



A fast task planning system for 6R articulated robots based on inverse kinematics

Yuan-Lung Lai¹

Received: 29 August 2022 / Accepted: 25 January 2023 / Published online: 13 February 2023
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2023

Abstract

Robots bring eventful impacts to the workplace, benefiting from the advantages of implementing any technology should be based on the premise of safety. This work proposes a systematic method to establish a postprocessor module for any specified 6R articulated robot. Instead of obsessively emphasizing achieving the desired locations and orientations by hand guiding grab and dragging a robotic arm for operation in teaching mode, the significant end-effector poses are calculated to form a new path for the joints efficiently in this study. Since robotic motion control is usually a complex system whose users must be well trained and acquainted with using them. There is a need for a GUI solution that can provide intuitive robotic motion control on the current location by the user independently, easy setup, arrangement, adjustment, and monitoring robot motion tasks. The proposed system simplifies the interaction between the technician and the industrial robotic arm in the case of robotic motion control and tracking at a distant location. The presented method is fully adapted to alternate between joint angles and end-effector poses on a graphic user interface system. After developing the capabilities of the solver, a functional postprocessor is programmed inside the proposed GUI system. Examples with specified posture and predefined movements are demonstrated for corroborating the algorithm. The results show considerable efficiency and reliability in task planning and are fully supported for automatic path generation.

Keywords Inverse kinematics · Six-axis robotic arm · Articulated · Postprocessor · D-H transformation

1 Introduction

Robots are becoming increasingly popular in workplaces to work next to humans and may always need humans to program them for tasks. Technicians may make mistakes that cause unexpected consequences resulting from a lack of understanding, or control of work processes. In the face of increasing demands in rapidly changing markets, the most important challenges for manufacturing industries are creating high-quality products and minimizing invalid operations. To improve fabrication capacity and reduce production expenses, adequate and systematic technologies must be developed for manipulators, which will ensure advanced automation, increased productivity, and effective

and economical production processes, all while lowering prices and time constraints.

Without requiring advanced knowledge, the hand-guided operation is currently the easiest method of programming for technicians to teach the robot [1]. Recently, collaborative robot is prevalent in smart factories [2]. It is still a challenge to have robots and technicians sharing the same space for the risk of injuries [3], not to mention being in an extremely harsh working environment. The outbreak of the coronavirus COVID-19 has affected the world in varying ways in the manufacturing industry. The fact is that employers tend to increase automation and robots rather than hiring new workers in many cases.

Owing to the simultaneous interpolation movement on the revolving axes on a six-axis robotic arm, the derivation of the six-axis trajectory path is more complex than that of the lower DOF trajectory path. For that reason, a specified postprocessor has to be used to convert end-effector pose posture data from a task planning system to CNC units. Despite superior control units that can accept end-effector pose locations to accomplish a task in real-time

✉ Yuan-Lung Lai
lyllaiber@cc.ncue.edu.tw; laiberlai@gmail.com

¹ Department of Industrial Education and Technology,
National Changhua University of Education, 2, Shida Rd,
Changhua 500, Taiwan, Republic of China

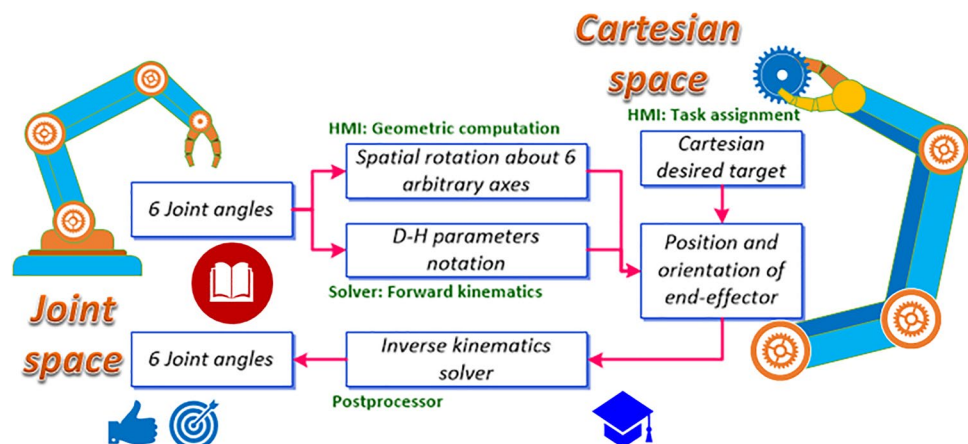
without post-processing support, they are relatively costly and utilized in some specific operations only. A typical problem facing the singularity is that a robotic arm fails one or more degrees of freedom. Specific configurations make the robot end-effector block in certain postures. Another severe problem that will cause very high joint velocities is close to the singular positions. In order to reach the desired posture, the revolving axes may lead to a quick return while the end-effector traverse closes singular positions. Facing such a situation without any protection, the unpredictable motion will cause harmful results during the task. A technician can apply an empirical way to guide the end-effector to retract along the end-effector axis at a secure distance away from the reserved position, revolve joints to the appropriate angles, and approach along the end-effector axis back to the intended position. Invalid operations result in a loss of productivity. To ensure a smooth trajectory in robotic arm motion control, the linearization method may be used. The cost of such a minor deviation operation is that accuracy must be compromised by dividing small step movements. On the contrary, the target position data is refined and regenerated to traverse in joint space for 6-axis robotic arms by an intuitive kinematics transformation system proposed in this work.

There are many candidate solutions for a 6R robot that are comparable to the same end-effector position. Traditionally, analytical and numerical methods are two main directions widely used to develop inverse kinematic (IK) techniques on robotic arms. Analytical closed-form solutions with analytical expressions can be achieved either algebraically or geometrically [4, 5]. Husty et al. [6] provided polynomial solutions of general 6R-chains to solve inverse kinematic problems. The numerical solution implementations are computational and iterative while using multidimensional Newton-Raphson or similar techniques to provide a solution [7, 8]. Apparently, numerical implementations are much slower in converging to all the admissible solutions. The human machine interface

(HMI) system provides a perfect tool for speeding task planning. As shown in the flowchart in Fig. 1, the relationship between forward and inverse kinematics illustrates the consideration that the D-H transformation is not the only way to find postures of arbitrary end-effectors. By using the analytical method to build a solution domain provides all potential solutions and selects one default by the minimum rotary angle in this paper.

Some efficient methods of inverse kinematics procedure based on D-H representation have been proposed for 6R robot [9–12]. At present, several different studies using various robots and methods have been developed in the field of robotics. To find closed-form solution of a hybrid robot, [13] proposed a numerical method to convert the higher order kinematic equation to a 4th-order polynomial. A modified product-of-exponential method is proposed to formulate a kinematic calibration method [14]. A further kinematic calibration method through machine vision for 6R robot can be found in [15]. The method with extra constraints for robotic machining is employed time-optimal motion planning [16]. Another path smoothing and feed rate planning based on B-spline is presented by [17]. The robot configuration is evaluated to improve the machining accuracy by a stiffness-based method [18, 19]. Applying the quaternion interpolation algorithm and concluding all cutting tool angles influence, an efficient model is developed to predict achievable precision for drilling [20]. A new parallel robot for minimally invasive surgery is constructed [21]. The singularity analysis for this parallel robot is presented in the robot workspace for design optimization of the robot [22]. Recently, some intelligent techniques to solve the IK problem have been proposed. These soft-computing methods have attracted more attention to solving the IK problem [23–30]. These methods can explore the answer without prior knowledge of the robot models. Most of the processes solve IK problems using the mapping function from the Cartesian space to joint space based on the intelligence attained through several training phases.

Fig. 1 A flowchart of the forward and inverse kinematics solver



Numerical techniques can achieve a solution by successive approximation algorithms, and the accuracy depends on iterations required for a suitable convergence from a starting point [31]. Increasing the step size in numerical methods reduces the error at the cost of computational iterations. This time-consuming process is hard to be used in the controller to real-time compute the position of the end-effector for online monitoring. The numerical methods [32, 33] find approximate solutions iteratively, which are close to the initial estimations, and it searches only one angle solution. Numerical solvers usually do not return all the solutions. Some numerical solutions may not correspond to physical postures. To avoid the limitation that comes from the numerical method, analytical solutions are desired to have complete forward kinematics solutions for different configurations of robotic arms [34]. Compared with numerical methods, direct and analytic computations of IK are extremely arduous because of the non-linearity of forward kinematics for high-DOF robotic arms [35]. Moreover, the structure of the robotic arm has to comply with certain kinematics rules. Although obtaining all the intermediate joint coordinates can be very difficult due to the large size and complexity of the analytic expressions that must be handled in the calculations. This paper illustrates the first comprehensive expressions to tackle the IK for 6R robotic arms that find multiple distinct joint solutions for given target positions fast.

The rest of the paper is structured as follows. First, a fast path planning system based on an intuitive concept is pointed out. Section 2 depicts the industrial six-axis robotic arms and D-H transformation. In Section 3, the organization of kinematics by constructing a mathematical model is proposed. Section 4 describes the derivation of the analytical module to transform end-effector poses to the numerical control commands. A 3D intuitive GUI system is developed to create a path strategy for complicated tasks. Section 5 illustrates some practical examples to perform the effectiveness and feasibility of the proposed method is discussed. Finally, Section 6 concludes this work.

2 D-H notation and transformation

It has always been difficult to evaluate which type of mechanism is superior, parallel, or serial. Table 1 shows a simple summary of the essential concept between parallel and serial mechanisms [34]. It is complex and difficult to determine the mechanism differentiation measure. Advancement has always been proposed to improve modeling tasks. A new parameterization method was introduced to simplify the mathematical model of parallel robots straightforwardly [36]. It completely depends

Table 1 Summary of serial and parallel models

Item	Serial mechanism	Parallel mechanism
Workspace	Expected	Limited
Workspace/footprint ratio	High	Low
Solve forward kinematics	Easy	Very difficult
Solve inverse kinematics	Difficult	Relatively easy
Calibration	Relatively easy	Complicated
Modal analysis	Relatively simple	Very complex
Analysis of singularity	Common	Complex
Position error	Accumulates	Averages
Dynamic error	Averages	Accumulates

on geometrical considerations and the requirements of the tasks performed.

On simplifying transformations, a robotic arm can be regarded as links connected in a set of joints. Except for very high speed and under the concern of limited costs, the configuration of a fully serial system provides a noticeable potential benefit over the hexapod. Figure 2 illustrates the mechanical structure of the proposed six-axis robotic arm, which is the prototype of a planned reconfigurable robotic arm for a multi-task and multi-target unit.

In order to find the orientation of the end-effector in joint space, the relationships of every two connecting links should be developed. An open-loop kinematic chain can be constructed by the revolute joints linked together. The structural model of a six-axis robot can be therefore regarded as a combination of the individual kinematic models of the links. The revolute joint turns the propulsion of the link about the corresponding Z-axis. The proposed kinematic transformation of the six-axis robotic arm in this study is illustrated in Fig. 3. The desired joint positions are provided, and the end-effector Cartesian tip will be obtained through the forward kinematic. To determine the posture of each successive link, using symbols $O_1 - O_6$ represent the links of the proposed kinematic chain. One local coordinate system is connected to one link. All local coordinate systems are applied by the number of the link, which the i^{th} coordinate system attaches to link i . To represent spatial mechanisms, the most commonly used parametric convention is the Denavit–Hartenberg (D–H) notation.

To build a successive homogeneous transformation matrix, the representation is described by four parameters. They are two directions of translation and two axes of rotation, including link length, angle, offset, and twist angle. Homogeneous transformation matrices have been used to describe the position and orientation of the end-effector. One transformation merges translation and rotation into one matrix. As shown in (1), the linear transformation from one coordinate system to the successive

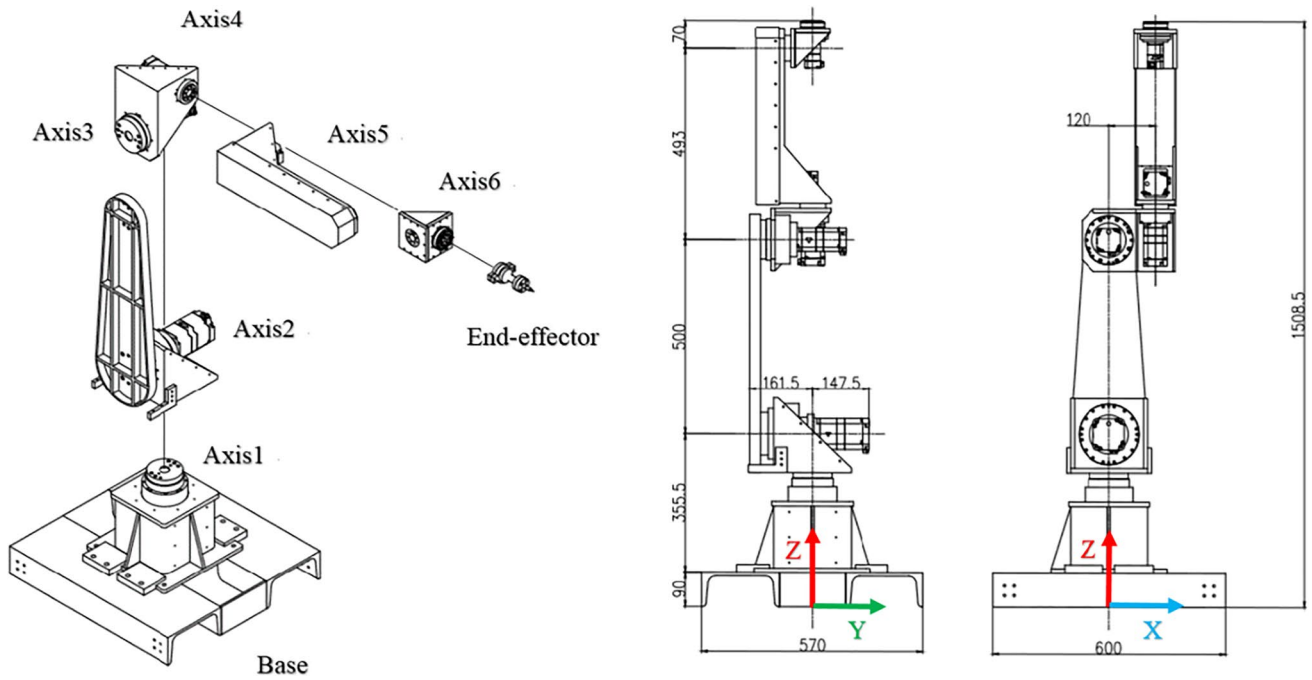
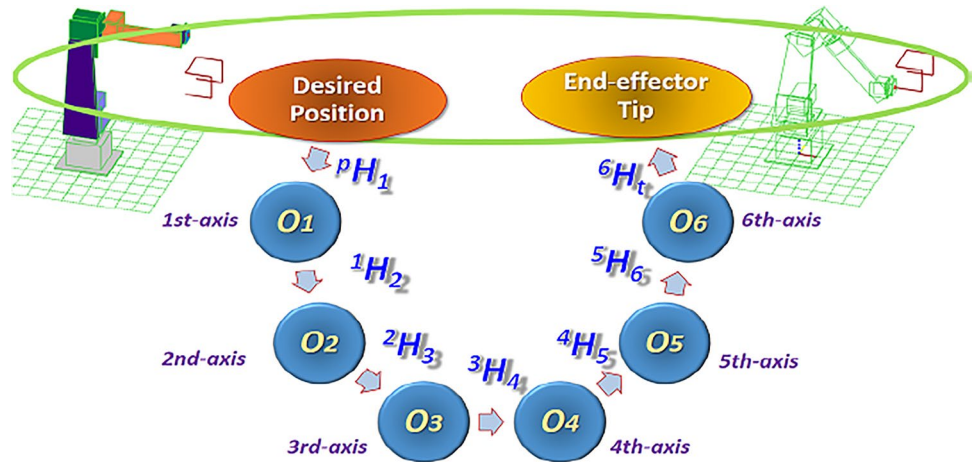


Fig. 2 Proposed six-axis robotic arm

Fig. 3 Chain of the proposed kinematic transformation of the six-axis robotic arm



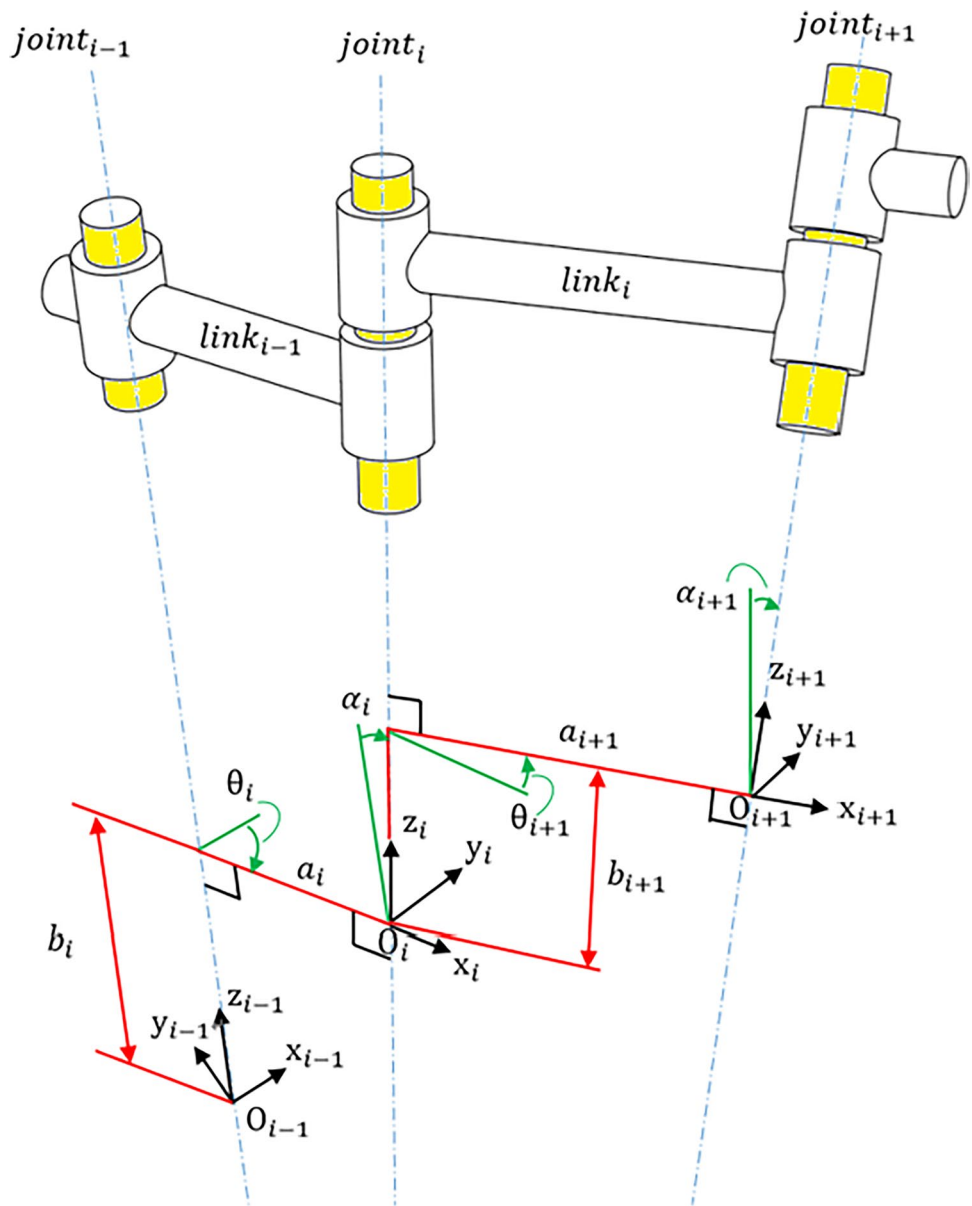
coordinate system is built by combining four homogeneous transformation matrices. The transformation matrix between adjacent coordinate frames $(i - 1)^{th}$ and i^{th} for the D–H representation changes in the proposed convention as follows:

$$\begin{aligned}
 {}^{i-1}H_i &= Trans(0, 0, b_i)Rot(z, \theta_i)Trans(a_i, 0, 0)Rot(x, \alpha_i) \\
 &= \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & b_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}
 \end{aligned}$$

As illustrated in Fig. 4, O_i and O_{i+1} are the two adjacent coordinate systems presented as follows. O_i is the origin and placed on the intersection of Z_i and common normal to both the $(i - 1)^{th}$ and i^{th} joint axes. Where Z_i –axis represents the direction of the i^{th} joint axis, X_i –axis is onward the common normal vector, and Y_i –axis is defined by the cross product of the Z_i –axis and X_i –axis.

The derivation of forward kinematics of the proposed algorithm follows the D-H convention. To begin with, define the $(i - 1)^{th}$ link. Next, label and place the $(i - 1)^{th}$ and i^{th} joints. Step three, build the local coordinate system. Specify the Z_i –axis to perform the rotary motion. Step four, place the base O_i between links $i - 1$ and i . Both X_i

Fig. 4 D-H notation for link transformation



and Y_i axes are determined to construct a right-hand coordinate system. Step five, define the X_i -axis onwards the common normal of Z_{i-1} -axis and Z_i -axis; the X_i -axis perpendicular to Z_i -axis and passes through O_i , where $\vec{x}_i = \pm(\vec{z}_{i-1} \times \vec{z}_i) / \|\vec{z}_{i-1} \times \vec{z}_i\|$. Step six, determine the Y_i -axis to form a right-hand coordinate system completely, $\vec{y}_i = \pm(\vec{z}_i \times \vec{x}_i) / \|\vec{z}_i \times \vec{x}_i\|$. Step seven, create a parameter table for b_i , θ_i , a_i , and α_i . Step eight, compute the homogeneous transformation matrices from the above parameter table into (1). Step nine, calculate ${}^0H_7 = {}^0H_1 {}^1H_2 {}^2H_3 {}^3H_4 {}^4H_5 {}^5H_6 {}^6H_7$. The desired position and orientation of the end-effector based on Cartesian space can be obtained.

The D-H parameters are defined sequentially as:

- b_i : offset length from location O_{i-1} to the common normal along Z_{i-1} -axis,
- θ_i : rotate angle about the Z_{i-1} -axis, between the X_{i-1} -axis and the X_i -axis,
- a_i : span length from Z_{i-1} -axis to Z_i -axis along the common normal,
- α_i : twist angle about the X_i -axis, between the Z_{i-1} -axis and the Z_i -axis.

The geometric dimensions of the proposed robotic arm are described in Fig. 5 and Table 2. In this case, each two links i and $i + 1$ employ a rotary mechanism. To the final, link 6 and link 7 implement the end-effector tool length.

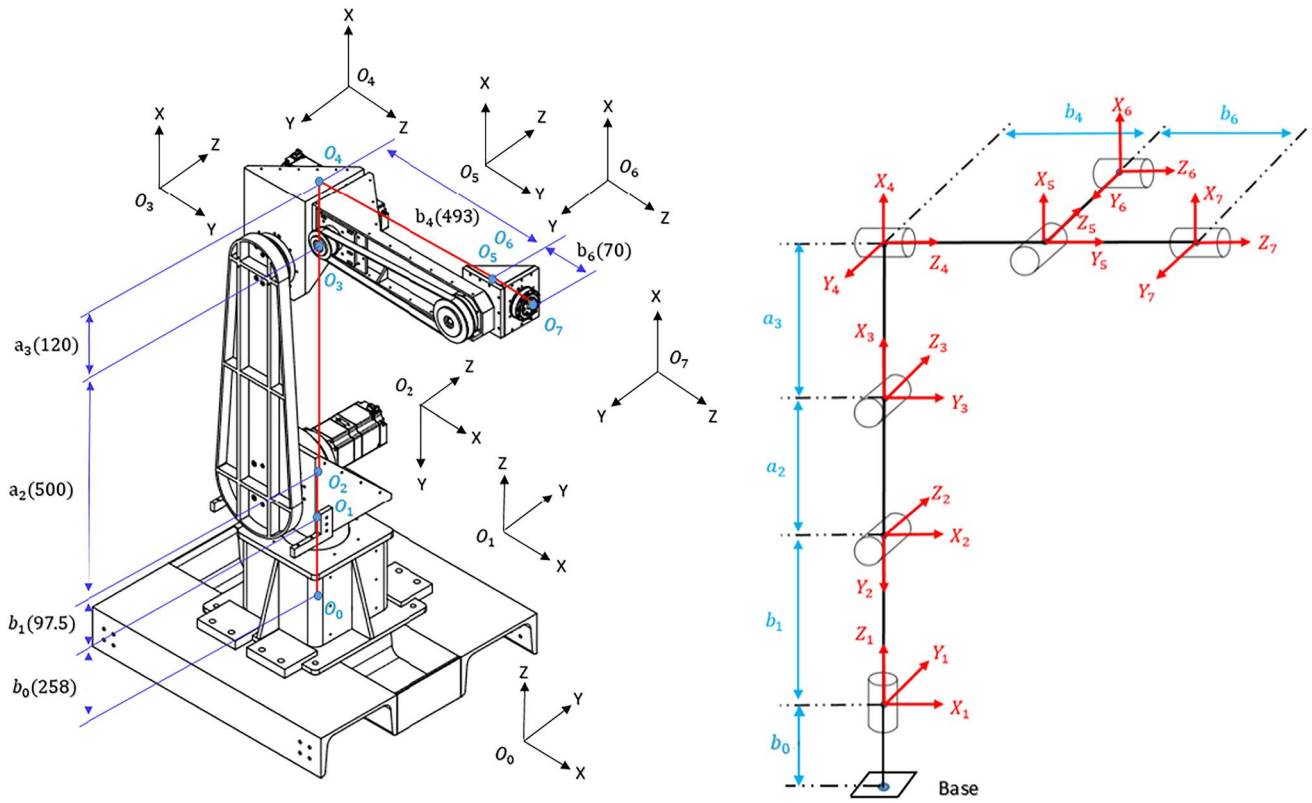


Fig. 5 Illustration of the geometric dimensions of the proposed six-axis robotic arm

Table 2 D-H parameters of the proposed example in Fig. 5

Link	b_i	θ_i	a_i	α_i
0/1	b_0	0	a_0	0
1/2	b_1	θ_1	a_1	$-\pi/2$
2/3	b_2	$-\pi/2 + \theta_2$	a_2	0
3/4	b_3	θ_3	a_3	$-\pi/2$
4/5	b_4	θ_4	a_4	$\pi/2$
5/6	b_5	θ_5	a_5	$-\pi/2$
6/7	b_6	θ_6	a_6	0

$$\begin{aligned}
 {}^6H_7 &= \text{Trans}(0, 0, b_6) \text{Rot}(z, \theta_6) \text{Trans}(0, 0, 0) \text{Rot}(x, 0) \\
 &= \begin{bmatrix} \cos \theta_6 & -\sin \theta_6 & 0 & 0 \\ \sin \theta_6 & \cos \theta_6 & 0 & 0 \\ 0 & 0 & 1 & b_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} ({}^6H_7)^{-1} \\
 &= \begin{bmatrix} \cos \theta_6 & \sin \theta_6 & 0 & 0 \\ -\sin \theta_6 & \cos \theta_6 & 0 & 0 \\ 0 & 0 & 1 & -b_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)
 \end{aligned}$$

The matrix 6H_7 , for example, indicating the D-H movements is computed by post-multiplying the four matrices specifying four transfer steps from frame 6 to 7 as shown in

(2). Provide the significant geometric structure of the manipulator and the prescribed values of the joints. The forward kinematics is implemented to obtain the relative distance and orientation of any two adjacent members. The transformation for the proposed serial-chain robot shown in Fig. 5 is

$${}^0H_7 = \prod_{i=1}^7 {}^{i-1}H_i = {}^0H_1 {}^1H_2 {}^2H_3 {}^3H_4 {}^4H_5 {}^5H_6 {}^6H_7 \quad (3)$$

It indeed needs a compact tool to achieve the pose of the robot. The homogeneous transformations provide such a compact representation. The computational efficiency for calculating the forward kinematics seems to underperform in computation. A reduction can be obtained by performing the separation of the position and orientation portions of the transformation. Since 0H_1 only indicates the relative displacements between base and foundation, to simplify the calculation and reduce the computation burden, it may be possible to remove 0H_1 first and then apply back in the final.

$${}^1H_7 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^1n_7 & {}^1o_7 & {}^1a_7 & {}^1p_7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

3 Mathematical solvers of inverse kinematics

Prior to modeling a mathematical solver of inverse kinematics, computing forward kinematics has to be solved first. A set of active-joint variables can define the pose of the end-effector uniquely. According to the matrix multiplications of (4), the final positions and orientations of the end-effector can be obtained.

The position vector is

$${}^1\mathbf{p}_7 = [p_x \ p_y \ p_z]^T \tag{5}$$

where the position of three components:

$$p_x = -((C_1C_2S_3 + C_1S_2C_3)C_4 + S_1S_4)S_5 + (C_1C_2C_3 - C_1S_2S_3)C_5b_6 + (C_1C_2C_3 - C_1S_2S_3)b_4 + (C_1C_2S_3 + C_1S_2C_3)a_3 + C_1S_2a_2 \tag{6}$$

$$p_y = -((S_1C_2S_3 + S_1S_2C_3)C_4 - C_1S_4)S_5 + (S_1C_2C_3 - S_1S_2S_3)C_5b_6 + (S_1C_2C_3 - S_1S_2S_3)b_4 + (S_1C_2S_3 + S_1S_2C_3)a_3 + S_1S_2a_2 \tag{7}$$

$$p_z = -(C_2C_3 - S_2S_3)C_4S_5 + (-C_2S_3 - S_2C_3)b_4 + (-C_2S_3 - S_2C_3)C_5b_6 + (C_3a_3 + a_2)C_2 - S_2S_3a_3 + b_0 + b_1 \tag{8}$$

To simplify the expressions in this study, $\cos\theta_i$, $\sin\theta_i$, $\cos(\theta_i + \theta_j)$, and $\sin(\theta_i + \theta_j)$ are abbreviated to C_i , S_i , C_{ij} , and S_{ij} , respectively.

The unit normal vector is

$${}^1\mathbf{n}_7 = [n_x \ n_y \ n_z]^T = [r_{11} \ r_{21} \ r_{31}]^T \tag{9}$$

where the normal direction of three components:

$$r_{11} = ((C_1C_2C_3 + C_1S_2C_3)C_4 + S_1S_4)C_5 + (C_1C_2C_3 - C_1S_2S_3)S_5C_6 + (-C_1C_2S_3 + C_1S_2C_3)S_4 + S_1C_4)S_6 \tag{10}$$

$$r_{21} = ((S_1C_2S_3 + S_1S_2C_3)C_4 - C_1S_4)C_5 + (S_1C_2C_3 - S_1S_2S_3)S_5C_6 + (-S_1C_2S_3 + S_1S_2C_3)S_4 - C_1C_4)S_6 \tag{11}$$

$$r_{31} = ((C_2C_3 - S_2S_3)C_4C_5 + (-C_2S_3 - S_2C_3)S_5)C_6 - (C_2C_3 - S_2S_3)S_4S_6 \tag{12}$$

The unit orientation vector is

$${}^1\mathbf{o}_7 = [o_x \ o_y \ o_z]^T = [r_{12} \ r_{22} \ r_{32}]^T \tag{13}$$

where the orientation direction of three components:

$$r_{12} = -((C_1C_2S_3 + C_1S_2C_3)C_5 + (C_1C_2C_3 - C_1S_2S_3)S_5)S_6 + (-C_1C_2S_3 + C_1S_2C_3)S_4 + S_1C_4)C_6 \tag{14}$$

$$r_{22} = -((S_1C_2S_3 + S_1S_2C_3)C_4 - C_1S_4)C_5 + (S_1C_2C_3 - S_1S_2S_3)S_5)S_6 + (-S_1C_2S_3 + S_1S_2C_3)S_4 - C_1C_4)C_6 \tag{15}$$

$$r_{32} = -((C_2C_3 - S_2S_3)C_4C_5 + (-C_2S_3 - S_2C_3)S_5)S_6 - (C_2C_3 - S_2S_3)S_4C_6 \tag{16}$$

The unit approach vector is

$${}^0\mathbf{a}_7 = [a_x \ a_y \ a_z]^T = [r_{13} \ r_{23} \ r_{33}]^T \tag{17}$$

where the approach direction of three components:

$$r_{13} = -((C_1C_2S_3 + C_1S_2C_3)C_4 + S_1S_4)S_5 + (C_1C_2C_3 - C_1S_2S_3)C_5 \tag{18}$$

$$r_{23} = -((S_1C_2S_3 + S_1S_2C_3)C_4 - C_1S_4)S_5 + (S_1C_2C_3 - S_1S_2S_3)C_5 \tag{19}$$

$$r_{33} = -(C_2C_3 - S_2S_3)C_4S_5 + (-C_2S_3 - S_2C_3)C_5 \tag{20}$$

For serial-chain manipulators, the inverse kinematic problem involves the values of all the geometric link parameters and the joint variables. This simplified statement is only applicable to serial-chain robots. That can determine the position and orientation of the end-effector relative to the base in the Cartesian coordinate system. The more general statement is to solve the values of all the joint directions and positions for any kind of manipulator. It is more complex to solve inverse kinematics than forward kinematics, not to mention modeling the parameter table is always a challenge for engineers. Once the formulation of the results of (5)–(20) is examined, the inverse kinematics solution applies the positions and directions ($p_x, p_y, p_z, \varphi_n, \varphi_o, \varphi_a$) of the end-effector can be obtained. Roll (φ_a), pitch (φ_o), and yaw (φ_n) are basically a sequence of three rotations about Z, Y, and X axis respectively for free orientation in space. The final matrix in (4) has been known to determine the joint variables ($\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$). It is possible to group the variables into two categories. The first three variables form the arm-like structure, while the last three variables perform the wrist-like mechanism. By inspection of the illustration and focus on the last three joints, the DOF axes of these three local coordinate systems intersect at a point that is independent of the three variables θ_4, θ_5 , and θ_6 .

Inspired by this feature, only the first three joints θ_1, θ_2 , and θ_3 should be considered in advance when determining the position of a certain point. Apply the results of the forward kinematics, the inverse kinematic of the proposed robot can be derived as below. Follow the analytic procedure, the inverse kinematics solution is calculated for all the joint parameters θ_i through the formulation matrix (4). To derive θ_1 , (4) may firstly reorganized as

$${}^1H_6 = {}^1H_2{}^2H_3{}^3H_4{}^4H_5{}^5H_6 = [m_{ij}] \quad (21)$$

where

$$m_{14} = (-S_1S_2S_3 + S_1C_2C_3)b_4 + S_1S_2C_3a_3 + S_1C_2S_3a_3 + S_1S_2a_2 = S_1(k) \quad (22)$$

$$m_{24} = (-C_1S_2S_3 + C_1C_2C_3)b_4 + C_1S_2C_3a_3 + C_1C_2S_3a_3 + C_1S_2a_2 = C_1(k) \quad (23)$$

$$H_7^1(H_7^6)^{-1} = \begin{bmatrix} r_{11} \cos \theta_6 - r_{12} \sin \theta_6 & r_{11} \sin \theta_6 + r_{12} \cos \theta_6 & r_{13} & -r_{13}b_6 + p_x \\ r_{21} \cos \theta_6 - r_{22} \sin \theta_6 & r_{21} \sin \theta_6 + r_{22} \cos \theta_6 & r_{23} & -r_{23}b_6 + p_y \\ r_{31} \cos \theta_6 - r_{32} \sin \theta_6 & r_{31} \sin \theta_6 + r_{32} \cos \theta_6 & r_{33} & -r_{33}b_6 + p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (24)$$

It is noted that the elements m_{14} and m_{24} can be utilized to define θ_1 .

$$\theta_1 = \tan^{-1} \left(\frac{p_y - r_{23}b_6}{p_x - r_{13}b_6} \right) \quad (25)$$

Once θ_1 is obtained, the first link inverse transformation matrix is pre-multiplied by the second part of Eq. (12) to find the inverse kinematics solution for θ_3 as a function of the known elements as follows:

$$\left(({}^1H_2)^{-1} \right) {}^1H_2{}^2H_3{}^3H_4{}^4H_5{}^5H_6 = {}^2H_3{}^3H_4{}^4H_5{}^5H_6 = {}^2H_6 = [m_{ij}] \quad (26)$$

where

$$m_{14} = (-S_2S_3 + C_2S_3)b_4 + S_2C_3a_3 + C_2S_3a_3 + S_2a_2 = b_4C_{23} + a_3S_{23} + a_2S_2 \quad (27)$$

$$m_{24} = (S_2C_3 + C_2S_3)b_4 - C_2C_3a_3 + S_2S_3a_3 - C_2a_2 = b_4S_{23} - a_3C_{23} - a_2S_2 \quad (28)$$

With the known elements,

$${}^2H_6 = ({}^1H_2)^{-1} ({}^1H_7) ({}^6H_7)^{-1} = [m_{ij}] \quad (29)$$

where

$$m_{14} = C_1(-r_{13}b_6 + p_x) + S_1(-r_{23}b_6 + p_y) \quad (30)$$

$$m_{24} = r_{33}b_6 - p_z + b_1 \quad (31)$$

Square both sides and add of (27), (28), (30), and (31) it yields as

$$A \cos(-\theta_3) + B \sin(-\theta_3) = D \quad (32)$$

The joint θ_3 can be obtained,

$$\theta_3 = \pm \cos^{-1} \left(\frac{D}{C} \right) + \varphi \quad (33)$$

where $C = \sqrt{(2a_2a_3)^2 + (2a_2b_4)^2}$, $A = C \cos \varphi$, $B = C \sin \varphi$, $\varphi = \tan^{-1} \left(\frac{2a_2b_4}{2a_2a_3} \right)$,

$$D = (C_1(-r_{13}b_6 + p_x) + S_1(-r_{23}b_6 + p_y))^2 + (r_{33}b_6 - p_z + b_1)^2 - b_4^2 - a_3^2 - a_2^2. \quad (34)$$

Once θ_1 and θ_3 are obtained, the similar process of inverse transformation matrixes is pre-multiplied to find the second joint variable solution for θ_2 as a function of the known elements as follows:

$$\left(({}^2H_3)^{-1} ({}^1H_2)^{-1} ({}^1H_7) ({}^2H_3) ({}^3H_4) ({}^4H_5) ({}^5H_6) \right) {}^3H_4{}^4H_5{}^5H_6 = {}^3H_6 = [m_{ij}] \quad (35)$$

where

$$m_{14} = -\cos \theta_3 \sin \theta_4 \quad (36)$$

With the known elements,

$${}^3H_6 = ({}^2H_3)^{-1} ({}^1H_2)^{-1} ({}^1H_7) ({}^6H_7)^{-1} = [m_{ij}] \quad (37)$$

where

$$m_{14} = (-b_6r_{13} + p_x) + \sin \theta_1(-b_6r_{23} + p_y) - \cos \theta_2(b_6r_{33} + b_1 - p_z) - a_2 \quad (38)$$

By rearranging the above equations, (36) and (38) can be rewritten as

$$A \cos(\theta_2) - B \sin(\theta_2) = D \quad (39)$$

The joint θ_2 can be obtained,

$$\theta_2 = \pm \cos^{-1} \left(\frac{D}{C} \right) - \varphi \quad (40)$$

where $C = \sqrt{(b_1 - p_z + r_{33}b_6)^2 + (\cos \theta_1(p_x - r_{13}b_6) + \sin \theta_1(p_y - r_{23}b_6))^2}$, $A = C \cos \varphi$, $B = C \sin \varphi$,

$$\varphi = \tan^{-1} \left[\frac{\cos \theta_1(p_x - r_{13}b_6) + \sin \theta_1(p_y - r_{23}b_6)}{b_1 - p_z + r_{33}b_6} \right], \quad D = b_4 \sin \theta_3 - a_3 \cos \theta_3 - a_2. \quad (41)$$

Once θ_1 , θ_2 , and θ_3 are obtained, the corresponding link inverse transformation matrixes are pre-multiplied to find the inverse kinematics solution for θ_4 as a function of the known elements as follows:

$$\left(({}^3H_2)^{-1} ({}^2H_3)^{-1} ({}^1H_2)^{-1} ({}^1H_7) ({}^2H_3) ({}^3H_4) ({}^4H_5) ({}^5H_6) ({}^6H_7) \right) {}^4H_5{}^5H_6{}^6H_7 = {}^4H_7 = [m_{ij}] \quad (42)$$

where

$$m_{13} = -\cos \theta_4 \sin \theta_5 \tag{43} \quad m_{23} = S_1 r_{13} - C_1 r_{23} \tag{50}$$

$$m_{23} = -\sin \theta_4 \sin \theta_5 \tag{44} \quad m_{31} = (C_3 C_2 - S_3 S_2) C_1 r_{11} + (C_3 C_2 - S_3 S_2) S_1 r_{21} - (S_3 C_2 + C_3 S_2) r_{31} \tag{51}$$

$$m_{31} = \sin \theta_5 \cos \theta_6 \tag{45} \quad m_{32} = (C_3 C_2 - S_3 S_2) C_1 r_{12} + (C_3 C_2 - S_3 S_2) S_1 r_{22} - (S_3 C_2 + C_3 S_2) r_{32} \tag{52}$$

$$m_{32} = -\sin \theta_5 \sin \theta_6 \tag{46} \quad m_{33} = (C_3 C_2 - S_3 S_2) C_1 r_{13} + (C_3 C_2 - S_3 S_2) S_1 r_{23} - (S_3 C_2 + C_3 S_2) r_{33} \tag{53}$$

$$m_{33} = \cos \theta_5 \tag{47}$$

With the known elements,

$${}^4H_7 = ({}^3H_4)^{-1} ({}^2H_3)^{-1} ({}^1H_2)^{-1} ({}^1H_7) = [m_{ij}] \tag{48}$$

where

$$m_{13} = (S_3 C_2 + C_3 S_2) C_1 r_{13} + (S_3 C_2 + C_3 S_2) S_1 r_{23} + (C_3 C_2 - S_3 S_2) r_{33} \tag{49}$$

Calculating the division of (43), (44), (49), and (50) on both side it yields as

$$\tan \theta_4 = \frac{m_{23}}{m_{13}} \tag{54}$$

The joint θ_4 can be obtained,

$$\theta_4 = \tan^{-1} \left(\frac{m_{23}}{m_{13}} \right) = \tan^{-1} \left[\frac{S_1 r_{13} - C_1 r_{23}}{(S_3 C_2 + C_3 S_2) C_1 r_{13} + (S_3 C_2 + C_3 S_2) S_1 r_{23} + (C_3 C_2 - S_3 S_2) r_{33}} \right] \tag{55}$$

Since (51), (52), and (53) are known, comparing the elements of (47) and (53) the θ_5 yields as

$$\theta_5 = \pm \cos^{-1}(m_{33}) = \pm \cos^{-1}((C_3 C_2 - S_3 S_2) C_1 r_{13} + (C_3 C_2 - S_3 S_2) S_1 r_{23} - (S_3 C_2 + C_3 S_2) r_{33}) \tag{56}$$

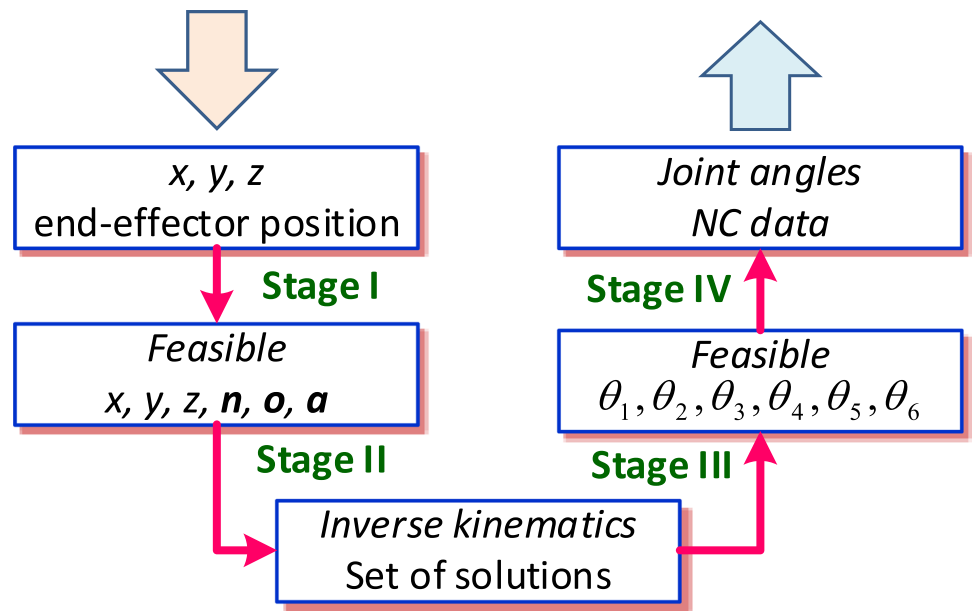
By dividing the element m_{32} by m_{31} , the joint θ_6 can be obtained,

$$\theta_6 = \tan^{-1} \left(-\frac{m_{32}}{m_{31}} \right) = \tan^{-1} \left(-\frac{(C_3 C_2 - S_3 S_2) C_1 r_{12} + (C_3 C_2 - S_3 S_2) S_1 r_{22} - (S_3 C_2 + C_3 S_2) r_{32}}{(C_3 C_2 - S_3 S_2) C_1 r_{11} + (C_3 C_2 - S_3 S_2) S_1 r_{21} - (S_3 C_2 + C_3 S_2) r_{31}} \right) \tag{57}$$

Inverse problems are the opposite to forward problems, and the aim is to recover the causes given the task with an end-effector. The goal is to find one or more configurations to perform the desired pose of end-effector in inverse kinematics. If there exists a set of joint angles that satisfies the target positions and directions of the end-effector, there is expected a distinct inverse to the forward kinematics exists. This well-defined inverse can be built as the function of forward kinematics and is well-posed. In practical, as the target position is given, the challenge is to discover the causes shortly. The task is to detect a certain set of joint angles to meet the desired end-effector position, to find a well-defined inverse to the

forward kinematic function. However, without pre-defined constraints, the inverse kinematics problem may result in ill-posed because of no solution or infinite solutions. Owing to the geometric limits, if the target location exceeds the reachable workspace of the robotic arm, there is no solution performed by inverse kinematics because the point is infeasible. In this work, the inverse kinematics problem is solved by the proposed analytical closed-form solutions. Meanwhile, inverse kinematic equations can provide multiple solutions. Given the desired Cartesian description of end-effector within the reachable workspace, six-axis robotic arms can perform up to eight possible configurations for the same end-effector pose.

Fig. 6 Illustration of the main procedure of the proposed method



4 Postprocessor for a six-axis robotic arm

A flowchart is shown in Fig. 6, which illustrates the main procedure of the proposed method to find a superior solution. First of all, within the frame refer to machine base point and provide desired Cartesian space coordinates. At stage I, the pre-processor plays the role to provide positions and directions for the end-effector in Cartesian space. Then, after target coordinates x , y , and z are defined, refine and yield the normal and orientation vectors as approach vector is known already at stage II. Feasible positions and orientations for

the end-effector are verified at this stage. At stage III, a solution domain provides all potential solutions and selects one default by minimum rotary angle. The consideration that performs the determination of joint solutions in coding is interpreted in Fig. 7. Finally, all of the appropriate numerical control commands are generated for the controller system at stage IV.

A solid model of the industrial six-axis robotic arm proposed in this study is illustrated in Fig. 8. This device comprises six axes, whereas reconfigurable end-effectors are mounted on the sixth link of the manipulator. The

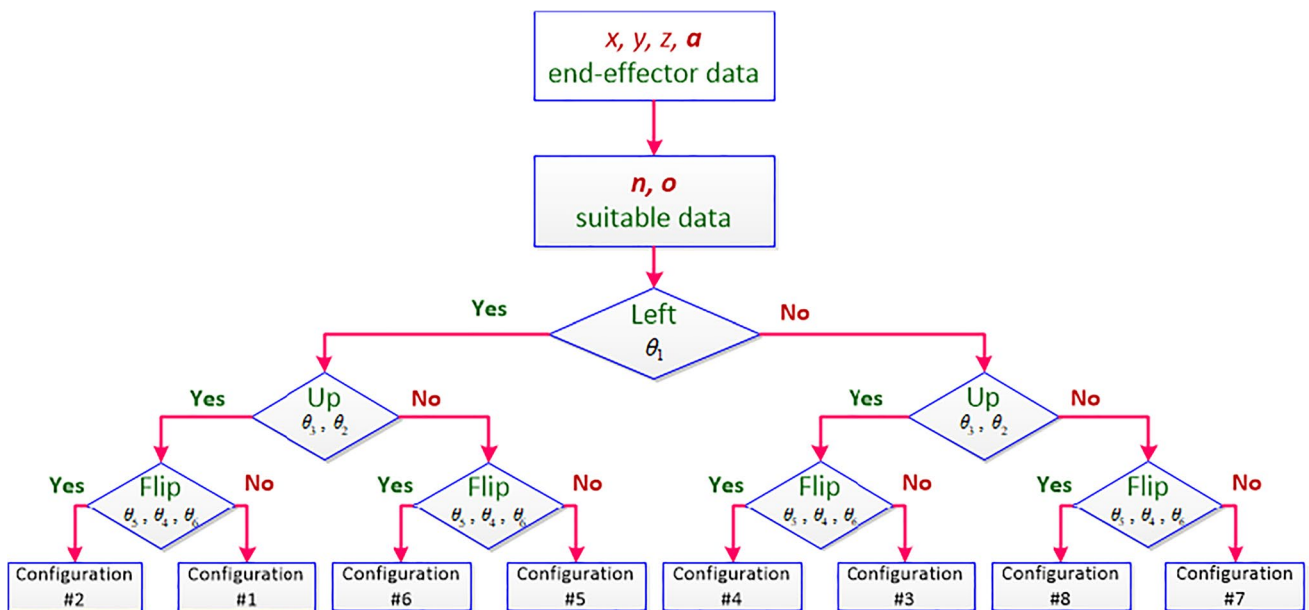
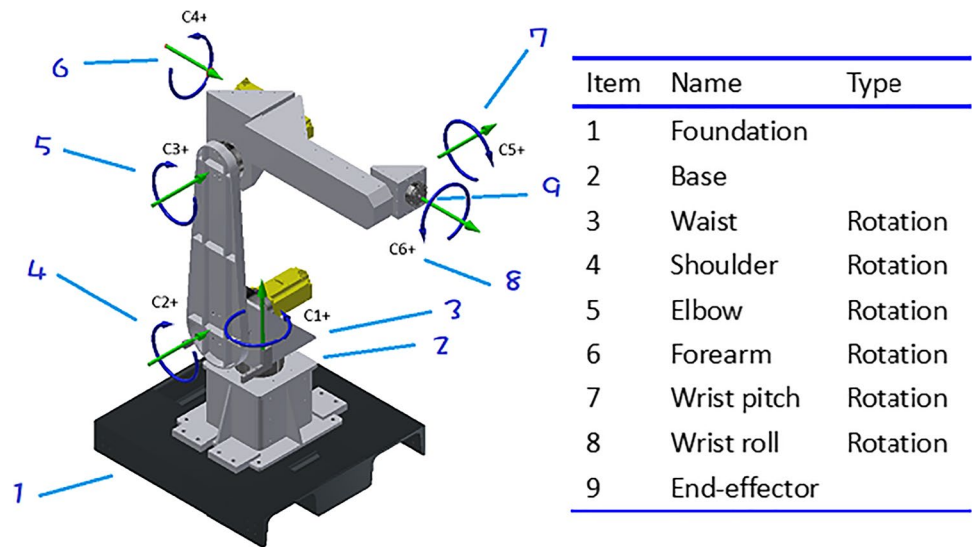


Fig. 7 Determination flowchart of joint angles

Fig. 8 Schematic of the proposed robotic mechanism



transformation data are obtained from the Cartesian space in the software GUI system to the joint space in the CNC system through a postprocessor. The corresponding commands are converted to the controller and executed to propel the end-effectors to the desired positions. As described in Fig. 1, postprocessors primarily use inverse kinematics to obtain the joint parameters that perform the desired end-effector posture given in Cartesian space. The post-processor is a unidirectional implementation. Any corrections made in the commands at the controller are executed immediately. Instructions provided to the manipulator are applied immediately. Postprocessors are used to embed forward kinematics and maintain the end-effector’s trajectory accuracy. To examine the effective performance of the proposed method, a Windows-based path planning system was programmed under the Windows 10 coding environment. The kernel and GUI are developed by Visual Studio, which displays the mathematical solver of kinematics. The model can transfer variable lengths of changeable end-effector immediately. By building a robotic library, users may select the model type and choose the corresponding parameter table. This proposed system will generate the appropriate NC commands for the configured robotic arm automatically.

As the proposed example in Fig. 5, the complete geometric model has been configured based on Cartesian coordinate convention. All significant parameters are defined completely, as illustrated in Table 3. To organize path planning, the desired postures have to move from the fundamental origin to the center of the first joint and then from the center of the first joint to the center of the second joint. The rests follow the same steps as describe above. Finally, the target pose moves from the center of the sixth joint to the center of the end-effector tip. Abiding by the procedure, all desired locations are determined. The entire NC codes are created thereby.

In Fig. 9, an illustration is prepared to show the movements and geometric positions of the arranged path with a 10 mm end-effector on the robotic arm. The off-line path consists of eight significant positions are provided to verify the transformation between Cartesian location inputs and joint NC outputs. In the proposed methodology, joint NC format uses the machine coordinate system as the reference and Cartesian location format is referenced to the work coordinate system. By equipping a 10 mm long end-effector, Fig. 10 shows the transformation of six joint angles on position 1, where the Y coordinate value is zero and maintain the

Table 3 D-H parameters table of the proposed six-axis robotic arm

Link	b_i	θ_i	a_i	α_i	Limits		Axis
0/1	258	0	0	0	0°	0°	n/a
1/2	97.5	θ_1	0	-90	+360°	-360°	R
2/3	0	$-90 + \theta_2$	500	0	+120°	-120°	R
3/4	0	θ_3	120	-90	90°	-270°	R
4/5	494	θ_4	0	90	+200°	-200°	R
5/6	80	θ_5	0	-90	+200°	-200°	R
6/7	70 + 10	θ_6	0	0	+360°	-360°	R

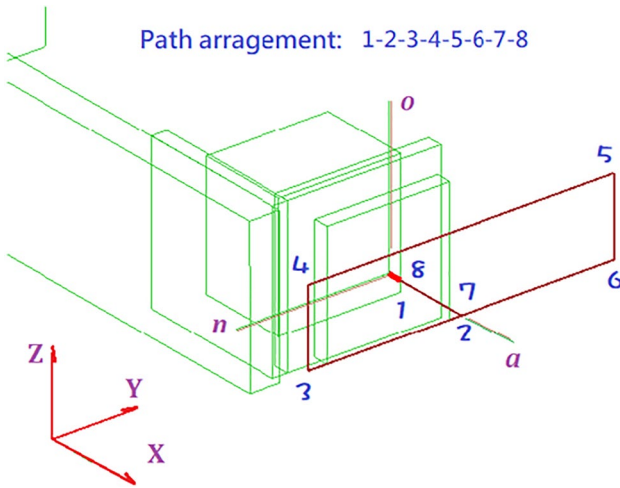


Fig. 9 Test path for the proposed robotic arm

same orientation as home position, whereas Fig. 11 shows the orthographic views of home position and position #1 (220, 0, 830). By substituting the corresponding parameters in Table 3 into (3), at the home position, the computed value of the distance from the end-effector tip to the center of

the bottom face of the base performs (574, 0, 975.5). This is verified by geometric measurement of the robotic arm. The geometric measurement of the robotic arm verifies the coordinates, and Fig. 12 shows the locations of the home position.

There are two portions of the data combine together to form the robot posture, end-effector tip positions and directions in the work coordinate system. The following verification will describe how a pre-defined target point transformed to the end-effector tip position. From the illustration in Fig. 11, position number one hold the end-effector location data (x, y, z, n, o, a) , $(x, y, z) = (220, 0, 830)$ for position on X, Y, and Z axes. The direction is formed by three unit vector. $n = [0, 0, 1]$ is the normal vector, $o = [0, -1, 0]$ is the orientation vector, and $a = [1, 0, 0]$ is the approach vector. According to the proposed algorithm, the normal, orientation, and approach vectors have to be refined as the input variables before implementing the DH transformation. In contrast to end-effector tip location data, NC commands for six-axis robotic arm uses coordinates of joint rotary angles represented in machine coordinate system. The relevant NC data of the above position for six-axis robotic arm is “MOVJ C1 = 0.0 C2 = -45.0281 C3 = 44.8982 C4 = 0.0 C5 =

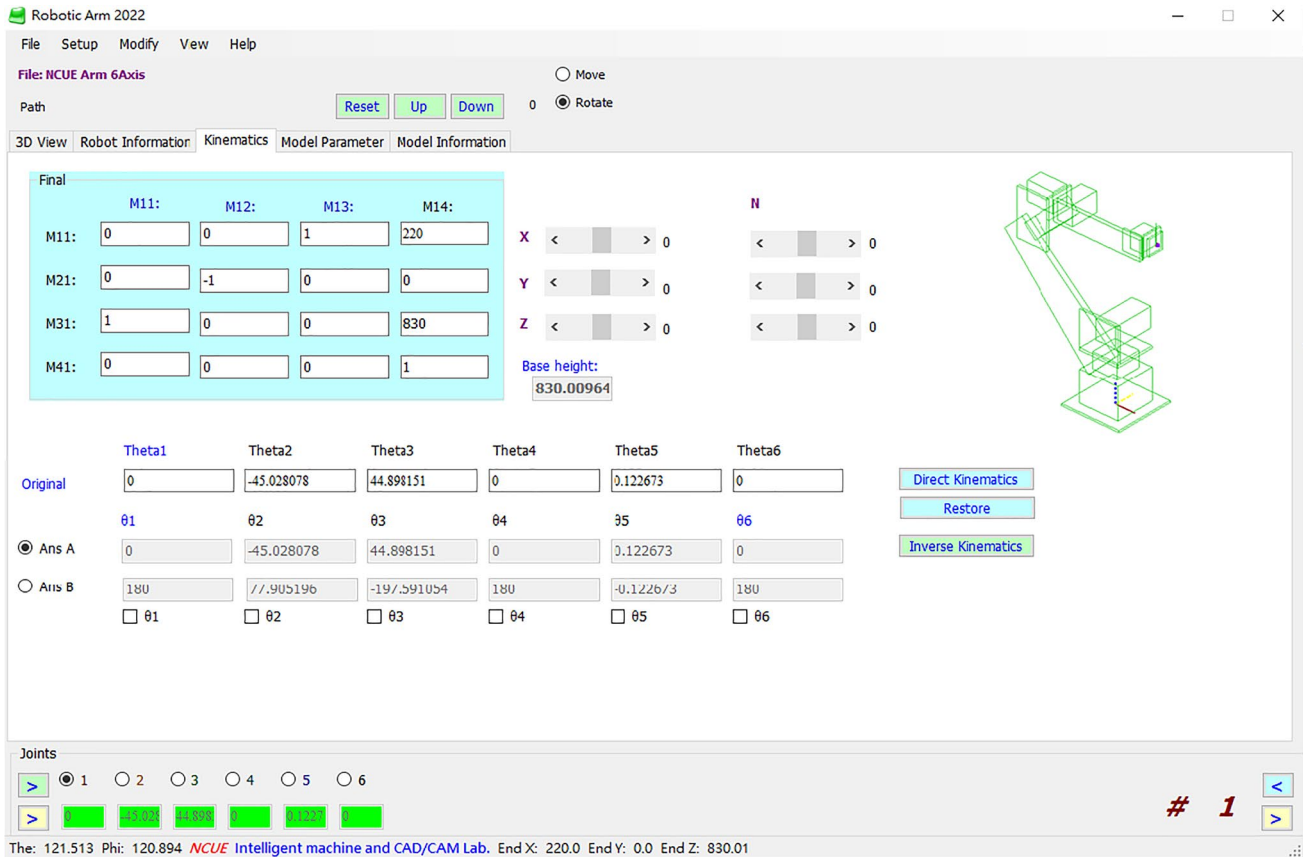
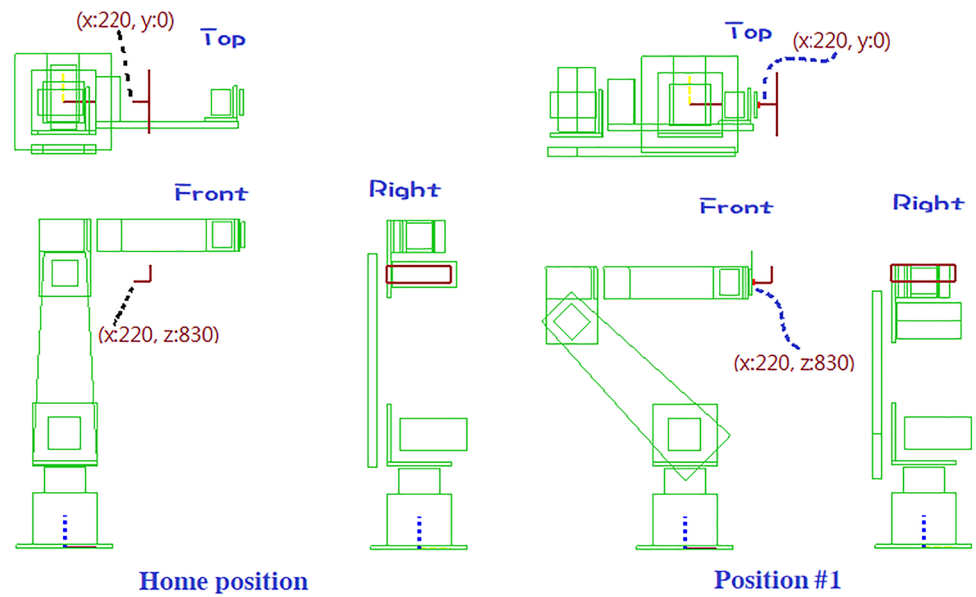


Fig. 10 Homogeneous matrix and transformation interface of the proposed system

Fig. 11 Extract the coordinate values on orthographic views



0.1227 C6 = 0.0 FJ50 PL10.” The graphical user interface of the proposed system for the transformation information between the end-effector tip Cartesian point and the joint space NC data is illustrated in Fig. 10.

5 Implementation of simulation and verification

Since the proper desired path arises from the complicated computation in each step of the process as shown in Fig. 1. Therefore, all corresponding parameters must be examined to guarantee the validity of the overall task trajectory. The robot motion planning of desired locations relies on the calculation of an end-effector, which builds up the tasks based on Cartesian coordinate space applied on a computer-aided planning model. Drive the end-effector to the target postures. A specified task may serve as an example to evaluate the performance of the proposed system. For the purpose to verify the effectiveness and feasibility, the restricted end-effector must maintain the posture toward the normal direction of the side face of a hexahedron was designated. This path positioning strategy on this task is located in front of a box with a 10 mm long end-effector. Six-axis robotic arm positioning was applied on the specified CNC equipped with a Syntec 81R NC unit. The predefined desired targets are shown in Table 4. The final solution for all joints through the proposed inverse kinematic calculation is shown in Table 5. The third position is displayed as shown in Fig. 13 with a wireframe representation, and the total eight different target locations are demonstrated as shown in Fig. 14. The proposed solver was evaluated to show the effectiveness and performance of the specified robotic arm. The detail at each target location

of the end-effector was also shown in Fig. 15. Constructing the end-effector location conditions for the positioning task is a prerequisite; such constraint conditions are generated in the GUI system by using the proposed solver. Figures 16 and 17 illustrate the positioning locations and path NC codes, and Fig. 18 displays the simulated positioning movements. The demonstration shows the proposed kinematic solver can be applied to practical industrial six-axis robotic arms successfully.

Figure 19 depicts the corresponding trajectories of joint angle for the default solution, i.e., trajectory for all the joints of the proposed robotic arm during its motion along desired target locations. A common typical robotic arm has three large links follows by three smaller links near the end-effector. The last three arms are used primarily to tune the attitude of the end-effector about the target. Given the desired Cartesian description of end-effector within reachable workspace, there can be up to eight different possible options to approach the destination. In this proposed case,

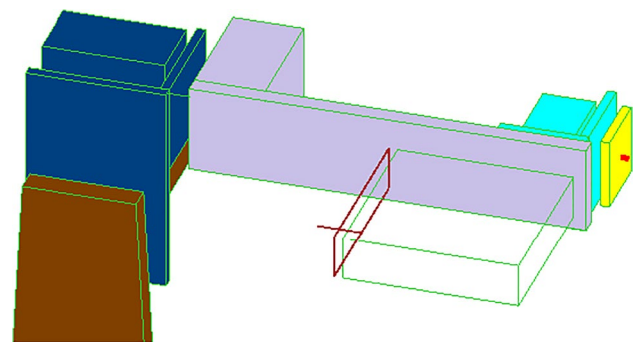


Fig. 12 Illustration of the test locations about home position

Table 4 The pre-defined eight targets of the desired task

Value\point	#1	#2	#3	#4	#5	#6	#7	#8
X	220	270	270	270	270	270	270	220
Y	0	0	-100	-100	100	100	0	0
Z	830	830	830	880	880	830	830	830
Movement	Start	Forward	Right	Up	Left	Down	Right	Back

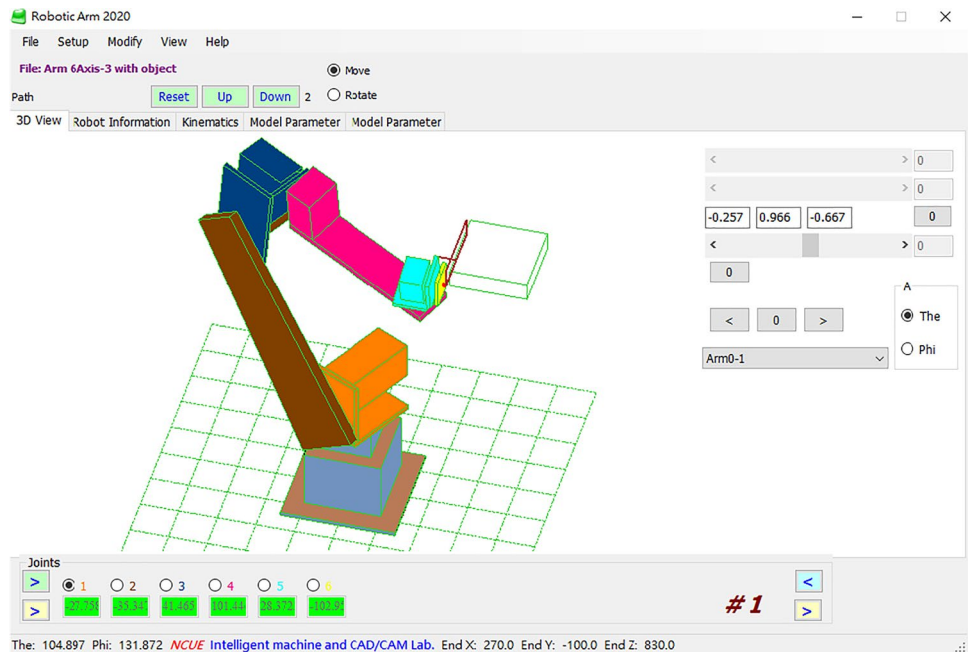
Table 5 Final solution for pre-defined targets with 10 mm long end-effector

Configura-tion	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1	0	-45.0281	44.8982	0	0.1227	0
2	0	-38.5330	42.7470	0	-4.2141	0
3	-27.7585	-35.3474	41.4651	101.4468	28.3723	-102.9598
4	-27.7585	-34.2431	35.2649	91.9407	27.7759	-92.1932
5	27.7585	-34.2431	35.2649	88.0593	-27.7759	-87.8068
6	27.7585	-35.3474	41.4651	78.5532	-28.3723	-77.0402
7	0	-38.5330	42.7470	0	-4.2141	0
8	0	-45.0281	44.8982	0	0.1227	0

weight is applied in consideration of determining the closest solution. The potential selection favors smaller joints rather than large joints to reduce moving loading as little as possible. The eight solutions for position #1 through its inverse kinematic calculation is shown in Table 6, and the corresponding postures are shown in Fig. 20. Follow the inverse kinematic solution tree as shown in Fig. 7, one can examine the behavior according to the configurations in Table 6. The

first joint provides two possible options, left (Fig. 20(a, b, e, and f)) or right (Fig. 20(c, d, g, and h)). The next two joints also provide two possible options, up (Fig. 20(a, b, c, and d)) or down (Fig. 20(e, f, g, and h)). These first three joints yield four different configurations with a combination of waist, shoulder, and elbow. Similarly, the last three joints can behave two different options, flip or not, with a combination of forearm, wrist pitch, and wrist roll. Equation

Fig. 13 The pre-defined third position with 10 mm long end-effector



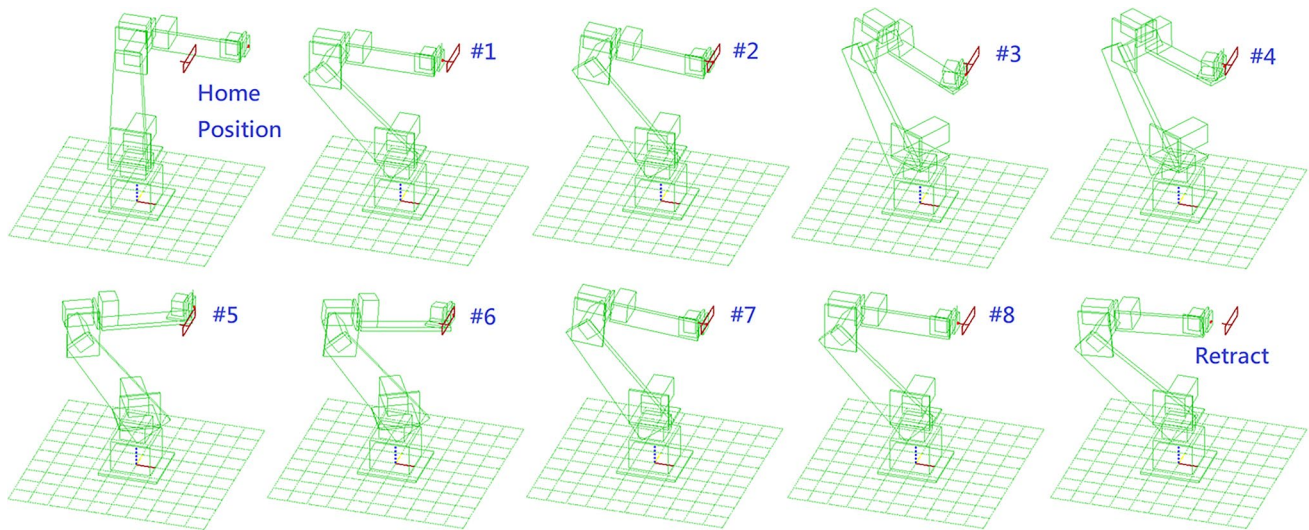


Fig. 14 Wireframe illustration of the pre-defined eight target locations

(58) shows the relationship between wrist orientations, flip or not. Equation (59) shows the relationship of θ_4 and θ_6 with flip or not according to Eq. (58). The eight solutions for position #3 through its inverse kinematic calculation is shown in Table 7.

$$\theta_{5(\text{Flip})} = -\theta_{5(\text{NoFlip})} \tag{58}$$

$$\theta_{4(\text{Flip})} = \theta_{4(\text{NoFlip})} \pm 180^\circ, \theta_{6(\text{Flip})} = \theta_{6(\text{NoFlip})} \pm 180^\circ \tag{59}$$

It is obviously suitable for quick positioning with automatic end-effector change system; a 100 mm long end-effector was changed to mount on the sixth link as shown in Fig. 21. Perform the same pre-defined targets are shown in Table 4. Compare to manual operations, the effects of the fast planning system are demonstrated by post-processing Cartesian desired locations for multi-axis paths. Incorporating the position refinement, an advanced postprocessor is presented. Figure 22 reveals a schematic flowchart of the proposed fast planning system. Under the implementation, after all the original end-effector locations are confirmed, the smoothness and connection conditions are detected. The final solution for all joints through the proposed inverse kinematic calculation is shown in Table 8. The corresponding trajectories of joint angle for the default solution, i.e., trajectory for all the joints of the proposed robotic arm during its motion along desired target locations is shown in Fig. 23.

In multi-axis CNC articulated robotic arm positioning, it is often causing singularities while equips propelling

axes with more degrees of freedom. Such an error occurs when the axes of joint 4 and joint 6 turn into coincident. This singularity is quite commonly encountered in end-effector traveling with the condition corresponds to $\theta_5 = 0$. At the wrist singularity, the wrist pitch angle crosses zero position and the end-effector remains motionless while the wrist turns about the coaxial axes of joint 4 and joint 6. Owing to numerical problems, if the end-effector cannot cross the singularity without stopping, it is also impossible for the end-effector to do this crossing in Cartesian space. In such a situation, the inverse kinematics-based post-processing module plays a very important role in quick off-line path planning. The module dominates the transformation from Cartesian space to joint space to generate a more robust positioning path. The difficulties are associated with solving processes for forward and inverse kinematics transformation. Passing through such singular locations using common strategies may result in disconnected trajectories, causing a delay in the real executing feed rate during the positioning path. Inevitably affect processing quality and production efficiency, reduced competitiveness may become unavoidable. Although the rearrangement of the current sequence to appropriate paths is the easiest way to avoid a singular situation, the better way is to create the suitable joint space movements for each propelling axis robustly and smoothly. The above tasks may not be accomplished in Cartesian space, because some configurations will block the Cartesian movements of the end-effector. Although it is intuitive to give Cartesian space motion commands, it may impede the control calculation. The

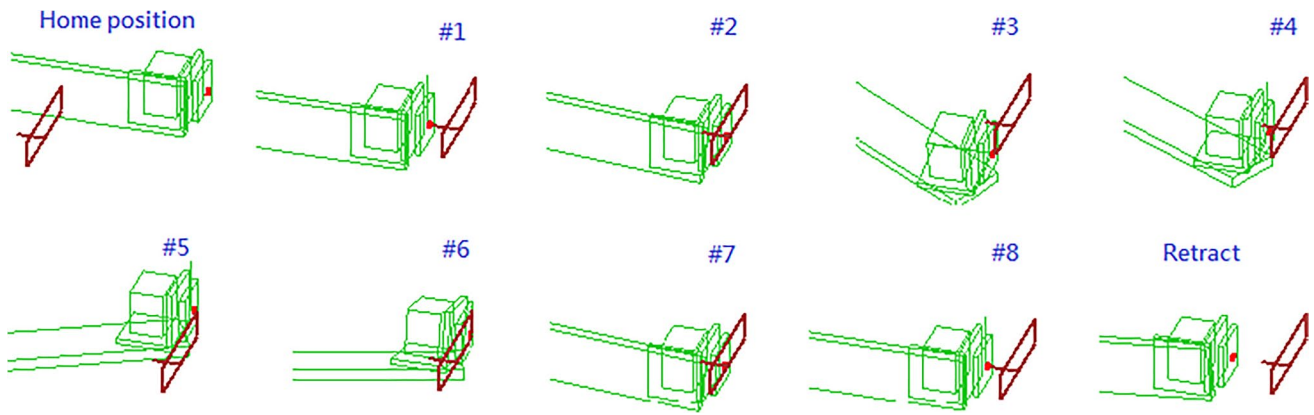


Fig. 15 Detailed illustration of the eight target locations

Fig. 16 Positioning locations of the proposed six-axis robotic arm



Fig. 17 A brief script of the NC commands of the six-axis articulated robot

```
G90;
MOVJ C1=0.0 C2=-45.0281 C3=44.8982 C4=0.0 C5=0.1227 C6=0.0 FJ50 PL10;
MOVJ C1=0.0 C2=-38.533 C3=42.747 C4=0.0 C5=-.2141 C6=0.0 FJ50 PL10;
MOVJ C1=-27.7585 C2=-35.3474 C3=41.4651 C4=101.4468 C5=28.3723 C6=-102.9598 FJ50 PL10;
MOVJ C1=-27.7585 C2=-34.2431 C3=35.2649 C4=91.9407 C5=27.7759 C6=-92.1932 FJ50 PL10;
MOVJ C1=27.7585 C2=-34.2431 C3=35.2649 C4=88.0593 C5=-27.7759 C6=-87.8068 FJ50 PL10;
MOVJ C1=27.7585 C2=-35.3474 C3=41.4651 C4=78.5532 C5=-28.3723 C6=-77.0402 FJ50 PL10;
MOVJ C1=0.0 C2=-38.533 C3=42.747 C4=0.0 C5=-.2141 C6=0.0 FJ50 PL10;
MOVJ C1=0.0 C2=-45.0281 C3=44.8982 C4=0.0 C5=0.1227 C6=0.0 FJ50 PL10;
MOVJ C2=0.0 C3=0.0 C5=0.0 FJ200 PL10;
M30;
```


Fig. 18 The proposed experimental six-axis robotic arm. First (left) and third (right) positions

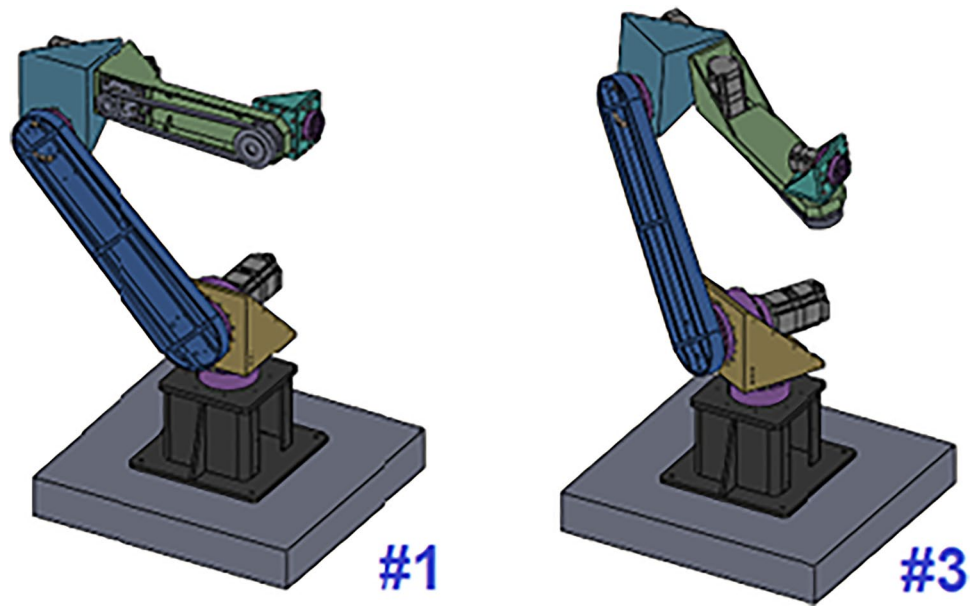


Fig. 19 Trajectories of joint angle with 10 mm long end-effector

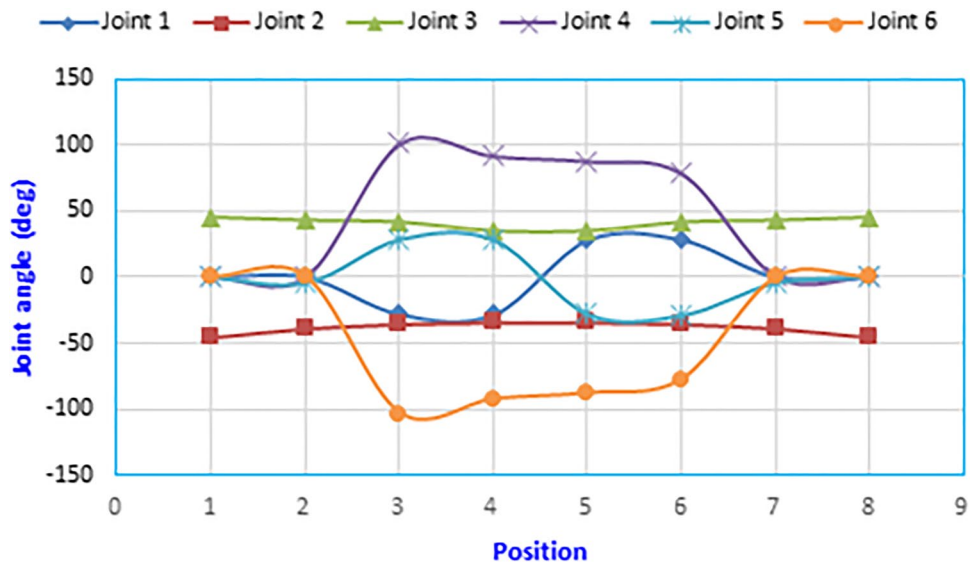


Table 6 All possible configurations for position #1

Configuration	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	U/D	L/R	N/F
1/a	0	-45.02808	44.898208	0	0.122673	0	Up	Left	No
2/c	0	-45.02808	44.898208	180	-0.122673	180	Up	Left	Flip
3/b	180	45.02808	-197.5910	180	27.437013	0	Up	Right	No
4/d	180	45.02808	-197.5910	0	-27.437013	180	Up	Right	Flip
5/e	0	77.90523	-197.5910	0	119.68586	0	Down	Left	No
6/g	0	77.90523	-197.5910	180	-119.68586	180	Down	Left	Flip
7/f	180	-77.90523	44.898208	180	146.99295	0	Down	Right	No
8/h	180	-77.90523	44.898208	0	-146.99295	180	Down	Right	Flip

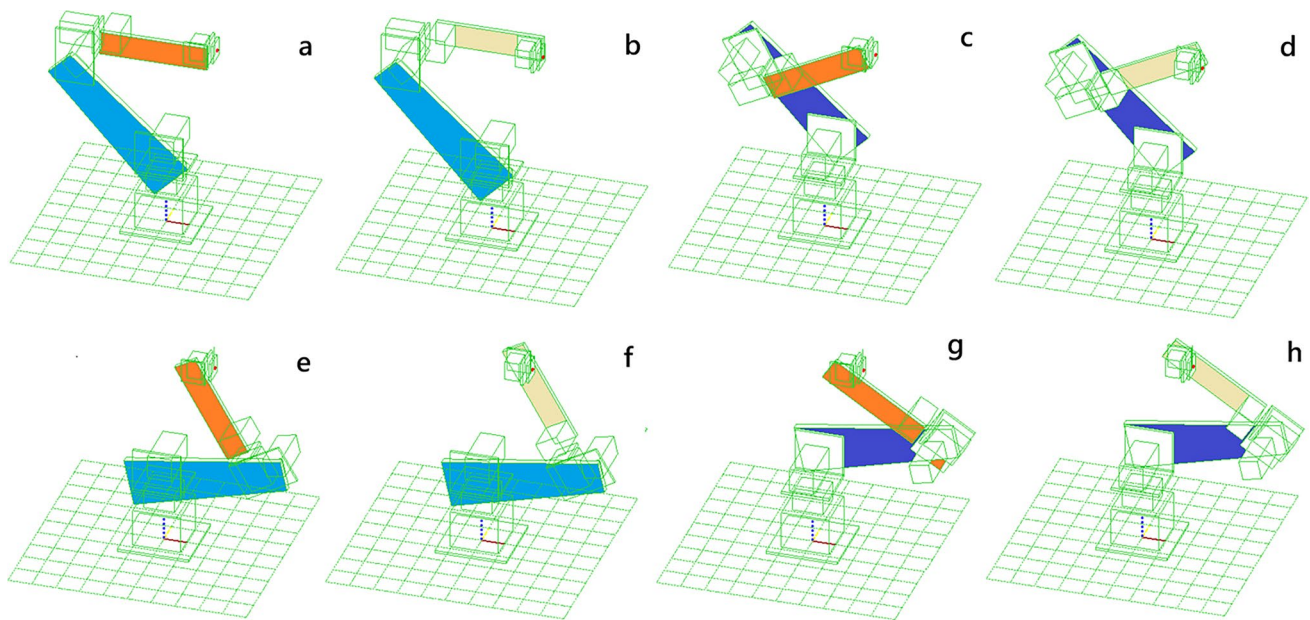


Fig. 20 Postures for the eight potential solutions of the 6-DOF robot at #1 position

Table 7 All possible configurations for position #3

Config-uration	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	U/D	L/R	N/F
1	-27.7585	-35.3474	41.4651	101.4468	28.3723	-102.9598	Up	Left	No
2	-27.7585	-35.3474	41.4651	-78.5532	-28.3723	77.0402	Up	Left	Flip
3	152.2415	35.3474	-194.158	55.5203	-34.4024	129.7721	Up	Right	No
4	152.2415	35.3474	-194.158	-124.4797	34.4024	-50.2279	Up	Right	Flip
5	-27.7585	84.0405	-194.158	29.2712	107.72	9.6815	Down	Left	No
6	-27.7585	84.0405	-194.158	-150.7288	-107.72	-170.3185	Down	Left	Flip
7	152.2415	-84.0405	41.4651	37.8804	-130.6656	-153.1181	Down	Right	No
8	152.2415	-84.0405	41.4651	-142.1196	130.6656	26.8819	Down	Right	Flip

posture of the end-effector determined by the kinematic mapping in terms of the joint variables may often lead to large joint velocities and restrict instantaneous mobility. The significant advantage of using the Joint space motion command is that the only time the robot will suspend is when a joint hits the limit or a mechanical interference occurs. The appropriate implementation recommended is to provide the desired set of joint variables to propel each joint to the desired position simultaneously and linearly.

Programming industrial robots can be tedious and monotonous. Efficiency may be improved by removing these steps from the manufacturing process. The fact is that a lot of experience and expert knowledge in the domain of robotics is required to realize even trivial operations. In order to save costs, reduce mistakes, and increase

efficiency, a novel robotic task assistant system was developed to allow non-experts in the domain of robotics to program robots to cope with numerous changeable tasks quickly. After applying the proposed system, quick path planning can be implemented to achieve the expected efficiency and robustness by specifying the position and orientation in a limited time. Once a better path is identified, a higher performance operating situation can be obtained, and a lot of operating time can be saved. The presented algorithm can realize auto-programming of the positioning path of the desired tasks and is easy to be customized for industrial six-axis robotic arms.

A graphical user interface provides a set of tools to facilitate the operation of determining desired locations and calibrating suitable orientations. Within the

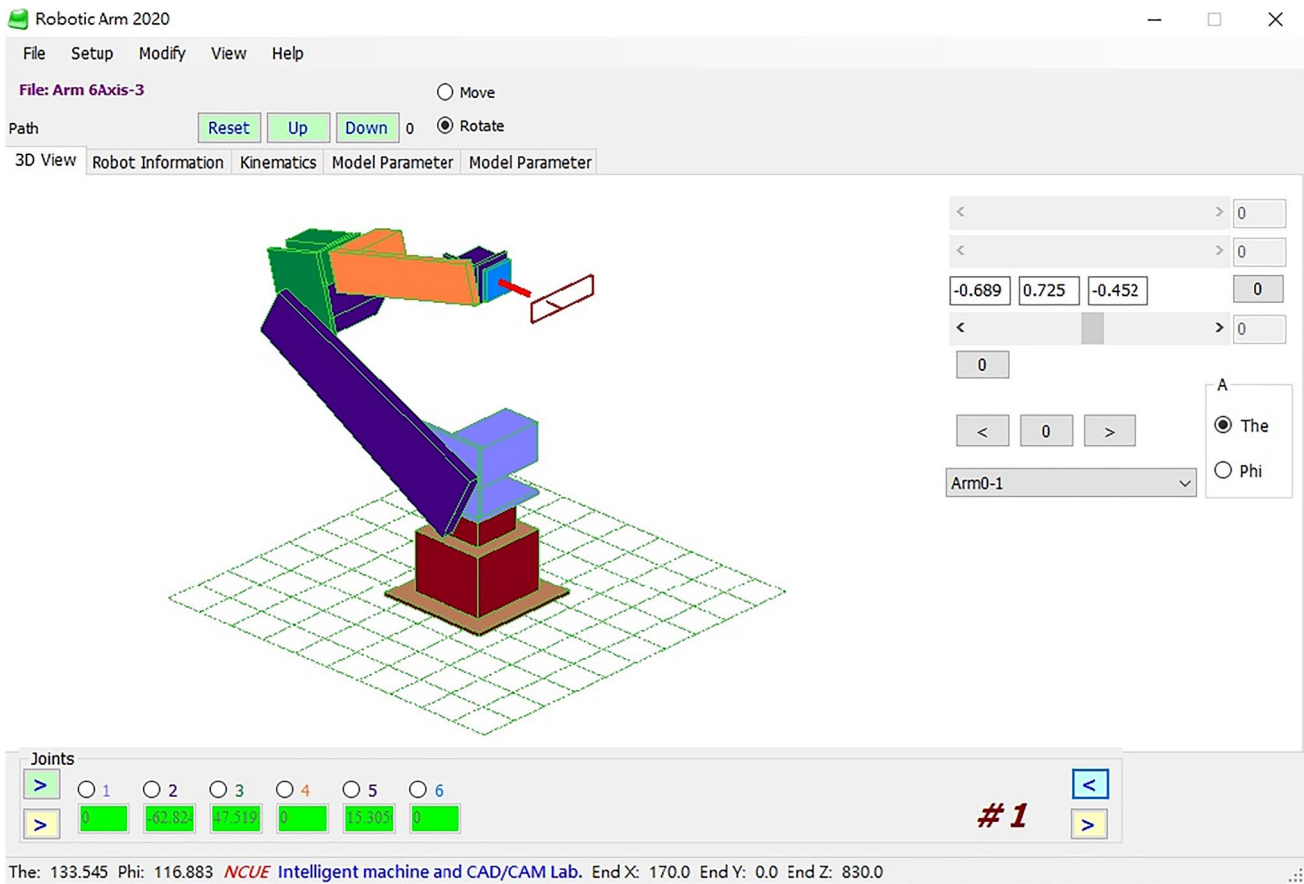


Fig. 21 The pre-defined retract position with 100 mm long end-effector

proposed interface, there are two primary coordinate systems, global and local (i.e., base and end-effector). The two coordinate systems are distinguished by their colors in the GUI as illustrated in Fig. 24. The location of the origin and three primary axes always remain fixed. The global coordinate system is unchangeable throughout the implementation of any specified model. Each link in the model, including the end-effector, has a local coordinate system. The figure shows the orientation vectors displayed in this tab and visualizes the local coordinate system. The target position can be defined manually by typing in its coordinates, or by clicking the intended direction in the simulation space with the slider. The target location can be updated manually, or can be adjusted by clicking and dragging on one of the primary-axis or approach vectors. The location display will update automatically.

Similar to the previous task, a fast implementation with a different length of end-effector was demonstrated to generate the NC positioning path codes, a 207 mm long

gripper was changed to mount on the sixth link as shown in Fig. 25. First, the path planning system is employed to guide the robotic arm to reach a specified location by a visual user interface in a short time. The rest is to manipulate the end-effector to follow the pre-defined positions and maintain the original orientation as shown in Fig. 26. The positioning path is employed directly on the specified model in the working space, and the poses of the robotic gripper are then defined along the path and visualized. The corresponding trajectories of the joint angle for the default solution are shown in Fig. 27. The case shows that all feasible postures are connected to complete a desired task cycle in a quick operating procedure, even changing the gripper frequently.

This proposed module system is an open architecture, and users can add new robot models and parameters by themselves. However, a professional technique is required to develop a new post-processor if it is an industrial six-axis robotic arm with a new configuration. Postprocessors are the bridge between the offline programming software

Fig. 22 A schematic flowchart of the proposed system

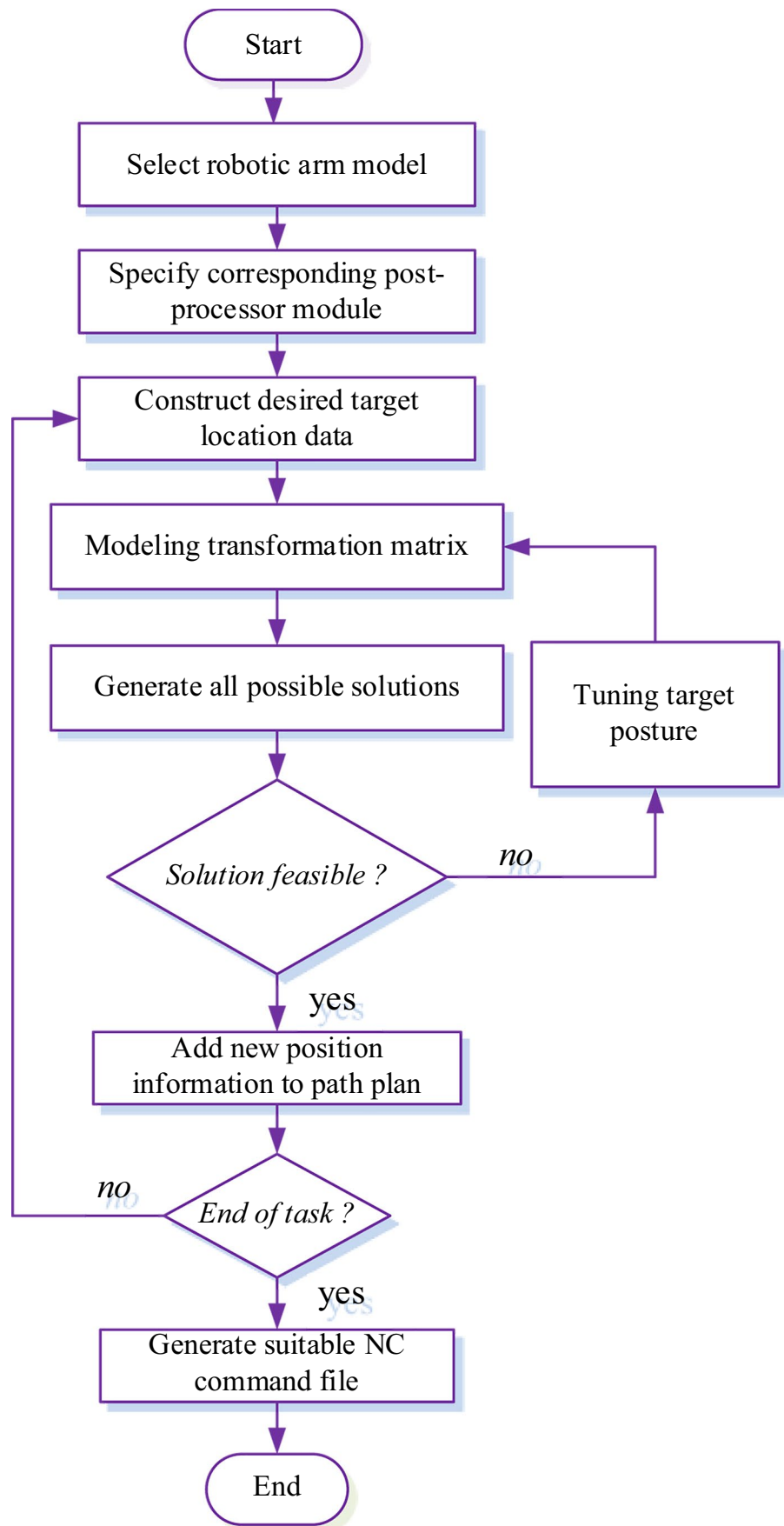
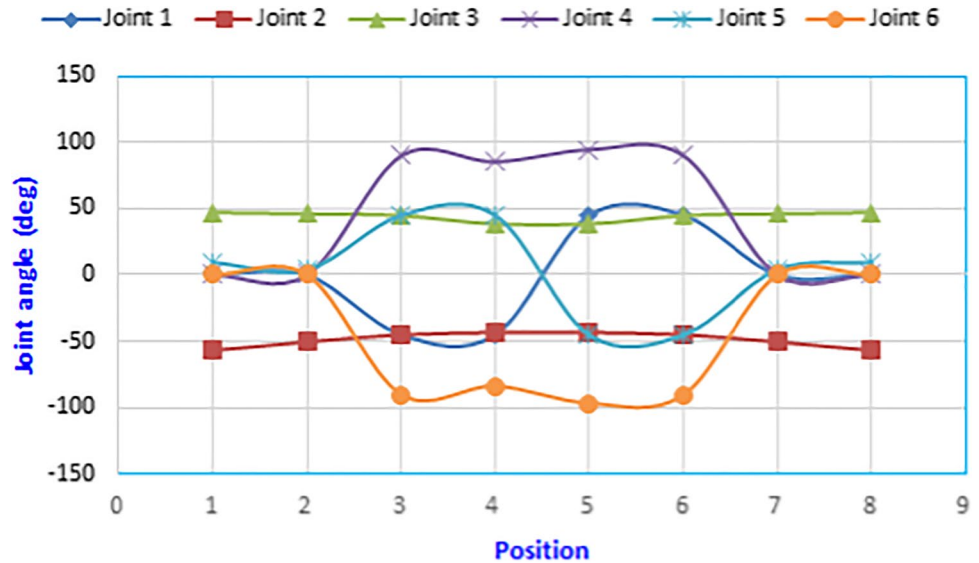


Table 8 Final solution for pre-defined targets with 100 mm long end-effector

Configuration	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1	0	-56.6333	47.1805	0	9.4528	0
2	0	-50.2253	46.1724	0	4.0529	0
3	-45	-44.8431	44.8454	90.0024	45	-90.0034
4	-45	-43.0598	38.466	85.4209	45.1838	-83.5173
5	45	-43.0598	38.466	94.5791	-45.1838	-96.4827
6	45	-44.8431	44.8454	89.9976	-45	-89.9966
7	0	-50.2253	46.1724	0	4.0529	0
8	0	-56.6333	47.1805	0	9.4528	0

Fig. 23 Trajectories of joint angle with 100 mm long end-effector



and the robotic arm. Once the right postprocessor is specified, the real robotic arm will move exactly as it did in the simulation. The presented manipulator is one of the models as an example to demonstrate the simulation for robot programmers to assess and predict the behavior of the robot and to verify and evaluate the process path planning. Other postprocessors can also be provided a complete process to solve IK in the proposed system as shown in Fig. 28.

6 Conclusion

The use of industrial robotic arms in automated applications is becoming increasingly common. They are more suitable than conveyors for moving objects from one place to another. The most difficult challenge is to develop a robotic arm that can traverse a trajectory over a long distance. Robotic arms are often used in dangerous and confined spaces; therefore,

long-distance control or programming of a robotic arm to quickly and safely pick up and manipulate objects is ideal solutions. In some circumstances, there is a need to customize an available postprocessor for a specific controller or application.

Not only requires knowledge and experience, but creating a postprocessor is a delicate and dedicated process. Without completely correct code that communicates with the controller, there is a risk of damage and failure. In this study, a novel postprocessor for an industrial six-axis robotic arm was proposed, with the aim of meeting the requirement for fast and robust computations of inverse kinematics. Despite the teaching mode being the most popular implementation in the robotic market, a tendency toward real-time postprocessing in the machine control units has become apparent. This has created a need for a trusted path planning strategy based on inverse kinematics that can facilitate complex tasks with a speedy and robust implementation. To precisely follow the programmed

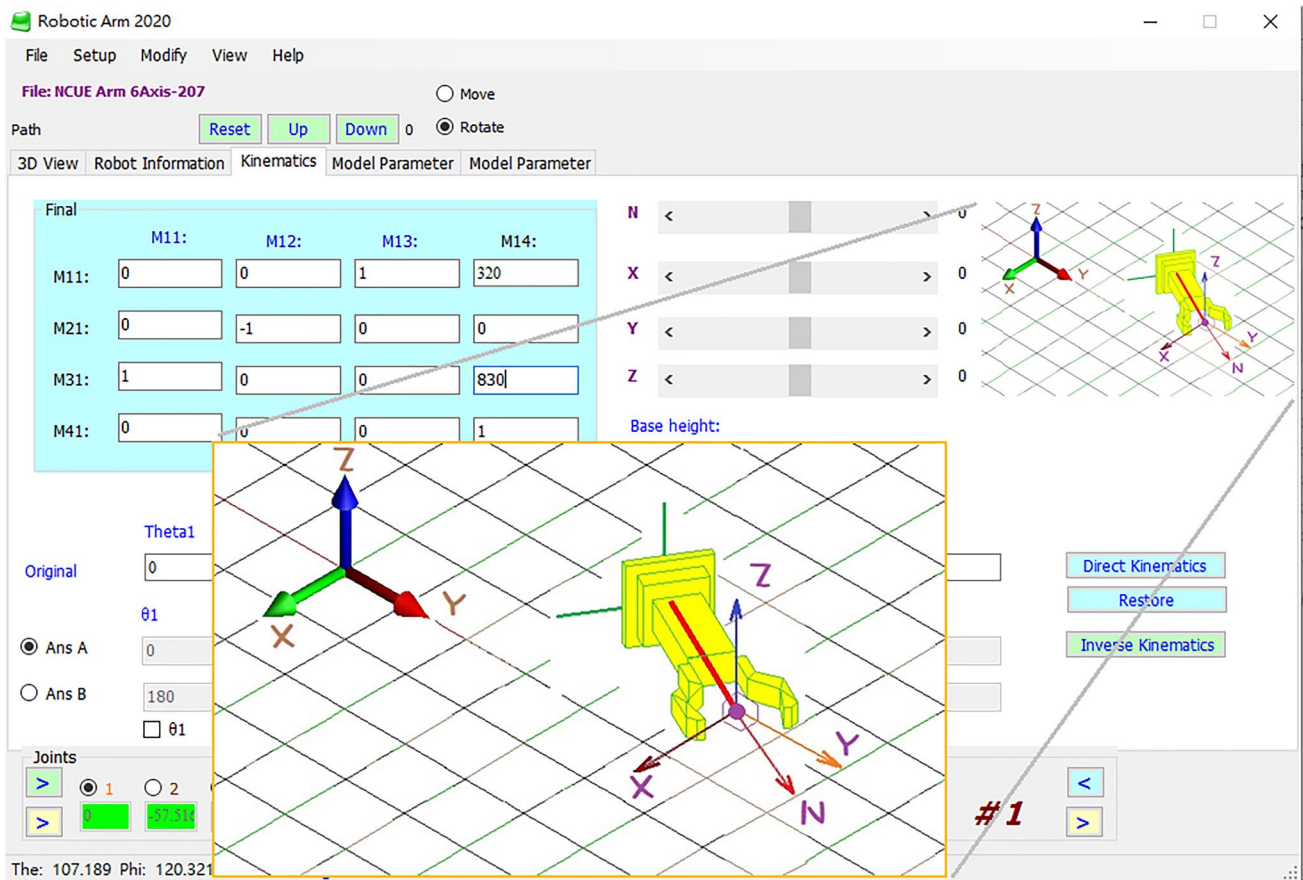


Fig. 24 The proposed GUI system for position modification

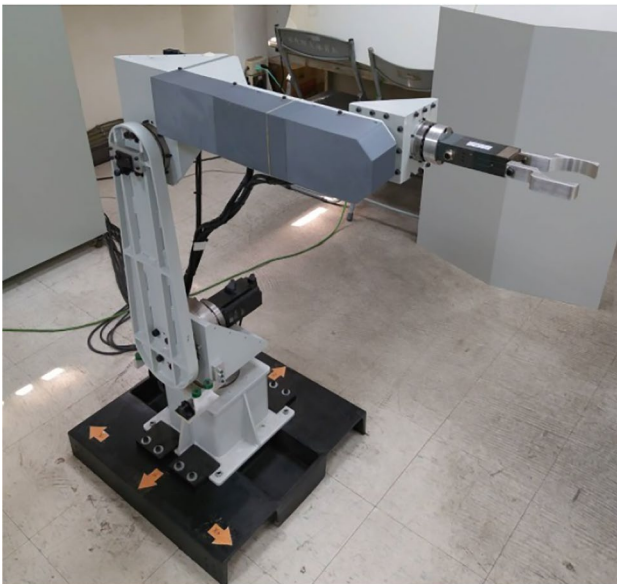


Fig. 25 The proposed six-axis robotic arm with 207 mm long gripper at home position

trajectories, the locations and directions of the end-effector must be accurately determined based on the feasible NC commands. The joint space NC codes were created by the proposed solver from the corresponding Cartesian coordinate target information. Each NC robotic arm is equipped with a unique postprocessor developed by using the geometric model. In this study, a methodology for six-axis articulated robotic arms based on the generalized kinematic model was proposed. To reduce task planning time and effort required to process changing conditions, a fast path planning system was developed and tested to meet the conditions of obstacles and path constraints. Additionally, an analytical procedure is presented that enables creating and adding new postprocessors more systematically and efficiently than before. Online robotic control systems can estimate the state of the real world based on signals received from their sensors, but the state of the real world is not always predicted accurately. Five-axis and six-axis computer numerical control (CNC) robotic arms must

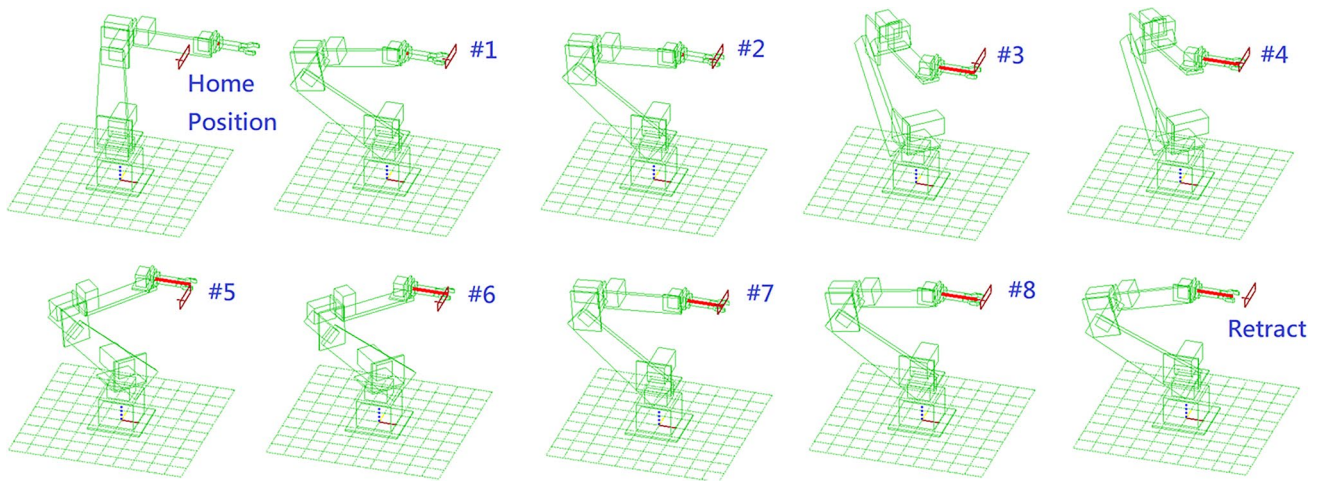
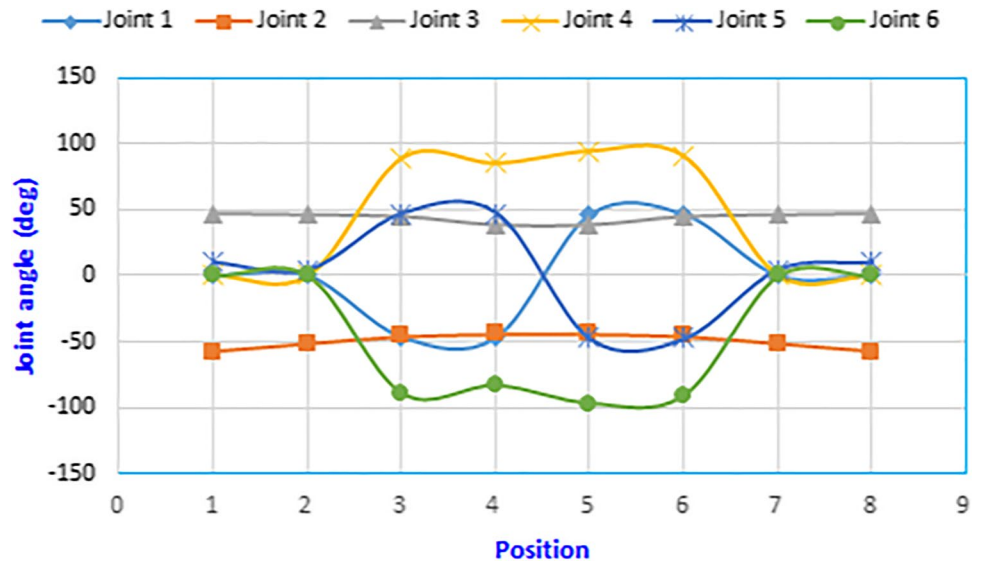


Fig. 26 Fast implementation with 207 mm long end-effector of the original eight target locations

Fig. 27 Trajectories of joint angle with 207 mm long end-effector



be equipped with postprocessors to maximize efficiency. This study presents a modular method for constructing a postprocessor system for a specific six-axis CNC robotic arm. This method is unique in its ability to create a post-processor architecture from the ground up and derive up to eight solutions simultaneously. Some limitations of the

proposed system are that it is not applicable for real-time tasks and is not available for parallel robots currently. The future work will focus on performance improvement techniques, such as heuristic and genetic algorithms, to optimize robot control tasks in the proposed system.

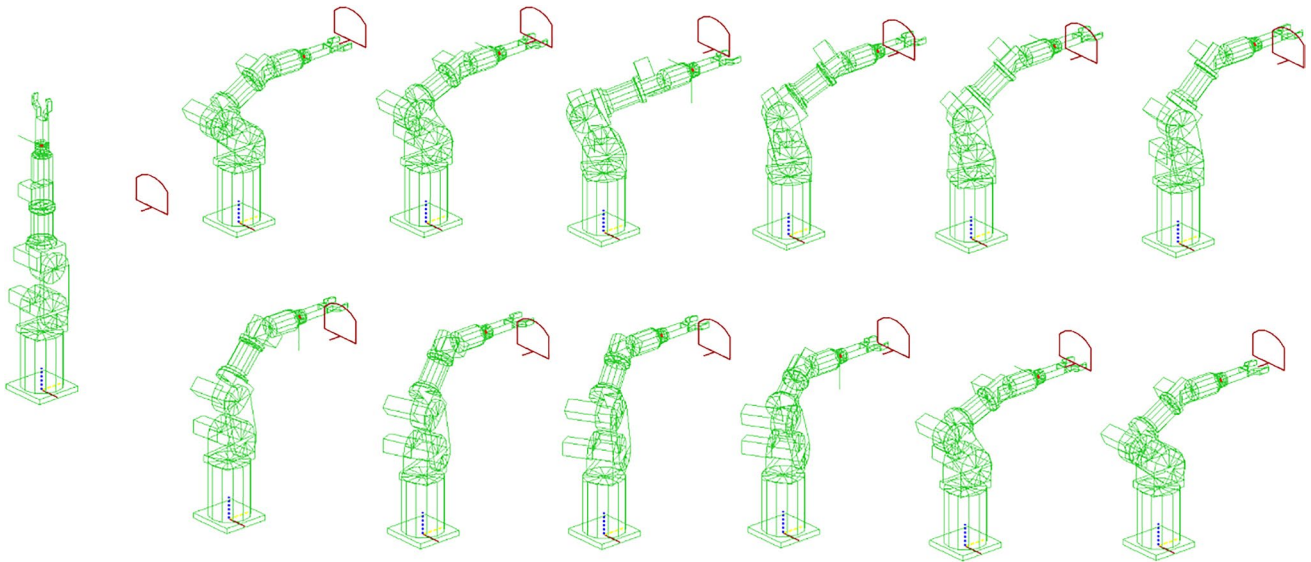


Fig. 28 Wireframe illustration of a pre-defined contour path

Authors' contributions YL Lai conceived, designed, and executed this study and wrote the manuscript. No other person is entitled to authorship.

Declarations

Competing interests The author declares no competing interests.

References

- Lin HI (2020) Design of an intelligent robotic precise assembly system for rapid teaching and admittance control. *Rob Comput Integr Manuf* 64:101946
- Benotsmane R, Kovács G, Dudás L (2019) Economic, social impacts and operation of smart factories in Industry 4.0 focusing on simulation and artificial intelligence of collaborating robots. *Soc Sci* 8:143
- Haddadin S, Albu-Schaer A, Hirzinger G (2009) Requirements for safe robots: measurements, analysis and new insights. *Int J Rob Res* 28(11/12):1507–1527
- Fu Z, Wenyu Y, Zhen Y (2013) Solution of inverse kinematics for 6r robot manipulators with offset wrist based on geometric algebra. *J Mech Robot* 5(3):0310081–0310087
- Liao Z, Jiang G, Zhao F, Mei X, Yue Y (2020) A novel solution of inverse kinematic for 6R robot manipulator with offset joint based on screw theory. *Int J Adv Robot Syst* 17(3):1–12
- Husty ML, Pfurner M, Schröcker HP (2007) A new and efficient algorithm for the inverse kinematics of a general serial 6R manipulator. *Mech Mach Theory* 42(1):66–81
- Wei Y, Jian S, Jian S, He S, Wang Z (2014) General approach for inverse kinematics of nR robots. *Mech Mach Theory* 75:97–106
- Doan NCN, Lin W (2017) Optimal robot placement with consideration of redundancy problem for wrist-partitioned 6R articulated robots. *Rob Comput Integr Manuf* 48:233–242
- Liu H, Zhang Y, Zhu S (2015) Novel inverse kinematic approaches for robot manipulators with Pieper-criterion based geometry. *Int J Control Autom Syst* 13(5):1242–1250
- Flanders M, Kavanagh RC (2015) Build-a-robot: using virtual reality to visualize the Denavit–Hartenberg parameters. *Comput Appl Eng Educ* 23:846–853
- Guzmán-Giménez J, Valera Fernandez A, Mata Amela V, Diaz-Rodríguez MA (2020) Synthesis of the inverse kinematic model of non-redundant open-chain robotic systems using Groebner Basis Theory. *Appl Sci* 10:2781
- Wang Z, Liu R, Sparks T, Chen X, Liou F (2018) Industrial robot trajectory accuracy evaluation maps for hybrid manufacturing process based on joint angle error analysis. *Adv Robot Autom* 7:183
- Ye H, Wang D, Wu J, Yue Y, Zhou Y (2020) Forward and inverse kinematics of a 5-DOF hybrid robot for composite material machining. *Rob Comput Integr Manuf* 65:101961
- Cho Y, Do HM, Cheong J (2019) Screw based kinematic calibration method for robot manipulators with joint compliance using circular point analysis. *Rob Comput Integr Manuf* 60:63–76
- Wang R, Wu A, Chen X, Wang J (2020) A point and distance constraint based 6R robot calibration method through machine vision. *Rob Comput Integr Manuf* 65:101959
- Lu L, Zhang J, Fuh JYH, Han J, Wang H (2020) Time-optimal tool motion planning with tool-tip kinematic constraints for robotic machining of sculptured surfaces. *Rob Comput Integr Manuf* 65:101969
- Xie F, Chen L, Li Z, Tang K (2020) Path smoothing and feed rate planning for robotic curved layer additive manufacturing. *Rob Comput Integr Manuf* 65:101967
- Xiong G, Ding Y, Zhu L (2019) Stiffness-based pose optimization of an industrial robot for five-axis milling. *Rob Comput Integr Manuf* 55:19–28
- Chen C, Peng F, Yan R, Li Y, Wei D, Fan Z, Z Tang Z, Zhu Z (2019) Stiffness performance index based posture and feed orientation optimization in robotic milling process. *Rob Comput Integr Manuf* 55: 29–40
- Ferreras-Higuero E, Leal-Munoz E, de Jalon JG, Chacón E, Vizán A (2020) Robot-process precision modelling for the improvement of productivity in flexible manufacturing cells. *Rob Comput Integr Manuf* 65:101966
- Pisla D, Plitea N, Gherman BG, Vaida C, Pisla A, Suciu M (2010) Kinematics and design of a 5-DOF parallel robot used

- in minimally invasive surgery. *Advances in robot kinematics: motion in man and machine*. Springer Science, Springer Netherlands, pp 99–106
22. Gherman B, Vaida C, Pisla D, Plitea N, Gyurka B, Lese D, Glogoveanu M (2010) Singularities and workspace analysis for a parallel robot for minimally invasive surgery. In: *Proceedings of the 2010 IEEE International conference on automation, quality and testing, Robotics, Cluj-Napoca, Romania, 28–30 May 1–6*
 23. Duguleana M, Grigore F, Teirelbar A, Mogan G (2012) Robotics and computer-integrated manufacturing obstacle avoidance of redundant manipulators using neural networks based reinforcement learning. *Robot Comput Integr Manuf* 28(2):132–146
 24. Agarwal V (2012) Trajectory planning of redundant manipulator using fuzzy clustering method. *Int J Adv Manuf Technology* 61:727–744
 25. Dereli S, Koker R (2020) Simulation based calculation of the inverse kinematics solution of 7-DOF robot manipulator using artificial bee colony algorithm. *SN Appl Sci* 2(27)
 26. Abdor-Sierra JA, Merchán-Cruz EA, Rodríguez-Cañizo RG (2022) A comparative analysis of metaheuristic algorithms for solving the inverse kinematics of robot manipulators. *Results Eng* 16:100597
 27. Larsen L, Kim J (2021) Path planning of cooperating industrial robots using evolutionary algorithms. *Robot Comput Integr Manuf* 67:102053
 28. Liu F, Huang H, Li B, Xi F (2021) A parallel learning particle swarm optimizer for inverse kinematics of robotic manipulator. *Int J Intell Syst* 36:6101–6132
 29. Karahan O, Karci H, Tangel A (2022) Optimal trajectory generation in joint space for 6R industrial serial robots using cuckoo search algorithm. *Intell Serv Robot* 15:627–648
 30. Lopez-Franco C, Diaz D, Hernandez-Barragan J, Arana-Daniel N, Lopez-Franco M (2022) A metaheuristic optimization approach for trajectory tracking of robot manipulators. *Mathematics* 10:1051
 31. Uchida T, McPhee J (2012) Using gröbner bases to generate efficient kinematic solutions for the dynamic simulation of multi-loop mechanisms. *Mech Mach Theory* 52:144–157
 32. Sanjuan J, Serje D, Pacheco J (2018) Closed form solution for direct and inverse kinematics of a us-rs-rps 2-DOF parallel robot. *Scientia Iranica* 25(4):2144–2154
 33. Zhang D, Xu Y, Yao J, Zhao Y, Chen IM (2017) Rotational axes and inverse kinematics analysis of a novel 5-DOF hybrid manipulator, *Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), IEEE International Conference on, IEEE*, pp 350–355
 34. Lai YL, Liao CC, Chao ZG (2018) Inverse kinematics for a novel hybrid parallel-serial five-axis machine tool. *Rob Comput Integr Manuf* 50:63–79
 35. Xie S, Sun L, Wang Z, Chen G (2022) A speedup method for solving the inverse kinematics problem of robotic manipulators. *Int J Adv Robot* 19(3)
 36. Husty M, Birlescu I, Tucan P, Vaida C, Pisla D (2019) An algebraic parameterization approach for parallel robots analysis. *Mech Mach Theory* 140:245–257
- Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.
- Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.