

Research Article

A Hierarchical Structure Control Strategy Based on MPC for a Six-DOF Flexible Joint Manipulator

Ningbo Jing,¹ Ming Bu,² Qi Ni,³ Hongguang Pan ¹, Xuebin Qin ¹ and Xianmin Ma¹

¹College of Electrical and Control Engineering, Xi'an University of Science and Technology, Xi'an 710 054, China

²School of Electrical Engineering, Xi'an Jiaotong University, Xi'an 710 049, China

³Shaanxiety Regional Electric Power Group Co. Ltd., Baoji 721 006, China

Correspondence should be addressed to Hongguang Pan; hongguangpan@163.com

Received 30 July 2021; Accepted 19 November 2021; Published 24 December 2021

Academic Editor: Ali Ahmadian

Copyright © 2021 Ningbo Jing et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The six-degree-of-freedom flexible joint manipulator is a complex system that suffers from the problem that the trajectory planning results are inconsistent with the control results. To keep the planned trajectory within the control range of the manipulator, a hierarchical structure control strategy is designed, which consists of a trajectory planning layer, a model predictive control layer, and a bottom control layer. Specifically, first, the target joint angles are obtained by a time-optimal trajectory planning algorithm based on a genetic algorithm in the trajectory planning layer. Second, in the model predictive control layer, considering the system physical constraints, the model predictive controller is adopted to provide the set points for the Proportion-Differentiation (PD) controllers. Finally, in the bottom control layer, the manipulator moves along the target trajectory under the PD controllers with the feedback control law. The simulation results show that, compared with the PD control strategy, the hierarchical structure control strategy can achieve better control performance and reduce the tracking error of the terminal trajectory by 33.70%.

1. Introduction

A flexible joint manipulator is light and small and consumes little energy, making this technology widely used in aerospace, medical, industrial assembly, and other fields [1, 2]. In practical applications, to meet different mission requirements, it is necessary to optimize the trajectory of the manipulator and implement precise control. Many achievements have been made in the trajectory planning and control implementation of manipulator systems [3, 4]. However, the existing results of manipulator control do not consider the consistency between the trajectory optimization results and the control results; that is, the trajectory optimization results may not be feasible in the control implementation.

The “hierarchical structure” approach is mainly used to deal with “multiperiod, multiobjective, multilevel model and optimization” complex systems in industrial production [5, 6]. In the hierarchical structure, each level has its own

division of labour, which makes complex problems easy to deal with [7]. The 6-DOF flexible joint manipulator control strategy is designed for industrial hierarchical structure systems [8]. Figure 1 compares the traditional control structure in the process industry and the hierarchical control structure designed in this paper (where FC, PC, TC, and LC represent flow, pressure, temperature, and liquid level control, resp.). The right portion indicates the basic components of the hierarchical structure, which consist of the trajectory planning layer, the MPC layer, and the bottom control layer. Compared with the existing hierarchical structure in the industry, the main function of the trajectory planning layer is to obtain the target trajectory, the MPC layer makes the terminal trajectory of the manipulator approximate the target trajectory, and the bottom control layer implements the final control.

Trajectory planning is the basis of manipulator motion control. When the trajectory planning of a manipulator is carried out, it is usually possible to maximally optimize some

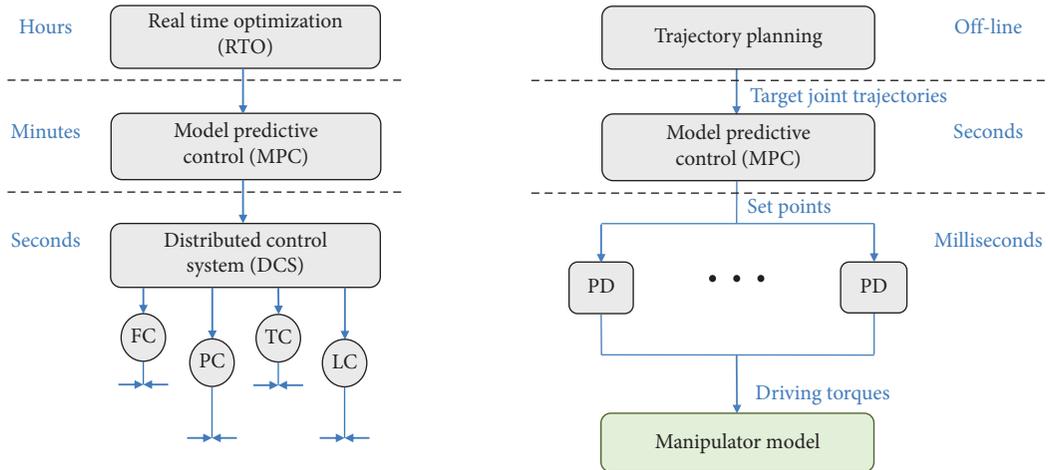


FIGURE 1: Comparison of hierarchical control structures.

of the indicators while satisfying the given constraints. There are many research results in trajectory planning, among which the time-optimal trajectory planning algorithm of manipulators is studied more [9, 10]. The position, velocity, acceleration, second-order acceleration, and other motion constraints of the manipulator are taken as the premise, and a higher-order polynomial is taken as the curve path between the path points [11]. However, the calculation of higher degree polynomials is complex. Bazaz et al. use the time-optimal trajectory planning method based on cubic spline interpolation, which is widely used and simple to calculate [12]. Because of the low order of the cubic spline interpolation function, the acceleration of the manipulator is discontinuous. The 3-5-3 polynomial interpolation is used to plan the trajectory of a manipulator [13]. Time-optimal trajectory planning is a nonlinear dynamic optimization problem. The constraints are generally nonlinear, and the calculation process is complex, and the calculation amount is large. For the optimization of such problems, 3-5-3 polynomial interpolation has no optimization attribute. It can be solved by an intelligent optimization algorithm, such as particle swarm optimization algorithm, artificial neural network algorithm, ant colony algorithm, and simulated annealing algorithm. These algorithms have advantages and disadvantages in the application of the robot trajectory optimization process [14–16]. At present, the genetic algorithm is the most possible general algorithm for trajectory planning and optimization. For example, L. Tian et al. propose the GA method, which is an effective optimization method for the trajectory planning of robot manipulators. The GA method can find an optimal trajectory for the robot even in a complicated environment [16]. Therefore, in this chapter, the optimal time in the process of manipulator motion is taken as the objective function, the path and the maximum angular velocity are taken as the constraints, and the genetic algorithm is used to optimize the interpolation time and time. The optimal trajectory planning method is designed.

The flexible joint manipulator is a multivariable, highly nonlinear complex system with strong coupling, which has

uncertainties, including parameter perturbation and external interference. The complexity of the system makes designing a control strategy difficult. Therefore, the problem of manipulator motion control has become an area of active research in recent years. PID control has been widely used in the control of manipulators because the approach is simple, flexible, and independent of the system model and shows other advantages [17, 18]. However, with the increasingly complex structure of manipulators, the PID controller method has been unable to meet the ever-increasing demand for control performance [19–21]. Therefore, a more efficient control strategy needs to be designed. Considering that the multidegree-of-freedom (DOF) manipulator is composed of several joints, model predictive control (MPC) can be applied to manipulator control [22]. For example, a real-time motion control strategy for the manipulator based on the constraint MPC is proposed in [23], which can control the achieved target position under the condition of physical constraints. A robust hierarchical multiloop method is proposed in [24]. The sliding mode control and MPC are used to design the internal and external loops, respectively. The optimal evolution of the manipulator control trajectory is guaranteed. The effectiveness of three control strategies based on sliding mode control, torque control, and MPC is compared, and MPC has the best control effect [25]. In general, considering the advantages of MPC, such as explicit constraints and flexible design of objective functions, it is obvious that its application in manipulator control is expanding.

However, the MPC approach requires much computation and yields a slow response speed, and it is not easy to control the multi-DOF manipulator. A PD controller has good dynamic performance and a faster response speed than the model predictive controller. Therefore, in the hierarchical structure control strategy designed in this paper, the set points obtained by the model predictive controller are sent to the PD controller.

To summarize, the hierarchical structure control strategy, that is, a control strategy of “trajectory planning layer + MPC layer + bottom control layer” for the 6-DOF

flexible joint manipulator, is designed. Among the layers, the trajectory planning layer is designed to obtain the target trajectory based on a GA, the MPC layer makes the terminal trajectory of the manipulator approach the target trajectory, and the bottom control layer realizes the final control.

The main contributions of this paper are as follows: (1) based on the appropriate assumptions in [26], a 6-DOF flexible joint model is established, and a dynamics model of a flexible joint manipulator is built based on a simplification of the Lagrange equation; (2) referring to the industrial hierarchical structure system, a hierarchical structure control strategy of a 6-DOF flexible joint manipulator is designed; and (3) the control strategy, which includes the trajectory planning layer, the MPC layer, and the bottom control layer, realizes stable operation of the manipulator under the optimal trajectory.

The organization of this paper is as follows. In Section 2, the dynamics model and state space model of the flexible joint manipulator are established. In Section 3, a time-optimal 3-5-3 polynomial interpolation trajectory planning method based on a GA is designed in the trajectory planning layer. Two controllers, the model predictive controller in the MPC layer and the PD controller in the bottom control layer, are designed in Section 4. In Section 5, the hierarchical structure control strategy is tested and analyzed, and Section 6 gives the conclusions of this paper.

2. The 6-DOF Flexible Joint Manipulator Model

The dynamics of the manipulator are very important in the control of manipulator. The dynamic model of the manipulator is the premise of the high-performance control strategy design. In this chapter, the Lagrange method is used to establish the dynamic equation of the manipulator.

2.1. Dynamics Modelling. The manipulator joint is composed of a power unit (motor), a transmission (reducer), a shaft, and a sensor. The model of the flexible link manipulator is complex, and the influence of flexibility on the motion performance is small. Therefore, the link flexibility is ignored, and the subject is simplified as a flexible joint manipulator. Spong suggested that the spring stiffness coefficient be used to describe the characteristics of flexible joints [26]. For the flexible joint in Figure 2, the following two assumptions are made during the modelling process.

Assumption 1. The link flexibility is ignored, and only the joint flexibility is considered.

Assumption 2. A flexible joint is a linear spring between the motor rotor and a link.

The relationship (Figure 2) between joint torque and stiffness coefficient can be stated as follows:

$$\tau_i = K_i(\theta_i - q_i), \quad (1)$$

where $i = 1, 2, \dots, 6$ and the definitions of $\tau_i, K_i, q_i,$ and θ_i are shown in Table 1. According to Assumptions 1 and 2, the

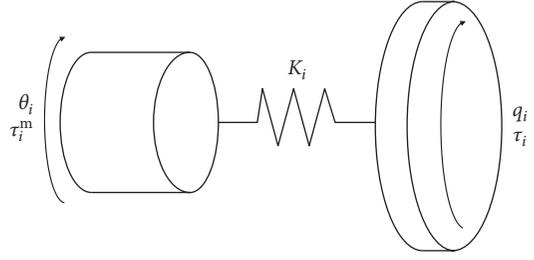


FIGURE 2: The model of the i -th flexible joint in the manipulator.

TABLE 1: Notations.

Notation	Meaning
q	Link angle vector, $q \in \mathbb{R}^{6 \times 1}$
\dot{q}	Link velocity vector, $\dot{q} \in \mathbb{R}^{6 \times 1}$
\ddot{q}	Link acceleration vector, $\ddot{q} \in \mathbb{R}^{6 \times 1}$
q_i	i -th link angle
θ	Joint angle vector, $\theta \in \mathbb{R}^{6 \times 1}$
$\ddot{\theta}$	Joint acceleration vector, $\ddot{\theta} \in \mathbb{R}^{6 \times 1}$
θ_i	i -th joint angle
τ^m	Joint driving torque vector, $\tau^m \in \mathbb{R}^{6 \times 1}$
τ_i	Torque acting on the i -th link
$H(q)$	Inertia matrix for the rigid links, $H = H^T, H \in \mathbb{R}^{6 \times 6}$
$F(q, \dot{q})\dot{q}$	Torque vector caused by Coriolis force and centrifugal forces, $F \in \mathbb{R}^{6 \times 6}$
$G(q)$	Torque vector caused by gravity, $G \in \mathbb{R}^{6 \times 1}$
J	Diagonal matrix of actuator inertias reflected to the link side of gears, $J \in \mathbb{R}^{6 \times 6}$
K	Diagonal matrix containing the joint stiffness coefficients, $K \in \mathbb{R}^{6 \times 6}$
K_i	i -th joint stiffness coefficients

dynamics model of a flexible joint manipulator is based on a simplification of the Lagrange equation

$$H(q)\ddot{q} + F(q, \dot{q})\dot{q} + G(q) = K(\theta - q), \quad (2a)$$

$$J\ddot{\theta} + K(\theta - q) = \tau^m, \quad (2b)$$

where (2a) is the dynamic equation for a rigid joint and (2b) is the mathematical model for a flexible joint. The meanings of the other notations are also listed in Table 1.

2.2. State Space Model. The dynamic equation (2) is a nonlinear strongly coupled multivariable equation, for which it is difficult to establish a state space equation. Therefore, it is necessary to linearize and discretize the equation to obtain the discrete state space equation. A Taylor expansion is used to linearize the input and output.

The state variable of the manipulator is defined as $x = [\theta, \dot{\theta}]^T$, which can be obtained by a Taylor expansion:

$$x(t + \Delta t) = \begin{bmatrix} \theta(t) + \Delta t \dot{\theta}(t) + \frac{\Delta t^2}{2} \ddot{\theta} \\ \dot{\theta}(t) + \Delta t \ddot{\theta} \end{bmatrix}. \quad (3)$$

According to (2), the relationship between the acceleration $\ddot{\theta}$ of the joint and the state variables of the manipulator can be obtained:

$$\ddot{\theta} = J^{-1}(\tau^m - H(q)\ddot{q} - C(q, \dot{q})\dot{q} - G(q)). \quad (4)$$

The state variable of the manipulator at the current moment is $x(n)$, and the sampling period is T . Each component of the state variable is discretized by a Taylor expansion, and the discrete state space equation can be obtained:

$$x(n+1) = x(t + (n+1)T) = Ax(n) + B\tau^m(n) + G_p, \quad (5a)$$

$$y(n) = Cx(n), \quad (5b)$$

where A , B , G_p , and C are matrices related to current state variables.

$$A = \begin{bmatrix} I_6 & TI_6 \\ O_{6*6} & I_6 \end{bmatrix}, \quad B = \begin{bmatrix} (T^2/2)^{-1} \\ TM^{-1} \end{bmatrix},$$

$$G_p = \begin{bmatrix} -T^2/2M^{-1}(H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q)) - TJ^{-1}(H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q)) \\ \end{bmatrix}, \text{ and } C = \begin{bmatrix} I_6 & O_{6*6} \end{bmatrix}. \text{ Equation (5) is the discrete state space model of the manipulator.}$$

3. Trajectory Planning

The control strategy of the 6-DOF manipulator designed in this paper adopts a hierarchical structure, that is, "trajectory planning layer + MPC layer + bottom control layer." In this structure, the role of the trajectory planning layer is to obtain the target trajectory through an intelligent algorithm. According to the requirements of the task, the trajectory planning of the manipulator can calculate the target trajectory and convert it into the parameters of the manipulator joint space, which lays the foundation for the accurate and stable execution of the task. In this section, a time-optimal trajectory planning method based on 3-5-3 polynomial

interpolation is designed, which uses a GA to optimize each interpolation time.

3.1. Construction of the 3-5-3 Polynomial Function. Given the spatial coordinates of the starting point, two trajectory points, and the terminal point of the manipulator in the rectangular coordinate system, the joint angles of each joint at the four interpolation points can be obtained by solving the inverse kinematics. ϕ_{ij} is the angle of i -th joint interpolation, where $i = 1, 2, 3, 4, 5, 6$ represents the joint number and $j = 0, 1, 2, 3$ represents the ordinal number of the four interpolation points. The constraints for trajectory planning are as follows: the given initial points ϕ_{i0} , ϕ_{i1} , and ϕ_{i2} are path points, the terminal point is ϕ_{i3} , the position between path points is given, the continuity of the velocity and acceleration is given, and the velocity and acceleration at the starting and ending points are 0. A 3-5-3 polynomial interpolation curve trajectory is adopted between points, and the general formula of the 3-5-3 polynomial of the segment i -th joint is

$$h_{i1}(t) = a_{i13}t^3 + a_{i12}t^2 + a_{i11}t^1 + a_{i10}, \quad (6a)$$

$$h_{i2}(t) = a_{i25}t^5 + a_{i24}t^4 + a_{i23}t^3 + a_{i22}t^2 + a_{i21}t^1 + a_{i20}, \quad (6b)$$

$$h_{i3}(t) = a_{i33}t^3 + a_{i32}t^2 + a_{i31}t^1 + a_{i30}, \quad (6c)$$

where $h_{i1}(t)$, $h_{i2}(t)$, and $h_{i3}(t)$ are the first cubic polynomial trajectories, the second polynomial trajectories, and the third joint cubic polynomial trajectories. The unknown coefficients a_{i1j} , a_{i2j} , and a_{i3j} are the j -th coefficients of the interpolation function of the i -th joint trajectory.

According to the above conditions, the unknown coefficients a_{ij} can be deduced and solved. t_i in (7) represents the interpolation time of the polynomial of segment 3 of the i -th joint, D is only related to t , and

$$D = \begin{bmatrix} t_1^3 & t_1^2 & t_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3t_1^2 & 2t_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6t_1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & t_2^5 & t_2^4 & t_2^3 & t_2^2 & t_2 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 5t_2^4 & 4t_2^3 & 3t_2^2 & 2t_2 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 20t_2^3 & 12t_2^2 & 6t_2 & 2 & 0 & 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_3^3 & t_3^2 & t_3 & 1 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3t_3^2 & 2t_3 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6t_3 & 2 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (7)$$

Equation (8) is the position matrix of the i -th joint angle,

$$a = [a_{i13}, a_{i12}, a_{i11}, a_{i10}, a_{i25}, a_{i24}, a_{i23}, a_{i22}, a_{i21}, a_{i20}, a_{i33}, a_{i32}, a_{i31}, a_{i30}]^T = \text{inv}(D) * b. \quad (8)$$

Equation (9) is the solution of unknown coefficients a_{ij} ,

$$b = [0, 0, 0, 0, 0, 0, \phi_{i3}, 0, 0, \phi_{i0}, 0, 0, \phi_{i2}, \phi_{i1}]^T. \quad (9)$$

3.2. Trajectory Planning Based on the GA. A method of time-optimal trajectory planning is designed that takes the optimal time in the working process of the manipulator as the objective function and the path and maximum angular velocity as the constraints. The time-optimal trajectory planning problem is a nonlinear dynamic optimization problem whose constraints are generally nonlinear, the calculation process is complex, and the calculation amount is large. For optimization of these kinds of problems, the 3-5-3 polynomial interpolation has no optimization property and can be solved by an intelligent optimization algorithm. In this paper, a time-optimal trajectory planning method that uses a GA to optimize each interpolation time is designed.

The optimization objective is that each joint runs in the shortest time within the scope of the kinematic constraints, and its fitness function is

$$f(t) = \min \sum_{i=1}^6 (t_{i1} + t_{i2} + t_{i3}), \quad (10a)$$

$$\text{s.t. (1), (2),} \quad (10b)$$

$$\max |V_{i1}| \leq V_{\max}, \quad (10c)$$

$$\max |V_{i2}| \leq V_{\max}, \quad (10d)$$

$$\max |V_{i3}| \leq V_{\max}, \quad (10e)$$

where $i = 1, 2, \dots, 6$ and V_{i1} , V_{i2} , and V_{i3} are the velocities of each polynomial of the i -th joint changing with time. If each joint is optimized separately, then the optimization objective function of the i -th joint is

$$f_i(t) = \min (t_{i1} + t_{i2} + t_{i3}), \quad (11)$$

$$\max |V_i| \leq V_{i\max}, \quad (12)$$

where V_i and $V_{i\max}$ are the velocity and maximum limit velocity of the i -th joint, respectively.

3.3. Trajectory Planning Algorithm. How to select independent variables to obtain the optimal time is the key problem to solve for the 3-5-3 polynomial interpolation function using a GA. In (7) to (9), times t_1 , t_2 , and t_3 are unknown variables to be optimized, and the coefficients a_{ij} are also unknowns to be solved. The unknown coefficients a_{ij} are taken as independent variables, and t_1 , t_2 , and t_3 are

taken as dependent variables. According to (7)–(9), the dimension of the variables to be optimized is 14. If the search space of time variables t_1 , t_2 , and t_3 is optimized directly, then the dimension of variables to be optimized is reduced to 3, decreasing the complexity and difficulty of population optimization.

To make the joint speed converge to the constraint as quickly as possible, two fitness function switching control is adopted. If any velocity does not conform to (12), then the fitness function of the individual is $|v(i, j)|$, where i is the i -th group of the M group population and j is the number of segments of polynomial interpolation. While finding the optimal individual, the genetic algorithm reduces $|v(i, j)|$ until the constraint body condition is satisfied. If the maximum speed of the three segments is consistent with (12), then (11) is used as the fitness function, and the genetic algorithm iteratively reduces the running time of each segment of the joint towards the optimization goal.

Specifically, the steps of the GA for time-optimal trajectory planning of the i -th joint of the manipulator are as follows:

Step 1: (initialize the population). M populations are randomly generated in the time search space of three interpolation functions of the i -th joint: populations POP_{t1} , POP_{t2} , and POP_{t3} .

Step 2: By substituting m groups of time variables t_{i1} , t_{i2} , and t_{i3} into (7)–(9), the unknown coefficients a_{ij} of the 3-5-3 polynomial are obtained.

Step 3: The coefficients a_{ij} of the 3-5-3 polynomial are substituted into (6), and the velocity function of the joint angle is obtained by using the time derivative to judge whether the real-time velocity satisfies (11).

Step 4: According to the constraints, the fitness function is selected to calculate the fitness value of each individual.

Step 5: The GA operation is as follows:

Step 6: Constitute new $M \times 3$ populations POP_{t1} , POP_{t2} , and POP_{t3} .

Step 7: If the termination condition (usually a good fitness value or a preset maximum number of iterations N_{\max}) is met, stop; else, go to Step 2.

Step 8: Complete the time optimization of each segment. The maximum value of each joint in each period of time is taken to ensure that kinematic constraints are met; that is, $t_1 = \max\{t_{i1}\}$, $t_2 = \max\{t_{i2}\}$, and $t_3 = \max\{t_{i3}\}$.

The GA requires cyclic iteration. The GA studied in this paper searches in the search space of the optimization target.

When (7) is calculated, the number of iteration steps is large, which may lead to nonfull rank matrices. Therefore, during each iteration, it is necessary to test whether there are any individuals with an actual value of zero in the newly generated population. If so, individual values are randomly regenerated.

4. Trajectory Tracking

The hierarchical structure control strategy of the six-degree-of-freedom manipulator is divided into three parts. First, the target joint angles of the six flexible joints are obtained in the trajectory planning layer. Then, in the MPC layer, the target joint angles are passed to the model predictive controller, and the optimized joint angles are obtained by solving (13). Further, in the bottom control layer, the optimized joint angles are transferred to the PD controller, the joint motion is driven by the torques obtained through (14), and the joint angles are measured by sensors. Finally, the joint angles are fed back to the model predictive controller for correction and input into forward kinematics to obtain the terminal trajectory.

4.1. Model Predictive Controller in the MPC Layer. Considering that MPC is a multivariable control strategy and the constraints can be explicitly processed in the optimization problem, this approach is widely used for manipulator control [27]. In this paper, the six-input-six-output MPC controller is selected for the 6-DOF manipulator.

The MPC strategy explicitly incorporates the model and constraints of the system for determining control inputs. At each step $k \geq 1$, the state space model is used to predict future outputs. At step k , an optimization problem involving future horizon prediction information (predictive horizon N_p) is solved online, and the control input of the future horizon (control horizon N_c) is calculated. At step $k + 1$, the optimization problem is solved again with new measurements. Thus, the control and prediction horizon recede by one step as time moves ahead by one step.

The MPC strategy is adopted to design the model predictive controller and optimize the joint angle as follows:

$$\min_{\theta(k|k), \theta(k+1|k), \dots, \theta(k+N_c-1|k)} \mathbb{J}_p(k), \quad (13a)$$

$$\text{s.t. (5),} \quad (13b)$$

$$\theta(k+m|k) \in [-180^\circ, 180^\circ], 0 \leq m \leq N_c - 1, \quad (13c)$$

$$\theta(k+m|k) = \theta(k+N_c-1|k), N_c \leq m \leq N_p - 1, \quad (13d)$$

where $\theta(k+m|k) \in \mathbb{R}^{6 \times 1}$ is the control input; (Tex translation failed) is the cost function; and $\theta_i^d(\cdot)$ represents the target angle of the i -th joint, which is solved using inverse kinematics. Only the first obtained control input $\theta^*(k|k)$ is applied to the system as the given value of the PD controller.

4.2. PD Controller in the Bottom Control Layer. However, the MPC is computationally expensive and slow in response, and it is not easy to control the multidegree-of-freedom manipulator. Compared with model predictive controller, the PD controller has better dynamic performance and faster response speed. Therefore, in the hierarchical control strategy designed in this paper, the model predictive controller with six inputs and six outputs is used to optimize the joint angle, and then the set values are transmitted to the bottom PD controllers to achieve.

PD feedback control is simple and easily implemented in engineering practice. Here, the PD controller is designed as follows:

$$\tau_i^{m*} = -K_p(\theta_i - \theta_i^*) - K_d(\dot{\theta}_i), \quad (14)$$

where $\dot{\theta}_i$ is the actual i -th joint angular velocity; θ_i^* is the optimized i -th joint angle of the upper model predictive controller; τ_i^{m*} is the optimized i -th joint driving torque of the PD controller; and K_p and K_d are the proportional and differential terms, respectively, both of which are diagonal positive definite matrices.

4.3. The Overall Algorithm. The overall algorithm is as follows.

- (1) If $k = 0$ (off-line stage)
- (2) {
- (3) Set the D-H parameters of the 6-DOF flexible joint manipulator.
- (4) Set the maximum simulation step k_{\max} .
- (5) Set predictive horizon N_p and control horizon N_c in the model predictive controller.
- (6) }
- (7) else if $0 < k \leq k_{\max}$ (online stage)
- (8) {
- (1) Obtain the target joint angle θ_i^d in the trajectory planning layer.
- (2) Input the target joint angle θ_i^d into the model predictive controller, and obtain the optimized joint angle θ_i^* by solving (13).
- (3) Pass the optimized joint angle θ_i^* into the PD controller; the torque τ_i^{m*} is obtained by (14) to drive the joint action.
- (4) Feed back the joint angle θ_i to the model predictive controller for correction.
- (5) Inject θ_i into the forward kinematics, and obtain the terminal trajectory X .
- (9) }

5. Simulation and Experiment

To verify the effectiveness of the hierarchical structure control strategy, the designed strategy is simulated using the Robotics Toolbox of MATLAB 2015a.

5.1. Simulation of Trajectory Planning. The simulation object in this section is a 6-DOF flexible joint manipulator. The kinematics model of the manipulator is constructed by the D-H method. The D-H parameters of the joints are shown in Table 2, where α_i is the i -th link twist angle, a_i is the i -th link length, and d_i is the offset value of the i -th joint.

In the Cartesian coordinate system, the interpolation points of the manipulator are given as shown in Table 3.

Through the inverse kinematics solution, the joint angles corresponding to the initial point, path points, and terminal point of the joints are obtained, as shown in Table 4.

The 3-5-3^o polynomial interpolation method based on PSO is used to optimize the interpolation time. The parameters are set as follows: set the particle swarm to 20, the number of iteration steps to 50, the inertia weight to $w_{max} = 0.9$ and $w_{min} = 0.4$, and the weight factors to $c_1 = 2$ and $c_2 = 2$. The maximum allowable velocity of the joint is $|V_{max}| = 1$ rad/s, and the velocity and acceleration at the initial point and the terminal point are 0. After 50 iterations, the joint interpolation times are $t_1 = 11.3831$ s, $t_2 = 4.2968$ s, and $t_3 = 14.2823$ s. The joint position curve, velocity curve, and acceleration curve are shown in Figures 3–5.

The steps in Section 3.3 are followed to determine the best time for each interpolation segment. Among the parameters, the population size is 100, the number of evolution generations is 50, the hybridization rate is $p_c = 0.8$, and the variation rate is $p_m = 0.5$. The maximum allowable velocity of the joint is $|V_{max}| = 1$ rad/s, and the velocity and acceleration at the initial point and the terminal point are 0. According to Step 8, the joint interpolation times are $t_1 = 7.3008$ s, $t_2 = 2.5050$ s, and $t_3 = 4.2507$ s.

After time optimization, the curves of the joint position, velocity, and acceleration are as shown in Figures 6–8. The joint velocity of the manipulator is close to V_{max} , and the acceleration is relatively stable.

Compared with the PSO results in the previous section, the GA is 15.9057 s shorter than PSO in time. Although the two algorithms can reach the optimal time under the kinematic constraints, after the optimization of GA, the joint speed is closer to the maximum speed, and the convergence speed is faster.

Although the PSO has the characteristics of simple structure and easy parameter adjustment, it still needs to use a random number to generate particle speed when generating a new particle swarm, thus reducing the convergence speed of particles. Although the GA is also based on the mask crossover of random generation, this paper can quickly eliminate vulnerable individuals by setting the crossover rate so that the individuals in the population can maintain high adaptability.

The PSO has a fast convergence speed. However, when the particles are too concentrated, they are likely to fall into local minima. Therefore, as shown in Table 5, in 10 simulation tests, the optimization effect of most particle swarm optimization algorithms is basically consistent with that of the genetic algorithm and sometimes even better than the genetic algorithm. However, when the particle swarm optimization algorithm falls into the local minimum, its optimization effect is far less than that of the genetic algorithm,

such as the fifth simulation result. Because the mutation operator of the genetic algorithm can ensure that the genetic algorithm jumps out of the local minimum, the stability of the genetic algorithm is higher than that of the particle swarm optimization algorithm, so the stability of the genetic algorithm is higher.

5.2. Simulation of the Hierarchical Structure Control Strategy.

The rigidity coefficient is $K = 10^4$, while the predictive horizon and the control horizon are $N_p = 30$ and $N_c = 5$, respectively. The maximum simulation step is $k_{max} = 705$. The sampling period T is chosen as 0.02 s. Considering the complexity of MPC, in each sampling period T , MPC calculates a group of set points only once, while the PD controller calculates 30 times to track the set points. The simulation results are shown in Figures 9–16.

Compared with the PD control strategy, the terminal trajectory of the hierarchical structure control strategy is more stable, and the tracking performance is much better, which shows that the designed strategy can restrain the jitter of flexible joints more effectively. For convenience of analysis, the projection of the trajectories in Figure 9 on the xy plane is shown in Figure 10, and further, the angle trajectories of each joint are shown in Figures 11–16. From Figures 1–16, we obtain the same conclusions.

The sum of squared error (SSE) is adopted to evaluate the effectiveness of the two control strategy:

$$SSE = \sum_{k=1}^{k_{max}} (E(k) - E^d(k))^2, \quad (15)$$

where $E(k)$ is the actual value and $E^d(k)$ is the target value. The SSE comparison is shown in Table 6. For the joint trajectories, the SSE of the hierarchical structure control strategy is always smaller than that of the PD control strategy, and for the terminal trajectories, the SSE of the hierarchical structure control strategy is reduced by 33.70% compared with the PD control strategy. In general, the designed hierarchical structure control strategy can coordinate and control the 6-DOF flexible joint manipulator more effectively than the PD control strategy.

To further illustrate the anti-interference and anti-disturbance performance of the proposed method, here, we design two cases. In Case 1, we add the white noise ($[-0.001, 0.001]$) in the feedback path. In Case 2, we add the disturbance (the amplitude is 0.005, and the time of duration is 1s) in the feedback path. The simulation results are shown in Figures 17 and 18, respectively. From Figures 17 and 18, we can clearly find that the tracking performance of the proposed strategy is much better than that of the PD strategy in both of the above two cases.

5.3. Experiment of the Hierarchical Structure Control Strategy.

In this paper, the dynamics and control theory of 6-DOF manipulator are studied, and the hierarchical structure control strategy is verified by simulation, but there are differences between theory and practice. Therefore, this section that has carried on the experimental research to the

TABLE 2: D-H parameters of the manipulator.

Joints	θ	α	a	d
1	θ_1	90°	0	0
2	θ_2	0	431 mm	0
3	θ_3	-90°	20.3 mm	150 mm
4	θ_4	90°	0	431 mm
5	θ_5	-90°	0	0
6	θ_6	0	0	0

TABLE 3: The interpolation points of the manipulator in Cartesian space.

Starting point	First interpolation point	Second interpolation point	Terminal point
(-431, -150, 452.1)	(-444.1, -128.5, 452.2)	(-468.1, -130.5, 452.2)	(-478.7, -149.2, 452.1)

TABLE 4: The interpolation points in joint space.

Points	Joints					
	1	2	3	4	5	6
ϕ_0	0	0	0	0	0	0
ϕ_1	-2.7887	0.845 7	-0.8627	21.566 2	-2.0823	21.567 4
ϕ_2	-2.4565	4.138 5	-4.2978	-80.2392	-2.3538	80.237 3
ϕ_3	-0.1005	6.328 1	-6.6750	-140.7006	-0.3724	145.687 4

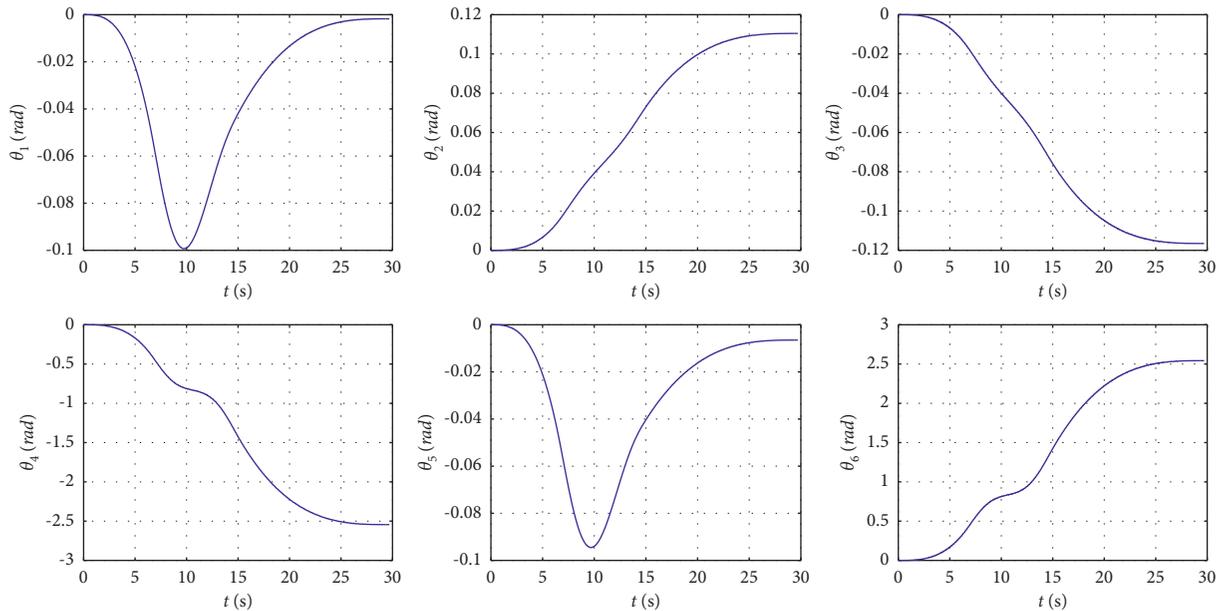


FIGURE 3: Joint position based on PSO.

manipulator, through the experiment, has verified the validity and the feasibility of the control strategy.

In this paper, we use the desktop manipulator, LeArm, as shown in Figure 19. The LeArm manipulator has six movable joints, with joint 1 moving left and right to realize base rotation; joint 2 and joint 3 moving back and forth; joint 4 moving up and down; joint 5 moving left and right to realize mechanical claw rotation; joint 6 closing the mechanical claw.

The D-H parameters of the arm are shown in Table 7. The joints of the LeArm manipulator are driven by actuators and

cannot directly feed back the actual joint angle. Therefore, in this experiment, the BWT901CL angle sensor is used to measure the joint angle, and the measured joint angle is fed back to the model predictive controller through a serial port for feedback correction, thus avoiding the error caused by the model mismatch.

In order to verify the effectiveness of the hierarchical structure control strategy of the 6-DOF manipulator, based on MPC, an experimental study was carried out on the LeArm manipulator. Firstly, the target position of each joint is given; then, the hierarchical structure control strategy

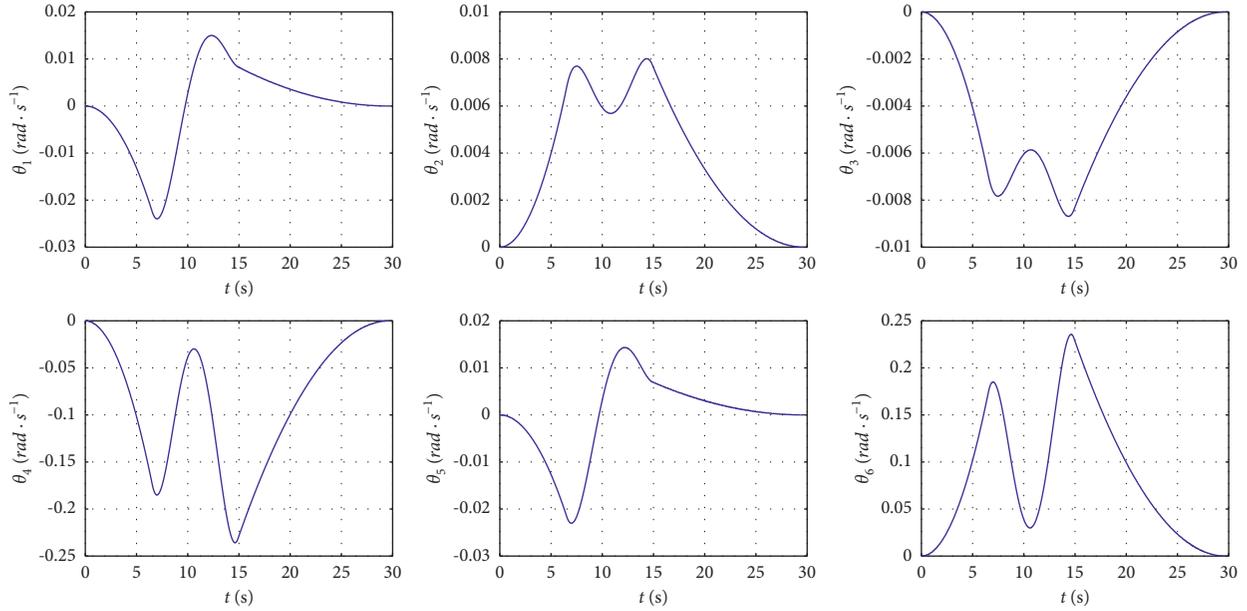


FIGURE 4: Joint velocity based on PSO.

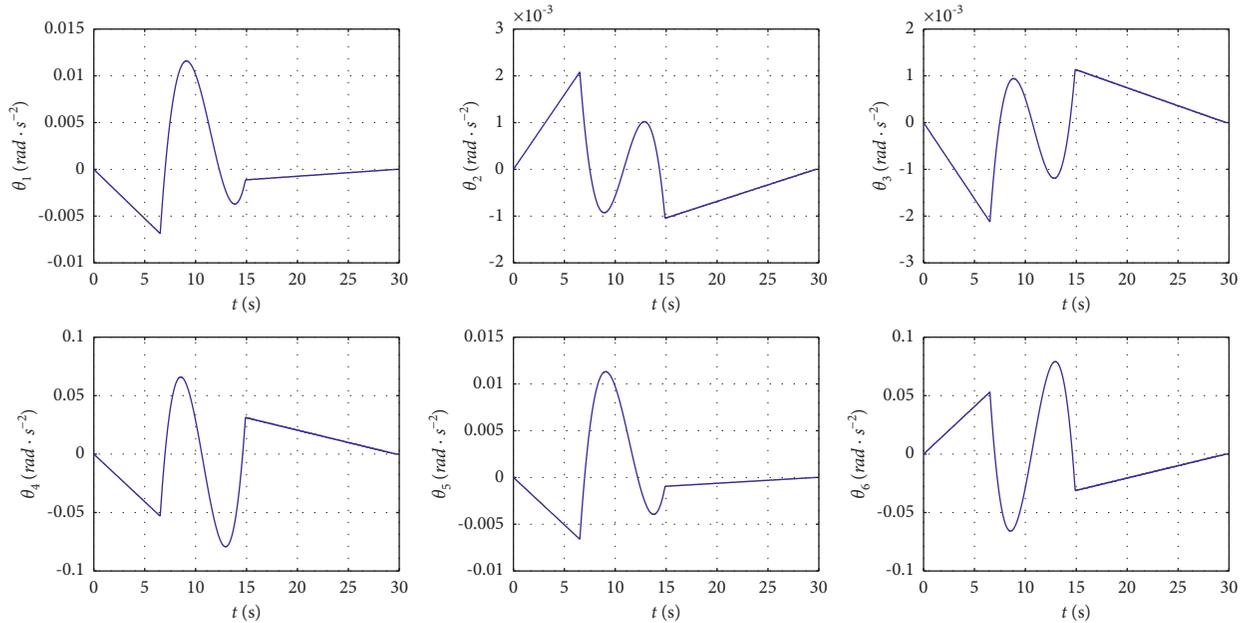


FIGURE 5: Joint acceleration curve based on PSO.

designed in this paper and the traditional PD control strategy are, respectively, used to make the manipulator reach the target position, as shown in Figure 19, which are three states in the process of robot arm motion; finally, taking joint 2 as an example and taking joint 3 and joint 4 as research objects, the joint angle curves under the two control strategies are compared in Figures 18–22.

As shown in Figures 20–22, by comparing and analyzing the curves under the two control strategies, the effect of the hierarchical control strategy is obviously better than that of the PD control strategy. Although these two control strategies can drive the manipulator to

the target position in a limited time, the difference is that the hierarchical structure control strategy can get the predicted output of the actuator in the time domain in advance. When the manipulator is close to the target position, the speed of the actuator is limited, which makes the manipulator fast and stable and reduces the overshoot.

Generally speaking, the hierarchical structure control strategy is outstanding in terms of rapidity and overshoot suppression, which verifies the effectiveness of the hierarchical structure control strategy designed in this paper for the manipulator control process. However, the hierarchical

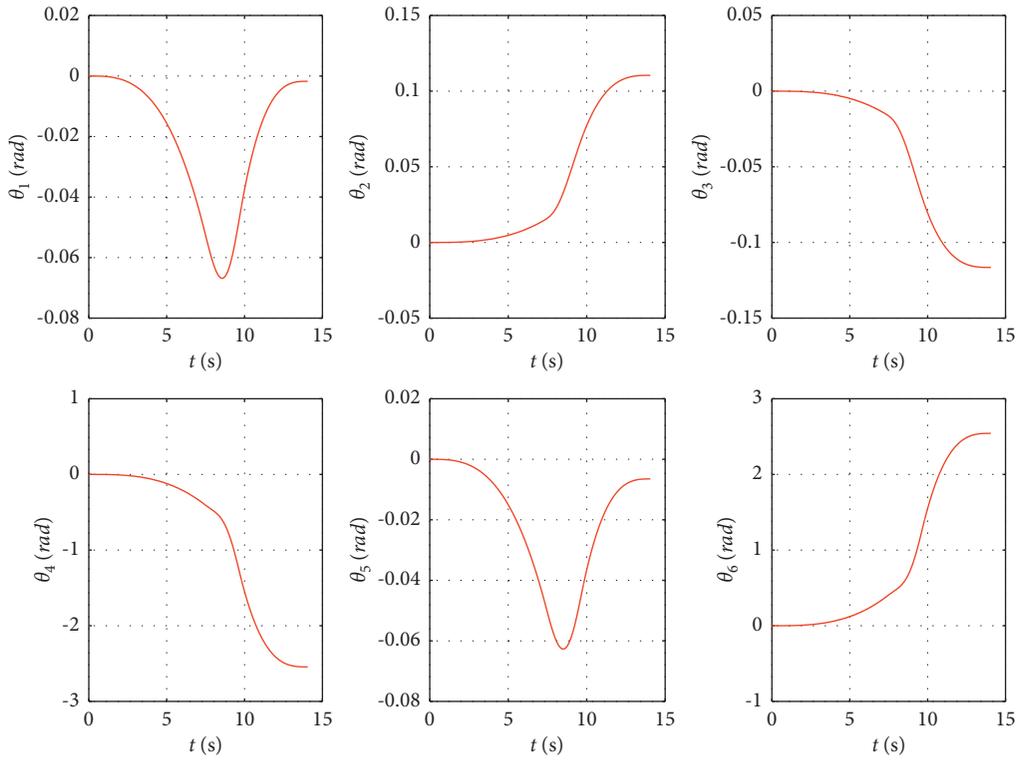


FIGURE 6: Joint position based on GA.

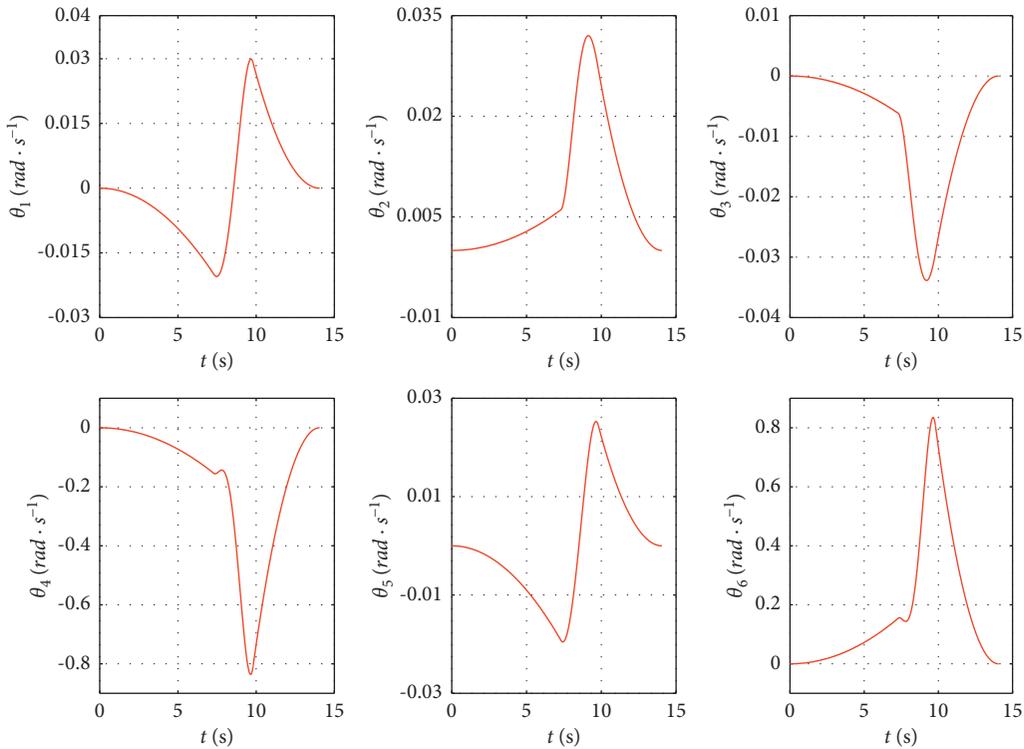


FIGURE 7: Joint velocity based on GA.

structure is more complex than that of the PD strategy, which brings in more challenges during the implementation in the actual application; at the same time, the computation

burden is much heavier, and the computing devices with higher performance are needed. In other words, the performance improvements of the proposed strategy come at

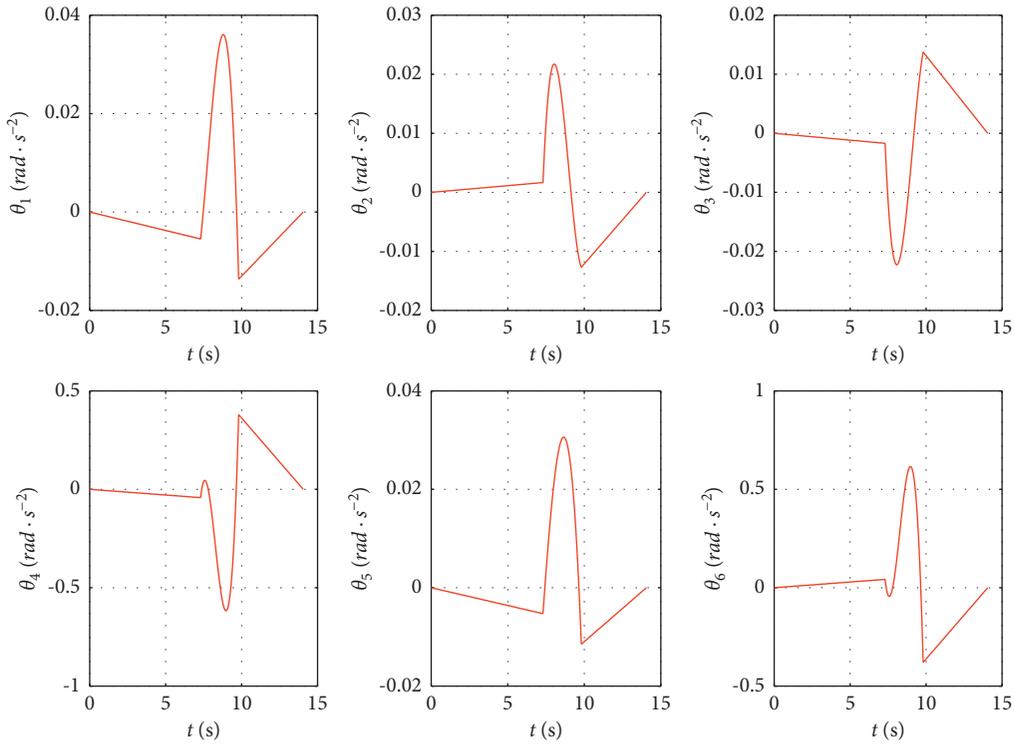


FIGURE 8: Joint acceleration curve based on GA.

TABLE 5: Trajectory optimization results statistics.

Algorithm	Simulation times									
	1	2	3	4	5	6	7	8	9	10
PSO	14.02s	14.25s	13.89s	18.23s	29.96s	15.31s	13.68s	14.12s	17.88s	15.55s
GA	14.98s	14.32s	15.01s	14.24s	14.06s	16.94s	18.55s	13.62s	17.50s	14.22s

Note. The PSO and GA are compared in joint angle, angular velocity, and angular acceleration curve.

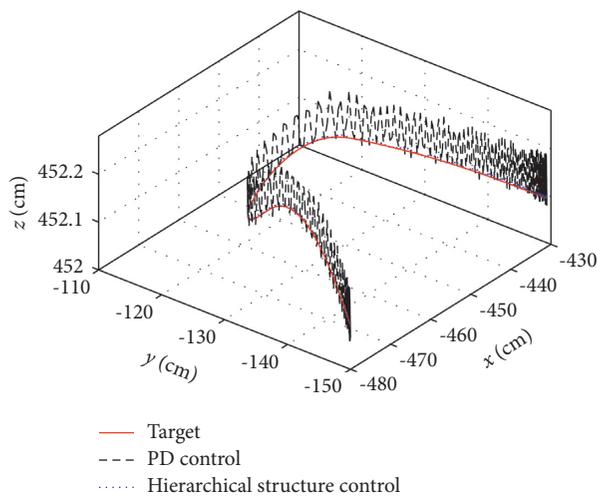


FIGURE 9: Terminal trajectories of the manipulator.

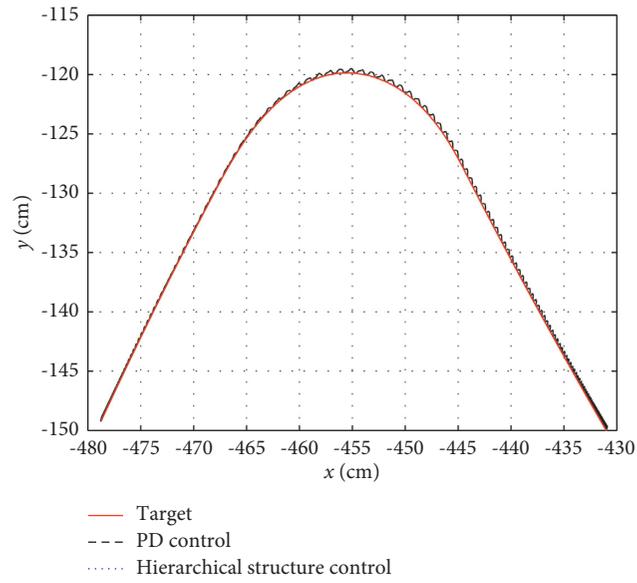
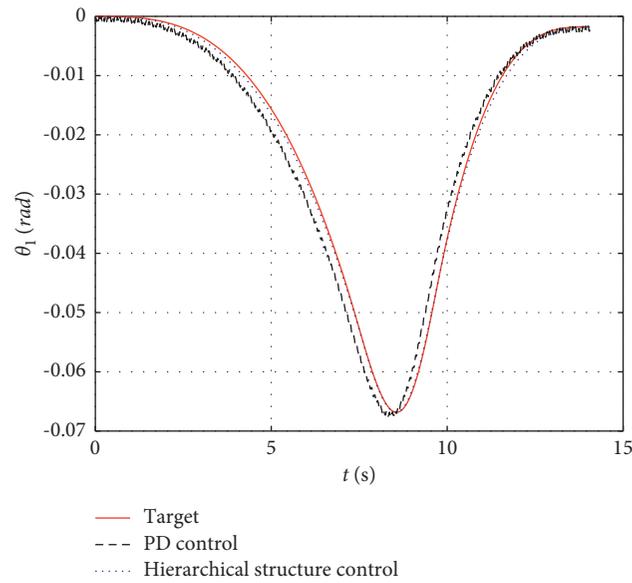
FIGURE 10: Terminal trajectories of the manipulator (xy plane).

FIGURE 11: Angle trajectories of joint 1.

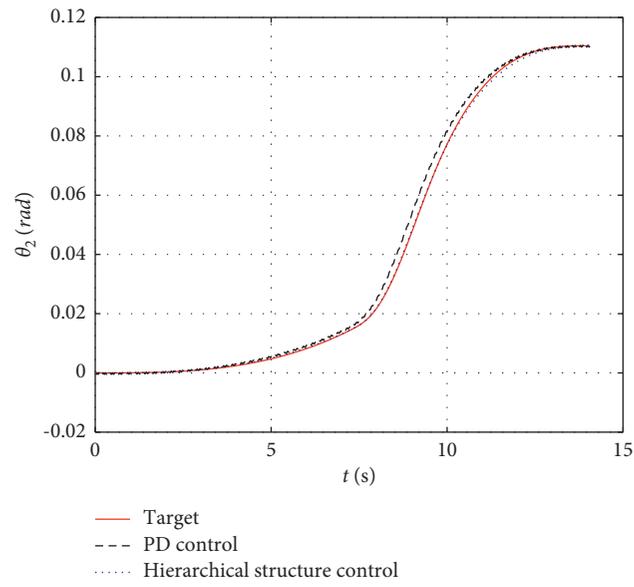


FIGURE 12: Angle trajectories of joint 2.

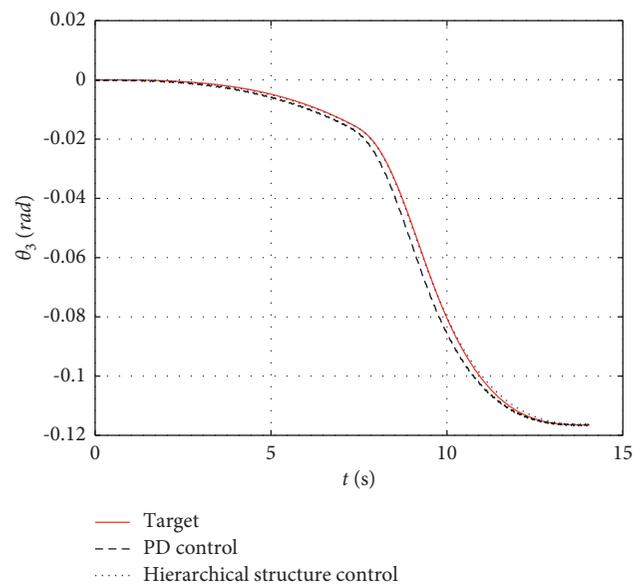


FIGURE 13: Angle trajectories of joint 3.

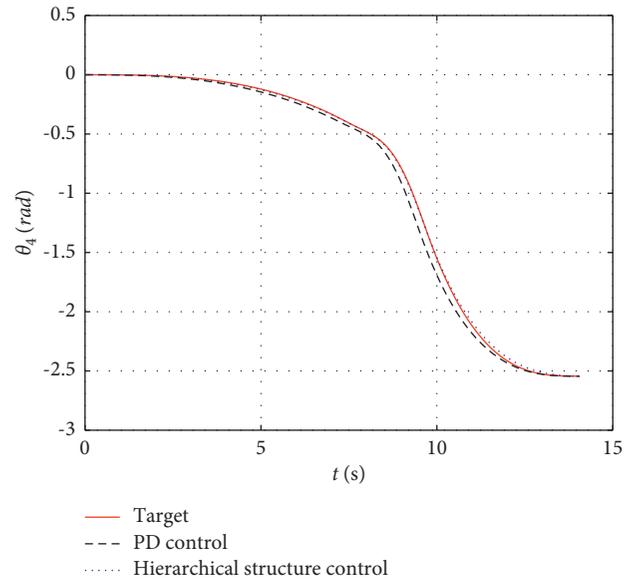


FIGURE 14: Angle trajectories of joint 4.

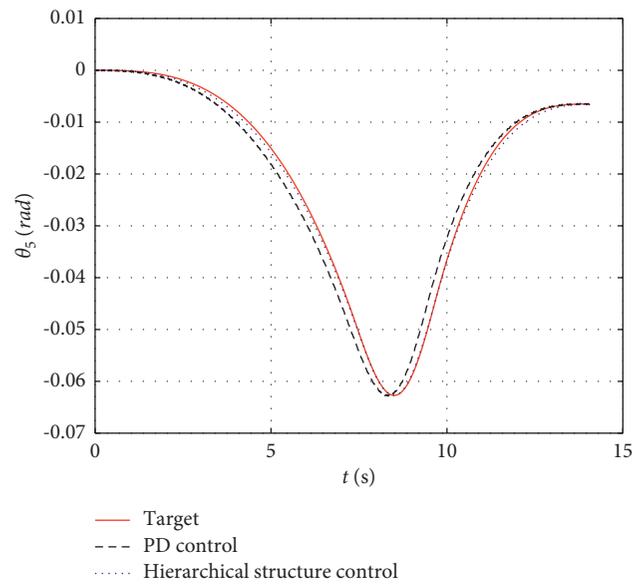


FIGURE 15: Angle trajectories of joint 5.

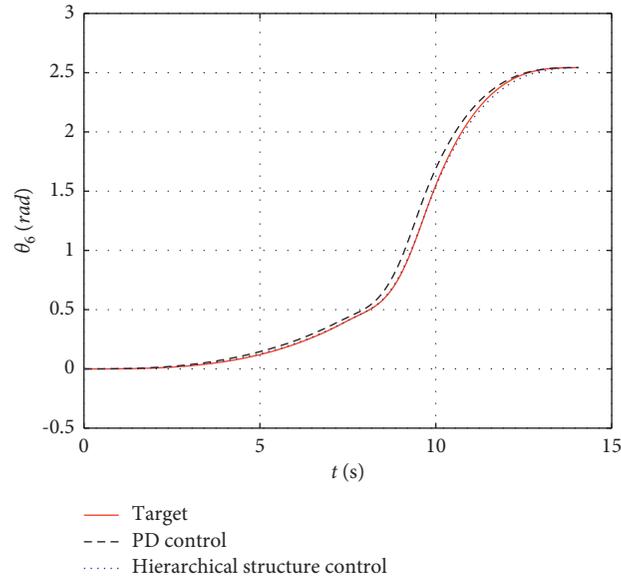


FIGURE 16: Angle trajectories of joint 6.

TABLE 6: SSE comparisons.

Strategy	Terminal trajectory	Joints					
		1	2	3	4	5	6
PD strategy	1841.900 0	0.131 4	0.109 3	0.122 3	59.671 4	0.108 0	59.584 9
Hierarchical structure strategy	1221.200 0	0.078 2	0.085 7	0.095 9	45.087 5	0.062 2	45.022 8
Improvement	33.70%	40.49%	21.59%	21.58%	24.44%	42.41%	24.44%

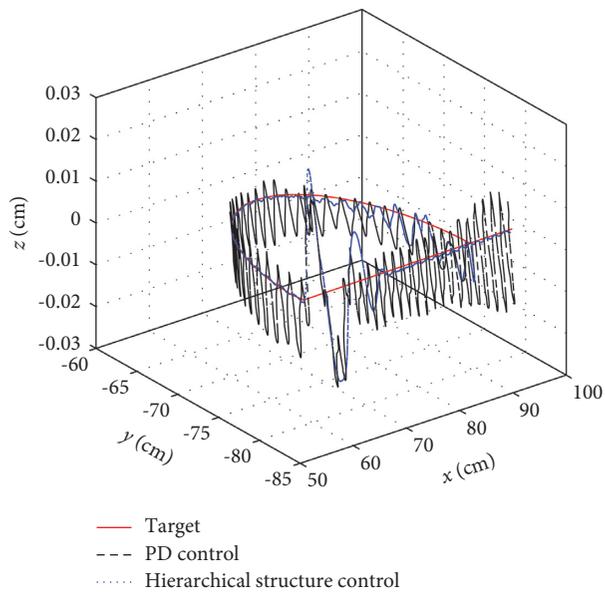


FIGURE 17: Terminal trajectories (noise).

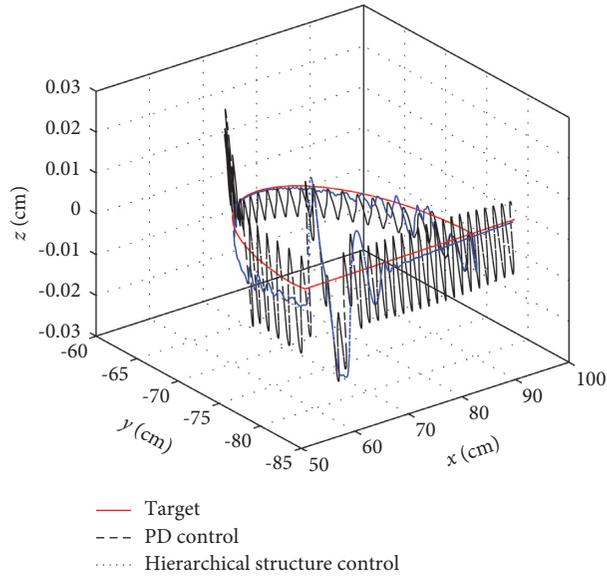


FIGURE 18: Terminal trajectories (disturbance).

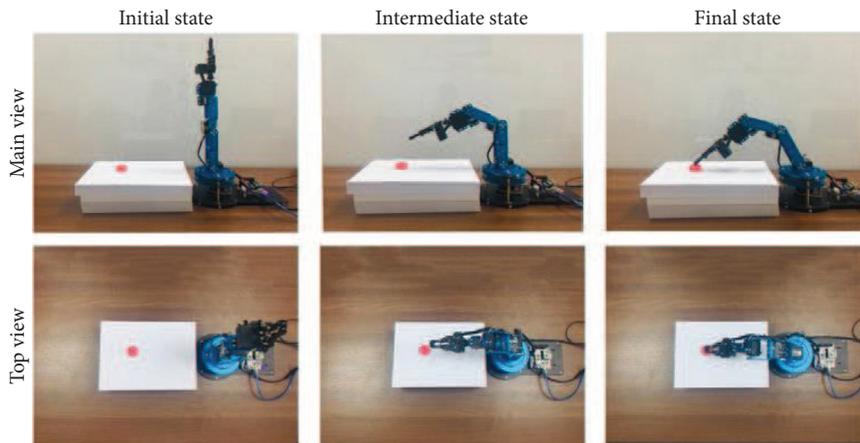


FIGURE 19: The LeArm manipulator.

TABLE 7: D-H parameters of LeArm manipulator.

Joints	θ	α	a	d
1	θ_1	0	0	149 mm
2	θ_2	90°	0	0
3	θ_3	0	100 mm	0
4	θ_4	0	90 mm	0
5	θ_5	0	156 mm	0
6	θ_6	0	0	0

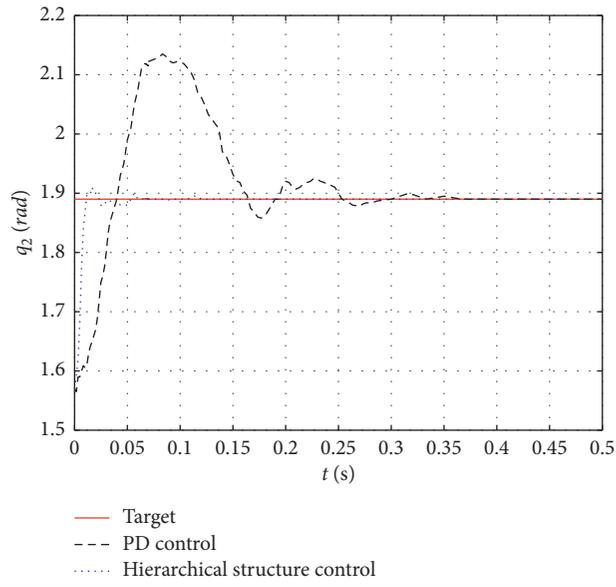


FIGURE 20: Joint 2 position curve.

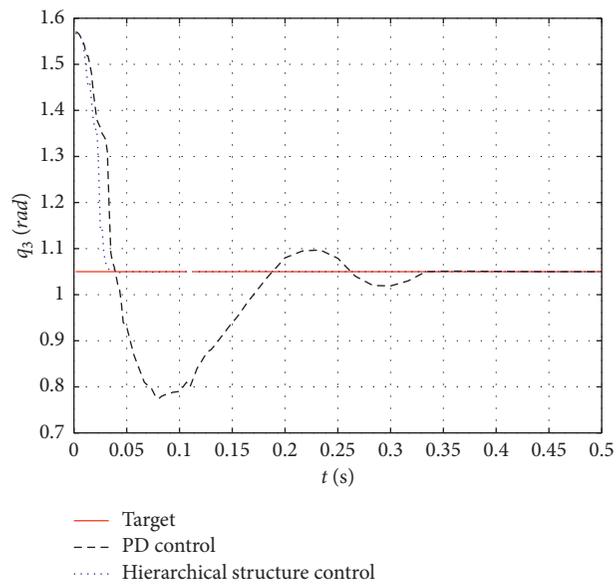


FIGURE 21: Joint 3 position curve.

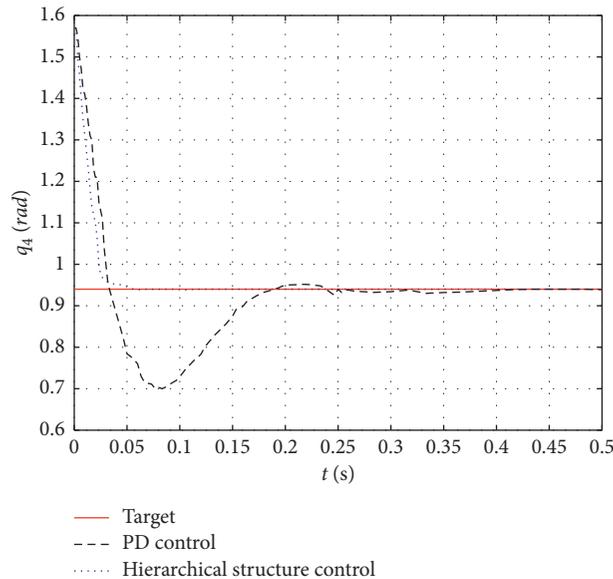


FIGURE 22: Joint 4 position curve.

the expense of the control system complexity and the computation increase.

6. Conclusions

In this paper, based on the dynamics analysis of a 6-DOF flexible joint manipulator, the trajectory planning and control implementation of the manipulator are thoroughly studied. By designing a hierarchical structure control strategy, the manipulator can achieve the target trajectory quickly and stably. Simulation results show that the control performance of the designed strategy is better than that of the PD control strategy. In the future, the following two points should be further paid more attention: (1) the research results should consider the actual condition, such as the system uncertainty and external disturbance; (2) the proposed strategy should be implemented online to the actual application scenarios through the embedded devices.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] J. Han, Y.-H. Chen, X. Zhao, and F. Dong, "Optimal design for robust control of uncertain flexible joint manipulators: a fuzzy dynamical system approach," *International Journal of Control*, vol. 91, no. 4, pp. 937–951, 2018.
- [2] Q. Yao, "Adaptive fuzzy neural network control for a space manipulator in the presence of output constraints and input nonlinearities," *Advances in Space Research*, vol. 99, pp. 1–13, 2021.
- [3] A. Lismonde, V. Sonnevile, and O. Bruls, "A geometric optimization method for the trajectory planning of flexible manipulators," *Multibody System Dynamics*, vol. 3, pp. 1–16, 2019.
- [4] S. Dian, Y. Hu, T. Zhao, and J. Han, "Adaptive backstepping control for flexible-joint manipulator using interval type-2 fuzzy neural network approximator," *Nonlinear Dynamics*, vol. 97, no. 2, pp. 1567–1580, 2019.
- [5] R. Scattolini, "Architectures for distributed and hierarchical model predictive control - a review," *Journal of Process Control*, vol. 19, no. 5, pp. 723–731, 2009.
- [6] V. Adetola and M. Guay, "Integration of real-time optimization and model predictive control," *Journal of Process Control*, vol. 20, no. 2, pp. 125–133, 2010.
- [7] Y. Xu, M. Xin, J. Wang, and S. Jayasuriya, "Hierarchical control of cooperative nonlinear dynamical systems," *International Journal of Control*, vol. 85, no. 8, pp. 1093–1111, 2019.
- [8] H. Pan, W. Zhong, and Z. Wang, "An on-line constraint softening strategy to guarantee the feasibility of dynamic controller in double-layered MPC," *Chinese Journal of Chemical Engineering*, vol. 25, no. 12, pp. 115–121, 2017.
- [9] L. Consolini, M. Locatelli, A. Minari, A. Nagy, and I. Vajk, "Optimal time-complexity speed planning for robot manipulators," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 790–797, 2019.
- [10] Y. Kim and B. K. Kim, "Time-optimal trajectory planning based on dynamics for differential-wheeled mobile robots with a geometric corridor," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 7, pp. 5502–5512, 2017.
- [11] C. Lin, P. Chang, and J. Luh, "Formulation and optimization of cubic polynomial joint trajectories for industrial robots," *IEEE Transactions on Automatic Control*, vol. 28, no. 12, pp. 1066–1074, 1984.
- [12] S. A. Bazaz and B. Tondu, "Minimum time on-line joint trajectory generator based on low order spline method for industrial manipulators," *Robotics and Autonomous Systems*, vol. 29, no. 4, pp. 257–268, 1999.
- [13] S. Behzadipour and A. Khajepour, "Time-optimal trajectory planning in cable-based manipulators," *IEEE Transactions on Robotics*, vol. 22, no. 3, pp. 559–563, 2006.

- [14] C.-T. Chen and H.-V. Pham, "Trajectory planning in parallel kinematic manipulators using a constrained multi-objective evolutionary algorithm," *Nonlinear Dynamics*, vol. 67, no. 2, pp. 1669–1681, 2012.
- [15] M. Wang, J. Luo, and U. Walter, "Trajectory planning of free-floating space robot using particle swarm optimization (PSO)," *Acta Astronautica*, vol. 112, pp. 77–88, 2015.
- [16] L. Tian and C. Collins, "An effective robot trajectory planning method using a genetic algorithm," *Mechatronics*, vol. 14, no. 5, pp. 455–470, 2004.
- [17] L. Angel and J. Viola, "Fractional order PID for tracking control of a parallel robotic manipulator type delta," *ISA Transactions*, vol. 79, pp. 172–188, 2018.
- [18] R. R. Ardehshiri, M. H. Khooban, A. Noshadi, N. Vafamand, and M. Rakhshan, "Robotic manipulator control based on an optimal fractional-order fuzzy PID approach: SiL real-time simulation," *Soft Computing*, vol. 24, no. 5, pp. 3849–3860, 2020.
- [19] J. Kumar, V. Kumar, and K. P. S. Rana, "Fractional-order self-tuned fuzzy PID controller for three-link robotic manipulator system," *Neural Computing & Applications*, vol. 32, no. 11, pp. 7235–7257, 2020.
- [20] Y. Pan, X. Li, and H. Yu, "Efficient PID tracking control of robotic manipulators driven by compliant actuators," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 2, pp. 915–922, 2019.
- [21] D. Guo, F. Xu, and L. Yan, "New pseudoinverse-based path-planning scheme with PID characteristic for redundant robot manipulators in the presence of noise," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 6, pp. 2008–2019, 2018.
- [22] L. Zong, J. Luo, M. Wang, and J. Yuan, "Obstacle avoidance handling and mixed integer predictive control for space robots," *Advances in Space Research*, vol. 61, no. 8, pp. 1997–2009, 2018.
- [23] X. Zhang, W. Li, Y. Xi, and W. Chen, "Design of constrained predictive control system for 6 degrees-of-freedom manipulators," *Control Theory & Applications*, vol. 31, no. 11, pp. 1464–1472, 2014.
- [24] G. P. Incremona, A. Ferrara, and L. Magni, "MPC for robot manipulators with integral sliding modes generation," *IEEE*, vol. 22, no. 3, pp. 1299–1307, 2017.
- [25] L. Shi, S. Kayastha, and J. Katupitiya, "Robust coordinated control of a dual-arm space robot," *Acta Astronautica*, vol. 138, pp. 475–489, 2017.
- [26] M. W. Spong, "Modeling and control of elastic joint robots," *Journal of Dynamic Systems, Measurement, and Control*, vol. 109, no. 4, pp. 310–318, 1987.
- [27] H. G. Pan, W. Zhong, and Z. Y. Wang, "Economic optimization and control based on multi priority rank RTO and double layered MPC," *Asian Journal of Control*, vol. 20, no. 6, pp. 2271–2280, 2018.