Analysis of Grasping Target by Simulated Six-Axis Robot

Guanxiang Jing
School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing, China
20222008@bjtu.edu.cn

Abstract. A six-axis robot is a type of robotic manipulator with six degrees of freedom (DOF) that enable it to move and rotate in six different directions. Because of its flexibility and high-precision operation, it is widely used in industrial production lines, semiconductor manufacturing, medical surgery and other fields. In the future, with the development of artificial intelligence, sensor technology and other fields, six-axis robots will achieve more intelligent and autonomous operation, and will also be more widely used in industry, medical treatment, service and other fields. In this paper, the important applications of six-axis robots in industry and other fields are introduced by introducing the relevant applications of artificial intelligence and robots. Then the DH representation method and inverse solution method are introduced, and a six-axis robot is modeled and the relevant data are calculated in MATLAB software using the DH representation method and inverse solution method. Finally, the trajectory of the robot gripper was simulated and the simulation results were obtained, which preliminarily verified the feasibility and reliability of the six-axis robot grasping objects.

Keywords: Six axis robots, DH method, inverse solution.

1. Introduction

Artificial intelligence and robot technology have become one of the most significant topics in the scientific and technological domain. As an integral part of robot technology, the manipulator has found extensive usage in manufacturing, medical, aerospace, and other fields.

Initially, the mechanical arm was primarily used in industrial manufacturing. However, with the continual progress of technology, the application scope of the manipulator has broadened, including fields such as medical, military, and aerospace. In recent years, the application of robots has extended to logistics distribution, smart home automation, and other domains, owing to the rapid advancements in artificial intelligence technology.

In the current era of Industry 4.0, the high volume of industrial products necessitates labor-intensive activities such as order sorting and warehousing. Many large companies have assigned such tasks to robots to reduce the workload of human operators and enhance work efficiency [1]. Similarly, in the field of distribution, robots use navigation as one of their basic functions to avoid other robots and humans and reach their intended destination [2].

The medical field has also seen widespread adoption of robots to assist attending doctors during surgery [3]. In several surgical operations in Europe and the United States, it is common to see attending doctors operating the robotic arm to perform surgery [4]. Regardless of the tasks that robots replace or assist humans with, these behaviors highlight an essential idea-human-computer interaction and communication with robots are the fundamental prerequisites for robotic work [5].

The purpose of this article is to evaluate the performance of human-computer interaction by having the six-axis robot complete a simple task. However, in advanced industrial applications such as machining, the absolute positioning accuracy of the six-axis robot is crucial [6]. Moreover, as intelligent manufacturing continues to develop, manual polishing is being replaced by high-precision six-axis polishing robots because of the low production efficiency and poor production environment of manual polishing [7].

To improve the accuracy of the six-axis robot, this article discusses various methods, including the DH method and the inverse solution method. The DH method, proposed by Denavit and Hartenberg in the 1950s, is a commonly used approach in manipulator kinematics. It involves converting the kinematics model of the manipulator into DH parameters, which include four parameters (a, alpha, d, and theta) that completely describe the position and attitude information of the manipulator. Due to its characteristics of parametric simplification, clear model, and simple calculation, the DH method is widely used in kinematics modeling of manipulators.

The inverse kinematics method is essential in determining the joint angles and positions required for the robot's end effector to reach a desired target location and orientation [8]. The inverse kinematics method involves calculating the joint motion of the robot manipulator based on the position and orientation information of the end effector, enabling precise control of the robot. This method is useful for tasks such as trajectory planning and path planning of the robot manipulator.

Based on the above methods, relevant data are obtained and the feasibility and reliability of the six-axis robot to implement the grasping task are preliminarily confirmed.

2. Methods

2.1. DH Table Represent

The key question for a robot to grasp objects is how a specific position and orientation of the tool can be realized [9]. Generally, a coordinate frame is attached to every link of the robot. The coordinate of each frame relative to the previous one can be then calculated. The composite of coordinates relating the tool frame to the base frame provides the position and orientation of the tool. To achieve the task, DH Representation is used to help build the robot model. The establishment of DH representation method is basically subdivided into following steps.

Assignment of joint numbers: Number the six joints of the robot from the base to the tool's yaw, pitch, and roll as 1 to 6, respectively.

Establishment of coordinate frames: Establish a coordinate frame for each joint. Begin with L0 at the robot's base and end with Ln at the tool tip.

Alignment of the z-axes: Line up the z-axis of frame L0 with joint 1's axis. For each joint k (from 1 to 5), align the z-axis of frame L^k with the axis of joint k+1.

Determination of the frame origins: Find the origin of each frame. The intersection point of the z-axes of frames L^k and L^{k-1} is the origin of frame L^k. If the z-axes do not meet, use the intersection point of their common normal.

Setting of the x-axes: Orient the x-axis of each frame. The x-axis of frame L^k is perpendicular to the z-axes of frames L^k and L^{k-1}. If the z-axes are parallel, point the x-axis away from the z-axis of frame L^{k-1}.

Setting of the y-axes: Set the y-axis of each frame. The y-axis should complete a right-hand coordinate system, forming an orthonormal frame.

Determination of the origin of the last frame: Establish the origin of frame Ln at the tool tip. Align the z-axis of frame Ln with the approach vector, the y-axis with the sliding vector, and the x-axis with the normal vector.

Calculation of theta: Compute the joint variable theta for each joint. It can be measured by using the rotation angle from xk-1 to xk to calculate theta for joint k.

Calculation of d: Find the link offset d for each joint. Measure the distance between the origin of frame L^{k-1} and the point where the x^k and z^{k-1} axes intersect along z^{k-1}.

Calculation of a: Calculate the link length a for each joint. Measure the distance from the intersection point of the x^k and z^{k-1} axes to the origin of frame Lk along x^k.

Calculation of alpha: Determine the link twist alpha for each joint. It can be measured by using the rotation angle from z^{k-1} to z^k to calsulate alpha for joint k.

Creation of the DH table: Record the DH parameters (theta, d, a, alpha) in a table to use for the robot's kinematic calculations.

Through the above steps, the coordinate axis of each joint can be drawn according to the robot's model drawing to establish the DH table. Since this paper is aimed at the six-axis robot, the DH table will also select the six-axis robot model to build. Today's six-axis robot manufacturers produce a variety of six-axis robots. In this paper, RS005N (Figure 1) produced by Kawasaki will be selected as the research object to establish various joint coordinate systems.

Figure 1. RS005N six-axis robot produced by Kawasaki [10]

The coordinate axis for each joint of the robot can be established based on the model shown in Figure 1, which enables the transformation of coordinates from the robot's base mark to each joint.

The DH table is a crucial tool for solving the forward kinematics of a robot. It provides a clear representation of the joint relationships. Table 1 represents the DH table for the robot mentioned above.

Table 1. DH table of RS005N

Joint	$\alpha_i(°)$	$a_i(mm)$	$d_i(mm)$	$\theta_i(°)$
1	-90	105	295	0
2	0	280	0	-90
3	-90	80	0	0
4	90	0	310	0
5	-90	0	0	0
6	0	0	78	0

2.2. Forward Solution

The DH table provides the necessary information to obtain the transformation matrix between adjacent joints. This transformation matrix can be used to determine the position and orientation of the robot's end effector based on the angles of each joint.

$$
{}^{i-1}_{i}A = R(z_{i-1},\theta_i)Trans(0,0,d_i)Trans(a_i,0,0)R(x_i,\alpha_i) = R(z_{i-1},\theta_i)Trans(a_i,0,d_i)R(x_i,\alpha_i)
$$

$$
= \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
= \begin{bmatrix} \cos\theta_i & -\cos\alpha_i\sin\theta_i & \sin\alpha_i\sin\theta_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\alpha_i\cos\theta_i & -\sin\alpha_i\cos\theta_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}
$$

The data in DH table is brought into the transfer matrix to get the relationship matrix between adjacent joints.

$$ {}_1^0A=\begin{bmatrix} c_1 & 0 & -s_1 & a_1c_1 \\ s_1 & 0 & c_1 & a_1s_1 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ {}_2^1A=\begin{bmatrix} s_2 & c_2 & 0 & a_2c_2 \\ c_2 & -s_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ {}_3^2A=\begin{bmatrix} c_3 & 0 & -s_3 & a_3c_3 \\ s_3 & 0 & c_3 & a_3s_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}_4^3A=\begin{bmatrix} c_4 & 0 & s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ {}_5^4A=\begin{bmatrix} c_5 & 0 & -s_5 & 0 \\ s_5 & 0 & c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ {}_6^5A=\begin{bmatrix} c_6 & -s_5 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

Where $c_i = \cos\theta_i,\ s_i = \sin\theta_i$

The transfer matrix T can be obtained by multiplying the transfer matrices of adjacent joints.

$$ T_6^0 = A_1^0A_2^1A_3^2A_4^3A_5^4A_6^5,\ T_6^0 = \begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3} $$

A transformation matrix is a 4x4 matrix that describes the position and orientation of a point or frame in relation to another point or frame. The upper-left 3x3 sub-matrix of the matrix represents the rotation or orientation of the transformed frame relative to the original frame. The rightmost column of the matrix, which is the position or translation vector, represents the position of the transformed frame relative to the original frame. These two components together define the position and orientation of the transformed frame with respect to the original frame.

In robotics, the transformation matrix is commonly used to represent the position and orientation of the end effector relative to the robot base frame. The upper-left 3x3 sub-matrix of the transformation matrix represents the orientation of the end effector relative to the base frame, while the rightmost column represents the position of the end effector relative to the base frame. The position vector is typically represented in the coordinate system of the end effector, which is known as the tool frame, and must be transformed into the base frame using the homogeneous transformation matrix.

2.3. Inverse Solution

Once the transfer matrix is obtained, the next step is to multiply the inverse of the first transfer matrix to both sides of the equation to obtain the matrix. This process can be continued using a similar method for the remaining joints. To simplify the calculations, an easier to calculate T matrix is used in this part.

$$ ({}_1^0A)^{-1}T_6^0 = A_2^1A_3^2A_4^3A_5^4A_6^5 \tag{4} $$

$$ ({}_1^0A)^{-1}T_6^0 = \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -s_1 & c_6 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ = \begin{bmatrix} n_xc_1+n_ys_1 & o_xc_1+o_ys_1 & a_xc_1+a_ys_1 & p_xc_1+p_ys_1 \\ -n_z & -o_z & -a_y & -p_z \\ n_yc_1-n_xs_1 & o_yc_1-o_ys_1 & a_yc_1-a_xs_1 & p_yc_1-p_xs_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ = \begin{bmatrix} c_2(c_4c_5c_6-s_4s_6)-s_2s_5s_6 & -c_2(c_4c_5c_6-s_4s_6)+s_2s_5s_6 & c_2c_4s_5+s_2c_5 & s_2d_3 \\ s_2(c_4c_5s_6-s_4s_6)+c_2s_5c_6 & -s_2(c_4c_5s_6+s_4s_6)-c_2s_5s_6 & s_2c_4s_5-c_2c_5 & -c_2d_3 \\ s_4c_5c_6-s_4s_6 & -s_2c_5s_6+c_4c_5 & s_4s_5 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5} $$

Where $c_i = \cos\theta_i,\ s_i = \sin\theta_i$

Then corresponding equality of element (3,4) is calculated:

$$ p_yc_1 - p_xs_1 = d_2 \tag{6} $$

In the equation, assuming that $p_x = r\cos\phi$, $p_y = r\sin\phi$,

$$r = +\sqrt{p_x^2 + p_y^2}, \phi = tan^{-1}\left(\frac{p_x}{p_y}\right) \tag{7}$$

After replacement,

$$\begin{cases} sin\,\varphi\,cos\,\theta_1 - cos\,\varphi\,sin\,\theta_1 = sin(\varphi - \theta_1) = \frac{d_2}{r} \\ \cos((\varphi - \theta_1) = \pm\sqrt{1 - sin^2(\varphi - \theta_1)} = \pm\sqrt{1 - \frac{d_2}{r}} \end{cases} \tag{8}$$

and

$$\theta_1 = tan^{-1}\left(\frac{p_x}{p_y}\right) - tan^{-1}\frac{d_2}{\pm\sqrt{r^2 - d_2^2}} \tag{9}$$

can be obtained.
To get θ2, it can be easily obtained from the matrix:

$$s_2 d_3 = c_1 p_x + s_1 p_y \tag{10}$$

$$c_2 d_3 = p_z \tag{11}$$

Solving the equations,

$$\theta_2 = tan^{-1}\frac{c_1 p_x + s_1 p_y}{p_z} \tag{12}$$

is obtained.
Similar methods can also be used when solving θ_2,

$$(A_2^1)^{-1}(A_1^0)^{-1}T_6^0 = A_3^2 A_4^3 A_5^4 A_6^5 = T_6^2 \tag{13}$$

This equation is obtained:

$$\begin{bmatrix} f_{21}(n) & f_{21}(o) & f_{21}(a) & f_{21}(p) \\ f_{22}(n) & f_{22}(o) & f_{22}(a) & f_{22}(p) \\ f_{23}(n) & f_{23}(o) & f_{23}(a) & f_{23}(p) \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_4c_5s_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & s_4s_5 & 0 \\ s_4c_5s_6 + c_4s_6 & c_4c_5s_6 - s_4s_6 & s_4s_5 & 0 \\ -s_4s_6 & s_5s_6 & c_5 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

Solving the equations,

$$d_3 = s_2(c_1 p_x + s_1 p_y) + c_c p_z \tag{15}$$

is obtained.
Using the similar method above,

$$\begin{cases} \theta_{4,1} = tan^{-1}\frac{-s_1a_x + c_1a_y}{c_2(c_1a_x + s_1a_y) - s_2a_z} \\ \theta_{4,2} = tan^{-1}\frac{-(-s_1a_x + c_1a_y)}{-c_2(c_1a_x + s_1a_y) - s_2a_z} \end{cases} \tag{16}$$

is obtained, and
$\theta_{4,2}=\theta_{4,1}+180^o$,

$$\theta_5 = tan^{-1}\frac{c_4[c_2(c_1a_x + s_1a_y) - s_2a_z] + s_4(-s_1a_x + c_1a_y)}{s_2(c_1a_x + s_1a_y) + c_2a_z} \tag{17}$$

$$c_6 = -s_4[c_2(c_1o_x + s_1o_y) - s_2o_z] + c_4(-s_1o_x + c_1o_y) \tag{18}$$

$$s_6 = -c_5\{c_4[c_2(c_1o_x + s_1o_y) - s_2o_z] + s_4(-s_1o_x + c_1o_y)\} + s_5[s_2(c_1o_x + s_1o_y) + c_2o_z] \tag{19}$$

$$\theta_6 = tan^{-1}\frac{s_6}{c_6} \tag{20}$$

are obtained.

The aforementioned paragraphs describe the conventional inverse solution process. It is apparent that this process involves an enormous and intricate calculation, which is nearly impossible to perform manually. Nevertheless, as per the formula mentioned above, a six-axis robot generally has eight sets of solutions when the end reaches its desired position. In such instances, the robot's posture is typical. However, when the robot reaches a singularity position, the number of solutions available is often not eight. Hence, the number of solutions computed by the computer can determine whether the robot has reached a singularity position to avoid any interruptions in its operation.

2.4. Jacobian Matrix

The Jacobian matrix is a mathematical matrix that is constructed using the first partial derivatives of a particular function, typically in vector calculus. The determinant of the Jacobian matrix is known as the Jacobian determinant. The importance of the Jacobian matrix lies in the fact that it reflects the differentiability of a given equation and provides an optimal linear approximation of the point being considered. Therefore, the Jacobian matrix can be thought of as similar to the derivative of a multivariate function [11]. The robot.jacob0() function in MATLAB can directly calculate the Jacobian matrix, and it has been used in this project.

3. Results

The robot can be modeled through the DH table and the Robotics toolbox in MATLAB. The model is shown in Figure 2.

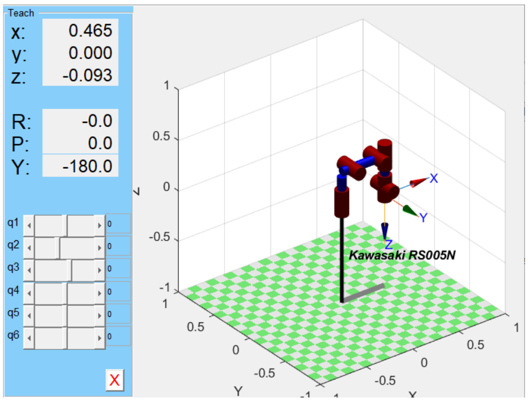

Figure 2. Successfully built robot models

Since MATLAB needs to bring in specific data during calculation, it may be assumed that the target is in the tool coordinate system (-0.3, -0.3, 0.1). Due to the characteristics of trigonometric function, generally, the six-axis robot can get eight sets of solutions. The calculation results are shown in Table 2.

Table 2. Angle of each joint

	Joint1(°)	Joint2(°)	Joint3(°)	Joint4(°)	Joint5(°)	Joint6(°)
G1	-135	-60.4468	11.56091	180	131.1141	113.383
G2	-315	146.0785	11.56091	0	22.36057	130.5394
G3	-135	33.92149	-162.62	180	51.30116	111.351
G4	-315	240.4468	-162.62	0	102.1735	92.54611
G5	-135	-60.4468	11.56091	360	-131.114	293.383
G6	-315	146.0785	11.56091	180	-22.3606	310.5394
G7	-135	33.92149	-162.62	360	-51.3012	291.351
G8	-315	240.4468	-162.62	180	-102.173	272.5461

Jacobian matrix can be calculated by the function in toolbox. Table 3 and Table 4 are the parameters for calculating the Jacobian matrix. Figure 2 and Figure 3 are the Jacobian matrices for the initial position of the robot and for grasping the target, respectively.

Table 3. Jacobian matrix of the initial point

0	-0.388	-0.388	0	-0.078	0
0.465	0	0	0	0	0
0	-0.36	-0.08	0	0	0
0	0	0	0	0	0
0	1	1	0	1	0
1	0	0	-1	0	-1

Table 4. Jacobian matrix of the target point

0.374	-0.125	0.046	-0.042	0.055	0
-0.374	-0.126	0.046	0.042	0.055	0
0	-0.424	-0.286	0	0	0
0	0.707	0.707	-0.533	-0.707	0
0	-0.707	-0.707	-0.533	0.707	0
1	0	0	-0.658	0	1

Finally, the trajectory can be drawn through the jtraj() function. Trajectory is shown in Figure 3, and Figure 4 is the image of angular velocity and angular acceleration of each joint in the whole process.

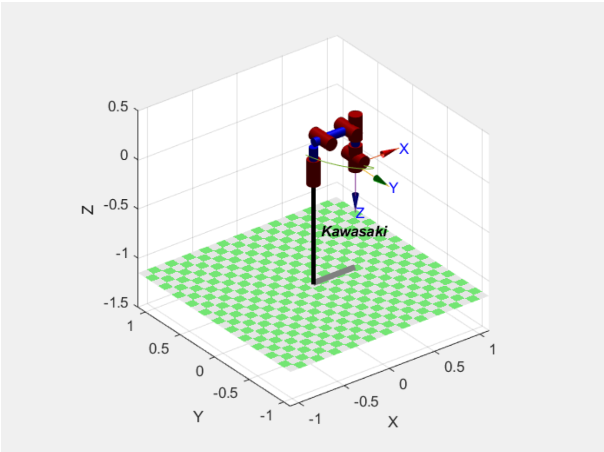

Figure 3. The simulated grab trajectory map

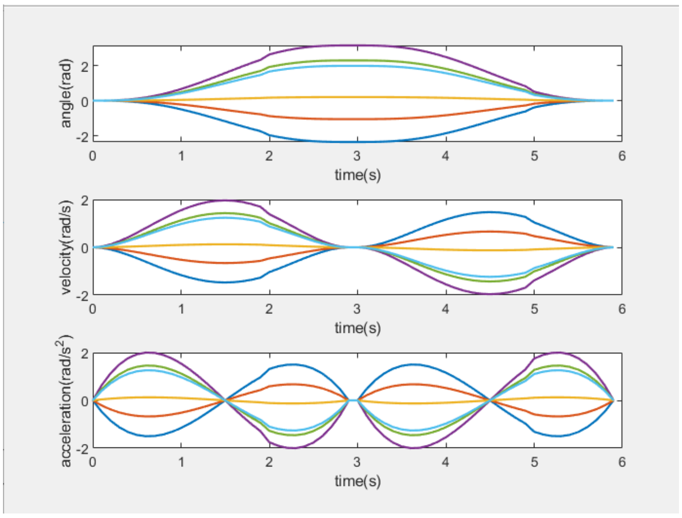

Figure 4. Speed curve analysis

4. Discussion

Based on the results, it is evident that the MATLAB modeling and simulation outcomes were relatively successful. By analyzing the icons in Figure 4, it can be concluded that at the third second, the speed and acceleration become zero, and each joint angle reaches its corresponding maximum position, indicating that the robot gripper has moved to the target position. Additionally, at the sixth second, each joint angle becomes zero, and the speed and acceleration are also zero, indicating that the robot has returned to its initial position.

However, it is important to note that the model structure used in this study is relatively simple, and there are several other factors that must be considered for more realistic simulations. In the future, the six-axis robot model will be optimized in terms of algorithm and modeling to make it more closely resemble the actual situation.

5. Conclusion

This article discusses a new algorithm for solving the forward and inverse kinematics problem of a six-axis robot using DH table. The algorithm is implemented and simulated in MATLAB, and the results show that it is highly efficient and accurate. The article also presents the computation of the Jacobian matrix, maximum motion speed, and acceleration of the robot in different workspaces, which are important for dynamic modeling and control of the robot.

The main contribution of the study is the introduction of a novel algorithm for solving the forward and inverse kinematics problem of a six-axis robot using DH table. The effectiveness of the proposed approach is validated through experimental results. Furthermore, the dynamic parameters of the robot are evaluated, which can be used as a reference for optimizing and controlling the robot.

References

[1] Wang, Z, Sheu, J, Teo, et al, G 2022, ‘Robot Scheduling for Mobile-Rack Warehouses: Human–Robot Coordinated Order Picking Systems’, Production and operations management, vol. 31, no. 1, pp. 98–116.

[2] Lam, C-P, Chou, C-T, Chiang, et al, L-C 2011, ‘Human-Centered Robot Navigation-Towards a Harmoniously Human-Robot Coexisting Environment’, IEEE transactions on robotics, vol. 27, no. 1, pp. 99–112.

[3] Sarchi, L, Mottaran, A, et al, 2022, ‘Robot-assisted radical prostatectomy feasibility and setting with the HugoTM robot-assisted surgery system’, BJU international, vol. 130, no. 5, pp. 671–675.

[4] Barbash, GI & Glied, SA 2010, ‘New Technology and Health Care Costs — The Case of Robot-Assisted Surgery’, The New England journal of medicine, vol. 363, no. 8, pp. 701–704.

[5] Bingol, MC & Aydogmus, O 2020, ‘Performing predefined tasks using the human–robot interaction on speech recognition for an industrial robot’, Engineering applications of artificial intelligence, vol. 95, p. 103903.

[6] Alam Md Moktadir, et al, 2021, ‘Kinematic Modeling of Six-Axis Industrial Robot and its Parameter Identification: A Tutorial’, International journal of automation technology, vol. 15, no. 5, pp. 599–610.

[7] Lu, Y & Wang, K 2021, ‘Kinematics Analysis and Trajectory Planning of Polishing Six-axis Robot’, IOP conference series. Earth and environmental science, vol. 632, no. 5.

[8] Wang, P, et al, N 2022, ‘An Inverse Solution Algorithm for Industrial Robot’, Journal of physics. Conference series, vol. 2173, no. 1, p. 012085.

[9] Zhang, T, Song, et al, Q 2021, ‘A novel method to identify DH parameters of the rigid serial-link robot based on a geometry model’, Industrial robot, vol. 48, no. 1, pp. 157–167.

[10] Kawasaki Robotics. https://kawasakirobotics.cn/products-robots/rs005n/.

[11] Hua Dai, Chengyong Yao, 2003, ‘Conditions for the existence of solutions to the inverse eigenvalue problem of Jacobian matrices’ [J] Journal of Computational Mathematics of Colleges and Universities, vol. 25, pp. 40-49.