White Shark Optimizer Based Time-optimal Trajectory Planning for Supernumerary Robotic Limbs

Jiayu Wang, Ruyue Sun, Yukun Zheng*, Rui Song*, and Yibin Li

***Abstract*—This paper presents an improved time-optimal trajectory planning algorithm leveraging the White Shark Optimizer (WSO), specifically designed for 6-degree-of-freedom (DOF) Supernumerary Robotic Limbs (SRL). This approach effectively addresses prevalent issues in trajectory planning, including low efficiency, premature convergence, and unstable operation. Beginning with an examination of the SRL's structural peculiarities, an auxiliary joint is incorporated to rectify the coordinate offset, facilitating subsequent inverse kinematics analysis. By utilizing the WSO algorithm, we develop a time-optimal trajectory optimization method within the joint space of the SRL. Moreover, to address the continuity challenges in the terminal movements of the SRL, we introduce a 3-5-3 piecewise polynomial trajectory planning method that integrates both 3-degree and 5-degree polynomials. Building on this foundational work, we present an optimization algorithm rooted in WSO to identify the most efficient time trajectory optimization strategy, considering multiple constraints. The effectiveness of the proposed optimization strategy is demonstrated through comprehensive co-simulation. A comparative analysis with current optimization algorithms highlights the advantages of the proposed time-optimal trajectory planning algorithm.**

***Index Terms*—Supernumerary Robotic Limbs, trajectory planning, time optimization algorithm, polynomial interpolation.**

I. Introduction

AS labor-centric industries experience progressive transformation and modernization [1], robotic technologies have expanded across various sectors, including automotive, electronics, food processing, chemical, and metal manufacturing. This growth is attributed to their increased efficiency and operational stability [2]. In the context of complex and labor-intensive tasks, Human-Robot Collaboration (HRC) has emerged as a leading trend within the rapidly evolving domain of robotics. To bring the concept of HRC to fruition, researchers and industry leaders worldwide have developed a range of collaborative robots, delving into the nuances of collaborative robot technologies [3]–[5].

Supernumerary Robotic Limbs (SRL), a subset of collaborative robots, represent wearable assistive devices designed to augment human limb functionality. They aim to enhance human activities, sensory perception, and manual dexterity [6]–[9]. Due to its outstanding operational efficiency and tangible economic benefits, SRL has found widespread application in defense and military operations, nuclear power management, medical rehabilitation [10], [11], and disaster relief and rescue efforts [12], [13]. Trajectory planning is a cornerstone of robot control research and plays a crucial role in the overall motion performance of robotic systems. With the integral human-machine interaction intrinsic in SRL applications, researchers globally have focused on trajectory optimization strategies. These efforts aim to enhance SRL's adaptability in complex and dynamic work environments [14]–[16], boost its operational effectiveness, and support its contribution to achieving operational objectives [17]. The field of manipulator trajectory planning employs a variety of methods, dictated by planning spaces, interpolation curve types, optimization objectives, and algorithmic strategies [18]. Numerous studies have applied interpolation functions to improve motion trajectory continuity and smoothness, with polynomial interpolation [19] and B-spline interpolations [20]–[22] being notable examples. Lin *et al.* [23] combined fifth-order polynomial interpolation, the S-curve velocity profile, and NURBS curve fitting to develop a three-segmented trajectory planning framework. This integration significantly enhances motion control precision and planning efficiency for manipulators. Concurrently, Wang [24] integrated angle sensors with depth cameras to develop a novel 3-5-7-5-3 multi-segment variable-order interpolation algorithm. This method is based on end-effector pose correction, enabling real-time calibration of the manipulator's terminal position and reducing trajectory planning errors. Simultaneously, Wang [25] suggested employing 3-5-3 segmented polynomials for joint trajectory interpolations, combining them with an enhanced cuckoo search algorithm to achieve time-optimal trajectory optimization under speed constraints. Lu and colleagues [26] applied discrete segmented fifth-order polynomials to interpolate sequences of joint angle position nodes, transforming the trajectory planning challenge into a constrained nonlinear multivariate optimization problem. Chen [27], in his pioneering work, introduced a 5-7-5 segmented polynomial interpolation approach, cleverly merging the computational simplicity of lower-order functions with the superior optimization performance of their higher-order counterparts. This resulted in smoother and more controllable

Manuscript received November 3, 2023; revised May 28, 2024. The work reported here was supported by the National Key Research and Development Program of China (2022YFC2604004) and Shandong Province Science and Technology-based Small and Medium-sized Enterprises Innovation Capacity Enhancement Project (2022TSGC2175).

J. Wang is a postgraduate student in the School of Control Science and Engineering, Shandong University, Ji'nan 250061, China. (e-mail: wangjiayu@mail.sdu.edu.cn).

R. Sun is an assistant engineer of the PLA Units No.92292, Qingdao 266000, China. (e-mail: Sruyue123@163.com).

Y. Zheng is a postdoctoral researcher of the School of Control Science and Engineering, Shandong University, Ji'nan 250061, China. (corresponding author to provide phone: 18563920008; e-mail: zhengyk@mail.sdu.edu.cn).

R. Song is a professor of the School of Control Science and Engineering, Shandong University, Ji'nan 250061, China. (corresponding author to provide phone: 13505319131; e-mail: rsong@sdu.edu.cn).

Y. Li is a professor of the School of Control Science and Engineering, Shandong University, Ji'nan 250061, China. (e-mail: liyb@sdu.edu.cn).

trajectories for manipulators.

Time-optimal trajectory planning has recently become a leading example in the field of single-objective trajectory optimization techniques, where researchers primarily utilize swarm intelligence optimization algorithms for enhancements. Wu and his team [28] proposed a robot trajectory method based on the fruit fly algorithm with a mean learning strategy. This method incorporates a focus on population mean learning, balancing the global and local search capabilities of the algorithm and thus improving both robot motion efficacy and trajectory smoothness. Yu and collaborators [29] adopted a novel time-optimal methodology that combines second-order continuous polynomial interpolation with cosine-weighted particle swarm optimization. By considering both kinematic and dynamic constraints, they achieved an impressive optimization of motion time for the PUMA 560 serial manipulator. Meanwhile, Zhao *et al.* [30] introduced a time-optimal trajectory strategy for manipulators, based on the multi-population competitive squirrel search algorithm. Relying on S-shaped curves for time interval estimation, they employed multi-population competitive iterations across all individuals of the algorithm to identify the optimal solution. Li *et al.* [31] proposed a 3-5-5-3 segmented polynomial interpolation trajectory planning method, leveraging an improved bat algorithm to demonstrate enhanced convergence and optimization capabilities. Furthermore, Esfandiar *et al.* [32] utilized the harmony search algorithm to navigate the complexities of path planning for flexible manipulators, especially under significant deformations.

This paper aims to apply the White Shark Optimization (WSO) algorithm to 3-5-3 polynomial interpolation trajectory planning, with the goal of refining trajectories in the joint space of SRL. Initially, we introduce an auxiliary joint into the SRL kinematic model to eliminate coordinate offsets. Subsequently, the 3-5-3 polynomial interpolation technique interpolates the joint space trajectories. Finally, the WSO algorithm is used to refine the interpolation time, achieving trajectory optimization for the SRL. The main contributions of this article include:

- The development of an auxiliary coordinate system, based on joint offset, skillfully addressing the complex issue of inverse kinematics resolution.
- The application of the WSO algorithm to propose a time-optimal trajectory optimization strategy in the joint space of the SRL.
- Simulation results demonstrate that the proposed optimization strategy delivers both smooth and time-efficient trajectories for the SRL.

The remainder of this paper is organized as follows: Section II provides a detailed examination of the SRL, including its mathematical model. Section III introduces the proposed time-optimal trajectory planning for SRL. Section IV brings to the simulation results, highlighting the efficacy of the proposed strategy, and the final conclusion with directions for future research is presented in Section V.

II. Problem Formulation and Model Design

A. Problem formulation

The pursuit of time-optimal trajectory planning for SRL is aimed at enhancing work efficiency through minimizing the operational duration of the manipulator. This trajectory approach ensures that the manipulator completes its movements in the most time-efficient manner possible, adhering to specific trajectory waypoints while complying with established kinematic constraints. The objective function for this study is formulated as follows:

$$f(t) = \min \sum_{j=0}^{n} (t_{j1} + t_{j2} + t_{j3}), \tag{1}$$

where t_{j1}, t_{j2}, t_{j3} represent the three-segment interpolation times of the 3-5-3 piecewise interpolation polynomial, respectively.

The constraints relating to the joint angle, angular velocity, and angular acceleration of the SRL are specified as follows:

$$\begin{cases} |q_j(t)| \le A_{j\max}, j = 1, 2, \cdots, 6 \\ |\dot{q}_j(t)| \le V_{j\max}, j = 1, 2, \cdots, 6 \\ |\ddot{q}_j(t)| \le W_{j\max}, j = 1, 2, \cdots, 6 \end{cases}, \tag{2}$$

where $q_j(t)$, $\dot{q}_j(t)$, and $\ddot{q}_j(t)$ represent the j-th joint angle, angular velocity, and angular acceleration, respectively. $A_{j\max}$, $V_{j\max}$, $W_{j\max}$ denote the maximum allowable values set by the manipulator itself, respectively.

B. Model design for SRL

The virtual prototype of the SRL, illustrated in Fig.1, comprises a foundational backboard and four intricately designed limbs. The backboard serves as the core of the system, housing essential devices such as the control module, power supply, and communication interface, thereby enabling precise interactions between the limbs and external systems. The limbs play a pivotal role in the functionality of the SRL, mirroring the biomechanical versatility inherent in human anatomy, thus offering both dexterity and mobility. Each limb features joints powered by cable-driven mechanisms, allowing for a wide range of movements and postures.

For this study, the emphasis is on trajectory optimization, abstracting from the specific actuation methods. Here, each limb is treated as an ideal jointed serial manipulator. With the limbs' design being uniform, one limb is selected for in-depth analysis. This single-limb model, shown in Fig.2(a), is divided into seven segments: the base, shoulder joint, upper arm, elbow joint, forearm, wrist joint, and end effector, featuring six rotational joints. The axes of the second, third, and fifth joints are parallel, while the others are orthogonal to these.

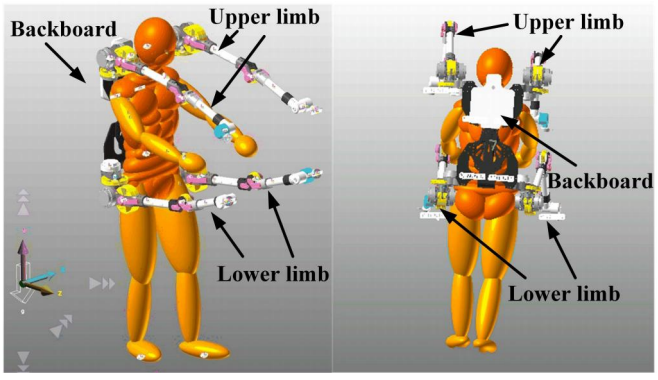
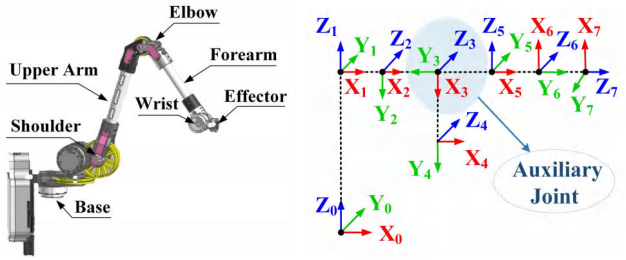


Fig. 1. Virtual prototype of SRL.



(a) SRL single-limb model (b) SRL coordinate system

Fig. 2. SRL single-limb model and coordinate system.

TABLE I
SRL MODIFIED D-H PARAMETERS.

i	α_{i-1}	a_{i-1}	d_i	θ_i	Range	Offset
1	0	0	0.09315	θ_1	$\left[-\pi, \frac{\pi}{6}\right]$	0
2	$-\frac{\pi}{2}$	0.08300	0	θ_2	$\left[-\frac{\pi}{3}, \frac{2\pi}{3}\right]$	0
3	0	0.39094	0	0	0	$\frac{\pi}{2}$
4	0	0.02958	0	θ_3	$\left[-\frac{5\pi}{6}, \frac{\pi}{6}\right]$	$-\frac{\pi}{2}$
5	$\frac{\pi}{2}$	0.33481	0.02958	θ_4	$\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$	0
6	$-\frac{\pi}{2}$	0.02400	0	θ_5	$\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$	$-\frac{\pi}{2}$
7	$-\frac{\pi}{2}$	0	0.06708	θ_6	$\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$	0

In consideration of the cable-driven design architecture of the SRL, the application of the traditional Denavit-Hartenberg (D-H) parameter method [33] for coordinate establishment results in offsets between the systems. This discrepancy arises due to the geometric centers of adjacent linkages not aligning along a straight trajectory. To overcome this issue, this paper introduces an auxiliary joint at the third joint position, further adjusting the first coordinate frame along its z-axis, thereby rectifying the offset observed in the Y-axis direction. The added auxiliary joint, labelled as the third joint, is kept in a fixed position, acting equivalently to a translational matrix in the calculations. Benefitting from the incorporating this auxiliary joint, a Modified Denavit-Hartenberg (M-DH) parameterization technique [34] is employed to construct the coordinate systems, as depicted in Fig.2(b). This approach enables a detailed kinematic representation of the robotic limb. The modified D-H parameters are meticulously listed in Table I.

The parameters defining the manipulator's linkages are described as follows: the link twist angle α indicates the angle between the axes of adjacent joints; the link length a signifies the distance between the two axes, specifically the length of the common perpendicular line between them; the link offset d denotes the distance along the direction of the shared axis between adjacent links, and the joint angle θ refers to the rotation angle around the shared axis of the adjacent links.

The forward kinematics of the SRL is primarily defined by the following functional form, indicating that the pose of the end effector is dependent on the joint coordinates. Through the homogeneous transformations, the resulting expression is simplified to the sequential product of transformation matrices, each corresponding to an individual linkage.

$$\begin{cases} \xi_{\mathrm{E}} = \mathrm{K}(\theta) \\ \xi_{\mathrm{E}} \sim {}^{0}T_{\mathrm{E}} = {}_{1}^{0}T \cdot {}_{2}^{1}T \cdot {}_{3}^{2}T \cdot {}_{4}^{3}T \cdot {}_{5}^{4}T \cdot {}_{6}^{5}T \cdot {}_{7}^{6}T \end{cases}, \tag{3}$$

where ξ_E represents the pose of the end effector, θ denotes the joint angles or the position vector of the robotic arm, and ${}_{i}^{i-1}T$ indicates the homogeneous transformation matrix from coordinate system $\{i-1\}$ to coordinate system $\{i\}$.

The inverse kinematics, crucial for the operation of the SRL, are formulated as follows:

$$\theta = K^{-1}(\xi). \tag{4}$$

The approach to computing the inverse kinematics of the SRL is primarily rooted in the analytical approach. A detailed derivation of these processes is presented in Appendix.

$$\begin{cases} \theta_1 = A\tan 2(A_1, B_1) \\ \theta_2 = A\tan 2(A_2, B_2) - A\tan 2(K, \pm\sqrt{A_2^2 + B_2^2 - K^2}) \\ \theta_3 = A\tan 2(A_3, B_3) - A\tan 2(L, \pm\sqrt{A_3^2 + B_3^2 - L^2}) \\ \theta_4 = A\tan 2(A_4, B_4) \\ \theta_5 = A\tan 2(\pm A_5, B_5) \\ \theta_6 = A\tan 2(A_6, B_6) - A\tan 2(N, \pm\sqrt{A_6^2 + B_6^2 - N^2}) \end{cases}. \tag{5}$$

It is evident that the SRL has eight distinct sets of inverse solutions refer to (5). The optimal solution should be selected based on the operational conditions, adhering to the principle of minimal joint angle variation. Such the inverse kinematic framework enables the conversion of interpolation points in the Cartesian space to their equivalent representations in the joint space.

III. TRAJECTORY PLANNING STRATEGY

This section proposes an optimal trajectory planning strategy for the SRL, aimed at minimizing trajectory time. This strategy is built on piecewise polynomial interpolation, enhanced by the WSO algorithm.

A. Polynomial fusion trajectory optimization

Utilizing low-degree interpolation polynomials may result in abrupt changes in acceleration. These changes have the potential to damage the manipulator's delicate components and reduce its operational efficiency. On the other side, high-degree interpolation polynomials are prone to the Runge phenomenon. This effect negatively impacts the quality of trajectory fitting and increases computational complexity, thus raising both time and financial costs associated with the process.

To overcome these challenges, we introduce the 3-5-3 piecewise polynomial interpolation method to the trajectory planning of the SRL. This method divides the total motion duration into three segments and establishes four critical interpolation points, as illustrated in Fig.3. The general formula representing the position, velocity, and acceleration characteristics of the 3-5-3 piecewise polynomial is as follows:

The first segment——Cubic Polynomial:

$$\begin{cases} \theta_{j1}(t) = \sum_{m=0}^{3} a_{j1m} t_1^m \\ \dot{\theta}_{j1}(t) = \sum_{m=1}^{3} m a_{j1m} t_1^{m-1} \\ \ddot{\theta}_{j1}(t) = \sum_{m=2}^{3} m\,(m-1)\, a_{j1m} t_1^{m-2} \end{cases}. \tag{6}$$

The second segment——Quintic Polynomial:

$$\begin{cases} \theta_{j2}(t) = \sum_{m=0}^{5} a_{j2m} t_2^m \\ \dot{\theta}_{j2}(t) = \sum_{m=1}^{5} m a_{j2m} t_2^{m-1} \\ \ddot{\theta}_{j2}(t) = \sum_{m=2}^{5} m\,(m-1)\, a_{j2m} t_2^{m-2} \end{cases}. \tag{7}$$

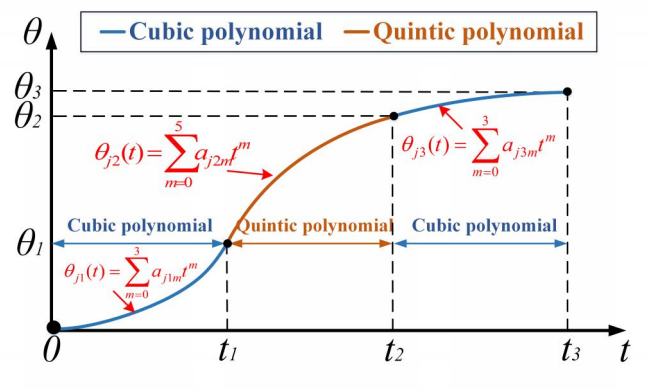


Fig. 3. 3-5-3 Segmented polynomial schematic diagram.

The third segment——Cubic Polynomial:

$$\begin{cases} \theta_{j3}(t) = \sum_{m=0}^{3} a_{j3m} t_3^m \\ \dot{\theta}_{j3}(t) = \sum_{m=1}^{3} m a_{j3m} t_3^{m-1} \\ \ddot{\theta}_{j3}(t) = \sum_{m=2}^{3} m\,(m-1)\, a_{j3m} t_3^{m-2} \end{cases} . \tag{8}$$

Where a_{j1m}, a_{j2m}, and a_{j3m} are the m-th coefficient of the interpolation function for the first, second, and third sections of the trajectory for the j-th joint, respectively, and θ_{jn} denotes the polynomial trajectory at the n-th section of the j-th joint.

The four interpolation points for the joints are designated as X_{j1}, X_{j2}, X_{j3}, and X_{j4}. During the trajectory planning, constraints are imposed to ensure the manipulator's motion trajectory remains continuous at these points. Typically, the initial (starting) and final (ending) interpolation points are characterized by a velocity and acceleration gradient of zero. Importantly, the end of the first segment smoothly transitions into the beginning of the subsequent one. Likewise, the end of the middle segment seamlessly connects to the start of the final segment, ensuring a continuous and uniform trajectory in both velocity and acceleration.

Based on the conditions and constraints of the three trajectory segments, the following relationships are deduced, where X_{jk} signifies the k-th interpolation point of the j-th joint, with $k = 1, 2, 3, 4$.

$$\begin{cases} X_{j1} = a_{j10} \\ \dot{X}_{j1} = a_{j11} = 0 \\ \ddot{X}_{j1} = 2a_{j12} = 0 \end{cases} . \tag{9}$$

$$\begin{cases} X_{j2} = \sum_{m=0}^{3} a_{j1m} t_1^m = a_{j20} \\ \dot{X}_{j2} = \sum_{m=1}^{3} m a_{j1m} t_1^{m-1} = a_{j21} \\ \ddot{X}_{j2} = \sum_{m=2}^{3} m\,(m-1)\, a_{j1m} t_1^{m-2} = 2a_{j22} \end{cases} . \tag{10}$$

$$\begin{cases} X_{j3} = \sum_{m=0}^{5} a_{j2m} t_2^m = a_{j30} \\ \dot{X}_{j3} = \sum_{m=1}^{5} m a_{j2m} t_2^{m-1} = a_{j31} \\ \ddot{X}_{j3} = \sum_{m=2}^{5} m\,(m-1)\, a_{j2m} t_2^{m-2} = 2a_{j32} \end{cases} . \tag{11}$$

Given the four known interpolation points of the trajectory, it is necessary to exclude both the velocity and acceleration terms from the formula due to their unknown values. Consequently, the coefficient matrix is structured as follows:

$$A = \begin{bmatrix} t_1^3 & t_1^2 & t_1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 3t_1^2 & 2t_1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 6t_1 & 2 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & t_2^5 & t_2^4 & t_2^3 & t_2^2 & t_2 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 5t_2^4 & 4t_2^3 & 3t_2^2 & 2t_2 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 20t_2^3 & 12t_2^2 & 6t_2 & 2 & 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_3^3 & t_3^2 & t_3 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3t_3^2 & 2t_3 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6t_3 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} . \tag{12}$$

The matrix of polynomial coefficients is:

$$a = \left[a_{j1}\ a_{j2}\ a_{j3}\right]^T , \tag{13}$$

where,

$$\begin{cases} a_{j1} = \left[a_{j13}\ a_{j12}\ a_{j11}\ a_{j10}\right]^T \\ a_{j2} = \left[a_{j25}\ a_{j24}\ a_{j23}\ a_{j22}\ a_{j21}\ a_{j20}\right]^T \\ a_{j3} = \left[a_{j33}\ a_{j32}\ a_{j31}\ a_{j30}\right]^T \end{cases} . \tag{14}$$

The joint angle can be articulated as follows:

$$B = \left[0\ 0\ 0\ 0\ 0\ 0\ X_{j4}\ 0\ 0\ X_{j1}\ 0\ 0\ X_{j3}\ X_{j2}\right]^T . \tag{15}$$

The specific configuration of the 3-5-3 piecewise polynomial is derived from $a = A^{-1} \cdot B$, so as to obtain the joint angle, angular velocity, and angular acceleration of the manipulator at each moment.

B. Trajectory optimization strategy based on WSO

While the 3-5-3 polynomial interpolation facilitates the generation of a smooth trajectory, it inherently restricts the manipulator to a fixed operational timeframe. To enhance the operational efficiency of the SRL, enabling it to navigate waypoints more quickly, it is essential to optimize the time allocations of the three segments inherent to the 3-5-3 polynomial interpolation. The WSO, noted for its exceptional local search capability, is employed for this purpose [35]. The algorithm primarily consists of four phases: 1) Rapid movement toward the target; 2) Encirclement of the optimal target; 3) Accurate navigation to the best attack position; 4) Grouping behavior as observed in fish. The equations below provide the mathematical basis for this algorithm.

$$\nu_{m+1}^i = \mu\left[\nu_m^i + h\left(w_{\text{gbest}_m} - w_m^i\right) \times c_1 + h_2\left(w_{best}^{\nu_m^i} - w_m^i\right) \times c_2\right]. \tag{16}$$

Where $i = 1, 2, \ldots n$ indicates the population of white sharks with a quantity of n; m represents the iteration count; ν_m^i and w_m^i refer to the velocity and position vectors of the i-th white shark in the m-th iteration, respectively; w_{gbest_m} denotes the globally best position vector achieved by all white sharks up to the m-th step; c_1 and c_2 are two coefficients generated randomly within the interval [0,1]; h_1 and h_2 signify the forces exerted by the white shark controlling w_{gbest_m} and $w_{best}^{\nu_m^i}$ on w_m^i, and μ is a contraction factor proposed in WSO to evaluate the convergence tendency of the algorithm.

The following equation captures the complex behavior of white sharks as they encircle their prey:

$$w_{m+1}^{i}=\begin{cases} w_{m}^{i}\cdot\neg\oplus w_{0}+u\cdot a+l\cdot b; & rand<mv \\ w_{m}^{i}+v_{m}^{i}/f; & rand\geq mv \end{cases}, \quad (17)$$

where $\neg$ is the negation operator; $\oplus$ denotes the bitwise XOR operation; a and b are one-dimensional binary vectors; l and u signify the lower and upper bounds of the search space, respectively. w_0 is a logical vector; f denotes the frequency of wave motion for the white shark; $rand$ is a random number within the range [0,1]. mv indicates the composite intensity of the auditory and olfactory sensations of the white shark as it approaches its prey, significantly influencing the shark's search strategy: notably, lower values favor localized searches, while higher values encourage a broader, global search.

Upon identifying the target, the white shark adjusts its trajectory towards the optimal attack position:

$$\vec{w}_{m+1}^{i}=w_{gbest_m}+d_1\vec{D}_w\,\mathrm{sgn}(d_2-0.5);\quad d_3<s_n, \quad (18)$$

where $\mathrm{sgn}(\bullet)$ takes a value of 1 or -1 to change the search direction; d_1, d_2, d_3 are random numbers from 0 to 1; $\vec{D}_w$ represents the distance between the prey and the white shark; s_n indicates the olfactory and visual intensity from other white sharks when tracking and approaching the optimal prey.

To mathematically simulate the collective behavior of the white shark population, the two previous best solutions are retained. These positions then guide the spatial updates of their counterparts:

$$w_{m+1}^{i}=\frac{w_{m}^{i}+\vec{w}_{m+1}^{i}}{2\times rand}. \quad (19)$$

On this basis, this paper proposes an Improved White Shark Optimization (I-WSO) algorithm, which introduces a function variation tolerance parameter, Δ_{tol}, to represent the desired accuracy level for the fitness function. During each iteration, after calculating the current white shark's global optimal position vector w_{gbest_m}, the change Δ_{fit} from its previous iteration's global best position vector $w_{gbest_{m-1}}$ is computed to verify whether the accuracy requirement has been met. Δ_{fit} is defined as follows:

$$\Delta_{fit}=\left|w_{gbest_m}-w_{gbest_{m-1}}\right|. \quad (20)$$

By comparing the magnitudes of Δ_{fit} and Δ_{tol}, if the former is smaller, it indicates stagnation in the current w_{gbest_m}. The algorithm evaluates w_{gbest_m} for stagnation at each iteration. To improve algorithm performance, the search concludes under any of the following conditions:

- If stagnation of w_{gbest_m} occurs for C_{stop} consecutive iterations without reaching the maximum iteration limit, the search is terminated. Here, C_{stop} represents the maximum allowed stagnation periods.
- The search ends upon reaching the maximum iteration count.

To briefly summarize the comprehensive procedure of the polynomial interpolation trajectory optimization strategy for SRL based on WSO, it can be outlined as follows:

Step 1: Initialize the parameters of WSO, including the population's dimensions Dim, the number of white sharks N, the maximum iteration count $Iter_{\max}$, white shark positions w, and velocities ν.

Step 2: Assess the objective function by calculating the fitness of each individual white shark. Concurrently, verify if the velocities of the SRL joints satisfy the constraints. If consistent, the fitness value of the current white shark is determined, alongside both the current optimal white shark position and the objective function's minimal value.

Step 3: Undertake positional updates for the white sharks, calculate the fitness of each updated white shark, and compare it with the objective function's minimal value. If reduced, the minimum value is replaced with this new fitness metric.

Algorithm 1 A pseudo code summarizing the iterative optimization process of I-WSO

Initialize the parameters of the problem and I-WSO
Initialize the population of whales with random positions and velocities
Evaluate the position of the initial population
while $m < Iter_{\max}$ **do**
  Update the related parameters
  while $c < C_{stop}$ **do**
    for $i = 1$ to n **do**
      $f(t) = \min \sum_{j=0}^{n}(t_{j1} + t_{j2} + t_{j3})$
      if $|q_j(t)| \le A_{j\max}, |\dot{q}_j(t)| \le V_{j\max}$, $|\ddot{q}_j(t)| \le W_{j\max}$ **then**
        Obtain current fitness value and position
        $\nu_{m+1}^{i} = \mu\left[\nu_{m}^{i} + h\left(w_{\mathrm{gbest}_m} - w_{m}^{i}\right)\times c_1 + h_2\left(w_{best}^{\nu_m^i} - w_{m}^{i}\right)\times c_2\right]$
      end if
    end for
    for $i = 1$ to n **do**
      if $rand < mv$ **then**
        $w_{m+1}^{i} = w_{m}^{i}\cdot\neg\oplus w_0 + u\cdot a + l\cdot b$
      else
        $w_{m+1}^{i} = w_{m}^{i} + v_{m}^{i}/f$
      end if
    end for
    for $i = 1$ to n **do**
      if $rand \le s_n$ **then**
        $\vec{D}_w = \left|rand \times \left(w_{gbest_m} - w_{m}^{i}\right)\right|$
        if $i == 1$ **then**
          $\vec{w}_{m+1}^{i} = w_{gbest_m} + d_1\vec{D}_w\,\mathrm{sgn}(d_2 - 0.5)$
        else
          $\vec{w}_{m+1}^{i} = w_{gbest_m} + d_1\vec{D}_w\,\mathrm{sgn}(d_2 - 0.5)$
          $w_{m+1}^{i} = \frac{w_m^i + \vec{w}_{m+1}^{i}}{2\times rand}$
          $\Delta_{fit} = \left|w_{gbest_m} - w_{gbest_{m-1}}\right|$
        end if
      end if
    end for
    Adjust the position of the white sharks that proceed beyond the boundary
    Evaluate and update the new positions
    $c = c + 1, m = m + 1$
  end while
end while
Return the optimal solution obtained so far

Step 4: Conduct a verification to ascertain if the iteration has reached its maximum stagnation or maximum iteration count. If either of these conditions is met, terminate the iteration and output the objective function's minimal value.

The pseudo-code of I-WSO is presented as Algorithm 1.

IV. Simulation Results and Discussion

This section is dedicated to evaluating the effectiveness of the proposed trajectory optimization strategy. This evaluation is achieved by comparing its performance in co-simulation scenarios with existing algorithms. These simulations were conducted using the CoppeliaSim and MATLAB platforms.

A. Polynomial interpolation algorithm simulation

To validate the robustness of the 3-5-3 polynomial interpolation methodology, a series of simulations was conducted within the MATLAB environment. These simulations covered three interpolation strategies: cubic polynomial interpolation, quintic polynomial interpolation, and the 3-5-3 segmented polynomial interpolation. The primary objective was to perform a comparative analysis assessing the operational advantages of each method.

The initial and terminal coordinates of the SRL were designated as (0.407, 0.125, 0.784) and (0.845, -0.132, 0.078), respectively, with interpolation points located at (0.413, 0.125, 0.782) and (0.846, -0.129, 0.085). Through inverse kinematics, the joint angles corresponding to the SRL's four interpolation points were meticulously derived and are detailed in Table II. Furthermore, the motion constraints applied to the SRL are outlined in Table III.

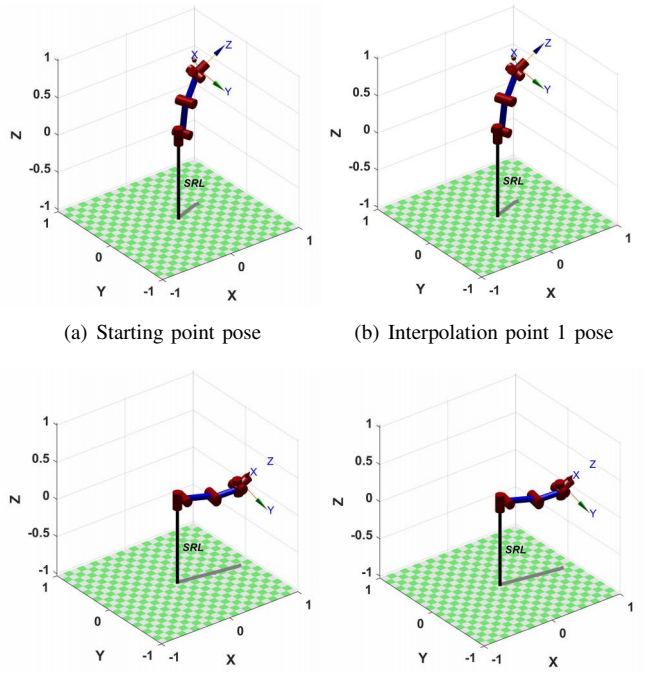
Fig.4 graphically illustrates the orientation of the SRL across the trajectory's four interpolation points. Based on a predetermined motion duration of $6s$, three trajectory plans were developed: one employing cubic polynomial interpolation, another using quintic polynomial interpolation, and the last implementing the 3-5-3 piecewise polynomial interpolation. The resultant data includes variations in joint angles, angular velocities, and angular accelerations, in addition to the position, velocity, and acceleration of the end-effector. These exhaustive results are showcased across Fig.5 through Fig.10.

TABLE II
Joint Angles of the Interpolation Point (rad).

Trajectory point	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
Starting point	0.3491	-1.3963	0.5236	-0.3491	0.2618	0
Interpolation point 1	0.3450	-1.3839	0.5174	-0.3443	0.2577	0.0055
Interpolation point 2	-0.1700	0.1610	-0.2550	0.2565	-0.2573	0.6921
End point	-0.1745	0.1745	-0.2618	0.2618	-0.2618	0.6981

TABLE III
Kinematic Constraints of Each Joint.

Variate	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
P(rad)	0.698	0.698	0.698	0.698	0.698	0.698
V(rad/s)	1.396	1.396	1.396	1.396	1.396	1.396
V(rad/s)	1.396	1.396	1.396	1.396	1.396	1.396



(a) Starting point pose (b) Interpolation point 1 pose

(c) Interpolation point 2 pose (d) End point pose

Fig. 4. Four interpolation point poses of SRL.

The analysis highlighted distinct operational characteristics associated with each interpolation method. Notably, cubic polynomial interpolation was found to cause abrupt changes in acceleration, potentially harming the manipulator's motor systems. On the other hand, while quintic polynomial interpolation promised continuous accelerations without sudden shifts, it occasionally resulted in excessive acceleration magnitudes. In contrast, the 3-5-3 piecewise polynomial interpolation was exemplary, offering smooth trajectories devoid of abrupt changes, thus establishing itself as the superior option among the evaluated methods.

These results collectively affirm the superiority of the 3-5-3 polynomial interpolation in trajectory planning. This method not only ensures continuity and stability but also protects the trajectory from unexpected disturbances, thereby enhancing the fluidity, safety, and efficiency of SRL's motion.

B. Time-optimal trajectory planning algorithm simulation

To ensure a comprehensive and unbiased evaluation, the following four distinct optimization algorithms were employed for comparative analysis, with uniform parameters consistently applied across all algorithms:

- The proposed **Improved White Shark Optimization (I-WSO)** algorithm, which encapsulates the concept of function variation tolerance.
- The classical **White Shark Optimization (WSO)** algorithm was adopted for comparison.
- The **Improved Particle Swarm Optimization (I-PSO)** algorithm was employed for its comparative evaluation.
- The **Dingo Optimization Algorithm (DOA)** was assimilated for analysis.

These simulations were conducted using MATLAB and CoppeliaSim platforms, as depicted in Fig.11. The primary objective was to verify the effectiveness of trajectory optimization for a single joint.

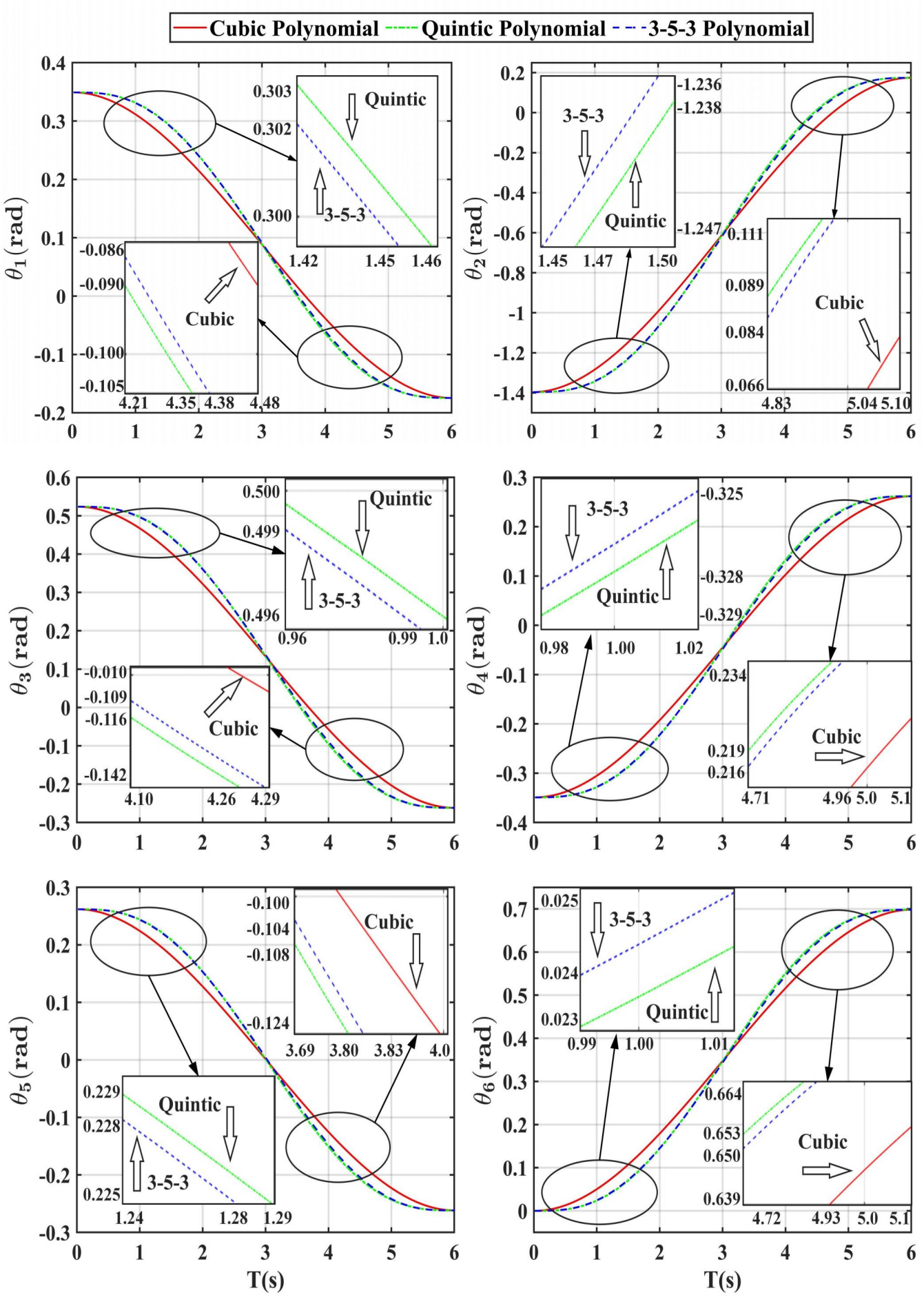


Fig. 5. Joint angle change of SRL.

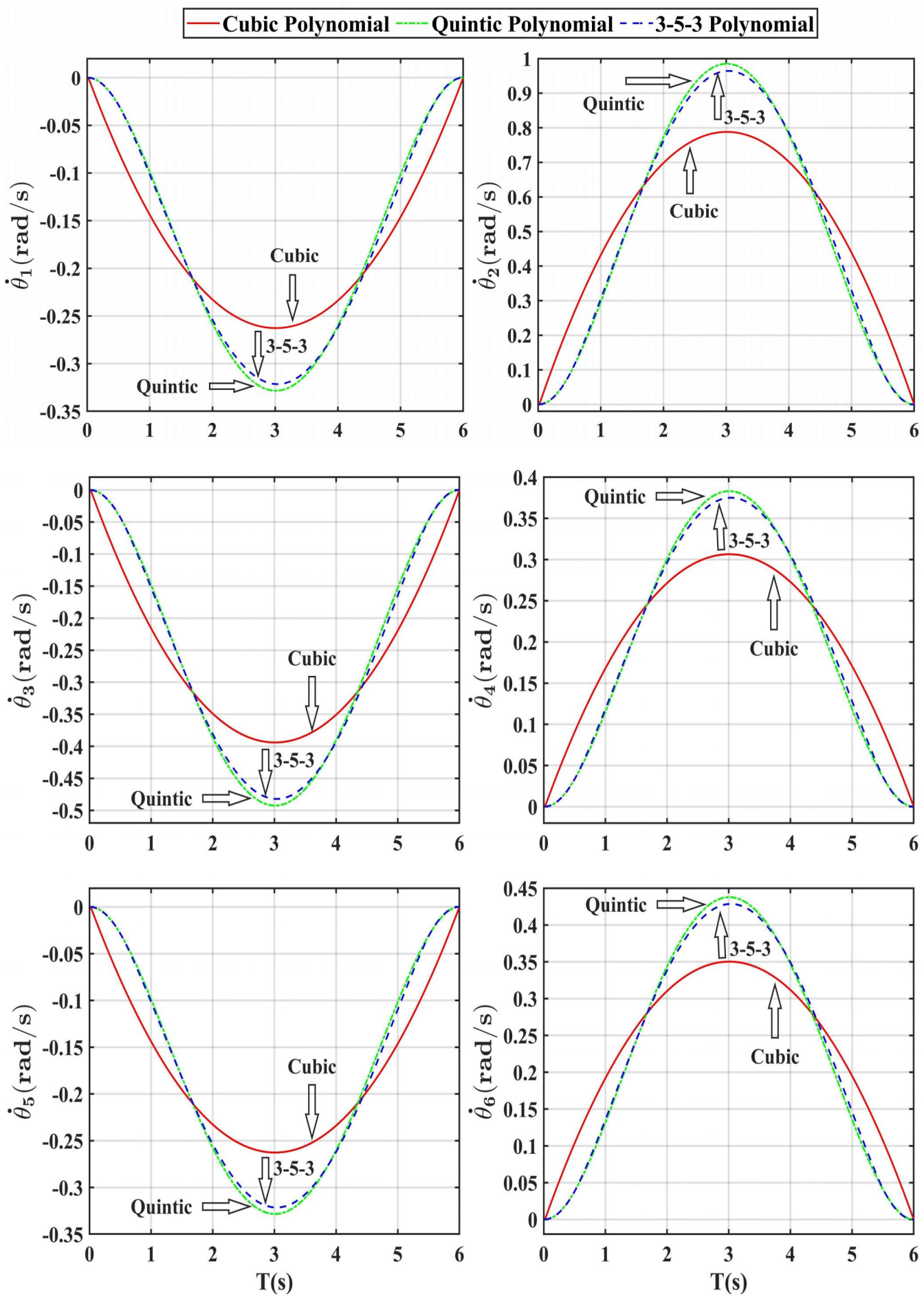


Fig. 6. Joint angular velocity variation of SRL.

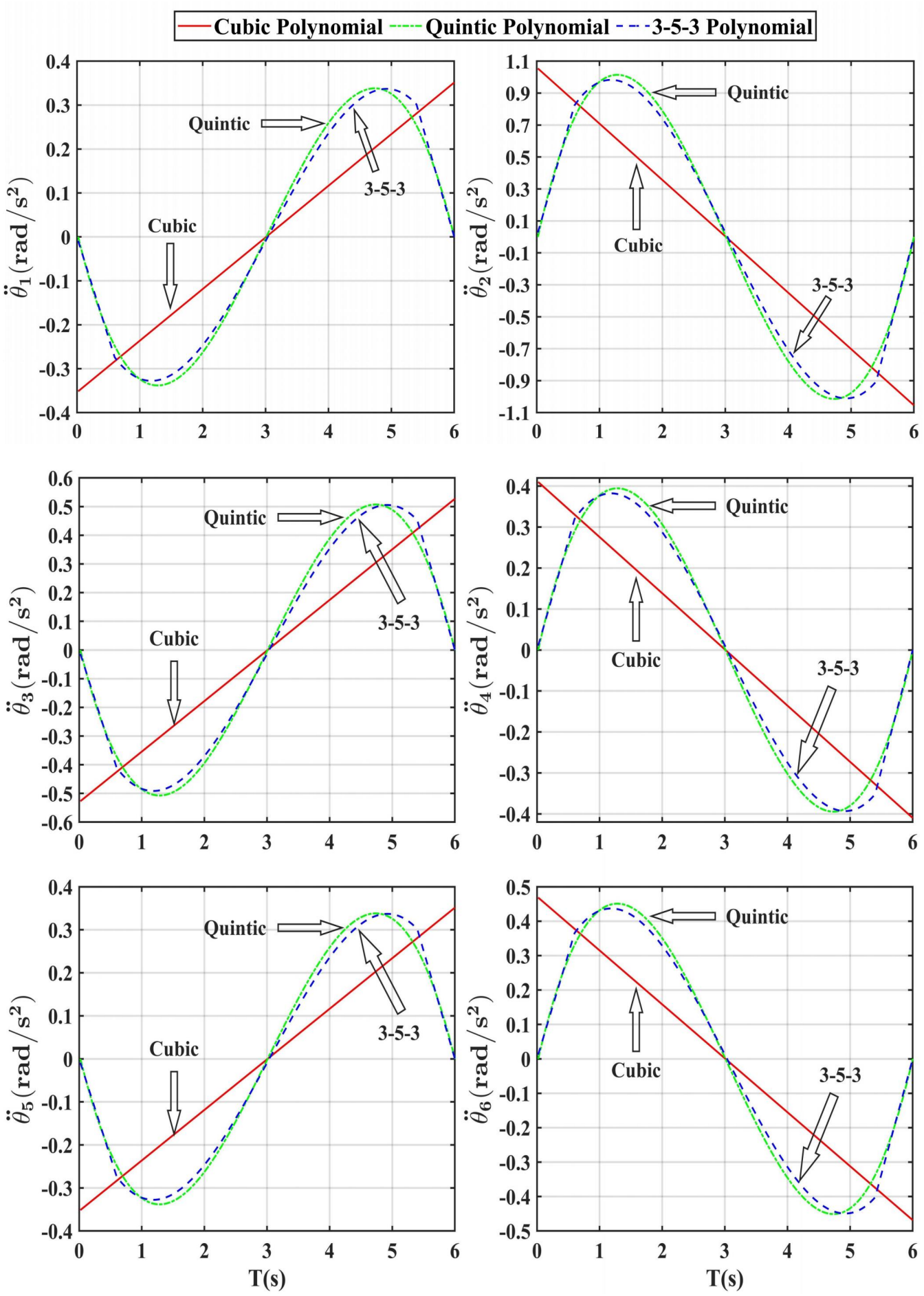


Fig. 7. Joint angular acceleration variation of SRL.

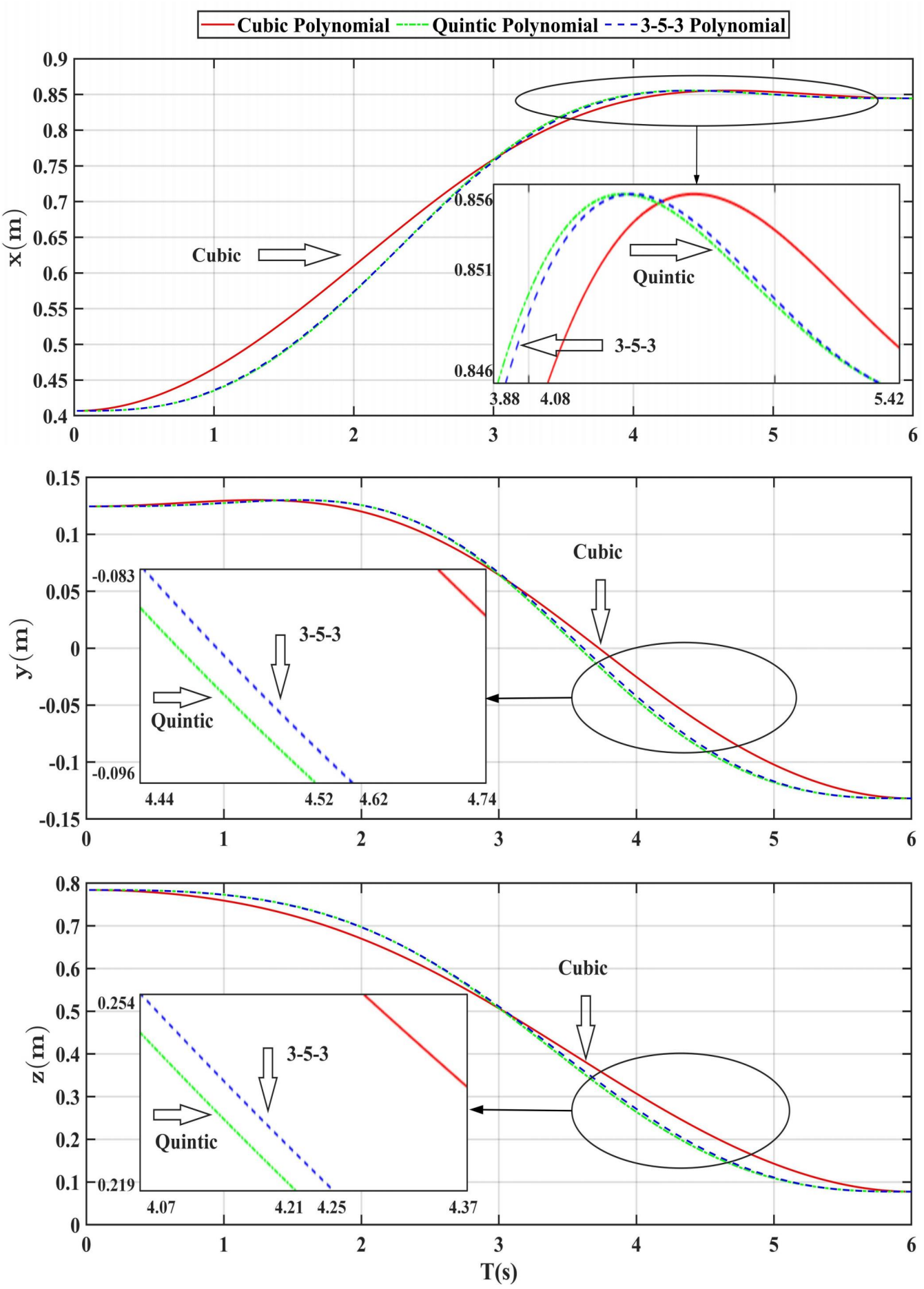


Fig. 8. Change in position of the end of SRL.

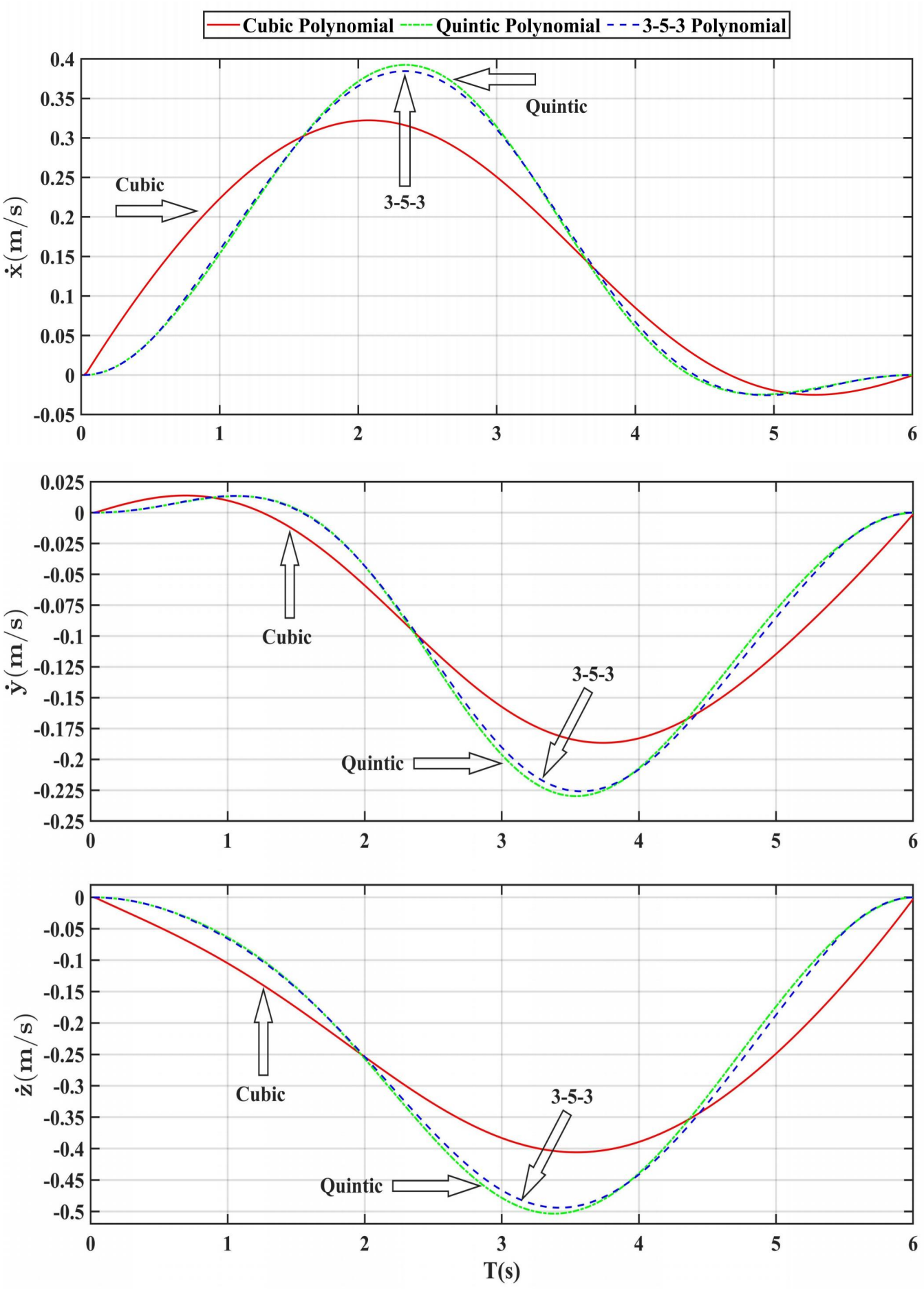


Fig. 9. Speed variation at the end of SRL.

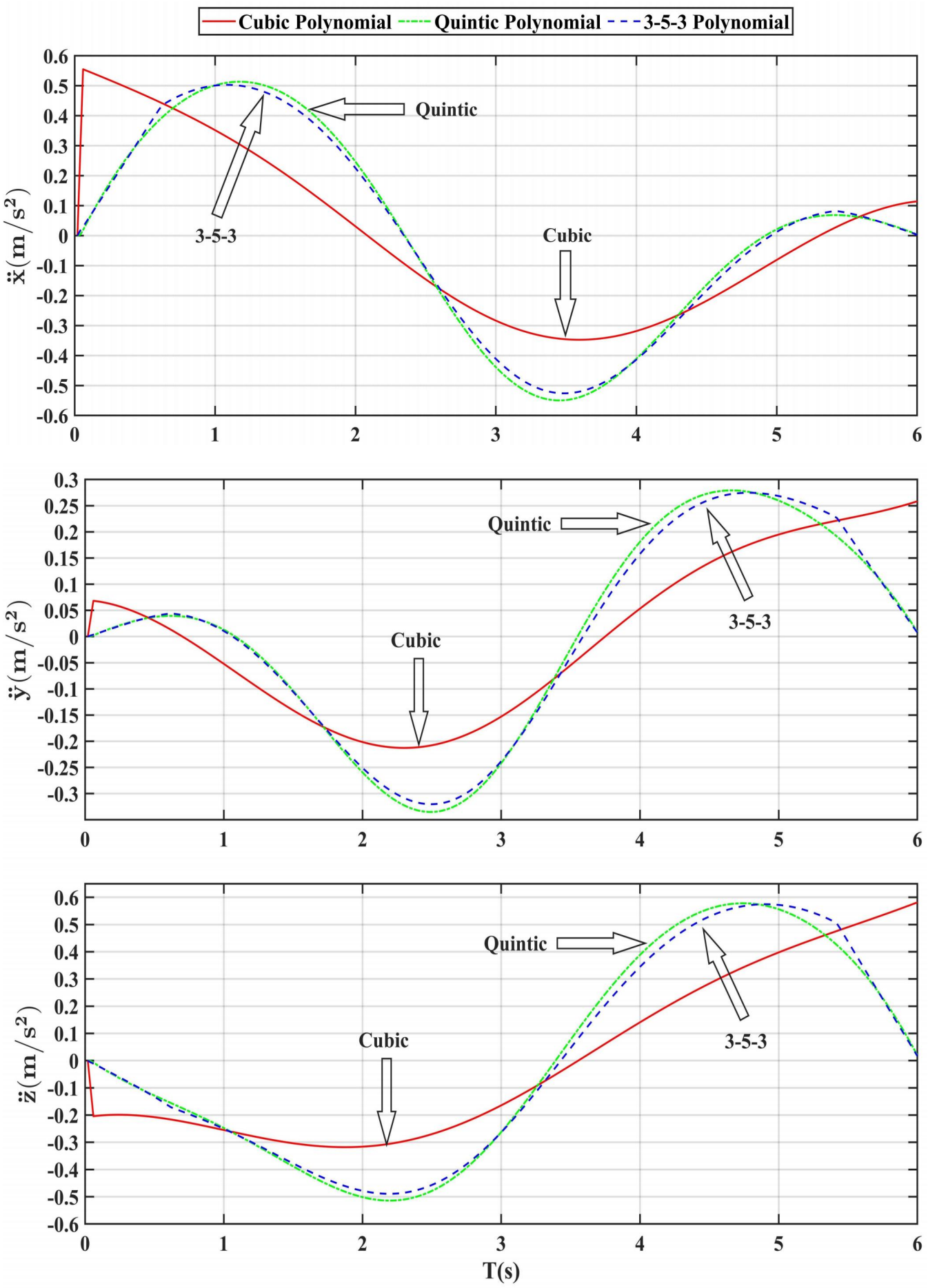


Fig. 10. Variation of acceleration at the end of SRL.

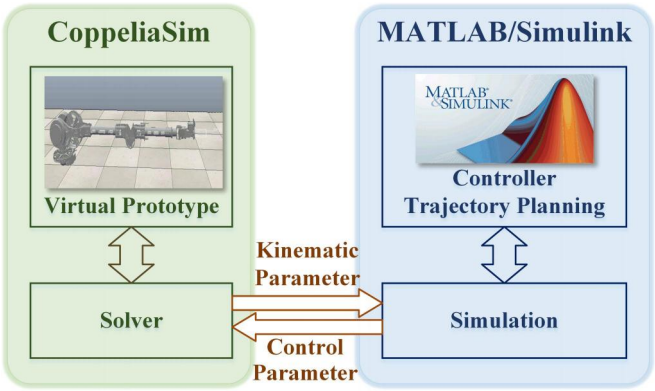


Fig. 11. Co-simulation of MATLAB and CoppeliaSim.

Parameter Settings: The angle interpolation points for the single joint are defined as (-1.2418, -0.8012, -0.4126, -0.5359)rad, with a maximum joint angular velocity of $1.396rad/s$. The basic parameters set for all four algorithms include: population size $N = 30$, maximum iteration count $Iter_{\max} = 1000$, population dimensions $Dim = 3$, and initial best fitness $gBest = 50$. Table IV offers a comprehensive summary of the specific configurations relevant to the algorithmic parameters.

TABLE IV
ALGORITHM PARAMETERS.

Algorithm	Parameter setting	Parameter names
WSO	$f_{\max} = 0.75$	Maximum frequency of wave motion
	$f_{\min} = 0.07$	Minimum frequency of wave motion
	$\tau = 4.11$	Coefficient of acceleration
	$h_{\min} = 0.5$	White shark initial velocity
	$h_{\max} = 1.5$	White shark subordinate velocity
I-WSO	$f_{\max} = 0.75$	Maximum frequency of wave motion
	$f_{\min} = 0.07$	Minimum frequency of wave motion
	$\tau = 4.11$	Coefficient of acceleration
	$h_{\min} = 0.5$	White shark initial velocity
	$h_{\max} = 1.5$	White shark subordinate velocity
	$C = 0$	Initial value of the counter
	$C_{stop} = 30$	Maximum value of the counter
	$\Delta_{tol} = 0.001$	Function variation tolerance
I-PSO	$c_{1\min} = 0.02$	Minimum self-learning factor
	$c_{1\max} = 0.05$	Maximum self-learning factor
	$c_{2\min} = 0.02$	Minimum social factors
	$c_{2\max} = 0.05$	Maximum social factors
	$w_{\min} = 0.3$	Minimum inertia weight
	$w_{\max} = 0.9$	Maximum inertia weight
DOA	$P = 0.5$	Probability of hunt strategy
	$Q = 0.7$	Probability of group attack strategy
	$N_{\min} = 2$	Minimum number of dingo attacks

TABLE V
RESULTS OF OPTIMIZATION OF FOUR ALGORITHMS.

Algorithm	I-WSO	I-PSO	WSO	DOA
Optimization results (s)	4.6597	6.4112	4.6592	4.9355
Program running time(s)	0.1873	0.1521	0.8338	0.9636

Aiming to refine the joint angle trajectory across the four optimization algorithms, the iterative count was limited to 10. Evaluation metrics focused on the optimal value of time optimization for the motion trajectory and the computational runtime of the program. The average values of these metrics are presented in Table V. Concurrently, Fig.12 displays the optimization results and program runtime curves for the four algorithms.

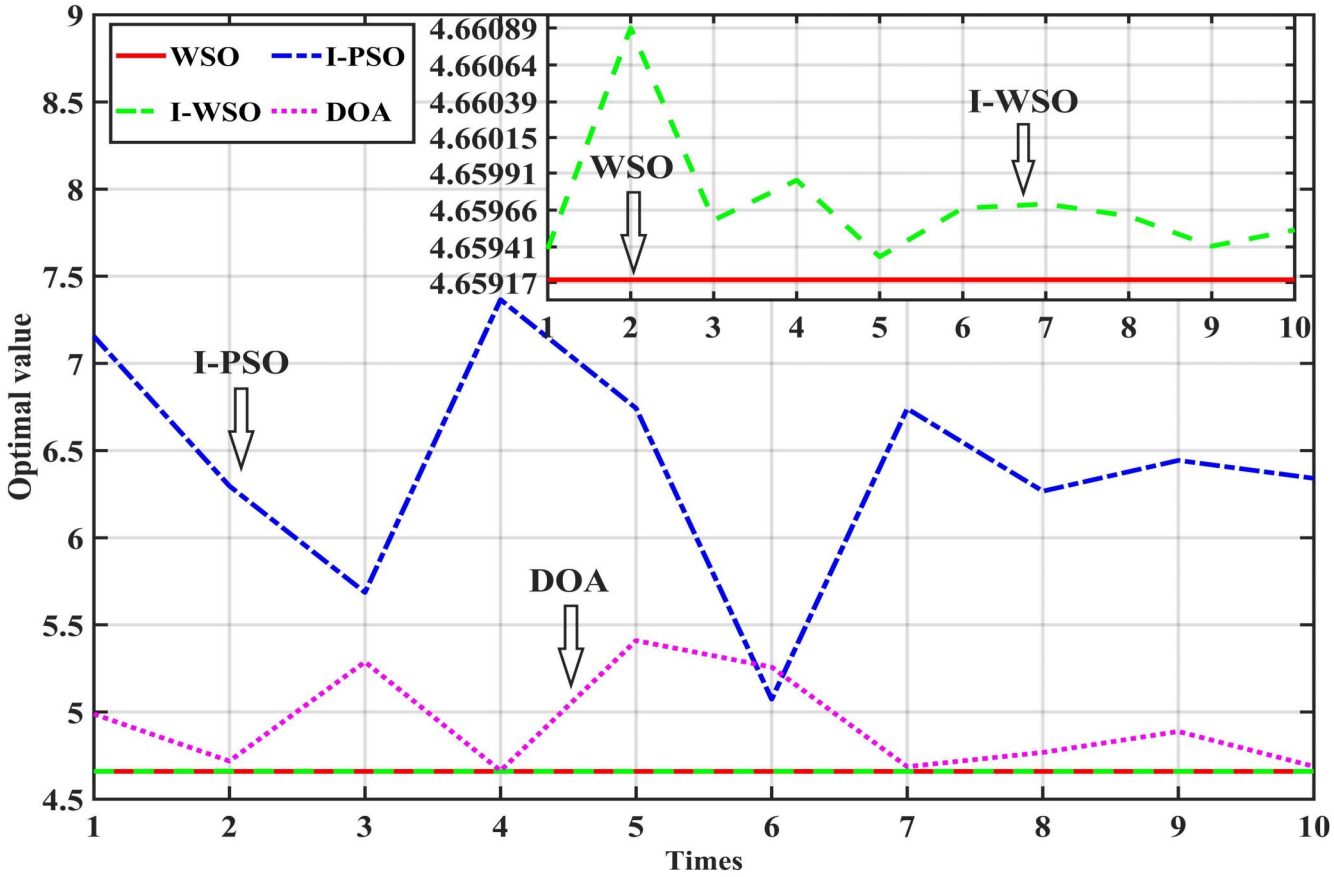
Meticulous analysis of the comparative data reveals that both the I-WSO and WSO algorithms demonstrate similar optimization outcomes. However, they significantly surpass the DOA's performance and markedly outperform the I-PSO, resulting in a noticeable reduction in trajectory execution time. Furthermore, it is evident that both the I-WSO and I-PSO achieve substantial reductions in computational runtime, distinctly outperforming the temporal metrics associated with the other two algorithmic alternatives. These enhancements significantly improve the responsiveness of the SRL system. Consequently, it is evident that trajectory planning using the I-WSO algorithm is not only effective for time optimization but also exhibits superior performance metrics compared to the I-PSO, WSO, and DOA algorithms.

Expanding the analysis to a 6-degree-of-freedom SRL and adhering to the 3-5-3 piecewise polynomial interpolation trajectory planning (with consistent simulation constraints and settings as previously outlined), the I-WSO algorithm was utilized for time-optimal trajectory planning of the SRL. Fig.13 illustrates the end-effector trajectory of the SRL as visualized in CoppeliaSim and MATLAB. The temporal variation profiles of joint angles, angular velocities, and angular accelerations are depicted in Fig.14. Additionally, Fig.15 shows the variation curves of end-effector positions, velocities, and accelerations over the same time frame.

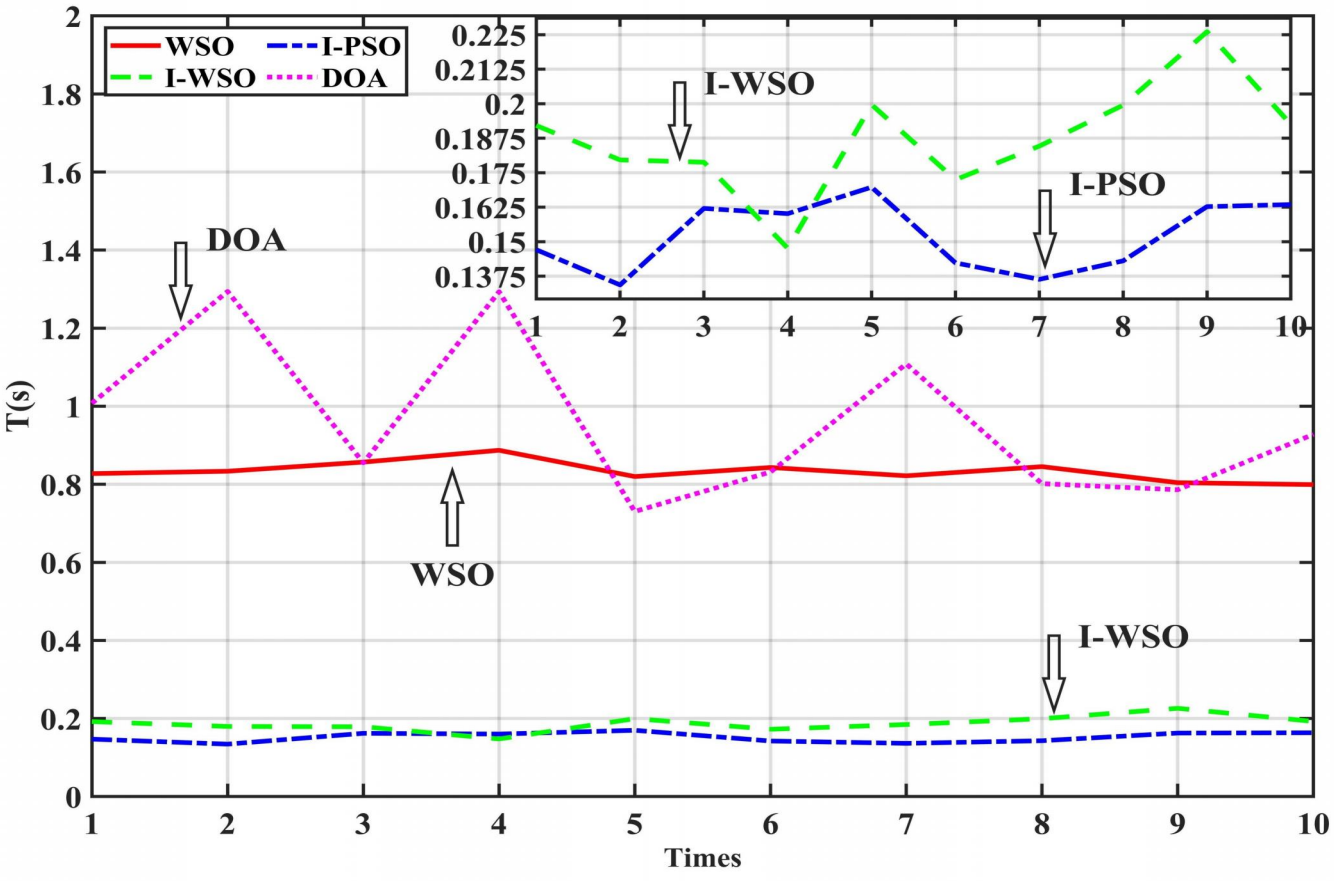
The data demonstrates that the implementation of the I-WSO has significantly enhanced the efficacy of the manipulator's joint space motion trajectory. There is a notable reduction in the motion time of the SRL, from 6 seconds to just 2.42 seconds, resulting in an improvement in overall efficiency by 59.7%. This significant improvement is crucial in increasing the operational productivity of the SRL. Moreover, the motion trajectories for each joint of the SRL are characterized by a continuous and smooth profile, free from any abrupt changes. This guarantees a seamless trajectory for both the joint angles and the end-effector motion, while complying with the predetermined constraints. The advanced trajectory optimization facilitated by the I-WSO algorithm further enhances the SRL's ability to follow the predetermined path with both stability and precision. These outcomes unequivocally highlight the efficacy and potential of the I-WSO algorithm in trajectory optimization.

V. CONCLUSION

This article introduces a time-optimal 3-5-3 piecewise polynomial interpolation trajectory optimization strategy supported by the WSO algorithm. The study focuses on the 6-degree-of-freedom SRL, incorporating auxiliary joints to mitigate coordinate offsets arising from its distinctive spatial architecture. Leveraging the WSO's simplicity and easy parameter adjustment, along with the SRL's kinematic constraints, this method balances the complexities of higher-order polynomial calculations, resulting in optimized trajec-

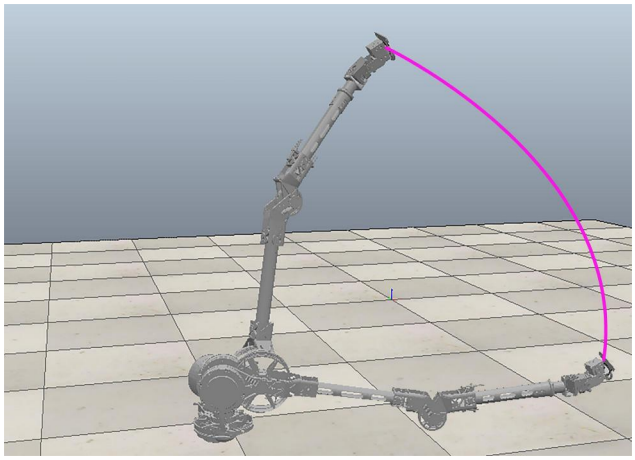


(a) Optimal value

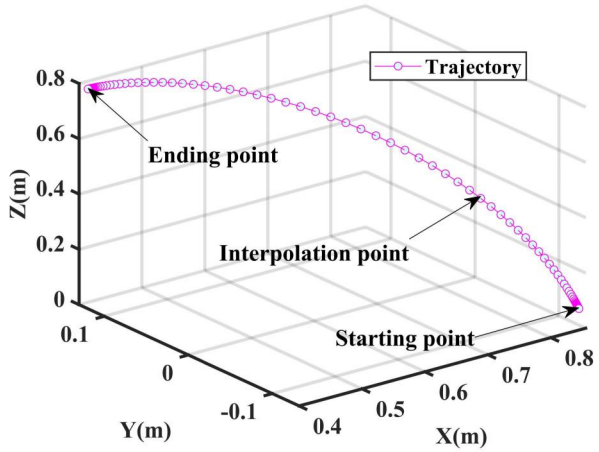


(b) Program runtime

Fig. 12. Optimization results and program runtime curves for four algorithm.



(a) The end-effector trajectory of SRL in CoppeliaSim



(b) The end-effector trajectory of SRL in MATLAB

Fig. 13. Trajectory of the end-effector of SRL in CoppeliaSim and MATLAB.

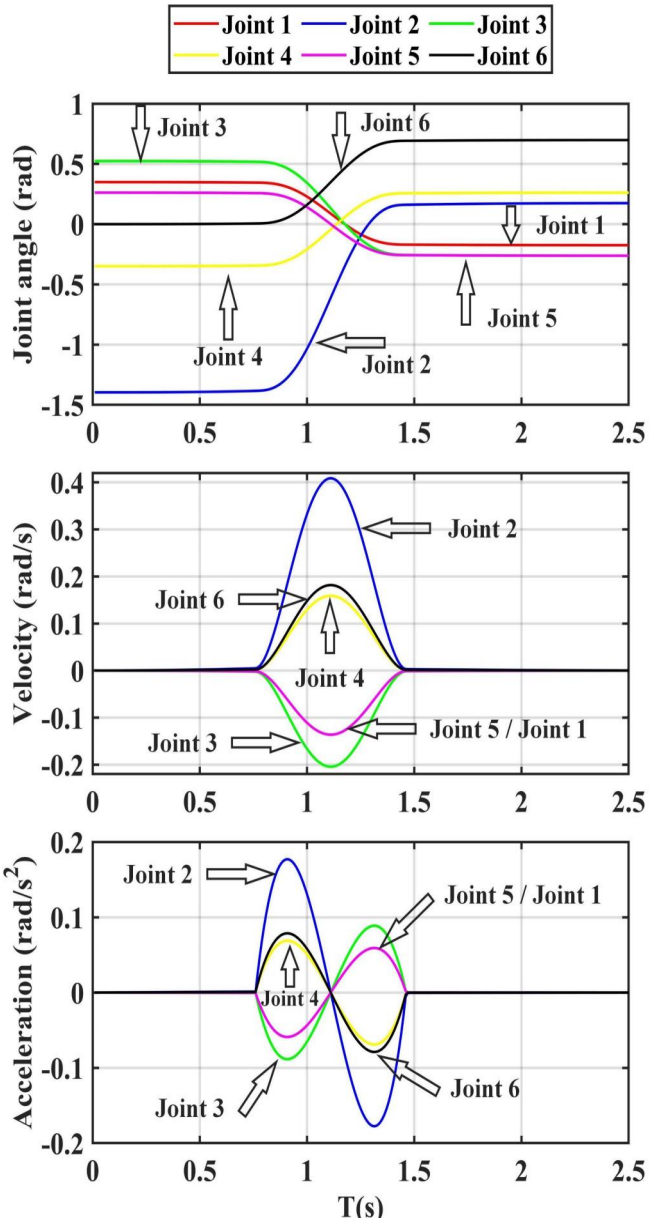


Fig. 14. Angular, angular velocity, and angular acceleration change curves of SRL.

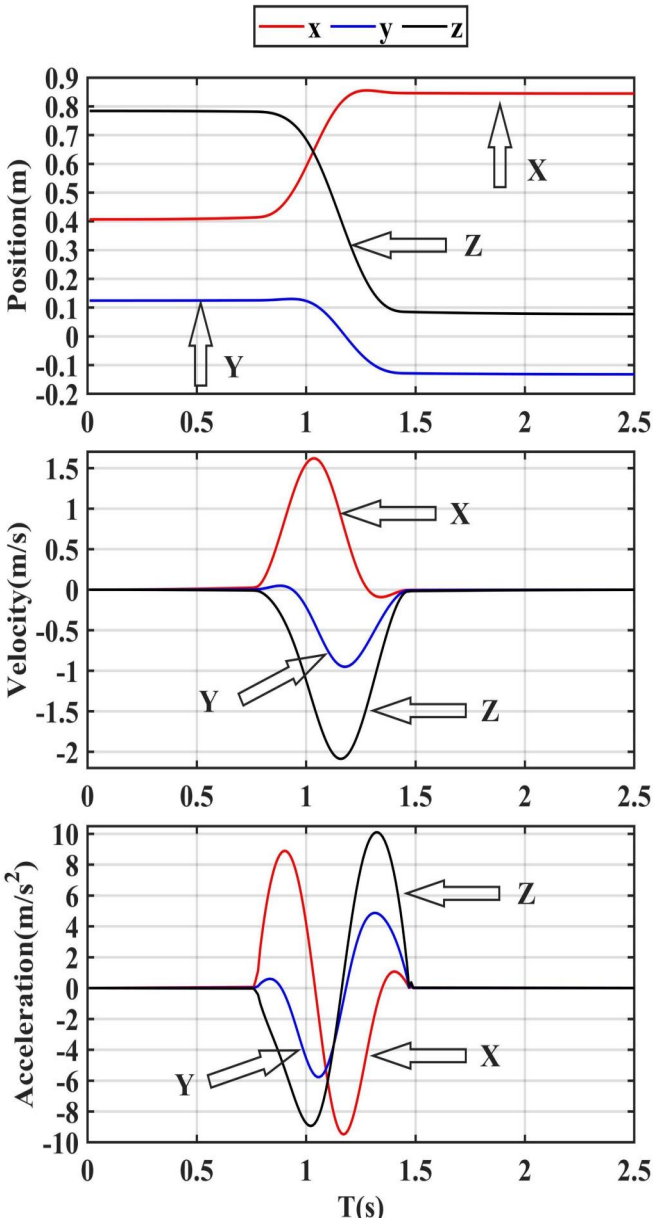


Fig. 15. Variation curves of end-effector position, velocity, and acceleration of SRL.

tory planning in joint space. Comparative simulation experiments have highlighted the method's benefits in terms of time efficiency and operational smoothness, offering an effective approach for real-world trajectory design and execution in the SRL with considerable practical implications.

Future research could extend the proposed algorithmic framework to include capabilities such as adeptly avoiding obstacles and coordinating multiple limbs in trajectory optimization for the SRL. These advancements promise to significantly enhance the effectiveness and real-world utility of SRL trajectory planning.

APPENDIX

The homogeneous transformation matrix of the SRL's end effector relative to its base is provided:

$$ {}^{0}_{7}T = {}^{0}_{1}T \cdot {}^{1}_{2}T \cdot {}^{2}_{3}T \cdot {}^{3}_{4}T \cdot {}^{4}_{5}T \cdot {}^{5}_{6}T \cdot {}^{6}_{7}T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (21) $$

Each element of the matrix is defined accordingly:

$$ \begin{cases} n_x = s_6(c_4 s_1 + s_4 c_1 c_{23}) - c_6(s_5(s_1 s_4 - c_4 c_1 c_{23}) - c_5 c_1 s_{23}) \\ n_y = c_6(s_5(c_1 s_4 + c_4 s_1 c_{23}) + c_5 s_1 s_{23}) - s_6(c_1 s_4 - s_4 s_1 c_{23}) \\ n_z = c_6(c_5 c_{23} - c_4 s_5 s_{23}) - s_4 s_6 s_{23} \\ o_x = c_6(c_4 s_1 + s_4 c_1 c_{23}) + s_6(s_5(s_1 s_4 - c_4 c_1 c_{23}) - c_5 c_1 s_{23}) \\ o_y = -c_6(c_1 c_4 - s_4 s_1 c_{23}) - s_6(s_5(c_1 s_4 + c_4 s_1 c_{23}) + c_5 s_1 s_{23}) \\ o_z = -s_6(c_5 c_{23} - c_4 s_5 s_{23}) - s_4 c_6 s_{23} \\ a_x = -c_5\,(s_1 s_4 - c_4 c_1 c_{23}) - s_5 c_1 s_{23} \\ a_y = c_s(c_1 s_4 + c_4 s_1 c_{23}) - s_s s_1 s_{23} \\ a_z = -s_5 c_{23} - c_4 c_5 s_{23} \\ p_x = c_1(a_2 + a_3 c_2 - a_4 s_2 + (d_5 - d_7 s_5) s_{23} + \\ (a_5 + a_6 c_4 + a_7 c_4 c_5) c_{23}) - (a_6 + d_7) s_1 s_4 \\ p_y = s_1(d_5 s_{23} + a_5 c_{23} + a_2 + a_3 c_2 - a_4 s_2) + \\ a_6(c_1 s_4 + c_4 s_1 c_{23}) + d_7(c_5(c_1 s_4 + c_4 s_1 s_{23}) - s_5 s_1 s_{23}) \\ p_z = d_1 - a_4 c_2 - a_3 s_2 + (d_5 - d_7) c_{23} - (a_5 + d_7 c_4 c_5 + a_6 c_4) s_{23} \end{cases} $$

In the above expression, for the sake of brevity, let $s_1 = \sin\theta_1$, $c_1 = \cos\theta_1$, $c_{23} = \cos(\theta_2 + \theta_3)$, $s_{23} = \sin(\theta_2 + \theta_3)$ and similarly for the others.

(1) Joint Angle θ_1

$$ \theta_1 = A\tan 2(A_1, B_1), \quad (22) $$

where $A_1 = p_y - d_9 a_y$, $B_1 = p_x - d_9 a_x$.

(2) Joint Angle θ_2

$$ \theta_2 = A\tan 2(A_2, B_2) - A\tan 2(K, \pm\sqrt{A_2^2 + B_2^2 - K^2}), \quad (23) $$

where,

$$ \begin{cases} A_2 = k_1 a_3 - k_2 a_4 \\ B_2 = -k_1 a_4 - k_2 a_3 \\ k_1 = p_x c_1 + p_y s_1 - a_y d_7 s_1 - a_x d_7 c_1 - a_2 \\ k_2 = p_z - d_1 - a_z d_7 \\ K = \dfrac{k_1^2 + k_2^2 + a_3^2 + a_4^2 - a_5^2 - d_5^2}{2} \end{cases}. \quad (24) $$

(3) Joint Angle θ_3

$$ \theta_3 = A\tan 2(A_3, B_3) - A\tan 2(L, \pm\sqrt{A_3^2 + B_3^2 - L^2}). \quad (25) $$

In equation,

$$ \begin{cases} A_3 = a_3 a_5 - a_4 d_5 \\ B_3 = -(a_3 d_5 + a_4 a_5) \\ L = \dfrac{k_1^2 + k_2^2 - a_3^2 - a_4^2 - a_5^2 - d_5^2}{2} \end{cases}. \quad (26) $$

(4) Joint Angle θ_5

$$ \theta_5 = A\tan 2(\pm A_5, B_5), \quad (27) $$

where,

$$ \begin{cases} A_5 = -s_{23}(a_x c_1 + a_y s_1) - c_{23} a_z \\ B_5 = \sqrt{k_3^2 + k_4^2} \\ k_3 = c_{23}(a_x c_1 + a_y s_1) - s_{23} a_z \\ k_4 = a_y c_1 - a_x s_1 \end{cases}. \quad (28) $$

(5) Joint Angle θ_4

When $c_5 = 0$, $\theta_5 = \frac{\pi}{2}$, the manipulator falls into singularity, leading to the inability to determine θ_4, while $c_5 \neq 0$, $\theta_5 \neq \frac{\pi}{2}$, it is possible to calculate θ_4.

$$ \theta_4 = A\tan 2(A_4, B_4), \quad (29) $$

where,

$$ \begin{cases} A_4 = \dfrac{a_y c_1 - a_x s_1}{c_5} \\ B_4 = \dfrac{c_{23}(a_x c_1 + a_y s_1) - s_{23} a_z}{c_5} \end{cases}. \quad (30) $$

(6) Joint Angle θ_6

$$ \theta_6 = A\tan 2(A_6, B_6) - A\tan 2(N, \pm\sqrt{A_6^2 + B_6^2 - N^2}), \quad (31) $$

where, $A_6 = -o_z$, $B_6 = n_z$, $N = s_{23} s_4$.

During the computational process, the expression for the inverse solution includes both positive and negative determinants, indicating the possibility of two potential values for the joint angles. As a result, the SRL can adopt eight distinct sets of inverse solution configurations.

REFERENCES

[1] Y. Shen, S. Zastrow, J. Graf, and G. Reinhart, "An uncertainty-based evaluation approach for human-robot-cooperation within production systems," *Procedia CIRP*, vol. 41, pp. 376–381, 2016.

[2] S. M. Ji and X. H. Huang, "Review of development and application of industrial robot technology," *Journal of Mechanical & Electrical Engineering*, vol. 32, no. 1, pp. 1–13, 2015.

[3] Y. Tong and J. Liu, "Review of research and development of supernumerary robotic limbs," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 5, pp. 929–952, 2021.

[4] F. Vicentini, "Collaborative robotics: a survey," *Journal of Mechanical Design*, vol. 143, no. 4, p. 040802, 2021.

[5] A. Hentout, M. Aouache, A. Maoudj, and I. Akli, "Human–robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017," *Advanced Robotics*, vol. 33, no. 15-16, pp. 764–799, 2019.

[6] M. Al Sada, M. Khamis, A. Kato, S. Sugano, T. Nakajima, and F. Alt, "Challenges and opportunities of supernumerary robotic limbs," 2017.

[7] B. Yang, J. Huang, X. Chen, C. Xiong, and Y. Hasegawa, "Supernumerary robotic limbs: a review and future outlook," *IEEE Transactions on Medical Robotics and Bionics*, vol. 3, no. 3, pp. 623–639, 2021.

[8] G. Gourmelen, A. Verhulst, B. Navarro, T. Sasaki, G. Gowrishankar, and M. Inami, "Co-limbs: An intuitive collaborative control for wearable robotic arms," in *SIGGRAPH Asia 2019 Emerging Technologies*, 2019, pp. 9–10.

[9] A. M. Dollar and H. Herr, "Lower extremity exoskeletons and active orthoses: Challenges and state-of-the-art," *IEEE Transactions on robotics*, vol. 24, no. 1, pp. 144–158, 2008.

[10] I. Hussain, G. Salvietti, L. Meli, C. Pacchierotti, D. Cioncoloni, S. Rossi, and D. Prattichizzo, "Using the robotic sixth finger and vibrotactile feedback for grasp compensation in chronic stroke patients," in *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2015, pp. 67–72.

[11] M. A. Vélez-Guerrero, M. Callejas-Cuervo, J. C. Álvarez, and S. Mazzoleni, "Assessment of the mechanical support characteristics of a light and wearable robotic exoskeleton prototype applied to upper limb rehabilitation," *Sensors*, vol. 22, no. 11, p. 3999, 2022.

[12] D. J. Gonzalez and H. H. Asada, "Design of extra robotic legs for augmenting human payload capabilities by exploiting singularity and torque redistribution," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4348–4354.
[13] D. J. Gonzalez and H. H. Asada, "Hybrid open-loop closed-loop control of coupled human–robot balance during assisted stance transition with extra robotic legs," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1676–1683, 2019.
[14] C. Véronneau, J. Denis, L.-P. Lebel, M. Denninger, J.-S. Plante, and A. Girard, "A lightweight force-controllable wearable arm based on magnetorheological-hydrostatic actuators," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 4018–4024.
[15] C. Véronneau, J. Denis, L.-P. Lebel, M. Denninger, V. Blanchard, A. Girard, and J.-S. Plante, "Multifunctional remotely actuated 3-dof supernumerary robotic arm based on magnetorheological clutches and hydrostatic transmission lines," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2546–2553, 2020.
[16] Y. Tong and J. Liu, "Review of research and development of supernumerary robotic limbs," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 5, pp. 929–952, 2021.
[17] V. Vatsal and G. Hoffman, "Design and analysis of a wearable robotic forearm," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5489–5496.
[18] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, "Path planning and trajectory planning algorithms: A general overview," *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches*, pp. 3–27, 2015.
[19] D. Sidobre and K. Desormeaux, "Smooth cubic polynomial trajectories for human-robot interactions," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 3-4, pp. 851–869, 2019.
[20] X. Y. Zhang, W. Q. Lu, M. Bao, H. L. Wang, and G. Z. Pan, "Research on multi-axis motion control method based on high-order non-uniform rational b-spline interpolation," *Machinery and Electronics*, vol. 40, no. 10, pp. 25–31, 2022.
[21] X. Wang, A. Wang, D. Wang, W. Wang, B. Liang, and Y. Qi, "Repetitive control scheme of robotic manipulators based on improved b-spline function," *Complexity*, vol. 2021, pp. 1–15, 2021.
[22] L. Yu, S. Zhou, and S. Huang, "Trajectory optimization of the redundant manipulator with local variable period under multi-machine coordination," *Robotica*, vol. 41, no. 1, pp. 292–305, 2023.
[23] M. H. Lin, L. Zhang, P. F. Li, and X. H. Wang, "Trajectory planning of segmented manipulator based on complex trajectory," *J. Xi'an Polytech. Univ.*, vol. 34, no. 4, pp. 43–50, 2020.
[24] L. F. Wang, "Trajectory planning and application of six-degree-offreedom manipulator arm for rail-mounted inspection robot, doctoral dissertation," Ph.D. dissertation, Shaanxi: Chang'an University, 2023.
[25] W. J. Wang, Q. Tao, Y. T. Cao, X. H. Wang, and X. Zhang, "Robot time-optimal trajectory planning based on improved cuckoo search algorithm," *IEEE access*, vol. 8, pp. 86 923–86 933, 2020.
[26] S. Lu, B. X. Ding, and Y. M. Li, "Minimum-jerk trajectory planning pertaining to a translational 3-degree-of-freedom parallel manipulator through piecewise quintic polynomials interpolation," *Advances in Mechanical Engineering*, vol. 12, no. 3, p. 1687814020913667, 2020.
[27] H. Chen, "Research on 6-dof robotic arm trajectory planning based on given key points," Master's thesis, Nanhua University, 2019.
[28] P. Wu and L. Tian, "Trajectory planning of industrial robots based on fruit-fly optimization algorithm with mean learning strategy," *Journal of Machine Design*, vol. 40, no. 2, pp. 126–133, 2023.
[29] Y. Yang, H. Z. Xu, S. H. Li, L. L. Zhang, and X. M. Yao, "Time-optimal trajectory optimization of serial robotic manipulator with kinematic and dynamic limits based on improved particle swarm optimization," *The International Journal of Advanced Manufacturing Technology*, vol. 120, no. 1-2, pp. 1253–1264, 2022.
[30] Y. H. Zhao, D. X. Liu, Z. Y. Liu, and J. R. Tan, "Time-optimal trajectory planning of manipulator based on multi-group competition squirrel search algorithm," *Journal of ZheJiang University (Engineering Science)*, vol. 56, no. 12, pp. 2321–2329, 2022.
[31] S. S. Li, S. Y. Wang, W. F. Zhang, and J. C. Li, "Time optimal trajectory planning of manipulator based on improved bat algorithm," *Light Industry Machinery*, vol. 40, no. 2, pp. 7–12, 2022.
[32] H. Esfandiar and M. H. Korayem, "Optimal point to point path planning of flexible manipulator under large deformation by using harmony search method," *Journal of Theoretical and Applied Mechanics*, vol. 54, no. 1, pp. 179–193, 2015.
[33] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *American Society of Mechanical Engineers*, vol. 22, no. 2, pp. 215–221, 1955.
[34] W. Khalil and J. Kleinfinger, "A new geometric notation for open and closed-loop robots," in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3. IEEE, 1986, pp. 1174–1179.
[35] M. Braik, A. Hammouri, J. Atwan, M. A. Al-Betar, and M. A. Awadallah, "White shark optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems," *Knowledge-Based Systems*, vol. 243, p. 108457, 2022.

Jiayu Wang received her B.S degree in automation from the School of Control Science and Engineering of Shandong University, Jinan, China, in 2021. She is currently working toward her M.S. degree at School of Control Science and Engineering, Shandong University, Jinan, China. She is engaged in special robot control research.

Ruyue Sun received her B.S degree in Measurement and Control Technology and Instrumentation Program from the naval aeronautical engineering institute, Yantai, China, in 2016; the M.S. degree in control science and engineering from Shandong University, Jinan, China, in 2023. She is currently working at PLA Units No.92292, Qingdao, China. Her research interests include special robot control and intelligent control.

Yukun Zheng received M.S. degree from the College of Mechanical Engineering and Applied Electronics Technology, Beijing University of Technology, Beijing, China, in 2017 and the Ph.D. degree in control theory and control engineering from Shandong University, China, in 2023. Since 2023, he has been with the School of Control Science and Engineering, Shandong University, China, as a Postdoctoral Researcher. His research interests include robot control, mobile manipulator control and intelligent control.

Rui Song received the B.S. degree in industrial automation and the M.S. degree in control theory and engineering from Shandong University of Science and Technology, in 1998 and 2001, respectively, and the Dr. Eng. degree in control theory and engineering from Shandong University, Jinan, China in 2011. He is currently a professor with the School of Control Science and Engineering, Shandong University. His research interests include intelligent robots and control theory.

Yibin Li received the B.S. degree in automation engineering from Tianjin University, Tianjin, China, in 1982; the M.S. degree in electrical engineering from Shandong University of Science and Technology, Jinan, China, in 1990; and the Dr. Eng. degree in automation engineering from Tianjin University in 2008. He is currently a professor with the School of Control Science and Engineering, Shandong University. His research interests include intelligent robots, special robots, intelligent vehicles, intelligent control, and mechatronic.