International Journal of Mechanical and Electrical Engineering
ISSN: 3005-9615 (Print), ISSN: 3005-7132 (Online) | Volume 7, Number 2, Year 2025
DOI: https://doi.org/10.62051/ijmee.v7n2.01
Journal homepage: https://wepub.org/index.php/IJMEE/index

Kinematic Analysis of SNR3-C30 Robotic Arm based on Matlab

Qia Tang *

Beijing University of Posts and Telecommunications, Beijing, China
* Corresponding Author Email: 2022213470@bupt.cn

ABSTRACT

With the continuous advancement of industrial automation, robotic arms have been widely adopted across various fields, making performance optimization a key research focus. As a critical component for enhancing robotic arm performance, kinematic analysis plays a vital role in achieving precise control. This study utilizes the MatlabRobotics toolbox to conduct kinematic analysis on the SNR3-C30 robotic arm. By establishing a mathematical model of the robotic arm and solving forward/reverse kinematic equations, we implemented a fifth-order polynomial interpolation algorithm for trajectory planning in the joint space. The research demonstrates that the established model accurately describes the robotic arm's motion characteristics, while the forward/reverse kinematic solution method proves effective and reliable. These findings provide crucial theoretical foundations for optimizing design and developing control strategies for the SNR3-C30 robotic arm, significantly improving operational precision and efficiency in real-world applications. This work holds great significance for advancing robotic arm technology development.

KEYWORDS

SNR3-C30 Robotic Arm; Matlab Robotics; Kinematic Analysis; Forward and Inverse Kinematics; Dynamics.

1. INTRODUCTION

1.1. Research Background

As a core component of modern automation technology, robotic arms have demonstrated extensive application prospects in industrial automation, aerospace, and medical fields. In industrial automation, they efficiently perform complex tasks such as assembly, welding, and material handling, significantly enhancing production efficiency and quality control standards. In aerospace applications, robotic arms leverage their high precision and flexibility for critical operations like satellite maintenance and space station construction, providing vital technical support for human space exploration [4]. Furthermore, in medical practice, robotic arms assist surgeons in performing minimally invasive procedures through precise control, substantially improving surgical safety and success rates. However, optimizing robotic arm performance requires in-depth study of their motion characteristics, with kinematic analysis serving as a crucial method to reveal operational principles. Systematic analysis of kinematic parameters including position, velocity, and acceleration can effectively enhance operational performance, meeting demands under complex real-world conditions [1]. Therefore, conducting research on robotic arm kinematics analysis holds significant theoretical value and practical significance.

1.2. Significance of the Study

The use of the MatlabRobotics toolbox for kinematic analysis of robotic arms offers distinct advantages. Firstly, it provides a comprehensive function library and visual interface that significantly simplifies computational processes in modeling and simulating robotic arm movements, allowing researchers to focus more on algorithm design and performance optimization. Secondly, MatlabRobotics enables intuitive visualization of 3D models, clearly demonstrating the motion trajectories and joint changes of robotic arms, providing clear reference points for design and control. Furthermore, Matlab-based kinematic analysis can seamlessly integrate with other disciplines such as dynamics analysis and trajectory planning, establishing a solid foundation for comprehensive performance optimization [2]. Therefore, applying the Matlab Robotics toolbox to analyze the SNR3-C30 robotic arm not only enhances research efficiency but also provides reliable theoretical support for practical applications.

1.3. Current Research Gaps

While existing research has achieved certain accomplishments in kinematic analysis and Matlab applications for robotic arms, there remain gaps in the study of motion analysis for the SNR3-C30 model using Matlab Robotics. First, current literature predominantly focuses on general-purpose robotic arm modeling and simulation, with relatively limited research dedicated to this specific model. Particularly, comprehensive analyses incorporating dynamic factors still lack in-depth exploration. Second, although multiple inverse kinematics algorithms have been proposed, they demonstrate limitations when handling multi-solution scenarios and singular points, failing to adequately address complex operational requirements in practical applications. To bridge these gaps, this study establishes precise mathematical models and efficient solution algorithms, providing innovative theoretical foundations and methodological support for motion analysis of the SNR3-C30 robotic arm.

1.4. Research Objectives

This study aims to establish an accurate mathematical model for the SNR3-C30 robotic arm through kinematic analysis, solve its forward and inverse kinematics problems, and consider the impact of dynamic factors on the arm's performance. Specifically, we first construct the linkage coordinate system and kinematic equations based on the D-H parameter method to ensure model accuracy and applicability. Next, forward kinematics analysis is performed using the MatlabRobotics toolbox to verify model correctness. Subsequently, we explore inverse kinematics solution methods, analyze multi-solution scenarios, and propose strategies for selecting appropriate solutions. Finally, by integrating dynamic model analysis, we examine force distribution under different motion states, providing theoretical foundations for control system design. Achieving these objectives will lay a solid foundation for performance optimization and practical applications of the SNR3-C30 robotic arm.

2. STRUCTURAL ANALYSIS OF SNR3-C30 ROBOT ARM

2.1. Introduction of SNR3-C30 Mechanical Arm

The SNR3-C30 six-degree-of-freedom robotic arm achieves precise positioning of any object within its workspace. Comprising a base, rotating platform, upper arm, lower arm, wrist joint, end effector, and hydraulic system, this device adheres to China's mainstream mechanical arm design standards. The base is securely mounted on the installation platform using four M16 hex socket screws and two φ12 positioning pins, effectively preventing operational instability. Its main structure functions as a serial linkage mechanism connected through six articulated joints. The end effector features versatile actuator compatibility to meet diverse production requirements, with each joint's internal linkage

integrated with drive motors. During operation, motion codes from the teaching pendant are processed by the control cabinet into precise motion commands, enabling coordinated sequential movements of corresponding joint motors to accomplish predetermined tasks. The compact design of the SNR3-C30 robotic arm delivers lightweight construction and exceptional operational flexibility.

2.2. Joint Types and Parameters

The SNR3-C30 robotic arm features all rotating joints, enabling complex spatial movements without requiring additional articulation. Each joint's angular range is precisely engineered to balance operational requirements with joint safety, preventing overextension or collisions. The mechanical parameters and maximum rotational speeds of each joint are detailed in Table 1 and Table 2 respectively.

Table 1. Mechanical parameters

Parameter	Parameter values
Type	SNR3-C30
Maximum load weight (kg)	3
Body weight (kg)	27
Repeatability accuracy (mm)	±2

Table 2. Maximum speed of each joint

Joint	1	2	3	4	5	6
Maximum Speed	250°/s	250°/s	300°/s	400°/s	360°/s	600°/s

As shown in Table 1, the SNR3-C30 robotic arm has a maximum payload of 3kg and a total weight of 27kg, making it easy to install and transport. Its repeat positioning accuracy is ±2mm, which helps improve processing precision and efficiency. Data in Table 2 indicates that the end effector motor can reach a maximum rotational speed of 600°/s, ensuring operational efficiency. The device is equipped with an international standard flange, enabling compatibility with various tools for tasks like welding and spraying, effectively expanding its application scope. Special note: When operating within the robotic arm's working area (safety zone), personnel must avoid entering to prevent injuries, which also serves as the workpiece positioning area. The SNR3-C30 has a maximum rotation radius of 637mm, offering a compact yet highly efficient workspace particularly suitable for confined spaces. Analysis of joint motion parameters provides foundational data support for subsequent kinematic modeling.

2.3. End Effector Characteristics

As the core component of SNR3-C30 robotic arm, the end effector plays a pivotal role in its design. Featuring tool interfaces for flexible module replacement, this component combines high rigidity with lightweight construction to ensure stability during high-speed operations. In kinematic analysis, the end effector's pose matrix serves as the fundamental basis for solving both forward and inverse kinematics problems. By precisely defining its spatial position and orientation, this matrix establishes

a mapping relationship between joint angles and end-effector positions, providing theoretical support for motion planning and control. Furthermore, the geometric characteristics of the end effector directly influence the complexity of inverse kinematics solutions and the occurrence of multiple solutions, requiring thorough analysis during robotic arm design and optimization processes.

3. ESTABLISHMENT OF MECHANICAL MATHEMATICAL MODEL BASED ON MATLABROBOTICS

3.1. DH Parameter Method

The DH parameter method (DH parameter method) is a widely used standard approach in mechanical arm kinematic modeling. Its core concept involves defining four parameters for each link: joint angle (θ), link length (a), joint offset (d), and link twist (α), which collectively describe the relative relationships between mechanical arm links. Specifically, the DH parameter method first establishes a local coordinate system for each joint and link, then converts these local coordinate systems into the base coordinate system through homogeneous transformation matrices to form a complete kinematic model. For the SNR3-C30 mechanical arm, its joint layout and link connection configuration determine the selection of DH parameters. Additionally, the DH parameter method demonstrates advantages in versatility and flexibility, enabling adaptation to various mechanical arm structures. The SNR3-C30 mechanical arm studied in this paper features six joints, each capable of rotational movement to facilitate diverse tasks. To facilitate intuitive analysis, the base is designated as link 0, while the remaining joints are labeled as link 1, 2, 3, 4, 5, and 6. A D-H coordinate system is established using SolidWorks software to model the geometric relationships between link coordinate systems.

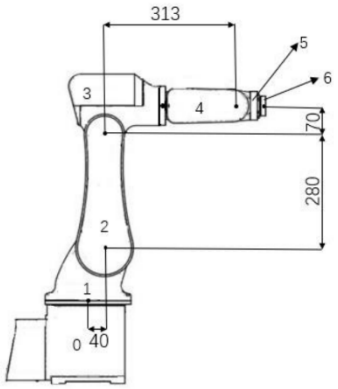

Figure 1. 3D Model of the SNR3-C30 Robotic Arm

3.2. DH Coordinate System Establishment

SNR3-C30 type mechanical arm is a series open chain mechanical arm, see Figure 2.

Step 1: Since the connecting rod 0 (base) is fixed on the mounting platform and remains stable during the movement of the robotic arm, to simplify the establishment of coordinate systems, coordinate systems {0} and {1} are combined. Both coordinate systems follow the right-hand principle, and their corresponding x, y, and z axes overlap pairwise;

Step 2: Establish coordinate systems {2} and {3} on link 2 and link 3 respectively. First, determine that the Z2-axis and Z3-axis are both perpendicular to the paper with their planes facing outward. Next, align the x2-axis with the common normal a2 of Z2 and Z3 axes, ensuring it runs along the a2 direction from the 2-axis toward the 3-axis. The y2-axis direction is established as horizontally leftward according to the right-hand rule.

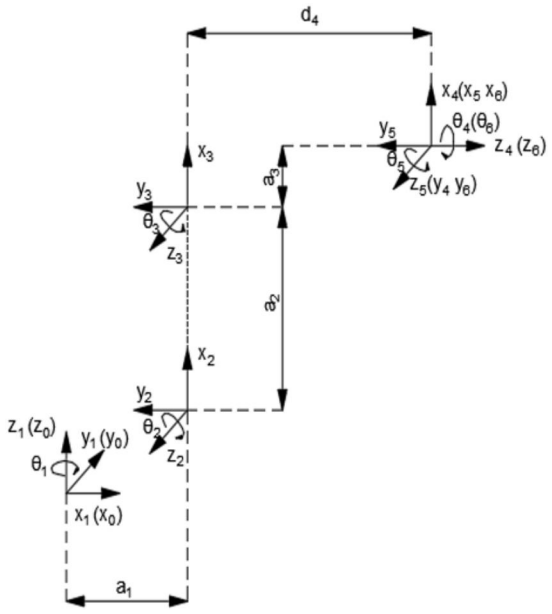

Figure 2. D-H coordinate system of SNR3-C30 robotic arm

Step 3: Since the axes of the three joints (4, 5, 6) intersect at a single point, the coordinate systems {4}, {5}, and {6} are aligned according to Pieper's principle. As the Z4-axis runs horizontally to the right, the x3-axis coincides with the common normal a3 of Z3 and Z4 axes, extending from the 3-axis toward the 4-axis while the y4-axis points outward perpendicular to the paper plane. Similarly, the Z5-axis extends outward perpendicular to the paper plane, with the x5-axis aligning with the common normal of Z4 and Z5 axes. Since all three axes share their origin, this common normal has zero length, so its direction is directly along the common perpendicular of Z4 and Z5 axes, with the y5-axis pointing horizontally to the left. The Z6-axis extends horizontally to the right. As there is no coordinate system {7}, the x6-axis can be oriented arbitrarily but is conventionally set as vertically upward in this study, making the y6-axis point outward perpendicular to the paper plane. When establishing the D-H coordinate system, clockwise rotation is defined as positive and counterclockwise as negative. Based on the 3D model of the SNR3-C3 robotic arm and the D-H coordinate system, the parameters *θi, di, αi-1,* and *ai-1* are calculated as shown in Table 3 below.

Table 3. D-H parameters of SNR3-C30 robotic arm

Joint *i*	*αi- 1* /º	*ai- 1*/mm	*di*/mm	*θi*	actuating range/º
1	0	0	0	0	-170~+170
2	-90	40	0	-90	-70~+120
3	0	280	0	0	-110~+70
4	-90	70	313	0	-180~+180
5	90	0	0	0	-120~+120
6	-90	0	0	0	-360~+360

For the SNR3-C30 robotic arm, its joint layout contains multiple non-orthogonal axis lines, which may cause errors when using traditional DH parameterization to describe joint transformations. Therefore, an improved DH parameterization method can better capture the actual motion behavior of the robotic arm by introducing virtual joints or adjusting joint offsets. Specifically, the enhanced

DH parameterization method can be implemented through the following steps: First, conduct a detailed analysis of the robotic arm's joint axes to identify any non-orthogonal configurations; Second, adjust relevant parameters in the DH parameter table based on the analysis results; Finally, validate the effectiveness of the improved model through simulation experiments.

3.3. Construction of Simulation Model

When conducting mechanical mathematical modeling in MATLAB's Robotics module, the first step is to determine the fundamental functions to be used. Since robotic arms consist of multiple serially connected linkages, we must first construct joint links using the LINK function. The general form of this function is L=LINK ([theta D A alpha sigma], CONVENTION), where theta represents the rotational angle of each joint; d denotes the distance between two linkages along their axis; a indicates the normal length; alpha signifies the twist angle of each joint linkage; and sigma represents translational/rotational transformations between joints (where 1 indicates translation and 0 indicates rotation). The CONVENTION parameter offers two options: standard D-H parameters or modified D-H parameters. In this study, we employ standard D-H parameters to establish the SNR3-C30 mechanical mathematical model. Therefore, when coding, it's essential to add a description after the LINK function. The specific code is as follows:

Table 4. Connecting rod codes for each joint

L1=Link([0	0	0.04	-pi/2	0	],' standard');
L2=Link([0	0	0.28	0	0	],' standard');
L3=Link([0	0	0.07	-pi/2	0	],' standard');
L4=Link([0	0.313	0	pi/2	0	],' standard');
L5=Link([0	0	0	-pi/2	0	],' standard');
L6=Link([0	0	0	0	0	],' standard');

robot=SerialLink ([L1 L2 L3 L4 L5 L6],'name','SNR3-C30');

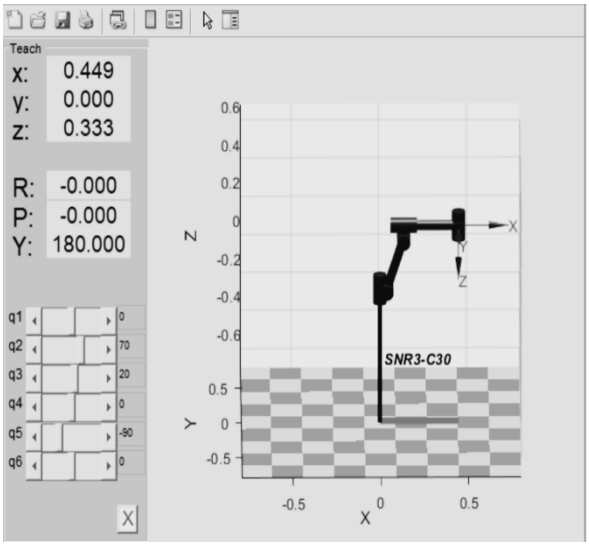

Figure 3. Mathematical model of SNR3-C30 robotic arm

On the basis of the establishment of joint link codes, the program was imported into MATLAB Robotics module, and the mechanical number mathematical model was established by using the robot function and named SNR3-C30. The initial position variables of six joints of the mechanical arm were preset as [0 0.38pi 0.11pi 0 -0.5pi 0], and the simulation model is shown in Figure 3.

4. FORWARD AND REVERSE KINEMATICS ANALYSIS AND SIMULATION OF ROBOTIC ARM

4.1. Kinematic Analysis

From the relationship between adjacent joints and linkages of the robotic arm, the pose coordinate matrix of linkages i relative to linkages i-1 is:

$$ {}^{i-1}_{i}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i\cos\alpha_{i-1} & \cos\theta_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} & -d_i\sin\alpha_{i-1} \\ \sin\theta_i\sin\alpha_{i-1} & \cos\theta_i\sin\alpha_{i-1} & \cos\alpha_{i-1} & d_i\cos\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1} $$

Because the mechanical arm moves when multiple connecting rods act together, the pose matrix of which is the end effector relative to the base coordinate system is equal to the sequential multiplication ${}^{6}_{0}T$ of the joint connecting rod matrices. The expression is as follows:

$$ {}^{0}_{6}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T\,{}^{4}_{5}T\,{}^{5}_{6}T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

Substitute the D-H parameters of the robotic arm in Table 3 into the above formula, and get the even-order variation matrices of each link coordinate system relative to the previous link coordinate system respectively:

$$ {}^{0}_{1}T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{1}_{2}T = \begin{bmatrix} c_2 & -s_2 & 0 & a_1 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{2}_{3}T = \begin{bmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{3}_{4}T = \begin{bmatrix} c_4 & -s_4 & 0 & a_3 \\ 0 & 0 & 1 & d_4 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{4}_{5}T = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{5}_{6}T = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3} $$

In order to simplify the formula, in the above formula, si represents $\sin\theta_I$, c_i represents $\cos\theta_I$, s12 represents $\sin(\theta_1+\theta_2)$, c12 represents $\cos(\theta_1+\theta_2)$,

Therefore, the calculation results of each element in ${}^{0}_{6}T$ matrix are as follows:

$$
\begin{aligned}
\mathrm{n}_x &= c_1\left[c_{23}\left(c_4c_5c_6 - s_4s_6\right) - s_{23}s_5c_6\right] + s_1\left(s_4c_5c_6 + c_4s_6\right)\\
\mathrm{n}_y &= s_1\left[c_{23}\left(c_4c_5c_6 - s_4s_6\right) - s_{23}s_5c_6\right] - c_1\left(s_4c_5c_6 + c_4s_6\right)\\
\mathrm{n}_z &= s_{23}\left(s_4s_6 - c_4c_5c_6\right) - c_{23}s_5c_6\\
o_x &= c_1\left[c_{23}\left(-c_4c_5s_6 - s_4c_6\right) + s_{23}s_5s_6\right] + s_1\left(c_4c_6 - s_4c_5s_6\right)\\
o_y &= s_1\left[c_{23}\left(-c_4c_5s_6 - s_4c_6\right) + s_{23}s_5s_6\right] - c_1\left(c_4c_6 - s_4c_5s_6\right)\\
\mathrm{o}_z &= s_{23}\left(s_4c_6 + c_4c_5s_6\right) + c_{23}s_5s_6\\
a_x &= c_1\left(-c_{23}c_4s_5 - s_{23}c_5\right) - s_1s_4s_5\\
a_y &= s_1\left(-c_{23}c_4s_5 - s_{23}c_5\right) + c_1s_4s_5\\
a_z &= \mathrm{s}_{23}c_4s_5 - c_{23}c_5\\
p_x &= c_1\left(a_1 + a_2c_2 + a_3c_{23} - d_4s_{23}\right)\\
p_y &= s_1\left(a_1 + a_2c_2 + a_3c_{23} - d_4s_{23}\right)\\
p_z &= -a_2s_2 - a_3s_{23} - d_4c_{23}
\end{aligned}
\tag{4}
$$

4.2. Simulation Verification

The initial angles of the six joints of SNR3-C3 0 robotic arm are set as 0, pi/2, pi/3, 0, -pi/2 and 0 respectively. They together form a joint vector q= [0 pi/2 pi/3 0-pi/2 0]. The elements of the end effector pose matrix are calculated according to Equation 4 as follows:

$$
{}^0_6T = \begin{bmatrix} 0.5000 & 0 & -0.8660 & -0.1771 \\ 0 & -1 & 0 & 0 \\ -0.8660 & 0 & -0.5000 & -0.04393 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}
$$

In MATLAB, the joint vector q is substituted into the function to solve the problem, and the end effector pose matrix is obtained. After comparison, the elements of this matrix are the same as Equation 5, which proves the correctness of the positive solution algorithm.

5. ANALYSIS AND SIMULATION OF INVERSE KINEMATICS OF MANIPULATOR

5.1. Reverse Kinematics Analysis

The inverse kinematics of a robotic arm is very important for real-time control and trajectory planning, and has significant practical value. This problem mainly covers three categories: solution diversity, solution existence and solution method:

5.1.1. Solution Diversity:

When an end effector reaches a target pose, multiple combinations of joint variables may satisfy the requirements. The diversity of solutions increases as the number of degrees of freedom in the joints or the range of motion expands. In practical applications, the optimal solution must be selected based on optimization criteria such as energy consumption, time, and path length, with specific evaluation metrics determined by actual operating conditions.

5.1.2. Existence of Solution:

It is necessary to determine whether the target point is located in the working space of the robotic arm. If this condition is met, the joint variables are further solved and verified whether the joint variables are within the allowable motion range. When both conditions are met, the solution exists.

5.1.3. Solution Method:

5.1.3.1 Valuation Method

For example, Newton iteration method, neural network method, genetic algorithm, etc.: rely on iterative convergence, the calculation complexity is high;

5.1.3.2 Analytical Solutions

For example, geometric method and algebraic method: they have closed form solution, but usually involve complex matrix operation and mathematical model construction.

5.2. Design of Geometric Inverse Optimization Algorithm

The geometric inverse optimization algorithm combines the rotation matrix characteristics of geometry and algebra, and can quickly find all the solutions. Unlike the traditional method, it does not need to carry out multiple inverse matrix operations, which greatly reduces the calculation amount.

The pose matrix of the end effector of the manipulator is:

$$ {}^{0}_{6}T=\begin{bmatrix} \mathrm{n}_x & \mathrm{o}_x & a_x & \mathrm{p}_x \\ \mathrm{n}_y & \mathrm{o}_y & a_y & \mathrm{p}_y \\ \mathrm{n}_z & \mathrm{o}_z & a_z & \mathrm{p}_z \\ 0 & 0 & 0 & 1 \end{bmatrix}=\begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix} \tag{6} $$

Here, R denotes the rotation matrix of the end effector's coordinate system relative to the base coordinate system, and P represents the position matrix in Cartesian space. The improved algorithm separates the rotation matrix from the pose matrix and utilizes the position matrix combined with geometric methods to independently solve for the first three joint angles θ_1, θ_2, θ_3. Subsequently, it solves the remaining joint angles through the rotation matrices of linkages 4, 5, and 6. The specific steps are as follows:

5.2.1. Step 1: Solve the First Three Joint Angles θ_1, θ_2, θ_3

According to the D-H coordinate system and the end effector pose matrix, the position matrix of the reference point PW of the robotic arm wrist can be obtained as $[px\ py\ pz]^T$. The geometric relationship of each joint link of the robotic arm is simplified into a rigid body, as shown in Figure 4.

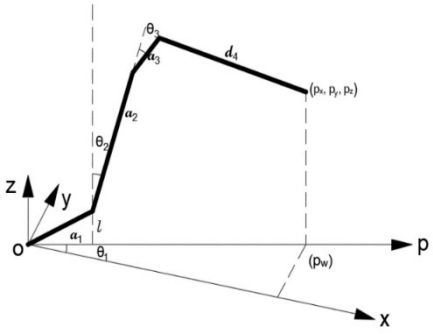

Figure 4. The geometric position relationship of the robot arm connecting rod

According to the figure above, the position relationship function of the end effector of the robotic arm is expressed as:

$$ \begin{aligned} P_x &= \left[a_1 + a_2\sin\theta_2 + a_3\sin\left(\theta_2+\theta_3\right) + d_4\cos\left(\theta_2+\theta_3\right)\right]\cos\theta_1 \\ P_y &= \left[a_1 + a_2\sin\theta_2 + a_3\sin\left(\theta_2+\theta_3\right) + d_4\cos\left(\theta_2+\theta_3\right)\right]\sin\theta_1 \\ P_z &= l + a_2\cos\theta_2 + a_3\cos\left(\theta_2+\theta_3\right) - d_4\sin\left(\theta_2+\theta_3\right) \end{aligned} \tag{7} $$

(1) Solve θ_1

From the projection of the robot wrist reference point PW on the x0y plane:

$$\theta_1 = \operatorname{atan2}\left(p_y, p_x\right) \text{ or } \theta_1 = \operatorname{atan2}\left(-p_y, -p_x\right) \tag{8}$$

(2) Solve θ_2

Add together the equations in Equation 7 after squaring:

$$\left(P_x / \cos\theta_1 - a_1\right)^2 + a_2^2 + \left(P_z - l\right)^2 - a_3^2 - d_4^2 = 2\left(P_x / \cos\theta_1 - a_1\right)a_2 \sin\theta_2 + 2\left(P_z - l\right)a_2 \cos\theta_2 \tag{9}$$

$$\text{Ream} \begin{cases} A = P_x / \cos\theta_1 - a_1 \\ B = P_z - l \\ \sin\varphi_1 = B / \left(\sqrt{A^2 + B^2}\right) \\ \cos\varphi_1 = A / \left(\sqrt{A^2 + B^2}\right) \end{cases}$$

Solutions

$$\theta_2 = \operatorname{asin} \frac{a_3^2 + d_4^2 - a_2^2 - A^2 - B^2}{-2a_2\sqrt{A^2 + B^2}} - a\tan 2\frac{B}{A} \tag{10}$$

(3) Solve θ_3

Substituting θ into Equation 7, we get:

$$P_z - l - a_2 \cos\theta_2 = a_3 \cos\left(\theta_2 + \theta_3\right) - d_4 \sin\left(\theta_2 + \theta_3\right) \tag{11}$$

$$\text{Ream} \begin{cases} \sin\varphi_2 = a_3 / \left(\sqrt{a_3^2 + d_4^2}\right) \\ \cos\varphi_2 = d_4 / \left(\sqrt{a_3^2 + d_4^2}\right) \end{cases}$$

Solutions:

$$\theta_3 = a\tan 2\left[P_z - l - a_2\cos\theta_2, \pm\sqrt{a_3^2 + d_4^2 - \left(P_z - l - a_2\cos\theta_2\right)^2}\right] - a\tan 2(a_3, d_4) \tag{12}$$

5.2.2. Step 2: Solve for the Three Joint Angles θ4, θ5, θ6

The rotation matrix between two adjacent linkages is:

$$R_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \sin\theta_i\cos\alpha_{i-1} & \cos\theta_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} \\ \sin\theta_i\sin\alpha_{i-1} & \cos\theta_i\sin\alpha_{i-1} & \cos\alpha_{i-1} \end{bmatrix} \tag{13}$$

From the orthogonal property of rotation matrix, we can get:

$$R_4R_5R_6 = \left(R_1R_2R_3\right)^{-1}R = R_3^T R_2^T R_1^T R \tag{14}$$

Solving θ_5

Since the left and right sides of Equation 14 are equal (2,3), we can get:

$$\theta_5 = \arccos\left(-a_x c_1 s_{23} - a_y s_1 s_{23} - a_z c_{23}\right) \tag{15}$$

Solving θ_4

Equations 14 on both sides (1,3) and (3,3) are equal, that is, the two equations:

$$\begin{cases} c_1 c_{23} a_x + s_1 c_{23} a_y - s_{23} a_z = -c_4 s_5 \\ -s_1 a_x + c_1 a_y = s_4 s_5 \end{cases}$$

1) When $0 < \theta_5 \leq 120$:

$$\theta_4 = \text{atan}\left(-s_1 a_x + c_1 a_y, -c_1 c_{23} a_x - s_1 c_{23} a_y + s_{23} a_z\right)$$

2) When-$120° \leq \theta_5 < 0$:

$$\theta_4 = \text{atan}\left(s_1 a_x - c_1 a_y, c_1 c_{23} a_x + s_1 c_{23} a_y - s_{23} a_z\right)$$

Solving θ_6

Equations 14 on both sides (2,1) and (2,2) are equal, that is, the two equations:

$$\begin{cases} -c_1 s_{23} n_x - s_1 s_{23} n_y - c_{23} n_z = s_5 c_6 \\ -c_1 s_{23} o_x - s_1 s_{23} o_y - c_{23} o_z = -s_5 s_6 \end{cases}$$

1) When $0 < \theta_5 \leq 120$:

$$\theta_6 = \text{atan}\left(c_1 s_{23} o_x + s_1 s_{23} o_y + c_{23} o_z, -c_1 s_{23} n_x - s_1 s_{23} n_y - c_{23} n_z\right)$$

2) When-$120° \leq \theta_5 \leq 0$:

$$\theta_6 = \text{atan}\left(-c_1 s_{23} o_x - s_1 s_{23} o_y - c_{23} o_z, c_1 s_{23} n_x + s_1 s_{23} n_y + c_{23} n_z\right)$$

5.3. Simulation Verification

The MATLAB simulation of robotic arm inverse kinematics consists of two main components. The first part involves verifying the correctness of the inverse solution obtained through the improved algorithm using MATLAB's ikine function. The second part demonstrates the superiority of the proposed algorithm by comparing it with geometric methods and analytical approaches.

5.3.1. Verify the Correctness of the Improved Algorithm

The pose matrix of the SNR3-C30 robotic arm end effector at the target point is shown in Equation 5. Through an improved inverse kinematics algorithm, the corresponding joint variables are determined as q= [0.0001 1.5699 1.0473 0.0011-1.5716 0.0019]. The MATLAB software directly calculates these joint variables using the ikine function, and the simulation results match those obtained from the optimization algorithm, validating the effectiveness of the enhanced approach.

6. SNR3-C30 ROBOTIC ARM TRAJECTORY PLANNING

Trajectory planning involves solving inverse kinematics for the end-effector's motion path during robotic arm operation, determining the joint motion states and generating continuous fitting curves. There are two primary approaches to trajectory planning: the first focuses on joint space trajectory planning, while the second employs Cartesian space trajectory planning. These two methods are respectively applied to coordinate robotic arm movements.

6.1. Trajectory Planning under Joint Space

Trajectory planning in joint space involves adjusting the initial and final joint angles as variables, converting them into time-dependent functions, and imposing constraints on angular velocity and acceleration. The jtraj function in MATLABRobotics Toolbox is used to generate the specified joint space trajectory. The jtraj function requires six joint angle vectors (representing both initial and target positions) along with a time duration vector. With the initial joint angle set to q0 and the final joint angle q1, the program sets a total sampling time of 5 seconds (equivalent to 100 steps). The resulting joint space trajectory is illustrated in Figure 5, while the corresponding curves of time versus joint angular velocity and acceleration are shown in Figures 6 and 7.

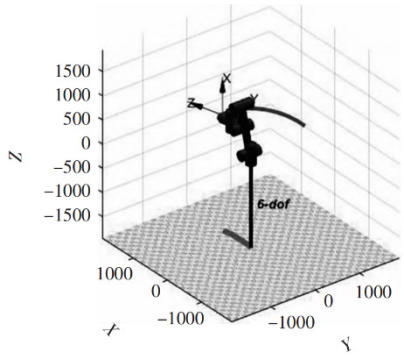

Figure 5. Space motion trajectory diagram of joint

The angular velocity and acceleration curves of the six joints in Figures 6 and 7 demonstrate smooth continuity without abrupt changes. As shown, the joint angular velocities start at zero and reach peak values during mid-motion, while the angular accelerations remain constant. Both velocity and acceleration curves maintain consistent smoothness throughout the motion, indicating stable operation of the SNR3-C30 robotic arm.

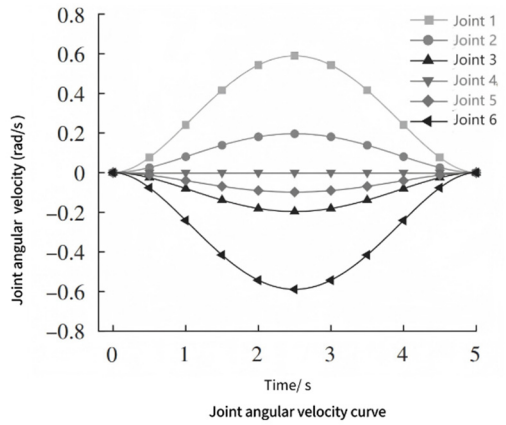

Figure 6. Joint angular velocity curve

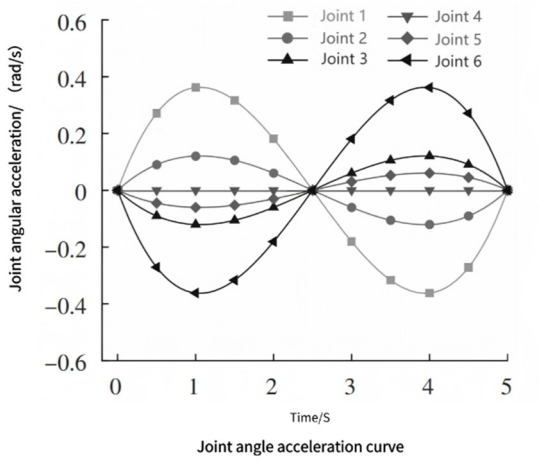

Figure 7. Joint angular acceleration curve

6.2. Cartesian Space Trajectory Planning

The starting and ending points for Cartesian space trajectory planning are determined by the initial and terminal coordinates of joint space trajectory planning: T_1 = transI (1136,0, 186) and T_2 = transI(-51,726,419). Using these coordinates as the mechanical arm's initial and final positions, we call the ctraj function while constraining its angular velocity and acceleration. Through five-order interpolation, we generate the trajectory plan for the robotic arm's end-effector displacement in Cartesian space, as shown in Figure 8. The actual motion path of the robotic arm's end-effector in Cartesian space is illustrated in Figure 9.

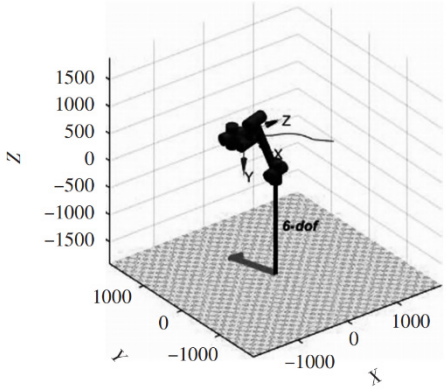

Figure 8. The trajectory diagram of the end point movement of the Cartesian space manipulator

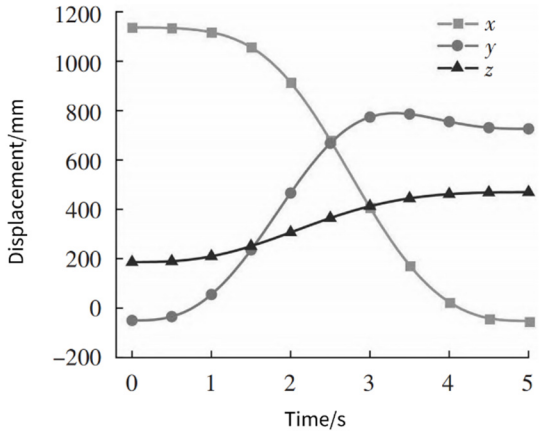

Figure 9. Diagram of Descartes' space motion Curve

As shown in Figures 7 and 9, trajectory planning in Cartesian space yields reasonable trajectories without abrupt changes, all reaching the designated positions. For joint space trajectory planning, we only need inverse solving for the starting point, several intermediate points, and the end effector. In contrast, Cartesian trajectory planning requires inverse solving for each path point, which must be transformed into joint space coordinates. This results in significantly higher computational load for Cartesian trajectory planning. Comparing the end-effector coordinate systems in Figures 5 and 8 reveals that Cartesian trajectory planning may achieve the target position of the end effector's flange but fails to attain the desired flange orientation. Therefore, joint space planning demonstrates superior precision.

7. CONCLUSION

7.1. Summary of Research Results

This study conducted a comprehensive kinematic analysis of the SNR3-C30 robotic arm using the MatlabRobotics toolbox, yielding significant research outcomes. First, in model development, we successfully established an accurate mathematical model for the robotic arm by employing the Denary Parameter Method (DPM) with its improved variant. This model not only precisely describes the geometric relationships between its joints but also lays a solid foundation for subsequent forward and inverse kinematic analyses. Second, through solving and simulating the pose of the robotic arm's end effector during forward kinematics analysis, we validated the model's accuracy and revealed motion trajectory characteristics at different joint angles. Additionally, the geometric inverse optimization algorithm proposed in reverse kinematics analysis significantly improved computational efficiency by avoiding extensive matrix operations in traditional methods while effectively addressing multi-solution issues, ensuring both physical validity and practical applicability of the results. Finally, incorporating dynamic factors like velocity and acceleration, we developed a complete dynamic model. Simulation analysis demonstrated joint force distribution under various motion conditions, providing crucial references for practical robotic arm control.

7.2. Limitations of the Study

While this study has achieved notable progress in robotic arm kinematic analysis, several limitations remain. First, although the improved DH parameter method was employed to enhance accuracy during model development, potential errors may still occur due to neglecting minor deformations or frictional effects in complex operational scenarios. Second, while the proposed optimization algorithm efficiently yields feasible solutions for inverse kinematics, its performance requires further refinement when handling extreme poses or multiple singular points. Additionally, the dynamic analysis primarily focuses on theoretical modeling and simulation experiments, lacking real-world application validation data. This gap somewhat limits the practical applicability and generalizability of the research findings.

REFERENCES

[1] Cheng Haotian, Zhu Xijing, Feng Xinyu, Zhao Jing, Cai Zhanpeng, and Ding Shuaishuai.6R Industrial Robot Geometric Inverse Optimization Algorithm and Simulation Analysis [J]. Journal of Combined Machine Tools and Automation Processing Technology, 2021(4):75-79.

[2] Ma Yanyan; Zeng Taiying; Jiang Hailin. Kinematic Analysis and Trajectory Planning of Robotic Arm Based on MATLAB [J]. Packaging Engineering, 2023,44(3):187-193.

[3] Yu Zijiang; Li Cunzhi; Yu Fei; Xie Aijun; Liu Guoliang. Research on Visualized Material Feeding for Robotic Arm Based on Simulink [J]. Manufacturing Technology and Machine Tools, 2024(2):80-84.

[4] Lin Lin. Structural Design and Motion Planning of Service Robot Arm [J]. Automation Application, 2024,65(3):22-24.

[5] Wang Chun and Han Qiushi. Kinematics and Working Space of a Six-Degree-of-Freedom Serial Robotic Arm [J]. Journal of Combined Machine Tools and Automation Processing Technology, 2020(6):32-36.