

Kinematics Modelling and Experimental Analysis of a Six-Joint Manipulator

Liang Li^{1,2*}, Ying Huang¹, Xuxia Guo^{1,2}

¹ School of Mechanical Engineering, Baoji University of Arts and Sciences, Baoji 721007, China

² Shaanxi Key Laboratory of Advanced Manufacturing and Evaluation of Robot Key Components, Baoji 721016, China

Corresponding Author Email: mendezlee@bjwlxy.edu.cn

<https://doi.org/10.18280/jesa.520513>

Received: 29 April 2019

Accepted: 7 August 2019

Keywords:

Denavit and Hartenberg (D-H) parameters, manipulator, kinematics modelling, simulation

ABSTRACT

This paper establishes a kinematics model for a six degree-of-freedom (DOF) manipulator, and verifies its correctness through simulation and experiment. First, the model was set up through the D-H method. Then, the homogenous matrix transform was introduced to perform forward kinematics analysis and calculate the attitude of the end effector. Then, the manipulator position and trajectory were simulated on Robotics Toolbox for MATLAB and RoboDK. Finally, the simulated trajectory was verified through an experiment on stacking operation in the lab environment. The results show that the established kinematics model of the manipulator is correct, laying a solid theoretical basis for offline programming and calibration of manipulators.

1. INTRODUCTION

With the introduction of Industry 4.0, industrial robots are increasingly widely used to satisfy the strong market demand. Many industrial robots adopt the 6 degree-of-freedom (DOF) manipulator [1]. The kinematics modelling of the 6DOF manipulator provides the theoretical basis for trajectory planning [2-4], offline programming [5, 6] and calibration [7-9] of the manipulator.

The kinematics analysis of the manipulator is fundamental to controlling the motions and planning the trajectories of robot. The forward kinematics analysis is the most basic problem [10]. The Denavit-Hartenberg (D-H) method, which is named after its creators, is a popular way for kinematics analysis [11, 12]. This method describes the spatial relationship between two adjacent links as a 4×4 homogenous transform matrix, and then establishes the kinematics equation of the manipulator [13]. If applied to model the kinematics of the manipulator, the D-H method can convert the complex link motions into intuitive equations and matrix operations.

The MATLAB has been extensively adopted to simulate the kinematics of the manipulator [14, 15]. On the upside, the MATLAB greatly facilitates the kinematic simulation with its powerful computing ability. On the downside, this tool falls short in the simulation accuracy of manipulator motions. To solve the problem, an emerging tool, the 3D modelling software of robot, has attracted much attention from researchers. For example, the ADAMS is often used as a virtual prototype to simulate the manipulator trajectory [16, 17]. However, this software mainly targets dynamic simulation of mechanical systems, and face certain difficulties in kinematics simulation. Similarly, the RobotStudio [18, 19], an industrial robot-specific simulation software, has won high recognition for its easy operability. But this software only works for ABB industrial robots.

To make up for the defects of MATLAB simulation of manipulator kinematics, this paper establishes a kinematics model of a 6DOF manipulator and simulates the trajectory of

the manipulator on RoboDK. Finally, an experiment on the manipulator was carried out in lab environment, aiming to verify the correctness of the establish kinematics model. The research lays the basis for offline programming and calibration of manipulators.

The remainder of this paper is organized as follows: Section 2 sets up a coordinate system for desktop 6DOF manipulator, and derives a kinematics model for the manipulator through the D-H method; Section 3 identifies the position and trajectory of the manipulator using Robotics Toolbox and RoboDK, respectively; Section 4 carries out a stacking operation with the manipulator in the lab environment to verify the established model; Section 5 wraps up this paper with several conclusions.

2. D-H MODELLING

The manipulator with six rotary joints was placed on the operation platform, and a workstation (Figure 1) was established to design programs for the manipulator to grab and transfer materials.



Figure 1. Workstation for manipulator experiment

To study the relative displacement between the links of the manipulator, a coordinate system was fixed rigidly onto each link according to the D-H method. The coordinate system fixed to the base is denoted as $\{0\}$, and that fixed to link i as $\{i\}$. In coordinate system $\{0\}$, axis Z_0 of coordinate system $\{0\}$ is collinear with the kinematics axis of joint 1, and points upward; axis X_0 points to the reverse direction of axis Z_{i-1} along the common normal of axes Z_i and Z_{i-1} ; axis Y_i is formulated according to the rules of righthanded cartesian coordinate system. The coordinate systems of the links on the manipulator are displayed in Figure 2.

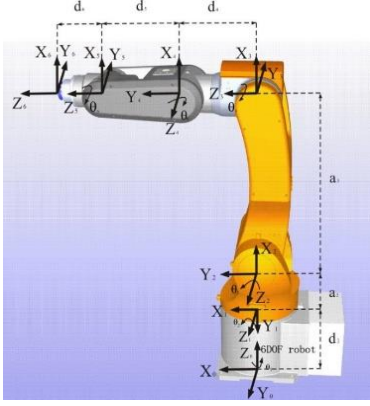


Figure 2. The coordinate systems of the links on the manipulator

In Figure 2, θ_i and d_i are the rotation angle and distance from X_0 to X_i about/along axis Z_0 , respectively; α_i and a_i are the rotation angle and distance from Z_0 to Z_i about/along axis X_0 , respectively; ...; θ_i and d_i are the rotation angle and distance from X_{i-1} to X_i about/along axis Z_{i-1} , respectively; α_i and a_i are the rotation angle and distance from Z_{i-1} to Z_i about/along axis X_{i-1} , respectively. The parameters a_i , α_i , d_i and θ_i are the D-H parameters of the links. They are used to describe each link and the relationship between the links. The D-H parameters of the 3kg 6DOF manipulator were obtained based on the coordinate systems of the links in Figure 2, and listed in Table 1 below.

Table 1. The D-H parameters of the manipulator

i	$\theta_i/^\circ$	d_i (mm)	a_i (mm)	$\alpha_i/^\circ$	Scope of work/ $^\circ$
1	$\theta_1(0)$	344	0	-90	-165~165
2	$\theta_2(-90)$	0	400	0	-180~0
3	$\theta_3(0)$	0	0	-90	-180~50
4	$\theta_4(0)$	366	0	90	-120~120
5	$\theta_5(0)$	0	0	-90	-120~120
6	$\theta_6(0)$	116	0	0	-360~360

According to the theory of homogenous transform, the link transform ${}^{i-1}T_i$ can be broken down into four sub-transforms:

- (1) Making X_{i-1} and X_i coplanar by rotating θ_i about axis Z_{i-1} .
- (2) Making X_{i-1} and X_i colinear by translating d_i along axis Z_{i-1} .
- (3) Making the origins of two coordinate systems coincidental by translating a_i along axis X_{i-1} .
- (4) Making X_{i-1} and X_i colinear by rotating α_i about axis X_{i-1} .

In this way, the transform matrix ${}^{i-1}T_i$ can be established as:

$${}^{i-1}T_i = Rot(z, \theta_i) * Trans(z, d_i) * Trans(x, a_i) * Rot(x, \alpha_i) \quad (1)$$

$$= \begin{bmatrix} c\theta_i & -s\theta_i * c\alpha_i & s\theta_i * c\alpha_i & a_i * c\theta_i \\ s\theta_i & c\theta_i * c\alpha_i & -c\theta_i * c\alpha_i & a_i * s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The transform matrix ${}^{i-1}T_i$ depends on the four parameters in Table 1. Here, a 6-dimensional set of joint vectors $q=[q_1 q_2 q_3 q_4 q_5 q_6]^T$ is introduced, with $q_i=\theta_i$ being the vector of joint i . Because all six joints are rotary, the transform matrix ${}^{i-1}T_i$ is a function about $q=[\theta_1 \theta_2 \theta_3 \theta_4 \theta_5 \theta_6]^T$.

The transform matrices of all links were multiplied to obtain the transform matrix of coordinate system $\{i\}$ relative to coordinate system $\{0\}$:

$${}^0T = {}^0T_1 * {}^1T_2 * {}^2T_3 * {}^3T_4 * {}^4T_5 * {}^5T_6 \quad (2)$$

The above transform matrix can also be described by four vectors $\{n, a, o, p\}$:

$${}^0T = [n, o, a, p] = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

where, a is the proximity vector; o is the orientation vector; n is the normal vector; p is the coordinates of the origin of coordinate system $\{i\}$ in coordinate system $\{0\}$.

Combining formulas (2) and (3), we have the kinematics equation of the 6DOF manipulator. Substituting the D-H parameters and joint vectors, we have:

$$n_x = s_6(c_4s_1 + s_4(c_1s_2s_3 - c_1c_2c_3)) + c_6(c_5(s_1s_4 - c_4(c_1s_2s_3 - c_1c_2c_3)) - s_5(c_1c_2s_3 + c_1c_3s_2))$$

$$n_y = -s_6(c_1c_4 - s_4(s_1s_2s_3 - c_2c_3s_1)) - c_6(c_5(c_1s_4 + c_4(s_1s_2s_3 - c_2c_3s_1)) + s_5(c_2s_1s_3 + c_3s_1s_2))$$

$$n_z = s_4s_6(c_2s_3 + c_3s_2) - c_6(s_5(c_2c_3 - s_2s_3) + c_4c_5(c_2s_3 + c_3s_2))$$

$$o_x = c_6(c_4s_1 + s_4(c_1s_2s_3 - c_1c_2c_3)) - s_6(c_5(s_1s_4 - c_4(c_1s_2s_3 - c_1c_2c_3)) - s_5(c_1c_2s_3 + c_1c_3s_2))$$

$$o_y = s_6(c_5(c_1s_4 + c_4(s_1s_2s_3 - c_2c_3s_1)) + s_5(c_2s_1s_3 + c_3s_1s_2)) - c_6(c_1c_4 - s_4(s_1s_2s_3 - c_2c_3s_1))$$

$$o_z = s_6(s_5(c_2c_3 - s_2s_3) + c_4c_5(c_2s_3 + c_3s_2)) + c_6s_4(c_2s_3 + c_3s_2)$$

$$a_x = -s_5(s_1s_4 - c_4(c_1s_2s_3 - c_1c_2c_3)) - c_5(c_1c_2s_3 + c_1c_3s_2)$$

$$a_y = s_5(c_1s_4 + c_4(s_1s_2s_3 - c_2c_3s_1)) - c_5(c_2s_1s_3 + c_3s_1s_2)$$

$$a_z = c_4s_5(c_2s_3 + c_3s_2) - c_5(c_2c_3 - s_2s_3)$$

$$p_x = 400c_1c_2 - 116s_5(s_1s_4 - c_4(c_1s_2s_3 - c_1c_2c_3)) - (116c_5 + 366)(c_1c_2s_3 + c_1c_3s_2)$$

$$p_y = 400c_2s_1 + 116s_5(c_1s_4 + c_4(s_1s_2s_3 - c_2c_3s_1)) - (116c_5 + 366)(c_2s_1s_3 + c_3s_1s_2)$$

$$p_z = -(366 + 116c_5)(c_2c_3 - s_2s_3) - 400s_2 + 116c_4s_5(c_2s_3 + c_3s_2) + 344$$

where, $c_i = \cos\theta_i$ and $s_i = \sin\theta_i$.

The above equation set represents the relationship between the link transform matrix 0T of the 6DOF manipulator and the set of joint vectors q and provides the basis for kinematics analysis of the manipulator. Solving the link transform matrix 0T based on joint vectors $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$ and θ_6 is called the forward kinematics analysis, while solving the joint vectors $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$ and θ_6 based on the link transform matrix 0T is called the inverse kinematics analysis.

3. KINEMATICS SIMULATIONS

Our simulations were carried out using Robotics Toolbox and RoboDK. Robotics Toolbox is a tool embed in MATLAB for modeling, trajectory planning and control of manipulators. RoboDK is a simulation software to create a virtual working scene of robot and simulate the actual applications. The software supports multiple brands of manipulators and self-developed manipulator models. RoboDK simulations can approximate the actual working conditions of manipulators. Note that the D-H parameters are required for manipulator simulation by both MATLAB and RoboDK. The simulation can be completed using the relevant functions and commands. Meanwhile, the kinematics model of the manipulator is written into the underlying program, and remains invisible to users.

3.1 MATLAB simulation of manipulator position

The forward kinematics simulation of the manipulator aims to analyze the attitude change of the end effector according to the link parameters and the variation of each joint [20]. Here, the manipulator position was simulated with Robotics Toolbox based on the established kinematics model. The simulation was carried out in the following steps:

Step 1. Based on the D-H parameters, the links were created on Robotics Toolbox, forming the manipulator, and a graph of the manipulator was produced.

Step 2. The set of joint vectors $q=[q_1 \ q_2 \ q_3 \ q_4 \ q_5 \ q_6]^T$ was adjusted randomly on the drive interface of the manipulator. Five different sets of joint vectors were selected: q^1, q^2, q^3, q^4 and q^5 . Then, the corresponding attitudes $(x, y, z, \varphi, \theta, \psi)$ of the end effector were recorded.

Step 3. The five randomly selected sets of joint vectors were inputted to MATLAB, and the link transform matrix and end effector attitude of each set were solved using the user-defined program for the kinematics equation.

Through the above steps, the drive interface of the manipulator was obtained as Figure 3. In the interface, the link configuration of the manipulator is displayed on the right, the attitude of the end effector $(x, y, z, \varphi, \theta, \psi)$ is shown on the upper left, and the adjustment region of joint vectors q_1, q_2, q_3, q_4, q_5 and q_6 is provided on the lower left. Because all joints are rotary, the joint vector q_i equals the joint angle θ_i . Depending on the inputs in the adjustment region, the manipulator moved to the corresponding attitude, and the attitude parameters were displayed on the upper left of the interface.

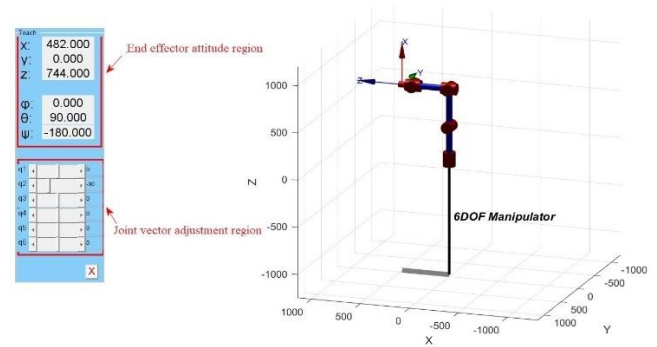


Figure 3. The drive interface of Robotics Toolbox

The six joint vectors in Figure 3 were adjusted, creating five sets of joint vectors (Table 2). The five sets of joint vectors were substituted into the program in Appendix 1. The results were collected and compared with the attitudes solved by MATLAB (Table 3).

As shown in Table 3, when five sets of joint vectors were randomly selected, the end effector attitudes obtained by Robotics Toolbox based on the D-H parameters were the same with those obtained by the kinematics equation, which was established based on the D-H method and homogenous matrix transform. Thus, the established kinematics model agrees well with the underlying model of Robotics Toolbox.

Table 2. The five randomly selected sets of joint vectors (unit: °)

q	$q_1(\theta_1)$	$q_2(\theta_2)$	$q_3(\theta_3)$	$q_4(\theta_4)$	$q_5(\theta_5)$	$q_6(\theta_6)$
q^1	-70	-20	-31.55	44.29	-17.45	26.0
q^2	-46.38	-42.44	20.85	-28.39	34.03	-18.07
q^3	8.39	-91.56	-129.44	104.37	-80.68	23.25
q^4	32.31	-53.47	-6.83	7.51	-32.65	23.25
q^5	159.25	-105.72	30.97	-110.56	104.69	-99.57

Table 3. Comparison between the end effector attitudes obtained by Robotics Toolbox and those obtained by MATLAB

Group	Compare	x/mm	y/mm	z/mm	$\varphi/^\circ$	$\theta/^\circ$	$\psi/^\circ$
q^1	Robotics Toolbox	238.703	-726.852	203.905	-83.376	115.157	-142.830
	Program calculation	238.703	-726.852	203.905	-83.376	115.157	-142.830
q^2	Robotics Toolbox	261.979	-319.654	163.202	-166.252	162.130	16.691
	Program calculation	261.979	-319.654	163.202	-166.252	162.130	16.691
q^3	Robotics Toolbox	-223.123	-144.995	1052.89	-76.910	73.567	64.748
	Program calculation	-223.123	-144.995	1052.89	-76.910	73.567	64.748
q^4	Robotics Toolbox	571.985	352.055	489.584	28.262	87.281	-150.224
	Program calculation	571.985	352.055	489.584	28.262	87.281	-150.224
q^5	Robotics Toolbox	-174.794	178.572	678.525	59.520	66.768	-20.137
	Program calculation	-174.794	178.572	678.525	59.520	66.768	-20.137

3.2 MATLAB simulation of manipulator trajectory

The first two segments of manipulator trajectory were planned and simulated. The set of joint vectors for a random initial configuration was set to $q^s = [-29.76, -26.987, -20.962, -2.252, 49.351, -30.017]^T$ (unit: $^\circ$). Through calculation, the initial attitude $(x_s, y_s, z_s, \varphi_s, \theta_s, \psi_s)$ of the end effector could be obtained as $(541.286, -313.477, 164.455, -157.958, 177.827, 20.296)$. The set of joint vectors for target configuration 1 was $q^{t1} = [-29.761, -46.084, -27.357, -1.773, 74.754, -31.01]^T$ (unit: $^\circ$), and the corresponding attitude $(x_{t1}, y_{t1}, z_{t1}, \varphi_{t1}, \theta_{t1}, \psi_{t1})$ of the end effector was $(541.383, -313.553, 411.914, -157.112, 177.848, 21.153)$; The set of joint vectors for target configuration 2 was $q^{t2} = [0, -90, 0, 0, 0, 0]^T$ (unit: $^\circ$), and the corresponding attitude $(x_{t2}, y_{t2}, z_{t2}, \varphi_{t2}, \theta_{t2}, \psi_{t2})$ of the end effector was $(482, 0, 744, 0, 90, -180)$. The relevant functions of Robotics Toolbox were adopted to plan the trajectory for the manipulator. The planned results are shown in Figure 4. The trajectory of the end effector and the angular displacement curve of each joint are presented in Figure 5.

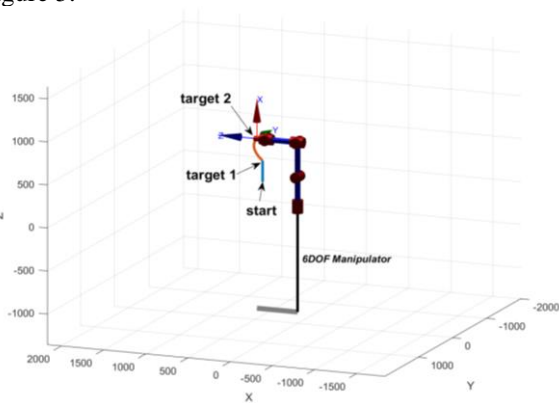


Figure 4. The manipulator trajectory simulated by MATLAB

trajectory diagram

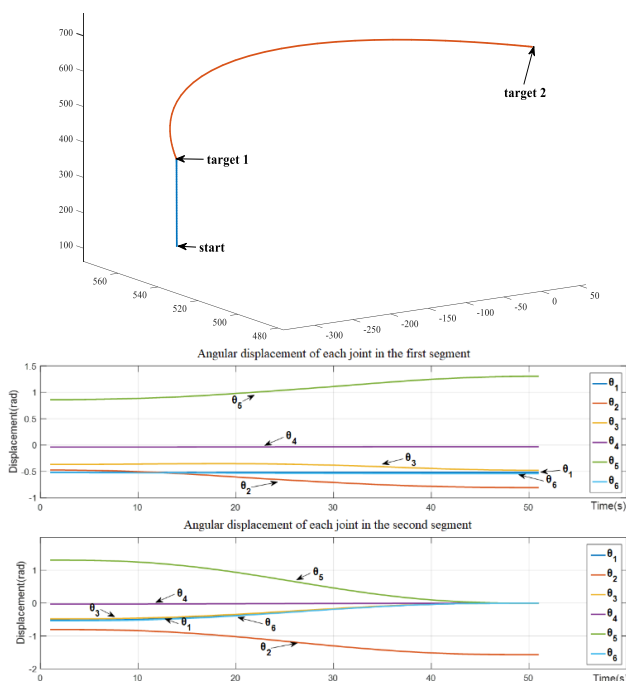


Figure 5. The trajectory of the end effector and the angular displacement curve of each joint

As shown in Figures 4 and 5, the first segment was a linear trajectory and the second segment, a curved trajectory; the manipulator ended up at the original position $q^{t2} = [0, -90, 0, 0, 0, 0]^T$ (unit: $^\circ$); all joints moved continuously and smoothly, the manipulator movement was stable, and the two segments were connected smoothly.

3.3 RoboDK simulation of manipulator trajectory

MATLAB integrates many basic algorithms of the manipulator. This greatly facilitates the research and simulation of traditional joint manipulator and mobile manipulator. However, the MATLAB simulation cannot reflect the collisions between links, for it overlooks the mechanical structure of the manipulator. To solve the defect, this paper simulates the first two segments of the trajectory in RoboDK again, using a real-scale 3D model of our manipulator.

The 3D structural map of the 6DOF manipulator were imported to RoboDK, and its D-K parameters were configured. Then, the sets of joint vectors at the initial configuration, target configuration 1 and target configuration 2 were demonstrated on the teach pendant screen, and named as q_start , $q_target1$ and $q_target2$, respectively. Through programming, the simulated manipulator trajectory is shown in Figure 6.

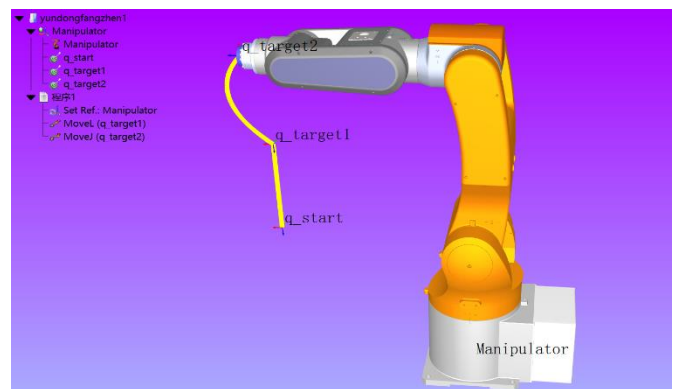


Figure 6. The manipulator trajectory simulated by RoboDK

It can be seen that the manipulator trajectory simulated by RoboDK is the same with that simulated by MATLAB, and the q_start , $q_target1$ and $q_target2$ are consistent with the attitude computed in Subsection 4.2. In addition, all joints moved smoothly without any collision throughout the movement.

4. EXPERIMENTAL ANALYSIS

This section plans a simple trajectory of the manipulator for stacking operation in the lab environment, and simulates the planned stacking trajectory on RoboDK. The simulation results were compared with the experimental data.

4.1 Trajectory planning in lab environment

The stacking operation aims to pick up a block from the right part of the workbench surface and place it on the left part of the workbench surface. To plan a suitable trajectory for the manipulator, it is necessary to identify the sequence of attitude nodes of the end effector, i.e. the sequence of $ap_0 \sim ap_6$ (Figure 7).

According to the sequence of attitude nodes, the stacking operation was described as a series of motions and actions. The joint angles of the manipulator were read from the controller and recorded.

As shown in Table 4, the joint angles at the original position read on the controller were different from those in MATLAB. Through investigation, it is learned that the joint angle of the

second link is -90° in MATLAB when the manipulator is at the original position. In actual operation, the controller will display 0° although the joint angle of the second link is inputted as -90° . Therefore, if the joint angle of the second link read from the controller is θ_2 , the corresponding value should be $-\theta_2-90^\circ$ in the MATLAB.

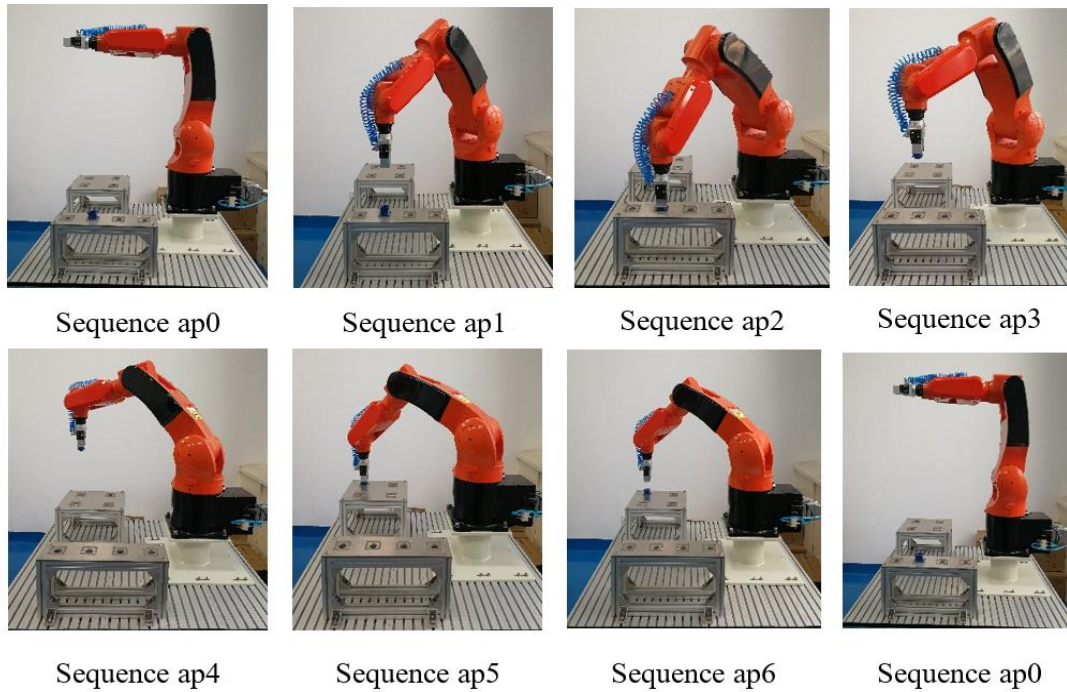


Figure 7. The planned trajectory for the manipulator

Table 4. The process from pickup to placement

Nodes	Joint angles ($\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$) / $^\circ$	Target
ap0	(0,0,0,0,0,0)	Original position
ap1	(36.802, -35.09, 3.191, 0.453, 53.697, 35.855)	Approaching the block
ap2	(36.801, -52.229, 4.455, 0.649, 35.31, 35.599)	Holding the block
ap3	(36.801, -30.057, -0.032, 0.432, 61.996, 35.92)	Lifting up the block
ap4	(-29.761, -43.916, -27.357, -1.773, 74.754, -31.01)	Approaching the destination
ap5	(-29.76, -63.013, -20.962, -2.252, 49.315, -30.017)	Placing the block
ap6	(-29.76, -57.656, -20.128, -2.116, 53.9, -30.232)	Raising the arm
ap0	(0,0,0,0,0,0)	Returning to the original position

4.2 RoboDK simulation

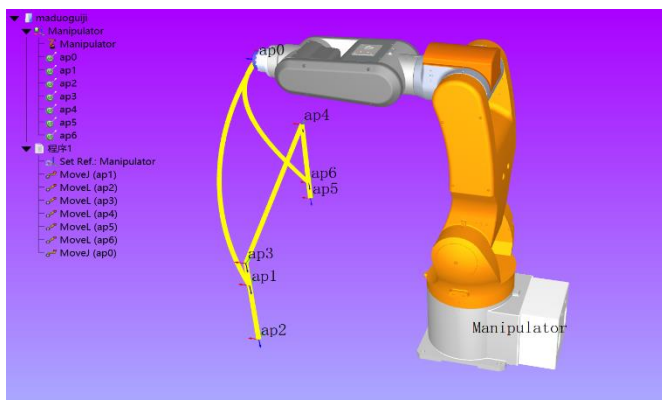


Figure 8. The stacking trajectory simulated by RoboDK

The joint angles of the above sequence of attitude nodes were inputted to RoboDK. The stacking trajectory was

simulated again (Figure 8), and the attitude was recorded at each node.

4.3 Comparative analysis

The attitudes at each node read from the controller were compared with those simulated by RoboDK (Table 5). The attitudes calculated by MATLAB based on the D-H method and homogenous matrix transform are also provided in Table 5. Each attitude is expressed as attitudes ($x, y, z, \varphi, \theta, \psi$); the coordinates (x, y, z) are given in mm; each posture (φ, θ, ψ) is given in $^\circ$.

As shown in Table 5, under the same inputs of joint angles, the attitudes simulated by RoboDK agree well with those computed by MATLAB. The attitudes read from the controller only had a slight difference from these two types of results. This means the manipulator motions are consistent with its kinematics model. Thus, the established kinematics model is correct, laying the basis for further research on manipulators.

Table 5. Comparison between attitudes

	Attitudes read from the controller	Attitudes simulated by RoboDK	Attitudes calculated by MATLAB
ap0	(482.02, -0.01, 743.96, 0, 90.01, -179.99)	(482.0, 0, 744.0, 0, 90.0, -180.0)	(482.0, 0, 744.0, 0, 90.0, -180.0)
ap1	(410.52, 308.05, 328.67, -153.65, 177.99, 25.69)	(410.52, 308.05, 328.63, -153.66, 177.99, 25.67)	(410.52, 308.05, 328.63, -153.66, 177.99, 25.67)
ap2	(410.45, 308.02, 167.21, -153.86, 177.97, 25.48)	(410.46, 308.02, 167.27, -153.86, 177.97, 25.47)	(410.46, 308.02, 167.27, -153.86, 177.97, 25.47)
ap3	(410.42, 308.01, 391.14, -153.88, 177.94, 25.45)	(410.42, 308.01, 391.15, -153.89, 177.94, 25.44)	(410.42, 308.01, 391.15, -153.89, 177.94, 25.44)
ap4	(541.38, -313.55, 411.91, -157.12, 17.85, 21.14)	(541.38, -313.55, 411.91, -157.11, 17.85, 21.15)	(541.38, -313.55, 411.91, -157.11, 17.85, 21.15)
ap5	(541.28, -313.48, 164.46, -157.96, 177.83, 20.30)	(541.29, -313.48, 164.46, -157.96, 177.83, 20.30)	(541.29, -313.48, 164.46, -157.96, 177.83, 20.30)
ap6	(541.15, -313.41, 219.14, -159.25, 177.79, 19.01)	(541.15, -313.41, 219.14, -159.25, 177.79, 19.01)	(541.15, -313.41, 219.14, -159.25, 177.79, 19.01)

5. CONCLUSIONS

This paper carries out modelling, simulation and experiment on a 3 kg removable 6DOF manipulator.

First, a kinematics model was set up for the 6DOF manipulator by the D-H method. A program was prepared in MATLAB according to the kinematics equation. Using this program, the transform matrix 0T_i of coordinate system $\{i\}$ relative to coordinate system $\{0\}$ and the attitude of the end effector ($x, y, z, \varphi, \theta, \psi$) can be derived from the 6 joint angles.

Second, the manipulator position and trajectory were simulated on Robotics Toolbox, and verified through a simulation on RoboDK. The results show that the established kinematics model agrees with the underlying model of RoboDK, and the manipulator movement was smooth and free of collisions.

Third, a simple stacking operation was performed in lab environment, and a sequence of attitude nodes was set up for the operation. The attitudes at each node were read from the controller, simulated by RoboDK and calculated by MATLAB based on the kinematics equation. The comparison between these results demonstrates the correctness of the established kinematics model.

The above work confirms that the established kinematics model is correct, and can serve as a reliable theoretical model for offline programming for the manipulator. The research results lay the basis for developing calibration techniques for manipulators.

ACKNOWLEDGMENT

This paper is supported by Key Research and Development Plan of Shaanxi Province, China (Grant No.: 2019GY-091); General Innovative Research Projects for Graduate Students, Baoji University of Arts and Sciences (Grant No.: YJSCX18YB37); Key Innovative Research Projects for Graduate Students, Baoji University of Arts and Sciences (Grant No.: YJSCX18ZD05).

REFERENCES

[1] Ranjbaran, F., Angeles, J., González-Palacios, M.A., Patel, R.V. (1995). The mechanical design of a seven-axes manipulator with kinematic isotropy. *Journal of*

Intelligent and Robotic Systems, 14(1): 21-41. <https://doi.org/10.1007/BF01254006>

[2] Wei, K., Ren, B. (2018). A method on dynamic path planning for robotic manipulator autonomous obstacle avoidance based on an improved RRT algorithm. *Sensors*, 18(2): 571. <https://doi.org/10.3390/s18020571>

[3] Chen, X.F. (2018). Research on the trajectory planning algorithm of palletizing robots. *Control Engineering of China*, 25(5): 925-929. <https://doi.org/10.14107/j.cnki.kzgc.14D4.0033>

[4] Fang, J., Song, Y., Zhu, M.F., Zhu, D.Q., Liu, Y.B., Jiao, J. (2018). Time-optimal trajectory planning for palletizing robots. *Control Engineering of China*, 25(1): 93-99. <https://doi.org/10.14107/j.cnki.kzgc.150879>

[5] Gołda, G., Kampa, A. (2017). Manipulation and handling processes off-line programming and optimization with use of K-Roset. *IOP Conference Series: Materials Science and Engineering*, 227(1): 1-6. <https://doi.org/10.1088/1757-899X/227/1/012050>

[6] Ni, W.B., Yang, W.C. (2017). The research on a method of off-line motion planning for dual-arm robot. *Manufacturing Technology & Machine Tool*, (4): 25-28. <https://doi.org/10.19287/j.cnki.1005-2402.2017.04.001>

[7] Joubair, A., Bonev, I.A. (2015). Kinematic calibration of a six-axis serial robot using distance and sphere constraints. *The International Journal of Advanced Manufacturing Technology*, 77(1-4): 515-523. <https://doi.org/10.1007/s00170-014-6448-5>

[8] Wu, L., Ren, H. (2016). Finding the kinematic base frame of a robot by hand-eye calibration using 3D position data. *IEEE Transactions on Automation Science and Engineering*, 14(1): 314-324. <https://doi.org/10.1109/TASE.2016.2517674>

[9] Zhou, X., Huang, S.F., Zhu, Z.H. (2019). TCP calibration model research and algorithm improvement of six joint industrial robot. *Journal of Mechanical Engineering*, 55(11): 186-196. <https://doi.org/10.3901/JME.2019.11.186>

[10] Liu, J.J., Peng, J.Q., Liu, X.H. (2015). Forward kinematics analysis and simulation for a series parallel hybrid 7-DOF humanoid mechanical arm based on MATLAB. *Journal of Mechanical Transmission*, 39(7): 59-62, 66. <https://doi.org/10.16578/j.issn.1004.2539.2015.07.014>

[11] Hartenberg, R.S., Denavit, J. (1955). A kinematic notation for lower pair mechanisms based on matrices. *Journal of applied mechanics*, 77(2): 215-221.

- [12] Zhang, X., Zheng, Z.L., Qi, Y. (2016). Parameter identification and calibration of D-H model for 6-DOF serial robots. *Robot*, 38(3): 360-370. <https://doi.org/10.13973/j.cnki.robot.2016.0360>
- [13] Shimizu, M., Kakuya, H., Yoon, W.K., Kitagaki, K., Kosuge, K. (2008). Analytical inverse kinematic computation for 7-DOF redundant manipulators with joint limits and its application to redundancy resolution. *IEEE Transactions on Robotics*, 24(5): 1131-1142. <https://doi.org/10.1109/TRO.2008.2003266>
- [14] Huang, Y., Wang, J., Tham, D.M. (2017). Kinematics modelling and adams matlab/simulink co-simulation for automated aerobridge docking process. In 2017 3rd International Conference on Control, Automation and Robotics (ICCAR), 304-309. <https://doi.org/10.1109/ICCAR.2017.7942708>
- [15] Li, X.H., Sun, Q., Zhang, L.G., Zhang, J. (2018). Research and simulation of the kinematics of dual-arm 6R service robot. *Journal of Mechanical Transmission*, 42(5): 129-134. <https://doi.org/10.16578/j.issn.1004.2539.2018.05.026>
- [16] Wen, G., Xu, L., He, F., Zhang, X. (2009). Kinematics simulation to manipulator of welding robot based on ADAMS. In 2009 International Workshop on Intelligent Systems and Applications, 1-4. <https://doi.org/10.1109/IWISA.2009.5072932>
- [17] Zhu, H.J. (2019). Kinematic analysis of 5-DOF welding robot based on screw theory. *Machine Building & Automation*, 48(1): 150-152. <https://doi.org/10.19344/j.cnki.issn1671-5276.2019.01.039>
- [18] Zhang, G.Q., Spaak, A., Martinez, C., Lasko, D.T., Zhang, B., Fuhlbrigge, T.A. (2016). Robotic additive manufacturing process simulation-towards design and analysis with building parameter in consideration. In 2016 IEEE International Conference on Automation Science and Engineering (CASE), 609-613. <https://doi.org/10.1109/COASE.2016.7743457>
- [19] Andrzej, B., Krzysztof, K., Dariusz, S., Magdalena, M., Jacek, N. (2017). Robot-operated quality control station based on the UTT method. *Open Engineering*, 7(1): 37-42. <https://doi.org/10.1515/eng-2017-0008>
- [20] Fu, Y.Y., Ge, A.P. (2013). Kinematics simulation study based on MATLAB on industrial robot. *Mechanical Engineering & Automation*, (3): 40-42. <https://doi.org/10.3969/j.issn.1672-6413.2013.03.018>