Comtr. **63** (2017), no. 2, 594–601. https://doi.org/10.1109/TAC.2017.2730480
31. D. Widmann and Y. Karayiannidis, *Human motion prediction in human-robot handovers based on dynamic movement primitives*, European Control Conference (ECC). IEEE, 2018, pp. 2781–2787. https://doi.org/10.23919/ECC.2018.8550170

32. Y. Yoon, H. J. Kim, J. H. Shin, T. Choe, and Y. Park, *Communication in distributed model predictive collision avoidance*, Proceedings of the 1st international conference on Robot communication and coordination, 2007, pp. 1–4. https://doi.org/10.5555/1377868.1377921
33. Y. J. Yoo, K. J. Oh, Y. J. Choi, and S. C. Won, *Obstacle avoidance for redundant manipulator without information of the joint angles*, Proceedings of 2013 9th Asian Control Conference (ASCC). IEEE, 2013, pp. 1–6. https://doi.org/10.1109/ASCC.2013.6606009
34. A. Altan and R. Hacıoğlu, *Model predictive control of three-axis gimbal system mounted on UAV for real-time target tracking under external disturbances*, Mech. Syst. Signal. Pr. **138** (2020), 106548. https://doi.org/10.1016/j.ymssp.2019.106548

AUTHOR BIOGRAPHIES

Qinwen Li was born in Shandong, China, in 1993. She received a BE degree from Shandong University, China, in 2016. She has been pursuing a PhD degree at the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, China, since 2016. She is currently a nongraduating student in mechanical engineering at the National University of Singapore. Her research interests include robotics and vision measurement.

Zhiqian Wang was born in Jilin, China, in 1969. He received a PhD degrees from Jilin University, China, in 2009. From 2000 to 2010, he was an associate professor at the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, China. Since 2010, he has been with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, where he is currently a professor in the Department of Optical-Electronic Measurement and a supervisor to several PhD candidates. His main research interests include photoelectric measurements and system control.

Wenrui Wang was born in Jilin, China, in 1993. He received a BE degree from Jilin University, China, in 2016. He has been pursuing a PhD degree at the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, China, since 2016. He is currently a nongraduating student in mechanical engineering at the National University of Singapore. His research interests include kinematic and dynamic control of robots and learning control.

Zhiyang Liu received a BS degree in electrical engineering and automation from Soochow University, Soochow, China, in 2019, and an MS degree in mechanical engineering from the National University of Singapore, Singapore, in 2020. He is currently pursuing a PhD degree in the Department of Mechanical Engineering, National University of Singapore, Singapore. He is also with the Advanced Robotics Centre, National University of Singapore. His research interests include robotics, autonomous vehicles, and intelligent systems.

Yiwen Chen received a BS degree in mechanical engineering from Chongqing University, Chongqing, China, in 2019. He received an MS degree from Mechanical Engineering, National University of Singapore in 2020. He is now pursuing a PhD degree in NUS with the topics of reinforcement learning, strategies, and robotics learning.

Xianyao Ng received the BS degree in mechatronics engineering from the University of New South Wales, Sydney, Australia, in 2016. He is currently pursuing the PhD degree with the School of Mechanical Engineering, National University of Singapore, Singapore. His research interests include robotic perception and computational fabrication.

Marcelo H. Ang Jr. is a professor in the Department of Mechanical Engineering, National University of Singapore. He received the BS degrees in Mechanical Engineering and Industrial Management Engineering from the De La Salle University, Manila, Philippines, in 1981; the MS degree in Mechanical Engineering from the University of Hawaii at Manoa, Honolulu, Hawaii, in 1985; and the MS and PhD degrees in Electrical Engineering from the University of Rochester, Rochester, New York, in 1986 and 1988, respectively. Now, Dr. Ang works in the Department of Mechanical Engineering of the National University of Singapore, where he is currently a professor. His research interests include robotics, mechatronics, and applications of intelligent systems methodologies.

How to cite this article: Q. Li, Z. Wang, W. Wang, et al. *A model predictive obstacle avoidance method based on dynamic motion primitives and a Kalman filter*, Asian J. Control, (2022), 1–16. https://doi.org/10.1002/asjc.2946