



Flexible manipulator trajectory tracking based on an improved adaptive particle swarm optimization algorithm with fuzzy PD control

Weiwei Sun¹, Yubin Jin¹, Kun Dai¹, Zhongyuan Guo², and Fei Ma¹

¹Mechanical Electrical Engineering School, Beijing Information Science & Technology University, Beijing 100192, China

²Beijing Institute of Remote Sensing Equipment, Beijing 100854, China

Correspondence: Fei Ma (mf@bistu.edu.cn)

Received: 27 August 2024 – Revised: 24 October 2024 – Accepted: 19 December 2024 – Published: 24 February 2025

Abstract. Trajectory planning for flexible manipulators is a critical area of research in robotics. A trajectory tracking controller can enhance the accuracy of the manipulator's path and reduce vibrations. However, current flexible manipulators remain largely in the research phase, with many studies revealing issues such as poor accuracy in dynamic modeling, weak tracking performance in controller design, and insufficient vibration suppression capabilities. To address these challenges and improve the trajectory tracking performance of the manipulator, this paper focused on vibration suppression and trajectory planning for a two-link flexible manipulator and proposed a novel control method that integrates a modified adaptive particle swarm optimization algorithm (MAPSO) with fuzzy proportional–derivative (PD) control to achieve effective trajectory tracking. Firstly, the dynamic equations of the two-link flexible manipulator system were derived using the assumed modal method in conjunction with Lagrangian dynamics. Next, a 3-5-3 hybrid polynomial algorithm based on MAPSO was proposed to optimize the trajectory of the manipulator. Simulation results demonstrated that the optimization algorithm significantly enhances efficiency. Specifically, the number of iterations required for the two joints was reduced by 33 % and 54 %, respectively, when compared to the original algorithm. Additionally, this optimization led to a total reduction in running time of 0.03 s. Subsequently, the MAPSO algorithm was utilized to enhance the fuzzy PD controller based on the previously obtained optimal trajectory, leading to the development of a trajectory tracking controller known as MAPSO-FuzzyPD. Simulation results indicated that the proposed algorithm significantly reduced the maximum starting torque for both joints. Specifically, the maximum starting torque of joint 1 was decreased by 61.3 % and 40.3 % when compared to PD control and fuzzy PD control, respectively. Additionally, the maximum starting torque of joint 2 was reduced by 57.9 % and 42.1 % in comparison to the same control methods. Finally, an experimental platform for the flexible manipulator was established, and the experimental results further validated the effectiveness and feasibility of the algorithm proposed in this paper concerning joint trajectory tracking.

1 Introduction

In recent years, flexible manipulators have been widely adopted in aerospace, biomedical (Lee et al., 2012), and industrial applications (Yu et al., 2016). The trend in robotic arm systems is towards lighter, more complex, and more precise designs, making the influence of component flexibility increasingly significant. Traditional multi-rigid-body models

often fail to accurately capture the dynamic characteristics of these flexible systems. Consequently, many researchers are focusing on the dynamic behavior of flexible multi-body systems. Meng et al. (2022) employed the assumed modal method (AMM) to model a rigid–flexible two-link manipulator. Similarly, Gao et al. (2019) utilized AMM to model a two-link flexible manipulator, demonstrating that their discrete model effectively mitigated data-related deficiencies,

resulting in more stable outcomes. Sun et al. (2024a, b) combined AMM with the Lagrangian method to characterize the dynamics of a rigid–flexible coupled manipulator. Shen and Fan (2024) integrated AMM with the finite-element method (FEM) to investigate the dynamic characteristics of a flexible arm system using Bessel interpolation. Additionally, several scholars, such as Wang et al. (2010), Liu et al. (2017), Wang et al. (2017), Shao et al. (2020), and Alandoli et al. (2021), have employed FEM to derive dynamic equations by segmenting the arm into a finite number of elements. At this stage, most research on flexible robotic arms focuses on modeling in the horizontal plane, often neglecting the effects of gravity. However, in many practical applications where the robotic arm operates in a vertical plane, accounting for gravity becomes essential. Thus, it is crucial to incorporate gravitational factors into the dynamic modeling of flexible manipulators. By implementing the modal truncation technique within the assumed modal approach, the modal order of the system can be condensed into a finite-dimensional structure. This method effectively decreases the system's degrees of freedom and the quantity of equations required. Moreover, by employing the Lagrangian method to determine the disparity between the overall kinetic and potential energies of the system, the intricate internal force terms that typically complicate the computational procedure can be circumvented. This simplification not only alleviates the complexity of solving the dynamic equations but also streamlines the subsequent formation of the control system. Consequently, this investigation seeks to apply the assumed modal method and Lagrangian equations in delineating the deformation of a flexible arm and formulating the system's dynamic model.

Flexible manipulators are susceptible to vibrations and other issues during operation, leading to significant degradation in the positioning accuracy of the end effector. To address this, trajectory tracking often employs active vibration suppression control methods which aim to reduce or eliminate the structural vibration energy through active system control. Sasaki et al. (2023) utilized reinforcement learning to control the vibrations of a vertical-plane double-link flexible manipulator, designing a reward function to minimize the gap between the manipulator's end and the target coordinates. While this approach effectively suppressed vibrations, discrepancies were observed between simulation results and physical validation. He et al. (2021) developed a reinforcement learning controller for a horizontal two-link flexible manipulator using two neural networks: one for behavioral learning and the other for reward evaluation. Experimental verification demonstrated the controller's applicability. Reinforcement learning had also been used by Nagesh Rao et al. (2014) and Tang et al. (2014) to achieve control over target robotic arms. However, despite its promise, reinforcement learning has limitations, including high computational demands, extended training times, and requiring significant human intervention and optimization. Fuzzy control, on the other hand, is a widely adopted method known for

its robustness and suitability for nonlinear systems. Numerous researchers have explored its applications. For instance, Alandoli et al. (2021) proposed a fuzzy logic control linear quadratic regulator (FLC-LQR) controller that combines optimal and fuzzy control strategies for a single-link flexible robotic arm, demonstrating effective vibration suppression and high robustness to uncertainties. Additionally, Beata (2018) introduced a fuzzy controller in a computer model of a rotating robotic arm joint with a localization control strategy, aimed at enhancing system performance by replacing traditional external controllers.

Traditional proportional–integral–derivative (PID) control is known for its simplicity and ease of implementation; however, its accuracy can decline when applied to complex systems. Integrating fuzzy control technology can enhance the capabilities of the original PID controller. Research by Anavatti et al. (2006), Liu et al. (2014), Zhang et al. (2016), and Chhabra et al. (2019) demonstrated the optimization of PID controllers through fuzzy control, with experimental evidence supporting the superiority of this combined approach. Despite its advantages, fuzzy control has limitations, particularly in its reliance on empirical design. The affiliation functions often require initial values that are determined through empirical and experimental methods. To address these drawbacks, Zhang et al. (2019) proposed a novel control scheme for a 3-degree-of-freedom robotic arm, utilizing a fuzzy proportional–derivative (PD) controller enhanced by a backpropagation neural network (BPNNBF-PD). This approach leverages the backpropagation neural network to calibrate the affiliation functions and optimize the performance of the fuzzy PD controller. In summary, while many researchers have explored fuzzy control, most have not fully mitigated its inherent drawbacks. Furthermore, there is a notable scarcity of studies focused on optimizing fuzzy PID controllers specifically for flexible manipulator applications.

The research direction of this paper is the vibration suppression trajectory planning and tracking control of a two-link flexible robotic arm, and a control method combining the improved particle swarm optimization algorithm with the fuzzy PD control is proposed to reach the purpose of trajectory tracking. Firstly, the dynamical equations of the robotic arm are obtained by assuming the modal method and the Lagrangian method on the non-inertial system. Then, a 3-5-3 hybrid polynomial algorithm based on the modified adaptive particle swarm optimization (MAPSO) is proposed to optimize the trajectory of the robotic arm. Finally, the MAPSO algorithm is utilized to modify the fuzzy PD controller on the basis of the obtained optimal trajectory to construct the MAPSO-FuzzyPD-based trajectory tracking controller, and the feasibility and superiority of the proposed method are demonstrated by theoretical and experimental analysis, respectively.

The focus of this paper is on vibration suppression, trajectory planning, and tracking control for a two-link flexible manipulator. A control method that combines an improved parti-

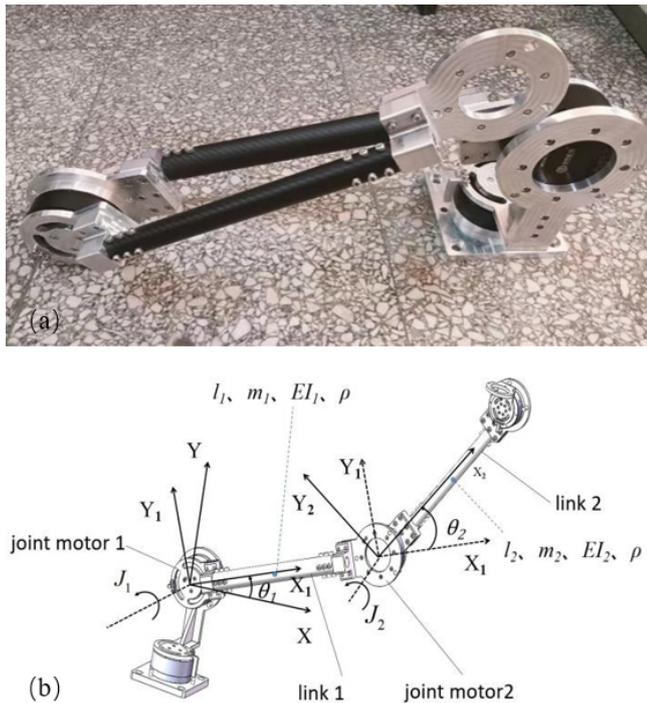


Figure 1. The flexible manipulator model. (a) The physical model. (b) The mathematical model.

cle swarm optimization algorithm with fuzzy PD control was proposed to achieve effective trajectory tracking. Initially, the dynamic equations of the manipulator were derived using the modal method and the Lagrangian approach within a non-inertial reference frame. Subsequently, a 3-5-3 hybrid polynomial algorithm based on the MAPSO algorithm was introduced to optimize the trajectory of the robotic arm. Finally, the MAPSO algorithm was employed to refine the fuzzy PD controller based on the optimal trajectory obtained, resulting in the development of the MAPSO-FuzzyPD trajectory tracking controller. The feasibility and superiority of the proposed method were validated through both theoretical and experimental analyses.

2 Dynamic modeling of the flexible manipulator

2.1 Flexible manipulator model

The physical object of the flexible manipulator studied in this paper is illustrated in Fig. 1a. This flexible robotic arm system consists of two flexible connecting rods, and its mathematical model is depicted in Fig. 1b. The parameters of the flexible manipulator are shown in Table 1.

To facilitate the modeling of the system dynamics, the following assumptions were made:

1. The influence of joint flexibility was neglected, and only the flexibility of the connecting rods was considered in the vibration analysis and control studies.

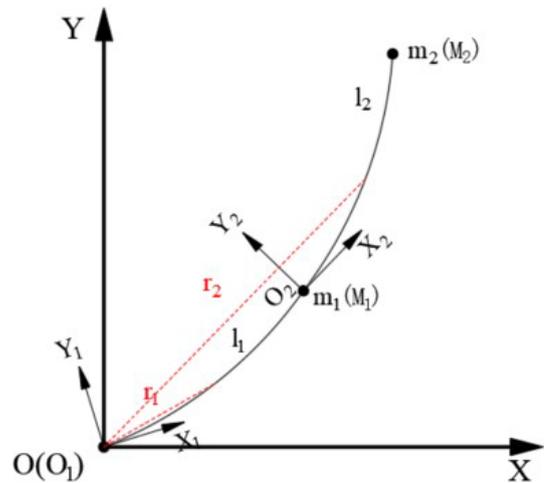


Figure 2. Coordinate system of the 2-degree-of-freedom flexible manipulator.

2. The analysis focused solely on the bending deformation of the flexible connecting rods, while shear deformation and axial tensile/compressive deformation were ignored.
3. The deformations of the end load and the joint motor components of the robotic arm were considered small enough to disregard their effects on the overall system, treating them as mass points.

Based on the above assumptions, the coordinate system depicted in Fig. 2 was established. In this system, $O-XY$ represents the inertial coordinate system, while $O_1-X_1Y_1$ and $O_2-X_2Y_2$ are the floating coordinate systems defined at the rotating joints of the flexible linkages L_1 and L_2 , respectively. And here, r_1 denotes the position vector of the point corresponding to the transverse coordinate x_1 ($0 \leq x_1 \leq l_1$) of the link L_1 with respect to the origin O of the inertial system. Similarly, r_2 represents the position vector of the point corresponding to the transverse coordinate x_2 ($0 \leq x_2 \leq l_2$) of the link L_2 with respect to the origin O of the inertial system.

2.2 Dynamic modeling of the flexible manipulator

According to the assumed modal method, the vibration mode function of the flexible manipulator (Tavasoli, 2018) could be expressed as

$$w_i(x_i, t) = \sum_{i=1}^n \Phi_i(x_i)q_i(t), \tag{1}$$

where n is the number of retained modes, $\Phi_i(x_i)$ is the n th-order modal function of link i , $q_i(t)$ represents the corresponding modal coordinate, and x_i denotes the axial position coordinate in the floating coordinate system of the link i .

Considering the two flexible connecting rods to be Euler-Bernoulli beams, the transverse vibration equations of mo-

Table 1. The main parameters of the flexible manipulator.

Title	Parameters	Values
Connecting rod L_1	Arm's length (l_1)	500 mm
	Section size (A_1)	External diameter d11: 35 mm. Internal diameter d12: 32 mm
	Moment of inertia (I_1)	0.1150 kg m^{-2}
	End quality (m_1)	1.38 kg
Connecting rod L_2	Arm's length (l_2)	500 mm
	Section size (A_2)	External diameter d21: 25 mm Internal diameter d22: 22 mm
	Moment of inertia (I_2)	0.0252 kg m^{-2}
	End quality (m_2)	0.303 kg
Density of flexible arms (ρ)		1760 kg m^{-3}
Modulus of elasticity of flexible arms (E)		$2.3 \times 10^{11} \text{ Pa}$

tion can be expressed as (Omidi and Mahmoodi, 2016)

$$EI \frac{\partial^4 w}{\partial x^4} + \rho A \frac{\partial^2 w}{\partial t^2} = 0. \quad (2)$$

According to the characteristics of the beam, the boundary conditions (Zhang et al., 2023) are defined as follows:

$$\begin{cases} w_i(0, t) = 0, \\ \frac{\partial w_i(0, t)}{\partial x_i} = 0, \\ E_i I_i \frac{\partial^2 w_i(l_i, t)}{\partial x_i^2} = 0, \\ E_i I_i \frac{\partial^3 w_i(l_i, t)}{\partial x_i^3} = m_{\text{end}} \frac{\partial^2 w_i(l_i, t)}{\partial t^2}. \end{cases} \quad (3)$$

The frequency equation, modal function, and intrinsic frequency of the beam are obtained by substituting Eq. (3) into Eq. (1), leading to the following results:

$$1 + \cosh(\lambda_j l) \cos(\lambda_j l) = \frac{m_{\text{end}}}{m} \lambda_j l \left(\sin(\lambda_j l) \cosh(\lambda_j l) - \cos(\lambda_j l) \sinh(\lambda_j l) \right), \quad (4)$$

$$\Phi_j(x_i) = \cosh(\lambda_j x_i) - \cos(\lambda_j x_i) - \frac{\cosh(\lambda_j l_1) + \cos(\lambda_j l_1)}{\sinh(\lambda_j l_1) + \sin(\lambda_j l_1)} (\sinh(\lambda_j x_1) - \sin(\lambda_j x_1)), \quad (5)$$

$$\omega_j = \lambda_j^2 \sqrt{(E_j I_j) / (\rho_j A_j)}, \quad (6)$$

where m is the mass of the flexible arm, m_{end} is the mass of the end of the flexible arm, and λ_j is the coefficient corresponding to the j th natural frequency.

According to our team's previous study, it is known that the first-order modes of this flexible manipulator play a major role in the deformation of the end elastic vibration. Therefore, the first-order modes were taken into account to calculate the dynamics model.

According to the geometric relationship, r_1 and r_2 can be expressed as

$$r_1 = \begin{bmatrix} x_1 c_1 - w_1(x_1, t) s_1 \\ x_1 s_1 + w_1(x_1, t) c_1 \end{bmatrix}, \quad (7)$$

$$r_2 = \begin{bmatrix} l_1 c_1 - w_1(l_1, t) s_1 + x_2 c_{12} - w_2(x_2, t) s_{12} \\ l_1 s_1 + w_1(l_1, t) c_1 + x_2 s_{12} + w_2(x_2, t) c_{12} \end{bmatrix}, \quad (8)$$

where $c_1 = \cos \theta_1$, $s_1 = \sin \theta_1$, $c_{12} = \cos(\theta_1 + \theta_2)$, and $s_{12} = \sin(\theta_1 + \theta_2)$.

To solve the kinetic equations, the Lagrangian method was used, the expression of which is as follows:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{p}} \right) - \frac{\partial L}{\partial p} = Q, \quad (9)$$

where the Lagrangian function is given by $L = T - U$, T being the total kinetic energy of the system and U the total potential energy of the system. p denotes the generalized coordinate of the system, and Q is the generalized force in generalized coordinates.

The total kinetic energy (T) of this manipulator system consisted of the kinetic energy (T_G) generated by the motion of the system itself and the kinetic energy (T_i) generated by the elastic deformation of the flexible linkage. The kinetic energy of the system could be defined as

$$T = T_G + T_i, \quad (10)$$

$$T_G = \frac{1}{2} J_0 \dot{\theta}_0^2 + \frac{1}{2} m_0 l_0^2 \dot{\theta}_0^2 + \frac{1}{2} J_1 (\dot{\theta}_0 + \dot{\theta}_1)^2 + \frac{1}{2} m_1 (\dot{r}_1|_{x_1=l_1})^2 + \frac{1}{2} J_2 (\dot{\theta}_0 + \dot{\theta}_1 + \dot{\theta}_2)^2 + \frac{1}{2} m_2 (\dot{r}_2|_{x_2=l_2})^2, \tag{11}$$

$$T_i = \frac{1}{2} \int_0^{l_1} \rho_1 A_1 \dot{r}_1 \cdot \dot{r}_1 dx_1 + \frac{1}{2} \int_0^{l_2} \rho_2 A_2 \dot{r}_2 \cdot \dot{r}_2 dx_2. \tag{12}$$

The total elastic potential energy (U) of the manipulator system can be expressed as the sum of the gravitational potential energy (U_G) and the elastic potential energy (U_i) due to deformation. Thus, we can write the following:

$$U = U_G + U_i, \tag{13}$$

$$U_G = (m_0 + m_1 + m_2) g L_0 s_0 + (m_1 + m_2) g L_1 s_{01} + m_2 g L_2 s_{012}, \tag{14}$$

$$U_i = \frac{1}{2} \sum_{n=1}^2 \int_0^{l_i} E_i I_i \left(\frac{\partial^2 w_i(x_i, t)}{\partial x_i^2} \right)^2 dx_i = \frac{1}{2} K_1 q_1^2 + \frac{1}{2} K_2 q_2^2, \tag{15}$$

where

$$K_1 = E_1 I_1 \int_0^{l_1} \left(\frac{\partial^2 \Phi_1(x_1)}{\partial x_1^2} \right)^2 dx_1,$$

$$K_2 = E_2 I_2 \int_0^{l_2} \left(\frac{\partial^2 \Phi_2(x_2)}{\partial x_2^2} \right)^2 dx_2.$$

The generalized coordinates are given as $p = (\theta_1, \theta_2, q_1, q_2)^T$, where θ_1 and θ_2 denote the rigidity angles that the flexible linkage turns through and q_1 and q_2 are the modal coordinates of the flexible linkage. And the generalized force is $Q = (\tau_1, \tau_2, 0, 0)^T$. Neglecting the coupling terms for the productization and the difference categories in the trigonometric equation, the kinetic expression is deduced as follows:

$$\mathbf{M} \times \ddot{p} + \mathbf{H}(\dot{p}, p) + \mathbf{K} \times p + G = \tau, \tag{16}$$

where \mathbf{M} denotes the mass matrix, $\mathbf{H}(p, \dot{p})$ is the cross-coupling matrix, and \mathbf{K} denotes the stiffness matrix.

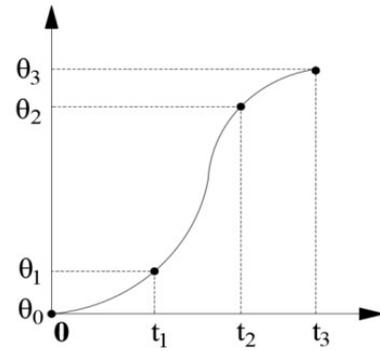


Figure 3. Illustration of the 3-5-3 schematic diagram of interpolation function.

3 Time-optimal trajectory planning for flexible manipulator

The manipulator has several possible paths to transition from the starting point to the designated destination. It is crucial to choose a seamless trajectory that guarantees system stability and precise control. To achieve this, a 3-5-3 interpolating polynomial is proposed to enhance the manipulator’s path, ensuring a smooth and viable trajectory.

The resolution of interpolating polynomials poses a complex challenge that necessitates adherence to specific constraints. As a solution, a customized adaptive particle swarm optimization technique, denoted as MAPSO, has been introduced to enhance the optimization process of the 3-5-3 polynomial algorithm. This advancement aims to generate a time-optimal trajectory curve, thereby focusing on the efficient utilization of time resources.

3.1 3-5-3 mixed polynomial interpolation

The study employs the 3-5-3 hybrid polynomial interpolation technique to address the challenge of trajectory planning for the manipulator’s end effector traversing various path points. Figure 3 illustrates the segmented polynomial interpolation diagram associated with this approach.

Any four known path points are selected in the motion trajectory for trajectory optimization design, which are noted as the start point $S_1(0, \theta_0)$, the path point $S_2(t_1, \theta_1)$, the path point $S_3(t_2, \theta_2)$, and the termination point $S_4(t_3, \theta_3)$, respectively. The trajectory is partitioned into three distinct segments by the specified path delineations, incorporating third-, fifth-, and third-degree polynomials successively.

1. Trajectory planning using cubic polynomials in the first stage (0 to t_1) is as follows:

$$\begin{cases} \theta_1(t) = a_{10} + a_{11}t_1 + a_{12}t_1^2 + a_{13}t_1^3, \\ \dot{\theta}_1(t) = a_{11} + 2a_{12}t_1 + 3a_{13}t_1^2, \\ \ddot{\theta}_1(t) = 2a_{12} + 6a_{13}t_1. \end{cases} \tag{17}$$

- 2. Within the second stage (t_1 to t_2), a fifth-degree polynomial is used for trajectory planning:

$$\begin{cases} \theta_2(t) = a_{20} + a_{21}t + a_{22}t^2 + a_{23}t^3 + a_{24}t^4 + a_{25}t^5, \\ \dot{\theta}_2 = a_{21} + 2a_{22}t + 3a_{23}t^2 + 4a_{24}t^3 + 5a_{25}t^4, \\ \ddot{\theta}_2 = 2a_{22} + 6a_{23}t + 12a_{24}t^2 + 20a_{25}t^3. \end{cases} \quad (18)$$

- 3. Trajectory planning using cubic polynomials in the third stage (t_2 to t_3) is as follows:

$$\begin{cases} \theta_3(t) = a_{30} + a_{31}t + a_{32}t^2 + a_{33}t^3, \\ \dot{\theta}_3 = a_{31} + 2a_{32}t + 3a_{33}t^2, \\ \ddot{\theta}_3 = 2a_{32} + 6a_{33}t. \end{cases} \quad (19)$$

At the initiation point S_1 and the conclusion point S_4 , assign zero values to their velocities and accelerations. The intermediate points S_2 and S_3 serve as connectors for the multi-segment interpolating polynomials, demanding consistent displacements, velocities, and accelerations at the boundary of these two points (Li, 2023).

The constraints are incorporated into Eqs. (17)–(19) and organized into matrix form as follows:

$$\mathbf{a} = \mathbf{A}^{-1}\theta, \quad (20)$$

where \mathbf{a} represents the coefficient matrix, θ denotes the combination matrix of four path points, and \mathbf{A} refers to the polynomial matrix with respect to time t .

3.2 Trajectory optimization based on MAPSO

The adaptive particle swarm optimization (APSO), as detailed by Zhang and Fu (2020), represents an optimization algorithm that leverages collective intelligence. Its fundamental concept involves emulating the social dynamics observed in bird flocks, facilitating efficient pursuit of optimal solutions through group-based collaborative search mechanisms.

The velocity and position formulation of the adaptive particle swarm optimization algorithm is as follows:

$$\begin{cases} v_{i,j}(t+1) = \omega \cdot v_{i,j}(t) + c_1 r_1 |pb_i(t) - x_{i,j}(t)| \\ \quad + c_2 r_2 |gb_i(t) - x_{i,j}(t)|, \\ x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1). \end{cases} \quad (21)$$

In this context, $v_{i,j}(t)$ represents the velocity of the i th particle during the j th interpolation process after t iterations, while it also signifies the position reached by the i th particle during the same process. ω denotes the inertia weight; c_1 and c_2 are the individual and population learning factors, respectively; and r_1 and r_2 are random numbers within the interval $[0, 1]$. $pb_i(t)$ indicates the individual best value found by the i th particle during the current search process, whereas $gb_i(t)$ represents the global best value discovered by all particles in the same process.

However, the inertia weight, as well as the individual and population factors in the APSO algorithm, is a fixed value.

This rigidity can lead to issues such as falling into local optima and a lack of adaptability to the search environment. To address these limitations, this paper proposes an enhancement to the APSO algorithm by optimizing the three parameters: inertia weight (ω), individual learning factor (c_1), and population learning factor (c_2). This results in the development of the MAPSO algorithm, which aims to improve the algorithm's flexibility and effectiveness in navigating the solution space.

3.2.1 Inertia weight (ω)

Transformation of constant inertia weights into nonlinear inertia weight functions using segmented tangent function:

$$\begin{cases} \omega = (\omega_{\max} - \omega_{\min}) \cdot \tan\left(0.875\left(1 - \frac{\text{iter}}{\text{iter}_{\max}}\right)^k\right) + \omega_{\min}, \\ \quad \text{iter} < 0.6 \cdot \text{iter}_{\max}, \\ \omega = 0.2 + 0.1 \cdot \text{rand}(), \\ \quad \text{iter} > 0.6 \cdot \text{iter}_{\max}, \end{cases} \quad (22)$$

where ω_{\max} is the maximum value of inertia weight, taking the empirical value of 0.9. ω_{\min} is the minimum value of inertia weight, taking the empirical value of 0.2. iter_{\max} is the number of iterations, and k is the control factor (generally taking 0.6).

3.2.2 Individual learning factor (c_1) and group learning factor (c_2)

A trigonometric function is used for dynamic adaptive adjustment, which closely combines the values of the two learning factors with the current iteration times of the particle. The function's representation is detailed as follows:

$$\begin{cases} c_1 = c_{\max} - (c_{\max} - c_{\min}) \cdot \sin^2\left(\frac{\pi}{2} \cdot \frac{\text{iter}}{v_{\max}}\right), \\ c_2 = c_{\min} + (c_{\max} - c_{\min}) \cdot \sin^2\left(\frac{\pi}{2} \cdot \frac{\text{iter}}{\text{iter}_{\max}}\right). \end{cases} \quad (23)$$

The optimization of the interpolation time for the 3-5-3 hybrid polynomial is achieved through the application of the MAPSO algorithm. Subsequently, the total time consumed by each joint, denoted as t_1 , t_2 , and t_3 , in executing the three-segment interpolation trajectory is aggregated to derive the fitness function for an individual joint. This fitness function is mathematically expressed as follows:

$$\min f = T = \sum_{t_0}^{t_f} t = \sum_{i=0}^1 (t_{i+1} - t_i). \quad (24)$$

3.2.3 Process of MAPSO

The precise procedures of the enhanced optimization algorithm are illustrated in Fig. 4.

1. Define the scope of particle attributes such as positions and velocities, the population magnitude denoted as m and the upper limit of iterations and establish the initial positions and velocities for each particle within the populace.
2. Log the most extreme values and present positions of the individual optimal and global optimal entities.
3. Utilize the N 3D temporal vectors derived from the initialization in Eq. (20) to determine the coefficients for each segmenting function. Subsequently, apply them to Eqs. (17) to (19) to compute the three-segment motion path, velocity, and acceleration for each articulation.
4. Determine the compliance of the velocity and acceleration exhibited by the three-segment trajectories pertaining to every joint of the robotic arm with the prescribed kinematic limitations. Should all three-segment trajectories adhere to the stipulated constraints, incorporate the duration of the three-segment trajectories into Eq. (24) for the computation of the fitness metric while documenting the prevailing positional data. Conversely, should the velocity or acceleration of any of the three-segment trajectories surpass the established constraints, assign a significantly elevated fitness value to the current particle to guarantee its exclusion during subsequent iterations.
5. Update the weight ω and learning factors c_1 and c_2 according to Eqs. (22) and (23), and incorporate them into the initial APSO equations to adjust the positions and velocities of the particles.
6. The sub-iteration loop is executed by assessing the fitness values of the individual pole and the global pole from which the particle exhibiting the highest fitness is chosen, representing the optimal pole.
7. If the termination criterion is not met, step 3 will be initiated, and the iteration will cease once the termination criterion is fulfilled.

3.3 Simulation analysis

The origin of the coordinate system was established at the joint motor 1, which controlled the flexible link L_1 . Initially, both flexible rods were aligned parallel to the ground. Four specific path points within the reachable space of the two flexible links were identified and are detailed in Table 2.

The initial total duration was established to be 6 s, with individual intervals of 2 s each. Utilizing MATLAB, a simulation was conducted to determine the convergence count and time for both the standard APSO algorithm and the novel MAPSO algorithm detailed in this study, as illustrated in Fig. 5.

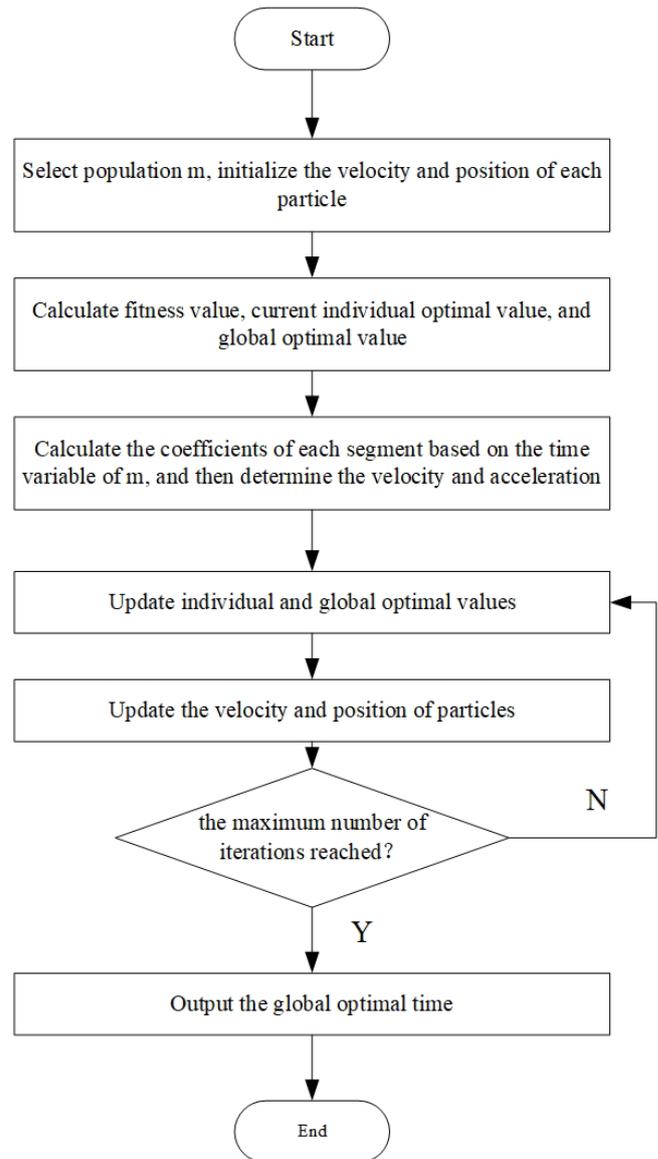


Figure 4. The process of the improved adaptive particle swarm optimization algorithm.

As shown in Fig. 5, comparing to the standard APSO algorithm, when the MAPSO algorithm optimized the target trajectories of the two flexible arms, the number of iterations of joint 1 decreased from 52 to 35, while the number of iterations of joint 2 decreased from 48 to 23. Meanwhile, the convergence time of the MAPSO algorithm was also better than that of the standard APSO algorithm. The specific comparison results regarding the number of global convergences and convergence times are presented in Table 3.

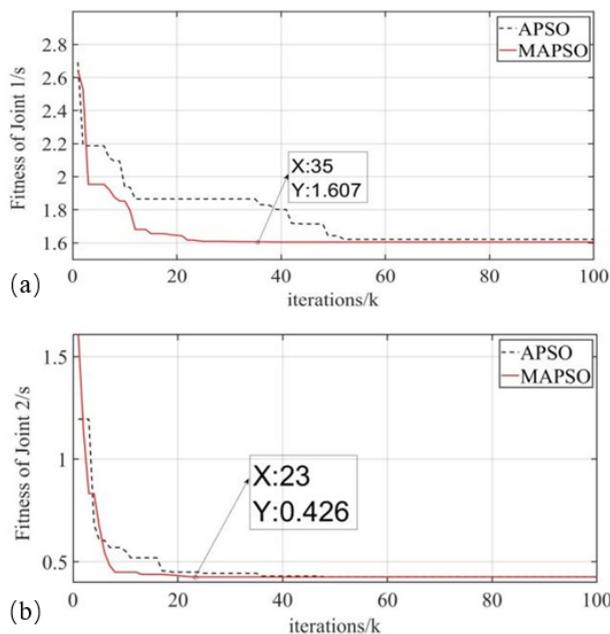
The time taken for each trajectory segment optimized by the MAPSO algorithm had been compared with the time required by the conventional setting method, as presented in Table 4.

Table 2. The target trajectory point parameter information of the flexible manipulator.

Control points ($S_i, i = 1$ to 4)	Coordinates of control points (x_i, y_i)	Angle of joint 1 (θ_1) [rad]	Angle of joint 2 (θ_2) [rad]
S_1 (starting point)	(1000, 0)	0	0
S_2 (path point 1)	(923.0678, 381.9104)	0.3491	0.0873
S_3 (path point 2)	(703.8806, 704.0296)	0.6981	0.1745
S_4 (endpoint)	(385.0977, 913.5447)	1.0472	0.2618

Table 3. Comparative analysis of two algorithms.

Joint (i)	The standard APSO algorithm		The MAPSO algorithm	
	The number of iterations	The convergence time [s]	The number of iterations	The convergence time [s]
1	52	1.622	35	1.607
2	48	0.427	23	0.426

**Figure 5.** Process of convergence of joints: (a) joint 1 and (b) joint 2.

As indicated in Table 4, the maximum values of the two joints for each segment trajectory were considered the overall running time for that segment: $t_1 = 0.6855$ s, $t_2 = 0.2354$ s, and $t_3 = 0.6845$ s. Consequently, the total running time of the robotic arm, t , was calculated to be 1.6054 s. This represented a reduction of 4.3946 s compared to the pre-optimization 3-5-3 segmented polynomial trajectory planning.

The variation curves of angular displacement, angular velocity, and angular acceleration for the action of interpolating polynomials optimized by the MAPSO algorithm are illustrated in Fig. 6.

The simulation results demonstrated that the two joints of the flexible manipulator can accurately navigate through each path point when driven by the MAPSO algorithm. Additionally, the velocities and accelerations of the joints adhered to the specified boundary constraints while achieving a time-optimal path. However, it is noteworthy that the values of the joints' velocities and accelerations were relatively high. Specifically, the maximum velocity for joint 1 was 1.56 rad s^{-1} , while for joint 2, it was 0.393 rad s^{-1} . The maximum accelerations recorded were 4.458 rad s^{-2} for joint 1 and 1.110 rad s^{-2} for joint 2. The graph clearly illustrates significant fluctuations during the intermediate stages, characterized by sharp changes in acceleration, which are attributed to the optimization of the MAPSO algorithm based on run-time.

4 Trajectory tracking strategy based on MAPSO-FuzzyPD

The studies indicated that while the MAPSO algorithm facilitates quicker attainment of target positions for the manipulator, it simultaneously elevated the speed and acceleration during operation, leading to increased vibrations. Consequently, it was essential to investigate vibration suppression techniques in trajectory tracking to mitigate the adverse effects of these vibrations on the robotic arm system. Implementing effective vibration control strategies will enhance the overall stability and performance of the manipulator, ensuring smoother operation and improved accuracy in task execution.

After trajectory planning, it is essential to employ algorithms and other methods to ensure that the robotic arm accurately follows the intended path (Chen et al., 2022). To address the inaccuracies in the dynamics model and reduce jitter in the tracking system, this paper proposed a MAPSO-FuzzyPD controller. This controller was designed with ro-

Table 4. Comparative table of time spent on each trajectory segment.

Joint	Total time [s]	First trajectory time [s]	Second trajectory time [s]	Third trajectory time [s]
General set points	6	2	2	2
1	1.6052	0.6855	0.2354	0.6845
2	0.4259	0.1591	0.1	0.1668

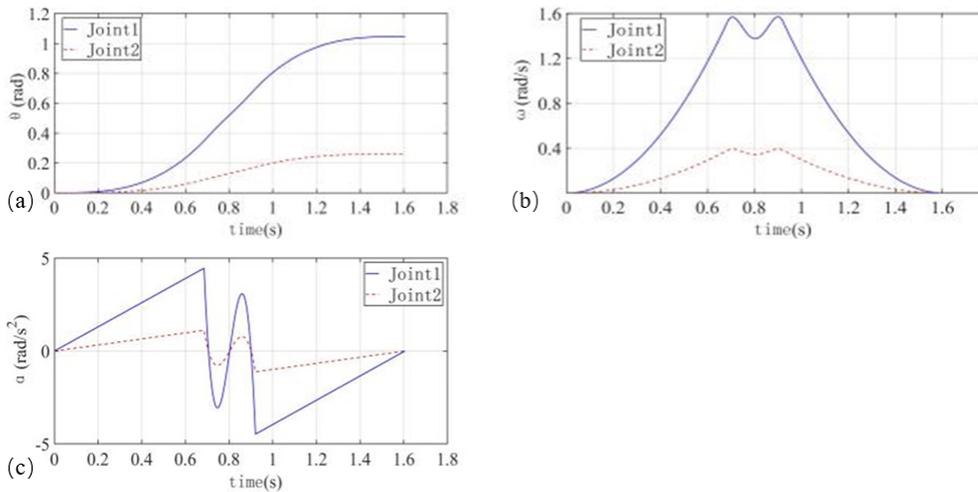


Figure 6. Improved algorithm 3-5-3 joint variation parameters under the action of hybrid polynomial interpolation. (a) Angular displacement, (b) angular velocity and (c) angular acceleration.

business to counteract the uncertainties affecting joint trajectory tracking control. By integrating the MAPSO algorithm with a Fuzzy-PD control strategy, the proposed approach aimed to enhance the tracking performance of each joint in the manipulator, ensuring smoother and more precise execution of the planned trajectory.

4.1 Fitting of joint angular displacements

The input for the trajectory tracking control consists of the joint angular displacements calculated in Sect. 3. To enhance the accuracy, smoothness, and stability of the controller’s output as well as to prevent potential issues during the solution process, the angular displacement curve is further refined.

A higher-order polynomial function was used to fit and optimize the angular displacements of the two joints, and the joint path points in Table 2 were added as constraints to obtain a smoother target trajectory curve. The errors in the fitted curves of angular displacements of the two joints with the original 3-5-3 interpolation are shown in Fig. 7.

Given that polynomials of a degree higher than nine were susceptible to overfitting, the analysis indicated that seventh- and eighth-degree polynomials yield smaller fitting errors. Additionally, the 3-5-3 mixed polynomials exhibit odd function properties, which may not be ideal for this application.

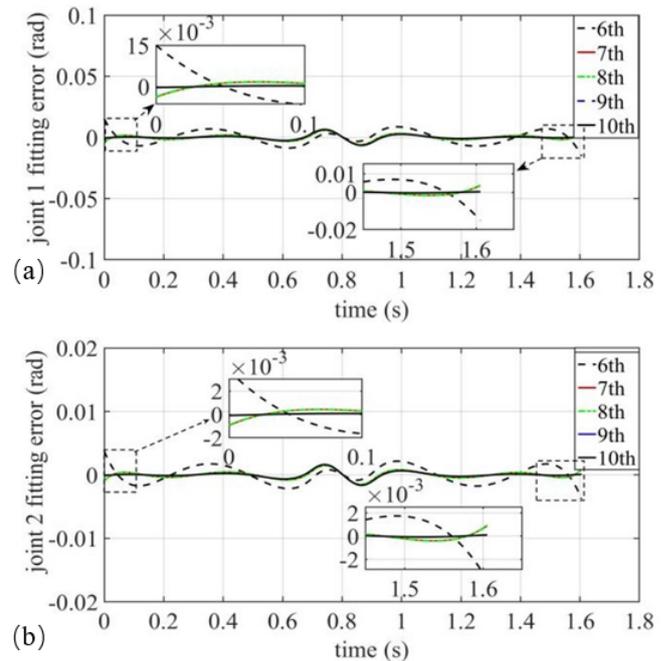


Figure 7. Fitting error in the *n*th-order polynomial fitted to the angular displacements of the two joints: (a) joint 1 and (b) joint 2.

Therefore, this paper selected the fitting curve of the seventh-degree polynomial as the final target curve. This choice balanced complexity and accuracy, ensuring a smooth trajectory while minimizing the risk of overfitting.

4.2 Fuzzy PD-based trajectory tracking controller

The integral component of PID control serves to eliminate steady-state error. However, it can also amplify noise and interference within the system. This amplification often leads to slow response times and unstable outputs, which can cause noticeable jitter in the operation of flexible robotic arms. To address these issues, this paper employed a fuzzy PD controller for trajectory tracking control. The fuzzy PD controller not only compensates for errors in trajectory tracking but also enhances control accuracy and increases the robustness of the system. The control block diagram for the fuzzy PD controller is illustrated in Fig. 8, demonstrating its application in improving trajectory tracking performance.

In Fig. 8, $e(t)$ is the joint angle following error, $e_c(t)$ is the rate of change in the following error of the joint angle, and ΔK_P and ΔK_D are the supplementary parameters corrected by the fuzzy controller. At this time, the fuzzy PD controller has two inputs, e and e_c , and two outputs, ΔK_P and ΔK_D . ΔK_P and ΔK_D are corrected by the fuzzy controller in real time by the action of the fuzzy rules, and the outputs are connected and externalized to the PD controller, which is combined with the given initial values of the PD parameters, K_{P0} and K_{D0} , and then finally applied to the robotic arm system. The fuzzy PD parameter self-tuning controller expression is obtained as follows:

$$\begin{cases} K_P = K_{P0} + \Delta K_P, \\ K_D = K_{D0} + \Delta K_D. \end{cases} \quad (25)$$

In this study, the triangular membership function was selected for the fuzzy control solution. Drawing from PD parameter tuning experiences and manual testing, the input variable domains for e and e_c were defined to be within $[-0.5, 0.5]$, while the output variable domains for ΔK_P and ΔK_D were set to $[-15, 15]$. The triangular membership function employs the center of gravity method, resulting in seven fuzzy sets: negative large (NB), negative medium (NM), negative small (NS), zero (ZO), positive small (PS), positive medium (PM), and positive large (PB). Based on expert knowledge and relevant literature regarding the regulation of these variables, as well as the fundamental principles of control quantity adjustment, the fuzzy rules governing the control of ΔK_P and ΔK_D parameters have been compiled and are presented in Tables 5 and 6. These rules facilitate effective tuning of the fuzzy PD controller, ensuring improved trajectory tracking performance.

Table 5. The fuzzy rule control table of ΔK_P .

U	E						
	NB	NM	NS	ZO	PS	PM	PB
NB	PS	PS	ZO	ZO	ZO	PB	PB
NM	NS	NS	NS	NS	ZO	NS	PM
NS	NB	NB	NM	NS	ZO	PS	PM
EC ZO	NB	NM	NM	NS	ZO	PS	PM
PS	NB	NM	NS	NS	ZO	PS	PS
PM	NM	NS	NS	NS	ZO	PS	PS
PB	PS	ZO	ZO	ZO	ZO	PB	PS

Table 6. The fuzzy rule control table of ΔK_D .

U	E						
	NB	NM	NS	ZO	PS	PM	PB
NB	PB	PB	PM	PM	PS	PS	
NM	PB	PB	PM	PM	PS	ZO	
NS	PM	PM	PM	PS	ZO	NS	
EC ZO	PM	PS	PS	ZO	NS	NM	
PS	PS	PS	ZO	NS	NS	NM	
PM	ZO	ZO	NS	NM	NM	NM	
PB	ZO	NS	NS	NM	NM	NB	

4.3 MAPSO-based fuzzy PD trajectory tracking controller

While the fuzzy controller effectively adjusts the parameters of the PD controller, the initial parameters K_{P0} and K_{D0} must still be determined manually through experiments, which can lack rigor. To address this, this paper employed an improved adaptive particle swarm optimization (MAPSO) algorithm, known for its straightforward principles, strong learning capabilities, and ease of convergence. The proposed approach combined the MAPSO algorithm with fuzzy PD control to formulate a MAPSO-FuzzyPD controller. This design first optimized the initial values of the proportional and derivative parameters, followed by fine-tuning these values using the fuzzy controller. This combination enhanced the overall control performance and robustness of the system. The improved schematic block diagram illustrating this methodology is presented in Fig. 9.

The optimized process of the improved adaptive particle swarm optimization algorithm discussed in Sect. 3.2.3, along with the modified structure diagram of the fuzzy PD controller, is illustrated in Fig. 10 for the fuzzy PD control system based on MAPSO.

The simulation model built using Simulink is shown in Fig. 11.

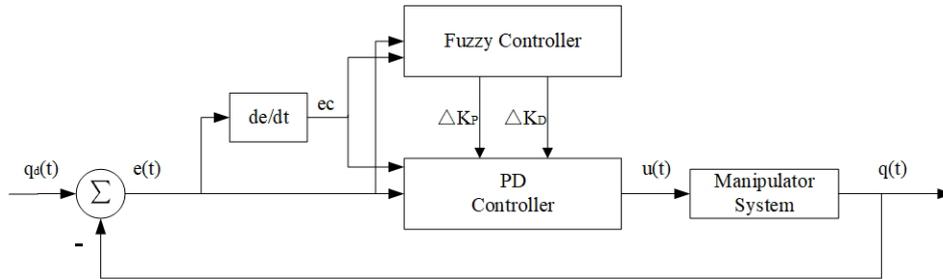


Figure 8. Fuzzy PD control block diagram.

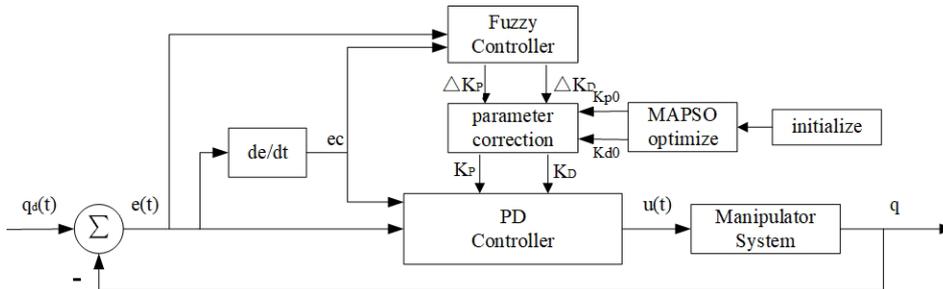


Figure 9. Block diagram of MAPSO-based fuzzy PD controller structure.

4.4 Simulation analysis of MAPSO-FuzzyPD controller

To validate the effectiveness of the control strategy proposed in this paper, a numerical simulation analysis was conducted. The population size (N) was set to 30, the maximum number of iterations ($iter_{max}$) was 100, and the dimension count was 4, representing the proportional control of two joints, along with the initial values of differential parameters, which were bounded within a range of $\pm[200, 200, 125, 125]$. The initial calculation of proportional and differential values was performed using the MAPSO algorithm, with corresponding code written for continuous iteration to obtain the optimal values based on the fitness function. These optimal values were then stored in the MATLAB workspace. Simulation was conducted using the trajectory tracking controller established in Fig. 11. During the trajectory tracking process, the PD control section automatically retrieved the optimal initial values of K_{P0} and K_{D0} from the workspace, incorporating fuzzy control for real-time adjustments. The results of the simulation, specifically the error between the theoretical and actual tracking angles of the two joints, are illustrated in Fig. 12.

The illustration in Fig. 12 demonstrates that the enhanced fuzzy PD controller exhibits superior performance in real-time tracking of joints 1 and 2 when contrasted with both the PD and the fuzzy PD controllers. Notably, it effectively aligns the joint trajectories with the desired targets, eliminating initial oscillations observed during trajectory tracking.

The angular velocity tracking and control torque curves for the three control strategies are shown in Fig. 13.

As shown in Fig. 13, the improved Fuzzy PD controller effectively eliminated velocity and moment oscillations during the initial phase compared to previous control strategies. Specifically, the MAPSO-FuzzyPD controller reduced the maximum tracking velocity for joint 1 from 0.22 to 0.14 rad s^{-1} , with the duration of this peak velocity being shortened to just 0.01 s. For joint 2, the maximum tracking velocity decreased from 0.067 to 0.04 rad s^{-1} , also with a duration reduction to 0.01 s. Regarding torque, the variation range of the starting torque for joint 1 was significantly improved, now fluctuating between -1.1 and 12.4 N m. This represented a reduction of 61.3 % compared to the maximum starting torque observed with the PD controller and a 40.3 % decrease compared to the fuzzy PD controller. For joint 2, the starting torque variation was reduced to between -3.8 and 0.52 N m, which was 57.9 % less than the maximum starting torque of the PD controller and 42.1 % less than that of the fuzzy PD controller. These results highlighted the enhanced stability and performance of the MAPSO-FuzzyPD controller during the onset phase, leading to smoother operation with reduced oscillations.

The simulation results clearly demonstrated the superiority of the MAPSO-FuzzyPD control strategy.

5 Experimentation and analysis

The physical platform of the 2-degree-of-freedom flexible manipulator built in this paper was mainly composed of three parts: mechanical system, electrical system, and control system, as shown in Fig. 14. The electrical system comprised

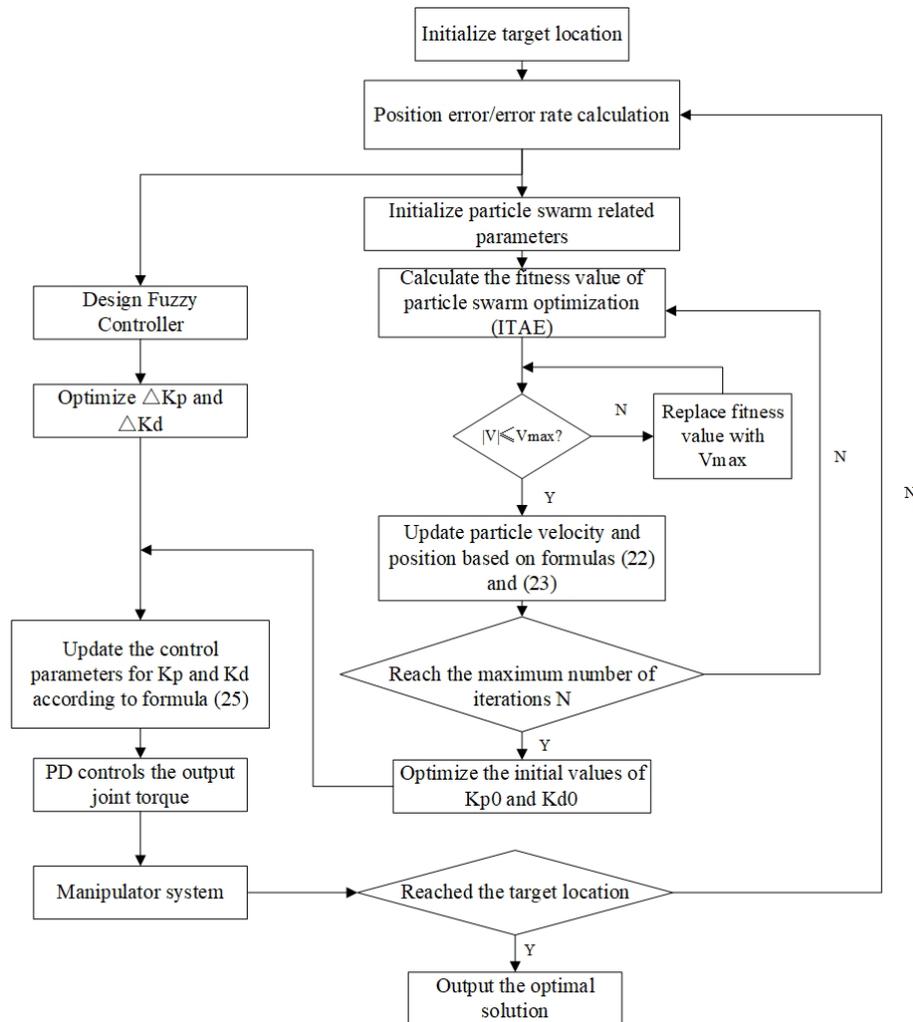


Figure 10. The fuzzy PD control system based on MAPSO.

several key components: articulated motor 1 (HT8115-J9), articulated motor 2 (HT8108-J9), a motor power supply, and an acceleration sensor (Weite Intelligent six-axis accelerometer gyroscope WT61C-TTL). The control system consisted of a PC host and an STM32 control board, which work together to manage and execute the control algorithms. This setup allowed for precise coordination of the motors and monitoring of system performance through the acceleration sensor, enabling effective control and feedback mechanisms essential for optimal operation.

The control strategy involved developing a control program for the articulated motors using MDK5. This program was then uploaded to the STM32 control board via an ST-LINK, facilitating communication between the PC host and the lower computer. Once the commands were received, the STM32 control board parsed them and executed the necessary arithmetic operations using its internal processor. Based on these calculations, it determined the appropriate control signals for each joint motor. These control signals were sub-

sequently transmitted to the respective joint motors through the controller area network (CAN) bus, ensuring precise and coordinated movement of the articulated system. This setup allowed for real-time control and adjustment, enhancing the overall responsiveness and performance of the system.

The experiment employed the optimized multi-path joint angle function from Sect. 4.1 as the target trajectory for the motor joint angles. The motor control program was integrated to enable the motor-controlled manipulator to follow four specified target paths, as detailed in Table 2. This movement was executed under three different control strategies: PD control, fuzzy PD control, and MAPSO-FuzzyPD control. The primary goals of the experiment were to assess the trajectory tracking performance of joint 1 and joint 2 and to evaluate the vibration characteristics at the end of flexible arm 2, specifically at the end effector.

The total duration of the experiment was 1.61 s, with a data collection precision set at 0.01 s. The data acquisition system extracted real-time data from various positions and

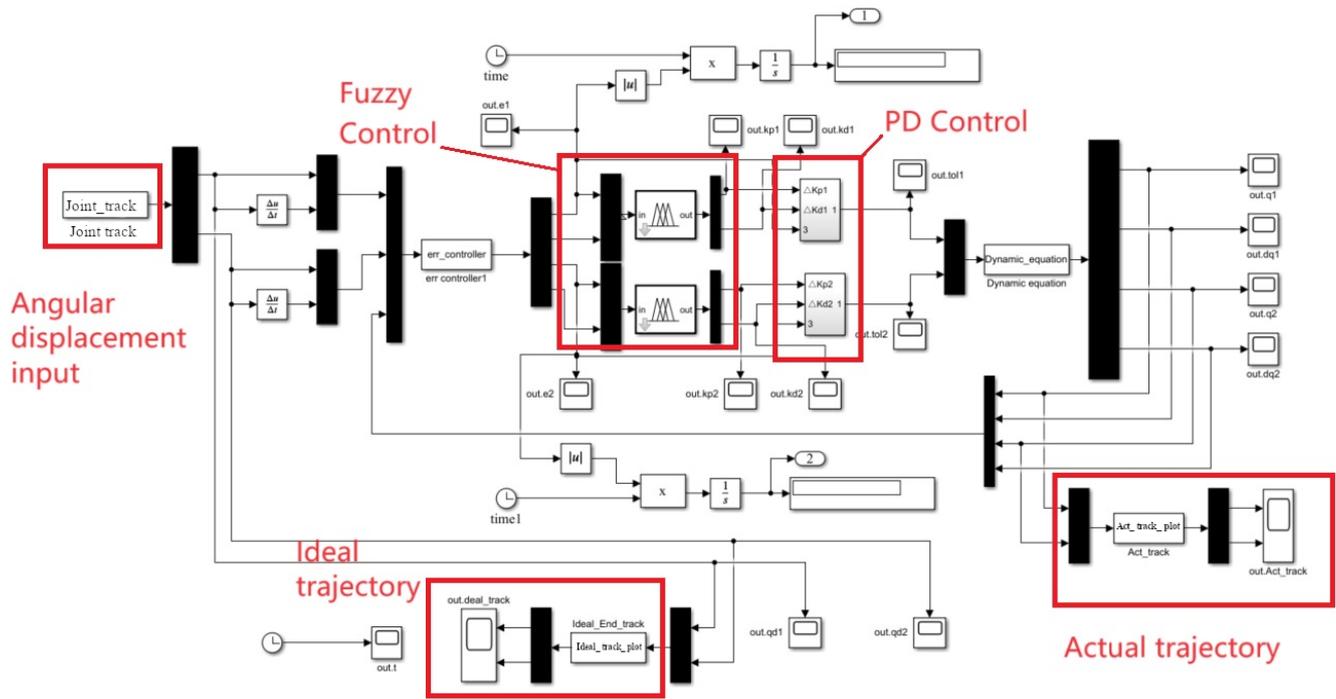


Figure 11. Simulink simulation model.

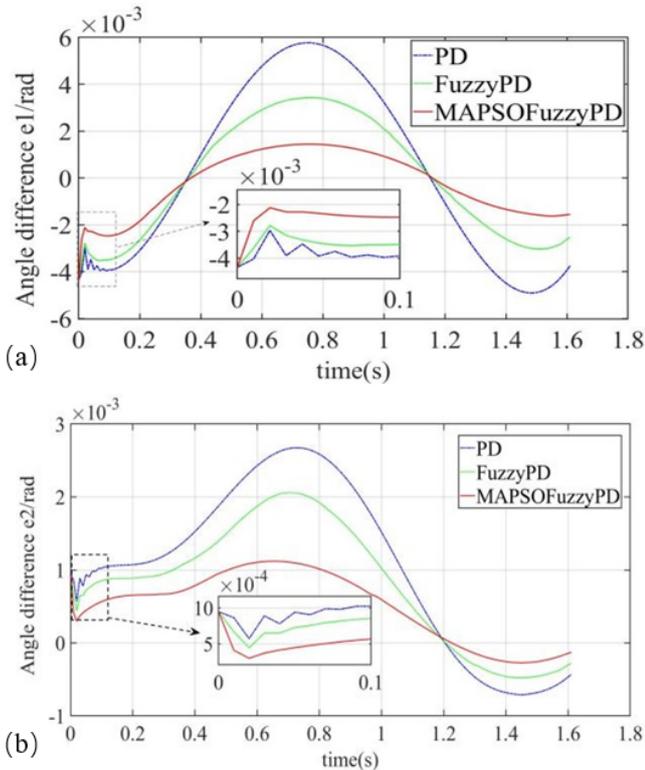


Figure 12. Angular difference from ideal trajectory with three control strategies: (a) joint 1 and (b) joint 2.

exported them to a spreadsheet for subsequent data fitting using MATLAB. And the angular trajectory following curve and the vibration curve at the end of the flexible manipulator is as shown in Figs. 15 and 16, respectively.

In Fig. 15, the overall trend indicated that under the influence of the three control strategies, the motors driving joints 1 and 2 were able to closely track the target trajectory in real time, maintaining smooth operation within an acceptable error range. However, during the trajectory tracking phase, it was observed that both PD control and fuzzy PD control exhibit a greater deviation from the target trajectory compared to the MAPSO-FuzzyPD control. This suggested that the MAPSO-FuzzyPD control provided better accuracy in trajectory tracking, resulting in reduced error and enhanced performance of the flexible manipulator throughout the experiment. Moreover, the results indicated that, compared to the desired trajectory, all three methods exhibit a degree of lag due to uncertainties such as friction and aerodynamic damping present in the physical model during motion. Additionally, the flexibility and large length-to-diameter ratio of the flexible manipulator contribute to this lag, potentially leading to deviations in end-effector positioning. Future studies could focus on optimizing performance by integrating vibration control techniques.

Figure 16 shows the real-time variation curves of the vibration acceleration components at the end of the manipulator under the three control actions, and the end vibration situations were very close to each other due to the small error between the actual tracking trajectories of the three con-

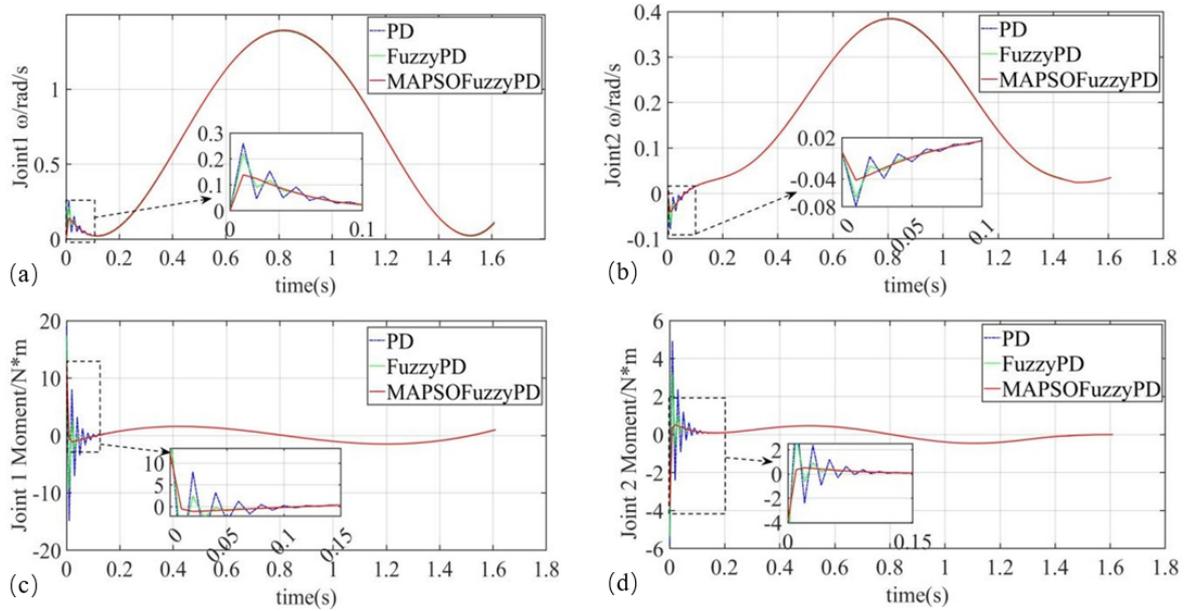


Figure 13. Angular velocity tracking and control torque curves. (a) Angle tracking velocity of joint 1. (b) Angle tracking velocity of joint 2. (c) Control torque of joint 1. (d) Control torque of joint 2.

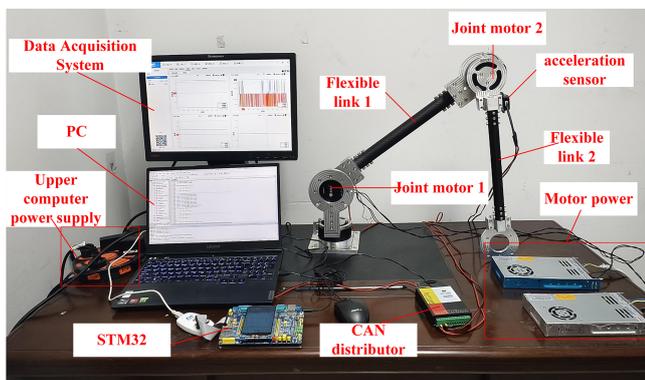


Figure 14. Experimental platform.

trollers. During the initial phase (0 to 0.4 s), the vibration acceleration amplitude at the end of the arm in the x direction for all three control strategies was nearly zero, indicating the successful implementation of a “soft start” in this direction. In contrast, in the y direction, the effects of gravity lead to a brief blocking torque during the instantaneous start, resulting in a significant amplitude of vibration at the arm’s end. However, all three controllers managed to return the end of the arm to smooth tracking within 0.02 s, with the MAPSO-FuzzyPD control demonstrating slightly lower acceleration amplitudes compared to the other two controllers. From 0.4 to 1.61 s, the tracking speeds of the three control strategies showed minimal differences, leading to similar vibration levels at the end of the flexible arm during this period. As the joints approached the target position, the vibration acceler-

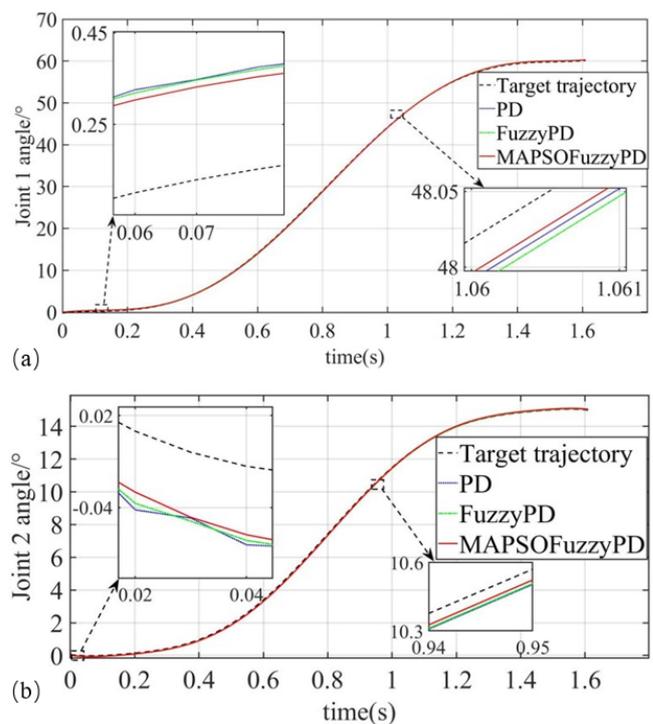


Figure 15. Joint angles under various control strategies: (a) joint 1 and (b) joint 2.

ation diminished due to the gradual reduction in speed. After 1.61 s, once the two joints reached their target positions smoothly, the absence of active and passive vibration suppression strategies at the end of the arm results in residual

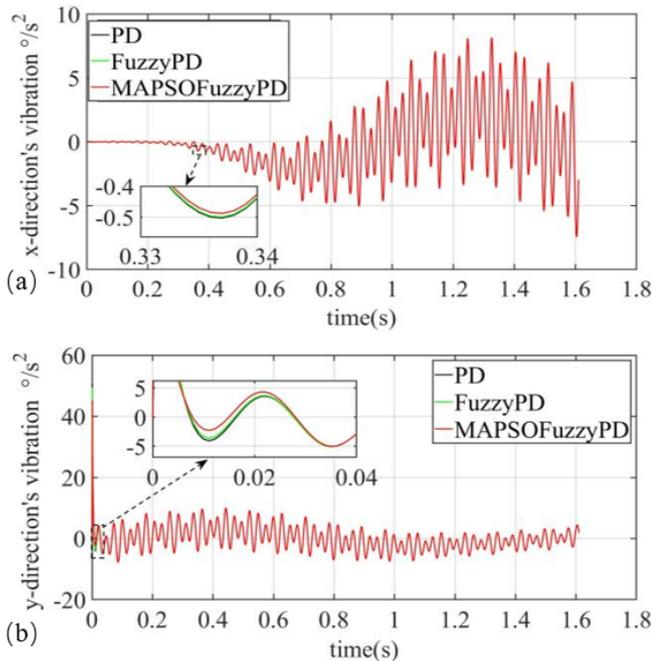


Figure 16. Vibration at the end of the manipulator. (a) Vibration of the end in the x direction. (b) Vibration of the end in the y direction.

vibrations of the flexible arm, highlighting an area for further investigation in future studies.

Based on the experimental results, both joints exhibit smooth trajectory tracking with minimal deviation. Notably, the MAPSO-FuzzyPD controller outperforms the other two control methods in terms of both trajectory tracking accuracy and vibration control. This demonstrates the effectiveness and advantages of the proposed control strategy in this paper, highlighting its potential for improved performance in robotic applications. The findings suggest that the MAPSO-FuzzyPD controller could be a valuable approach for enhancing the stability and responsiveness of robotic systems in dynamic environments.

6 Conclusions

In this paper, the 2-degree-of-freedom flexible manipulator was examined as the primary research object, focusing on three key aspects: dynamics modeling, trajectory planning, and trajectory tracking. The main conclusions drawn from this study are as follows:

1. The flexural deformation of the pliable manipulator was elucidated through the utilization of the assumed modal approach, and the dynamic equations governing the 2-degree-of-freedom manipulator were formulated through the amalgamation of the Lagrangian method.
2. Aiming at the problems of solving the coefficients of 3-5-3 interpolating polynomials and the difficulty of se-

lecting the optimal value, the adaptive particle swarm optimization algorithm was improved by a nonlinear segmented tangent inertia weight function and dynamically adjusting the learning factor based on a trigonometric function. A 3-5-3 hybrid polynomial algorithm based on MAPSO was proposed, and simulation experiments were carried out on the basic trajectory with time as the optimization objective. Simulation results indicated that both joints of the flexible manipulator accurately traversed each path point under the proposed algorithm. The resulting position, velocity, and acceleration profiles were smooth, continuous, and free of abrupt changes, thereby satisfying the specified boundary constraints.

3. A trajectory tracking controller based on MAPSO-FuzzyPD was constructed by modifying the fuzzy PD controller with the MAPSO algorithm and analyzed by simulation experiments. The outcomes of the simulation reveal notable improvements when comparing FuzzyPD with respect to tracking speed and torque control. Specifically, the maximum tracking speed of joint 1 decreased from 0.22 to 0.14 rad s^{-1} , with a reduced duration of 0.01 s . Similarly, the maximum tracking speed of joint 2 decreased from 0.067 to 0.04 rad s^{-1} , also with a duration reduction to 0.01 s . Consequently, the efficacy in determining the optimal PD control parameter initial value was significantly enhanced. In terms of torque, the range of starting torque for joint 1 decreased to -1.1 and 12.4 N m , representing a 61.3% and 40.3% decrease compared to the maximum starting torque of PD control and FuzzyPD, respectively. Likewise, the starting torque range for joint 2 decreased to -3.8 and 0.52 N m , indicating a 57.9% and 42.1% decrease compared to the maximum starting torque of PD control and FuzzyPD, respectively. These results demonstrated the algorithm's effectiveness in mitigating the initial vibrational phenomena observed in tracking angular displacement, angular velocity, and torque, thereby facilitating smoother motor and robotic arm control operations.

Finally, an experimental platform for the flexible manipulator was established, and the experimental results further validated the effectiveness and feasibility of the algorithm proposed in this paper concerning joint trajectory tracking.

Data availability. All the data used in this paper can be obtained from the corresponding author upon request.

Author contributions. WS, KD, and FM designed the proposed approach algorithm and experiments. YJ and KD carried them out and prepared the results. YJ, WS, and ZG prepared the paper with contributions from all co-authors.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors.

Review statement. This paper was edited by Daniel Condurache and reviewed by two anonymous referees.

References

- Alandoli, E. A., Lee, T. S., Lin, Y. J., and Vijayakumar V.: Dynamic model and intelligent optimal controller of flexible link manipulator system with payload uncertainty, *Arab. J. Sci. Eng.*, 46, 7423–7433, <https://doi.org/10.1007/s13369-021-05436-7>, 2021.
- Anavatti, S. G., Salman S. A., and Choi, J. Y.: Fuzzy + PID controller for robot manipulator, in: 2006 International Conference on Computational Intelligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA'06), Sydney, NSW, Australia, 28 November–1 December 2006, IEEE, 75–75, <https://doi.org/10.1109/CIMCA.2006.103>, 2006.
- Beata, J.: Fuzzy logic controller for robot manipulator control system, in: Proceedings of Applications of Electromagnetics in Modern Techniques and Medicine (PTZE), Raclawice, Poland, 9–12 September 2018, IEEE, 77–80, <https://doi.org/10.1109/PTZE.2018.8503205>, 2018.
- Chen, Y. H., Chen, Y. Y., Lou, S. J., and Huang, C. J.: Energy saving control approach for trajectory tracking of autonomous mobile robots, *Intell. Autom. Soft Co.*, 31, 357–372, <https://doi.org/10.32604/IASC.2022.018663>, 2022.
- Chhabra, H., Mohan, V., Rani, A., and Singh, V.: Robust nonlinear fractional order fuzzy PD plus fuzzy I controller applied to robotic manipulator, *Neural Comput. Appl.*, 32, 2055–2079, <https://doi.org/10.1007/s00521-019-04074-3>, 2019.
- Gao, H. J., He, W., Chen, Z., and Sun, C. Y.: Neural network control of a two-link flexible robotic manipulator using assumed mode method, *IEEE T. Ind. Inform.*, 15, 755–765, <https://doi.org/10.1109/TII.2018.2818120>, 2019.
- He, W., Gao, H., Zhou, C., Yang C., and Li, Z. Q.: Reinforcement learning control of a flexible two-link manipulator: an experimental investigation, *IEEE T. Syst. Man Cy.-S*, 51, 1–11, <https://doi.org/10.1109/TSMC.2020.2975232>, 2021.
- Lee, H., Choi, Y., and Yi, B. J.: Stackable 4-BAR manipulators for single port access surgery, *IEEE-ASME T. Mech.*, 17, 157–166, <https://doi.org/10.1109/TMECH.2010.2098970>, 2012.
- Li, H.: Research on trajectory planning and trajectory tracking control of multi-joint manipulator, Master thesis, Kunming University of Science and Technology, China, 102 pp., <https://doi.org/10.27200/d.cnki.gkmlu.2023.000221>, 2023.
- Liu, F. C., Liang, L. H., and Gao, J. J.: Fuzzy PID control of space manipulator for both ground alignment and space applications, *Int. J. Autom. Comput.*, 11, 353–360, <https://doi.org/10.1007/s11633-014-0800-y>, 2014.
- Liu, Z., Song, L. B., Ye, L., and Pan, B. Z.: Comparison of finite element and experimental modal analysis of multi-joint flexible robotic arm, in: 2017 International Conference on Mechanical, System and Control Engineering (ICMSC), St. Petersburg, Russia, 19–21 May 2017, IEEE, 96–100, <https://doi.org/10.1109/ICMSC.2017.7959450>, 2017.
- Meng, Q. X., Lai, X. Z., Ze, Y., Su, C. Y., and Wu, M.: Motion planning and adaptive neural tracking control of an uncertain two-link rigid-flexible manipulator with vibration amplitude constraint, *IEEE T. Neur. Net. Lear.*, 33, 3814–3828, <https://doi.org/10.1109/TNNLS.2021.3054611>, 2022.
- Nagesh Rao, S. P., Lopes, G. A. D., Jeltsema, D., and Babuška, R.: Passivity-based reinforcement learning control of a 2-DOF manipulator arm, *Mechatronics*, 24, 1001–1007, <https://doi.org/10.1016/j.mechatronics.2014.10.005>, 2014.
- Omidi, E. and Mahmoodi, S. N.: Vibration suppression of distributed parameter flexible structures by integral consensus control, *J. Sound Vib.*, 364, 1–13, <https://doi.org/10.1016/j.jsv.2015.11.020>, 2016.
- Sasaki, M., Muguro, J., Kitano, F., Njeri, W., Maeno, D., and Matsushita, K.: Vibration and position control of a two-link flexible manipulator using reinforcement learning, *Machines*, 11, 754, <https://doi.org/10.3390/machines11070754>, 2023.
- Shao, M., Huang, Y., and Silberschmidt, V. V.: Intelligent manipulator with flexible link and joint: modeling and vibration control, *Shock Vib.*, 2020, 4671358, <https://doi.org/10.1155/2020/4671358>, 2020.
- Shen, H. and Fan, J. H.: Dynamic modeling and simulation of a folding-link flexible manipulator based on the Bezier interpolation method, *J. Vib. Eng. Technol.*, 12, 1525–1535, <https://doi.org/10.1007/s42417-023-00923-7>, 2024.
- Sun, W. W., Dai, K., Liu, Y., Ma, F., Guo, Z. Y., Jiang, Z. X., and Suo, S. F.: Dynamic modeling and optimization analysis of rigid-flexible coupling manipulator based on assumed mode method, *Int. J. Struct. Stab. Dy.*, 2550047, <https://doi.org/10.1142/S0219455425500476>, 2024a.
- Sun, W. W., Dai, K., and Ma, F.: Review of modeling, control and trajectory planning of space flexible manipulators, *Science Technology and Engineering*, 24, 34–60, <https://doi.org/10.12404/j.issn.1671-1815.2209809>, 2024b.
- Tang, L., Liu, Y. J., and Tong, S.: Adaptive neural control using reinforcement learning for a class of robot manipulator, *Neural Comput. Appl.*, 25, 135–141, <https://doi.org/10.1007/s00521-013-1455-2>, 2014.
- Tavasoli, A.: Adaptive nonlinear boundary control of a hybrid Euler–Bernoulli beam with coupled rigid and flexible dynamics, *IJST-T. Mech. Eng.*, 42, 311–315, <https://doi.org/10.1007/s40997-017-0097-x>, 2018.
- Wang, J., Pi, Y. J., Hu, Y. M., Zhu, Z. C., and Zeng L. B.: Adaptive simultaneous motion and vibration control for a multi flexible-link mechanism with uncertain general harmonic disturbance, *J. Sound Vib.*, 408, 60–72, <https://doi.org/10.1016/j.jsv.2017.07.024>, 2017.
- Wang, W. R., Zhao, K., Lin, Z. Q., and Wang, H.: Erratum to: Evaluating interactions between the heavy forging process

- and the assisting manipulator combining FEM simulation and kinematics analysis, *Int. J. Adv. Manuf. Tech.*, 48, 481–491, <https://doi.org/10.1007/s00170-009-2285-3>, 2010.
- Yu, J. Z., Wang, C., and Xie, G. M.: Coordination of multiple robotic fish with applications to underwater robot competition, *IEEE T. Ind. Electron.*, 63, 1280–1288, <https://doi.org/10.1109/TIE.2015.2425359>, 2016.
- Zhang, S., Zhang, Y. H., Zhang, X. N., and Dong, G. X.: Fuzzy PID control of a two-link flexible manipulator, *J. Vibroeng.*, 18, 250–266, wos:000370627300022, ISSN 1392-8716, 2016.
- Zhang, S., Zhao, Y., and Lee, J.: Control of 3-DOF robotic manipulator by neural network based Fuzzy-PD, *Int. J. Precis. Eng. Man.*, 20, 2109–2118, <https://doi.org/10.1007/s12541-019-00203-z>, 2019.
- Zhang, Y. Y., Zhang, Y. H., Zhang, H., and Yu, J. B.: Effects of beam end constraint condition and corner haunch on bending simple harmonic vibration of box girder, *Journal of Vibration and Shock*, 42, 21–28+40, <https://doi.org/10.13465/j.cnki.jvs.2023.23.003>, 2023.
- Zhang, W. M. and Fu, S. X.: Time-optimal trajectory planning of dulcimer music robot based on PSO algorithm, in: 2020 Proceedings of Chinese Control and Decision Conference (CCDC), Hefei, China, 22–24 August 2020, IEEE, 4769–4774, <https://doi.org/10.1109/CCDC49329.2020.9164017>, 2020.