Hindawi Mathematical Problems in Engineering Volume 2021, Article ID 4516109, 11 pages https://doi.org/10.1155/2021/4516109



Research Article

Analysis of Open Architecture 6R Robot Forward and Inverse Kinematics Adaptive to Structural Variations

Chong Wang (b, 1 Dongxue Liu (b, 2 Qun Sun (b, 1 and Tong Wang (b) 1

¹School of Mechanical and Automotive Engineering, Liaocheng University, Liaocheng, China ²School of Mechanical and Automotive Engineering, Qilu University of Technology, Jinan, China

Correspondence should be addressed to Qun Sun; sunqun@lcu.edu.cn

Received 20 August 2020; Revised 7 January 2021; Accepted 5 March 2021; Published 16 March 2021

Academic Editor: Ioannis Kostavelis

Copyright © 2021 Chong Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a kinematic analysis for an open architecture 6R robot controller, which is designed to control robots made by domestic manufactures with structural variations. Usually, robot kinematic studies are often introduced for specific robot types, and therefore, difficult to apply the kinematic model from one to another robot. This study incorporates most of the robot structural variations in one model so that it is convenient to switch robot types by modifying model parameters. By combining an adequate set of parameters, the kinematic models, especially the inverse kinematics, are derived. The kinematic models are proved to be suitable for many classic industrial robot types, such as Puma560, ABB IRB120/1600, KAWASAKI RS003N/RS010N, FANUC M6iB/M10iA, and therefore be applicable to those with similar structures. The analysis and derivation of the forward and inverse kinematic models are presented, and the results are proven to be accurate.

1. Introduction

Kinematics analysis is one of the most important procedures for robot control. In the year 2018, the annual production volume of industrial robots reached 390 thousand; many of them are increasingly made by small or medium manufactures. A rising issue is that robots produced by different companies may have different structures, e.g., Figure 1; however, according to literature, robot kinematic models are often introduced for specific robot types rather than in a universal form. This leads to the fact that many robots with varying dimensional parameters must firstly derive a correct kinematic model in the controller.

For the purpose of reducing the inconvenience, this study aims to develop an open architecture controller in which the kinematic models are given a universal form so that they can apply to different robot types by changing parameters. The approach was based on the classic DH modelling method for well-known Puma560 analysis as presented in many text books [1]. This study introduced more link and offset parameters, which can be modified or set to zero for different robot structures, and the forward and

reverse solutions were rederived for this expended DH model. Compared with other analysis methods such as those based on screw theory, this approach is more understandable to general robot users and easy to implement in practice.

The Denavit–Hatenberg (DH) modelling approach is the most widely utilized method to analyze robot kinematics [1–11]. However, due to the variations in industrial robot structures as shown in Figure 1, many studies are associated with specific robot types [9–12] using specific relative link length and offset parameters, and therefore are difficult to adapt to different robot structures.

Today the large majority of industrial 6R robots are equipped with orthogonal wrists, where the axes of the last three joints intersect on one point. A distinctive example is a UR-5 robot that uses offset wrist, in which case joint 4 and joint 6 are parallel and vertical to joint 5. For common industrial 6R robots, as can be represented by ABB IRB120/IRB1600 as shown in Figure 1, the major structural differences lie in the arm configurations, especially the offset distances of joint 2 and joint 3. Studies on Puma560 robot are most widely presented in university textbook [1] and discussed in the literature [2–4, 11]. However, Puma560 is

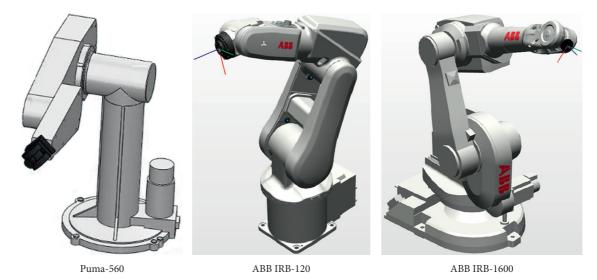


FIGURE 1: Variations in industrial robot structures.

somehow different from a normal industrial 6R robot since its link 3 is offset and the axis is not aligned with the axis of joint 1. For ABB IRB1600, the offset exists in link 2 and therefore, it cannot use the models for Puma560 and IRB120.

In terms of DH modelling, Puma560 is the most widely used example. The DH modelling approach takes robot section length, joint mounting angle, and offset distance to create a transformation matrix, from which the end point position can be calculated in forward kinematics, and on the contrary, the joint angles can be determined by known end status from inverse kinematics. More recent academic studies on robot kinematics are often based on screw theory [13–15], which provides better solutions, but are difficult to be utilized by domestic robot manufactures. For the robot controllers with fast CPUs, numerical solutions are always possible, and therefore, the studies on numerical approach are also well documented [16–18]. This study aims to develop open architecture kinematic models and controllers for small or medium robot manufactures, and therefore DH modelling remains in focus.

Robot dynamics is an important area besides analysis of kinematics, and relative studies are plenty [19–21], among which the Lagrange method is the most utilized approach. This study validates the robot on a controller developed on ARM and DSP, and control signals go through Panasonic A6 series motor drivers to drive motors. Since A6 drivers can be configured to use pulse mode and current control mode, and in the former case, it uses the internal PID controller to cancel out loading effects. For simplification purposes, this study uses pulse control mode in experiments and therefore, dynamics modelling is not required.

2. DH Models and Forward Kinematics

From the above investigations, a general 6R robot DH model should take adequate offset and length parameters into account, while the joint orthogonal status can be configured by one set of parameters, provided the input joint rotation angles can be adjusted to a specific robot type where necessary. Based on such premise, the DH parameters are listed in Table 1, where α stands for the joint orthogonal angle, a for link length, and d for link offset. Note the coordinate orientations in this form may be different from the definitions made by some original companies; however, the differences can always be cancelled out using some extra coordinate transformations after the model calculation.

Following this arrangement, the DH model diagram for the general 6R robot can be established as shown in Figure 2. It can be seen for Puma560 and ABB IRB120 that a_1 is zero. Thus, the axes of joint 1 and joint 2 intersect instead of being away. For Puma560, d_2 and d_3 both are present but are unequal, while for ABB IRB1600, d_2 and d_3 are identical, although its link appears to be on one side.

Besides, for industrial 6R robots, the axes of the last three joints intersect on a single point p, while the actual geometrical center of the robot end without considering tool apparatus is point p', which represents the genuine TCP of the robot. In many university textbooks and works of the literature, for Puma560 modelling, the calculations were often evaluated on point p. In many actual applications or simulations, such as in ABB Robot Studio, the evaluations can only be made on the genuine TCP p'. For later comparison purposes, this study presents results for Puma560 on point p, while for other robots, the results on p' will be evaluated. The origin of the coordinate systems was made on point O_0 , without considering the base height h_0 , which is included in Table 2 but not actually used in calculations.

The complete sets of parameters for some different robot types are given in Table 2. Note for Puma560, d_6 is taking 0 only because the evaluation is made on point p as mentioned above. Also, note apart from Puma560, d_2 and d_3 are always identical, which indicates for most industrial 6R robots, link 3 and the axis of joint 1 are in the same plane. Many robot DH models do not include d_2 and d_3 since the effects can be cancelled out. This study keeps all the parameters in the model in case variations are made by different manufactures.

Based on the DH model diagram in Figure 2, the forward DH transformation matrixes can be described in equations

Table 1: DH parameter	s of general 6R robots.
α_{i-1}	a_{i-1}

Link i	α_{i-1}	a_{i-1}	d_i
1	0°	0	0
2	-90°	a_1	$d_2 - d_3$
3	0°	a_2	$-d_3$
4	-90° 90° -90°	a_3	d_4
5	90°	0	0
6	-90°	0	d_6

(1) and (2), where T_{mn} stands for transformation from the mth coordinate system to the nth coordinate system.

$$T_{01} = T_{01}, T_{02} = T_{01}T_{12}, \dots, T_{06}$$

= $T_{01}T_{12}T_{23}T_{34}T_{45}T_{56},$ (1)

$$\mathbf{T}_{01} = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ \mathbf{T}_{12} = \begin{bmatrix} c_2 & -s_2 & 0 & a_1 \\ 0 & 0 & 1 & d_2 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ \mathbf{T}_{23} = \begin{bmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 1 & d_4 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ \mathbf{T}_{34} = \begin{bmatrix} c_4 & -s_4 & 0 & a_3 \\ 0 & 0 & 1 & d_4 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ \mathbf{T}_{45} = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ \mathbf{T}_{56} = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The matrix product of the above leads to the transformation from the origin to the last coordinate system of the robot, which can be described below, where n, o, and a correspond to the normal vector, the orientation vector, and the approaching vector.

$$\mathbf{T}_{06} = \mathbf{T}_{01} \mathbf{T}_{12} \mathbf{T}_{23} \mathbf{T}_{34} \mathbf{T}_{45} \mathbf{T}_{56} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$
(3)

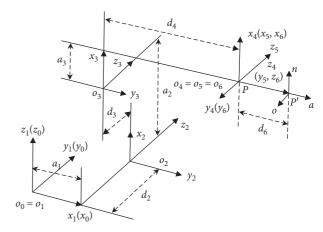


FIGURE 2: DH model diagram for general 6R robot.

TABLE 2: DH parameters for some different robot types.

Robot type	a_1	a_2	a_3	d_2	d_3	d_4	d_6	h_0
Puma560	0	431.8	20.32	149.09	0	433.07	0*	400
ABB IRB120	0	270	70	90	90	302	72	187
ABB IRB1600	150	475	108	215	215	600	65	486.5
FANUC M6iB	150	600	100	140	140	615	100	450
FANUC M10iA	150	600	200	207	207	640	100	450
Kawasaki RS003N	120	250	0	0	0	250	80	250
Kawasaki RS010N	100	650	0	207	207	700	88	430

In a robot control system based on a Python environment, the forward kinematic model can be directly calculated using matrix multiplication functions. In a different coding system, the equations might have to be configured manually. To simplify the trigonometric function representations, use symbols s and c in place of sin and cos and define $c_{23} = c_2c_3 - s_2s_3$, $s_{23} = c_2s_3 + s_2c_3$. Then, the equations of the forward kinematic model can be written as below.

$$n_{x} = c_{1} \left[c_{23} \left(c_{4}c_{5}c_{6} - s_{4}s_{6} \right) - s_{23}s_{5}c_{6} \right] + s_{1} \left(s_{4}c_{5}c_{6} + c_{4}s_{6} \right),$$

$$n_{y} = s_{1} \left[c_{23} \left(c_{4}c_{5}c_{6} - s_{4}s_{6} \right) - s_{23}s_{5}c_{6} \right] - c_{1} \left(s_{4}c_{5}c_{6} + c_{4}s_{6} \right),$$

$$n_{z} = -s_{23} \left(c_{4}c_{5}c_{6} - s_{4}s_{6} \right) - c_{23}s_{5}c_{6},$$

$$o_{x} = c_{1} \left[c_{23} \left(-c_{4}c_{5}s_{6} - s_{4}c_{6} \right) + s_{23}s_{5}s_{6} \right] - s_{1} \left(s_{4}c_{5}s_{6} - c_{4}s_{6} \right),$$
of the
$$o_{z} = s_{1} \left[c_{23} \left(-c_{4}c_{5}s_{6} - s_{4}c_{6} \right) + s_{23}s_{5}c_{6} \right] + c_{1} \left(s_{4}c_{5}s_{6} - c_{4}s_{6} \right),$$
of the
$$o_{z} = s_{23} \left(c_{4}c_{5}s_{6} + s_{4}c_{6} \right) + c_{23}s_{5}s_{6},$$

$$a_{x} = -c_{1} \left(c_{23}c_{4}s_{5} + s_{23}c_{5} \right) - s_{1}s_{4}s_{5},$$

$$a_{y} = -s_{1} \left(c_{23}c_{4}s_{5} + s_{23}c_{5} \right) + c_{1}s_{4}s_{5},$$

$$a_{z} = s_{23}c_{4}s_{5} - c_{23}c_{5},$$

$$p_{x} = c_{1} \left(a_{3}c_{23} - d_{4}s_{23} + a_{2}c_{2} + a_{1} \right) - s_{1} \left(d_{2} - d_{3} \right),$$

$$p_{y} = s_{1} \left(a_{3}c_{23} - d_{4}s_{23} + a_{2}c_{2} + a_{1} \right) + c_{1} \left(d_{2} - d_{3} \right),$$

$$p_{z} = -a_{3}s_{23} - d_{4}c_{23} - a_{2}s_{2}.$$

$$(4)$$

Since the output of T_{06} represents the position of the wrist intersection point p, while in many cases, users measure and evaluate the position of robot genuine TCP, i.e., point p', which should include parameter d_6 in the calculation. Without considering other robot tool configurations, the actual robot end point p' can be calculated using one more matrix transformation as follows:

$$\mathbf{T}_{6p'} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{T}_{0p'} = \mathbf{T}_{06} \mathbf{T}_{6p'} = \begin{bmatrix} n_x & o_x & a_x & p_x + a_x d_6 \\ n_y & o_y & a_y & p_y + a_y d_6 \\ n_z & o_z & a_z & p_z + a_z d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$
(5)

Clearly, when a specific tool is installed, the TCP position may be shifted by modifications in equation (5). For inverse calculation purposes, equation (5) should be firstly reversed to recover T_{06} so that the inverse analysis as presented in the following section may commence.

3. Inverse Kinematic Models

3.1. Calculate Joint 1 Angle θ_1 . The approach to calculate θ_1 is similar to that mentioned in the classic Puma560 model in the textbook [1]. In order to calculate θ_1 , it is necessary to construct an equation where the only unknown parameter is θ_1 , and all the other parameters are known constants. For this purpose, investigate the transformation matrix T_{16} , which can be obtained either by multiplying the inverse matrix of T_{01} with T_{06} , or by multiplying T_{12} , T_{23} , T_{34} , T_{45} , and T_{56} , as represented follows:

$$T_{01}^{-1}T_{06} = T_{12}T_{23}T_{34}T_{45}T_{56} = T_{16}, (6)$$

$$T_{01}^{-1}T_{06} = \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$
(7)

In the above, p_x and p_y are already known from equation (4). To calculate θ_1 , investigate the element (2, 4) of the matrix product in equation (7) and that of T_{16} , the below relationship can be established:

$$-s_1 p_x + c_1 p_y = d_2 - d_3. (8)$$

Differed from the Puma560 model, in equation (8), there is a parameter d_3 that is not used in Puma560 robot. To facilitate analysis, define the following functions:

$$p_x = \rho \cos \varphi,$$

 $p_y = \rho \sin \varphi.$ (9)

Substitute equation (9) into (8), then the solution of θ_1 can be obtained.

$$\theta_1 = a \tan 2(p_y, p_x) - a \tan 2[(d_2 - d_3), \pm \sqrt{p_x^2 + p_y^2 - (d_2 - d_3)^2}].$$
 (10)

3.2. Calculate Joint 3 Angle θ_3 . Letting the element (1, 4) and (3, 4) of matrix T_{16} and $T_{01}^{-1}T_{06}$ be identical, the below can be obtained:

$$c_1 p_x + s_1 p_y = a_3 c_{23} - d_4 s_{23} + a_2 c_2 + a_1, \tag{11}$$

$$-p_z = a_3 s_{23} + d_4 c_{23} + a_2 s_2. (12)$$

For equations (8), (11), and (12), sum up the squares of left and right sides, and the two sides should still be identical; therefore,

$$a_3c_3 - d_4s_3 = k, (13)$$

where *k* is as follows:

$$k = \frac{\left[p_x^2 + p_y^2 + p_z^2 + E - a_2^2 - a_3^2 - (d_2 - d_3)^2 - d_4^2\right]}{2a_2}.$$
 (14)

In the above equation, the term E is introduced due to the parameter a_1 , which is not present in the classic Puma560 model. From analysis the term E can be written as follows:

$$E = a_1^2 - 2a_1c_1p_x - 2a_1s_1p_y. (15)$$

Since equations (8) and (13) are in a similar format, θ_3 can be solved in a similar manner as follows:

$$\theta_3 = a \tan 2(a_3, d_4) - a \tan 2\left[k, \pm \sqrt{a_3^2 + d_4^2 - k^2}\right].$$
(16)

3.3. Calculate Joint 2 Angle θ_2 . To derive θ_2 , the method is new from the classic Puma560 model since it involves parameters a_1 and a_3 . The analysis is performed on the transformation matrix T_{26} , as follows:

$$T_{02}^{-1}T_{06} = T_{23}T_{34}T_{45}T_{56} = T_{26}. (17)$$

The principle of deriving θ_2 relies on searching for an equation where the unknown parameter is θ_2 while the rest parameters only include θ_1 and θ_3 , which are already worked out. Therefore, the above equation is rewritten in the following form:

$$T_{02}^{-1}T_{06} = \begin{bmatrix} c_1c_2 & s_1c_2 & -s_2 & -c_2a_1 \\ -c_1s_2 & -s_1s_2 & -c_2 & s_2a_1 \\ -s_1 & c_1 & 0 & -d_2^2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = T_{26}.$$

$$(18)$$

By looking into element (1, 4) of the above equation and T_{26} , the following relationship can be identified:

$$c_2(c_1p_x + s_1p_y - a_1) - s_2p_z = c_3a_3 - s_3d_4 + a_2.$$
 (19)

To help resolve equation (19), define the equations below:

$$A = c_1 p_x + s_1 p_y - a_1,$$

$$B = p_z,$$

$$C = c_3 a_3 - s_3 d_4 + a_2.$$
(20)

Then equation (19) can be transformed into the following form, which again has a similar structure as equation (8):

$$c_2 A - s_2 B = C. \tag{21}$$

Then, the final solution of θ_2 can be obtained as follows:

$$\theta_2 = a \tan 2(A, B) - a \tan 2\left(C, \pm \sqrt{A^2 + B^2 - C^2}\right).$$
(22)

3.4. Calculate Joint 4 Angle θ_4 . Similarly, to calculate θ_4 the method is to analyze the transformation matrix T_{36} , as follows:

$$T_{03}^{-1}T_{06} = T_{34}T_{45}T_{56}. (23)$$

By analyzing the (1, 3) and (3, 3) elements of the matrixes on the two sides, the following relationships can be identified:

$$a_x c_1 c_{23} + a_y s_1 c_{23} - a_z s_{23} = -c_4 s_5,$$

$$-a_x s_1 + a_y c_1 = s_4 s_5.$$
(24)

If the operational condition satisfies $s_5 \neq 0$, then there is a definite solution for θ_4 .

$$\theta_4 = a \tan 2(-a_x s_1 + a_y c_1, -a_x c_1 c_{23} - a_y s_1 c_{23} + a_z s_{23}).$$
(25)

If the condition is $s_5 = 0$, then the robot is in a singular posture. In this case, θ_4 can be taken with an arbitrary value that does not affect the end results.

3.5. Calculate Joint 5 Angle θ_5 . To derive θ_5 it is possible to investigate T_{46} as given by the following:

$$T_{04}^{-1}T_{06} = T_{45}T_{56} = T_{46},$$

$$T_{04}^{-1}T_{06} = \begin{bmatrix} c_{1}c_{23}c_{4+}s_{1}s_{4} & s_{1}c_{23}c_{4} - c_{1}s_{4} & -s_{23}c_{4} - a_{2}c_{3}c_{4+}d_{2}s_{4} - a_{3}c_{4} \\ -c_{1}c_{23}c_{4+}s_{1}s_{4} & -s_{1}c_{23}c_{4} - c_{1}s_{4} & s_{23}c_{4} & a_{2}c_{3}c_{4+}d_{2}s_{4} + a_{3}c_{4} \\ -c_{1}s_{23} & -s_{1}s_{23} & -c_{23} & a_{2}s_{3} - d_{4} \end{bmatrix} \begin{bmatrix} n_{x} & o_{x} & a_{x} & p_{x} \\ n_{y} & o_{y} & a_{y} & p_{y} \\ n_{z} & o_{z} & a_{z} & p_{z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= T_{46} = \begin{bmatrix} c_{5}c_{6} & -c_{5}s_{6} & -s_{5} & 0 \\ s_{6} & c_{6} & 0 & 0 \\ s_{5}c_{6} & -s_{5}s_{6} & c_{5} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$(26)$$

From the equality of elements (1, 3) and (3, 3) on the two sides, the following relationships can be derived:

$$a_{x}(c_{1}c_{23}c_{4} + s_{1}s_{4}) + a_{y}(s_{1}c_{23}c_{4} - c_{1}s_{4}) - a_{z}(s_{23}c_{4}) = -s_{5},$$

$$a_{x}(-c_{1}s_{23}) + a_{y}(-s_{1}s_{23}) + a_{z}(-c_{23}) = c_{5}.$$
(27)

These give the solution for θ_5 as follows:

$$\theta_5 = a \tan 2(s_5, c_5).$$
 (28)

3.6. Calculate Joint 6 Angle θ_6 . Finally calculating θ_6 requires working on T_{56} and the relationships derived from elements (3, 1) and (1, 1) are the following:

$$-n_{x}(c_{1}c_{23}s_{4}-s_{1}c_{4})-n_{y}(s_{1}c_{23}c_{4}+c_{1}s_{4})+n_{z}(s_{23}s_{4})=s_{6},$$

$$n_{x}[(c_{1}c_{23}c_{4}+s_{1}s_{4})c_{5}-c_{1}s_{23}s_{4}]$$

$$+n_{y}[(s_{1}c_{23}c_{4}-c_{1}s_{4})c_{5}-s_{1}s_{23}s_{5}]$$

$$-n_{z}(s_{23}c_{4}c_{5}+c_{23}s_{4})=c_{6}.$$
(29)

The above gives a unique solution for θ_6 as follows:

$$\theta_6 = a \tan 2(s_6, c_6).$$
 (30)

In addition, the symmetric solutions of the robot wrist angles do not affect end results. Therefore, there are extra solutions for θ_4 , θ_5 , and θ_6 as follows:

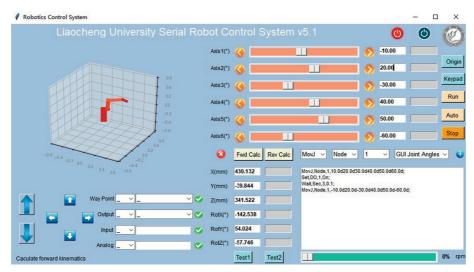


FIGURE 3: Example run on ABB IRB120 in the Python software.

Table 3: Forward kinematic	modelling results	s for some robot types.
PUMA560	ABB120	ABB120

	PUMA560	PUMA560	ABB120	ABB120	ABB1600	ABB1600
J1 (deg)	0	20	-10	72.53	0	-27.49
J2 (deg)*	0	20	20	29.82	0	62.47
J3 (deg)	0	20	-30	-54.97	0	-67.18
J4 (deg)	0	20	20 40 37.06		0	87.03
J5 (deg)	0	20	50	25.48	0	-21.26
J6 (deg)	0	20	-60	50.12	0	55.28
X (mm)	452.12	83.34	430.132	116.454	815.0	1071.912
Y (mm)	149.09	188.992	-39.844	432.2	0.0	-584.298
Z^* (mm)	-433.07	-492.497	341.522	431.209	583.0	382.645
RX (deg)	-180	160.588	-142.538	-85.816	-180	82.755
RY (deg)	0	-57.030	54.024	2.397	90	40.951
RZ (deg)	0	12.058	-57.746	-6.896	-0	-121.013

$$\theta'_4 = \theta_4 + 180^{\circ},$$
 $\theta'_5 = -\theta_5,$
 $\theta'_6 = \theta_6 + 180^{\circ}.$
(31)

4. Model Validations

4.1. Validation of the Forward Kinematic Model. The forward kinematic models were calculated in a robot control software developed in a Python environment, which was purposely made to construct an open architecture robot control system. A test run on ABB IRB120 is shown in Figure 3. For comparison purposes, ABB Robot Studio v.6.04 was employed with a IRB120 model. By configuring the input joint angles to the same degrees and then measuring the end coordinate values, the identical results can be obtained. Note in this ABB robot the original posture of link 2 is vertical upward, while the direct output of $\theta_2 = 0$ in the DH models results in link 2 horizontal, as for Puma560. Therefore, the control software performs an initial subtraction of 90 degrees for θ_2 for this ABB robot.

The forward kinematic modelling results for some robot types are shown in Table 3. For the ABB robots, the results were compared with those measured in ABB Robot Studio v.6.04 and proven to be consistent. The results for the Puma560 robot were compared with the literature and again showed consistency. Attention should be paid to joint 2 angles, which is the original value for Puma560, while for the ABB robots, the presented θ_2 values subtract 90 degrees were the actual inputs for the DH models.

In order to further support comparisons, the DH models for the mentioned robots were redeveloped in MATLAB Robotics Toolbox v10.4, and the end position values are consistent with those shown in Table 3. Below graphs were recorded in MATLAB Robotics Toolbox, where Figure 4 shows Puma560 at the two positions suggested in Table 3, and Figure 5 for ABB120 and Figure 6 for ABB1600, respectively.

There are other methods often utilized for robot validations, such as simulations in ADAMS and in ROS (Robot Operating System) environments. However, to practice these simulations, it is necessary to construct an environment identical to the current study and redevelop models in

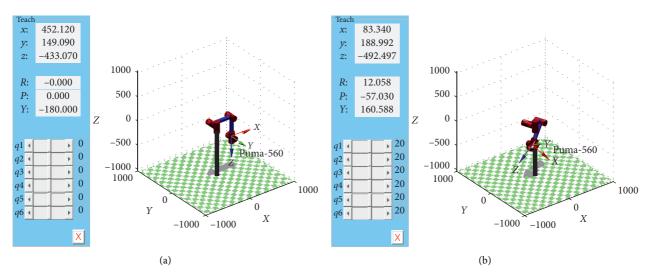


FIGURE 4: The forward kinematic modelling results for Puma560.

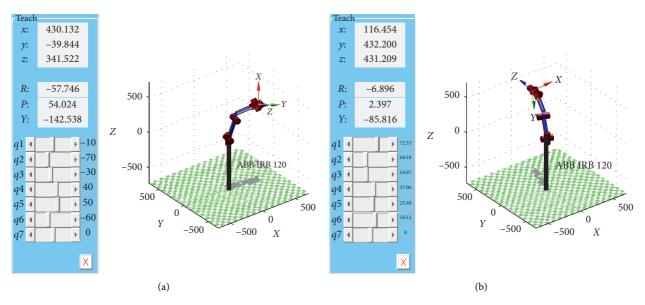


FIGURE 5: The forward kinematic modelling results for ABB IRB120.

these environments. Comparatively, MATLAB Robotics Toolbox is very simple to reconstruct DH models in different forms, and therefore, it is chosen as the major validation tool.

4.2. Validation of the Inverse Kinematic Model. By entering the end coordinate values and run inverse models in the Python software, all the possible solutions can be worked out, and the optimal solution can be selected by minimizing absolute angle changes comparing with the previous posture. For validation purposes, all the 8 solutions for Puma560 data set 2 were recorded and shown in Table 4. It can be seen that solution 1 is consistent with the original inputs.

For different robot types, the angle range of each joint may be different. This study assumes the angle range for each

joint is from -180° to 180° as for Puma560 configurations, although some specific robot types may have different angle ranges, and these should be altered accordingly in the real application. For the ABB IRB120 robot, there are also 8 solutions, and Table 5 shows the solutions for data set 1. It can be seen that solution 1 is consistent with the original inputs in Table 3.

For the ABB IRB1600 data set 2, the 8 inverse solutions are shown in Table 6, where one can recognize solution 5 is consistent with the original inputs.

5. Experiments

The experiments were carried out with an open architecture robot controller made with DSP and ARM computer as shown in Figure 7(a). The robot calibration was carried out

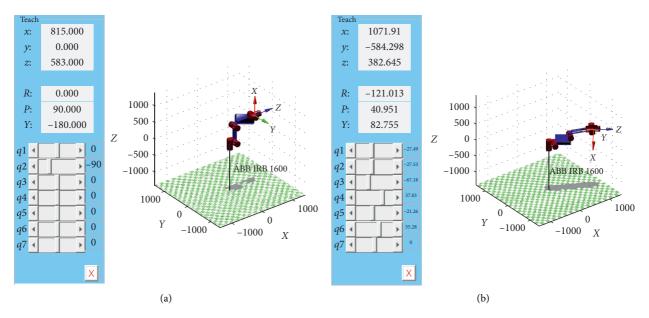


Figure 6: The forward kinematic modelling results for ABB IRB1600.

TABLE 4: Inverse kinematic solutions of Puma560 data set 2.

	J1 (deg)	J2 (deg)	J3 (deg)	J4 (deg)	J5 (deg)	J6 (deg)
Solution1*	20.00019	20.00004	19.99994	20.00043	20.00003	19.99989
Solution2	20.00019	20.00004	165.373	171.66486	126.2009	-146.0638
Solution3	-67.59241	52.37185	19.99994	-117.67587	72.49085	114.39521
Solution4	-67.59241	52.37185	165.373	-77.09362	119.95148	-101.11077
Solution5	20.00019	20.00004	19.99994	-159.9996	-20.00003	-160.00011
Solution6	20.00019	20.00004	165.373	-8.33514	-126.2009	33.9362
Solution7	-67.59241	52.37185	19.99994	62.32413	-72.49085	-65.60479
Solution8	-67.59241	52.37185	165.373	102.90638	-119.95148	78.88923

Table 5: Inverse kinematic solutions of ABB IRB 120 data set 1.

	J1 (deg)	J2 (deg)	J3 (deg)	J4 (deg)	J5 (deg)	J6 (deg)
Solution1*	-9.99997	19.99977	-29.99974	39.99967	49.99976	-59.99978
Solution2	-9.99997	19.99977	-123.90033	39.30978	128.99056	-4.40317
Solution3	169.99988	-70.38190	-29.99974	-148.60783	109.03982	-20.40059
Solution4	169.99988	-70.38190	-123.90033	-118.78936	34.18343	-88.06442
Solution5	-9.99997	19.99977	-29.99974	-140.00033	-49.99976	120.00022
Solution6	-9.99997	19.99977	-123.90033	-140.69022	-128.99056	175.59683
Solution7	169.99988	-70.38190	-29.99974	31.39217	-109.03982	159.59941
Solution8	169.99988	-70.38190	-123.90033	61.21064	-34.18343	91.93558

Table 6: Inverse kinematic solutions of ABB IRB 1600 data set 2.

	J1 (deg)	J2 (deg)	J3 (deg)	J4 (deg)	J5 (deg)	J6 (deg)
Solution1	-27.49005	62.46982	-67.17977	-92.97042	21.25992	-124.7196
Solution2	-27.49005	62.46982	-92.41242	-43.59768	31.67582	-178.8865
Solution3	152.5098	-112.26975	-67.17977	76.59041	21.85514	-113.50012
Solution4	152.5098	-112.26975	-92.41242	37.39259	36.60483	-69.43954
Solution5*	-27.49005	62.46982	-67.17977	87.02958	-21.25992	55.28039
Solution6	-27.49005	62.46982	-92.41242	136.40232	-31.67582	1.11348
Solution7	152.5098	-112.26975	-67.17977	-103.4096	-21.85514	66.49988
Solution8	152.5098	-112.26975	-92.41242	-142.6074	-36.60483	110.56046



FIGURE 7: The robot testing equipment. (a) Robot controller. (b) Dynalog CompuGauge calibration equipment.

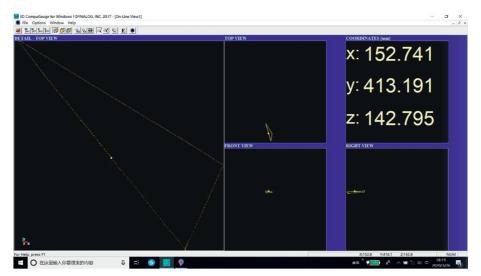


FIGURE 8: Testing trace of linear interpolation.

	TABLE 7. Repetited positioning accuracy testing data.								
	X_1	Y_1	Z_1	X_2	Y ₂	Z_2	X_3	Y ₃	Z_3
Calculated	150.917	571.526	115.126	331.109	265.102	322.413	-180.367	650.286	1.537
1	150.892	571.493	115.122	331.073	265.081	322.292	-180.412	650.383	1.532
2	150.934	571.491	115.121	331.082	265.088	322.381	-180.404	650.374	1.524
3	150.946	571.497	115.067	331.108	265.093	322.448	-180.395	650.328	1.531
4	150.936	571.501	114.944	331.162	265.149	322.383	-180.406	650.332	1.522
5	150.967	571.52	114.939	331.235	265.202	322.354	-180.405	650.375	1.447
6	150.954	571.456	115.115	331.16	265.205	322.295	-180.372	650.284	1.533
7	150.957	571.486	115.159	331.176	265.106	322.331	-180.394	650.263	1.432
8	150.965	571.494	115.109	331.11	265.136	322.233	-180.397	650.262	1.375
9	150.875	571.468	114.975	331.104	265.125	322.385	-180.408	650.277	1.477
10	150.874	571.496	115.004	331.031	265.049	322.369	-180.399	650.329	1.344
Mean	150.930	571.490	115.056	331.124	265.123	322.347	-180.399	650.321	1.472
Deviation	-0.013	0.036	0.071	-0.015	-0.021	0.066	0.032	-0.035	0.065
Standard deviations	0.036228	0.017599	0.082392	0.059283	0.051062	0.061111	0.011223	0.047084	0.06996

Table 7: Repeated positioning accuracy testing data.

on a Dynalog CompuGauge platform, as shown in Figure 7(b). The open architecture controller consists of a handheld ARM upper computer as the user command interface, a DSP board as the mid-layer signal processer, and a bottom layer Panasonic A6 series motor driver to generate current to drive A6 motors on the robot. The kinematic models were built in the ARM upper computer to enable users to define robot motion commands. The commands were sent to the DSP, which converts the requirements into a corresponding signal frequency and pulse numbers. Then, the processed control signals were sent to A6 drivers to generate the correct current to drive motors.

The A6 motor drivers can be configured to pulse control mode or current control model. In the pulse control mode, the AC server motors are controlled in a very similar way to stepper motors, i.e., the motor shaft ration is proportional to the number of input pulses. In this case, the driver uses an internal closed loop PID controller to cancel out the loading effects so that the proportion always remains constant. This means in the pulse control mode, the robot dynamics do not need to be modelled and the robot still produces good positioning accuracy. For this advantage, the pulse control mode was employed in the current stage of development. An example testing trace obtained in the Dynalog CompuGauge platform was shown in Figure 8.

The Dynalog CompuGauge platform can record robot motion data. A group of data for validating robot repeated positioning accuracy is given in Table 7. On average of each axis, the controller gives repeated positioning accuracy within 0.05 mm.

6. Conclusions

This paper has investigated the structural variations in industrial 6R robots with the aim of developing open architecture 6R robot controller. The main results of the paper are summarized below:

- (1) The analysis employed a classic Puma560 DH modelling approach but introduced more link offset and length parameters to suit different robot types. The major newly derived equations include equations (8)–(10), (14)–(16), (20)–(22).
- (2) The forward and inverse kinematic models have been developed and can apply to different robot types by modifying structural parameters.
- (3) Model validations have been carried out for Puma560, ABB IRB120, and ABB IRB1600 and showed satisfactory results.
- (4) The forward model outputs have been compared with ABB Robot Studio and MATLAB Robotics Toolbox and showed good consistency.
- (5) The inverse model outputs have been calculated in Python software and compared thoroughly to prove the validity of the model.
- (6) The model was employed on an open architecture controller based on ARM and DSP, and the average accuracy of each axis is within 0.05 mm.

The development of the kinematic models makes it possible to construct an advanced open architecture robot controller, which can adapt to different robot types and the environment with minimum changes in the control software. This progress would benefit small or medium robot manufacturers in the future.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by Shandong Key Research and Development Plan—Major Scientific and Technological Innovation Projects (grant no. 2017CXGC0907), Special Fund Plan for Local Science and Technology Development Lead by Central Authority, China, and Scientific Research Fund of Liaocheng University (grant no. 3180500), China.

References

- [1] Z. Cai, *Robotics*, Tsinghua University Press, Beijing, China, 3rd edition, 2015.
- [2] M. Raghavan and B. Roth, "Inverse kinematics of the general 6R manipulator and related linkages," *Journal of Mechanical Design*, vol. 115, no. 3, pp. 502–508, 1993.
- [3] D. Martins and R. Guenther, "Hierarchical kinematic analysis of robots," *Mechanism and Machine Theory*, vol. 38, no. 6, pp. 497–518, 2003.
- [4] A. Korzun and J. Röning, "An algorithm for the inverse kinematics problem of a calibrated robot," *IFAC Proceedings Volumes*, vol. 28, no. 5, pp. 79–83, 1995.
- [5] I. A. Vasilyev and A. M. Lyashin, "Analytical solution to inverse kinematic problem for 6-DOF robot-manipulator," *Automation and Remote Control*, vol. 71, no. 10, pp. 2195– 2199, 2010.
- [6] A. Pashkevich, "Real-time inverse kinematics for robots with offset and reduced wrist," *IFAC Proceedings Volumes*, vol. 30, no. 3, pp. 191–196, 1997.
- [7] I. Dulzba and M. Opałka, "A comparison of Jacobian-based methods of inverse kinematics for serial robot manipulators," *International Journal of Applied Mathematics and Computer Science*, vol. 23, no. 2, 2013.
- [8] S. Kucuk and Z. Bingul, "Inverse kinematics solutions for industrial robot manipulators with offset wrists," *Applied Mathematical Modelling*, vol. 38, no. 7-8, pp. 1983–1999, 2014.
- [9] F. Li, D. Zhao, and C. Ma, "Study on the inverse kinematics analysis and solution of FANUC_M-6iB robot," *Machinery Design Manufacture*, vol. 10, 2008.
- [10] X. Zhang and H.-l. Xu, "Analysis on structure and kinematics of FANUC M-6iB industrial robot," *Journal of Nantong University (Natural Science Edition)*, vol. 1, 2009.
- [11] H. Liu, W. Zhou, X. Lai et al., "An efficient inverse kinematic algorithm for a PUMA560-structured robot manipulator," *International Journal of Advanced Robotic Systems*, vol. 10, no. 5, pp. 323–338, 2013.

- [12] L. Hong, B. Ji, G. Cai et al., "Inverse kinematics algorithm for industrial robot of ABB1410 based on screw theory," Machinery Design & Manufacture, vol. 4, pp. 190–194, 2016.
- [13] E. Sariyildiz, E. Cakiray, and H. Temeltas, "A comparative study of three inverse kinematic methods of serial industrial robot manipulators in the screw theory framework," *Inter*national Journal of Advanced Robotic Systems, vol. 8, no. 5, 2011.
- [14] R. C. Rocha, C. P. Tonetto, and A. Dias, "A framework for kinematic modelling of cooperative robotic systems based on screw theory," in *Proceedings of the 21st International Con*gress of Mechanical Engineering, Natal, Brazil, October 2011.
- [15] R. Zhao, Z. Shi, Y. Guan et al., "Inverse kinematic solution of 6R robot manipulators based on screw theory and the Paden-Kahan subproblem," *International Journal of Advanced Ro*botic System, vol. 15, no. 6, 2018.
- [16] Y. Zhao, T. Huang, and Z. Yang, "A new numerical algorithm for the inverse position analysis of all serial manipulators," *Robotica*, vol. 24, pp. 1–4, 2005.
- [17] Z. Sui, L. Jiang, Y.-T. Tian, and W. Jiang, "Genetic algorithm for solving the inverse kinematics problem for general 6R robots," in *Proceedings of the 2015 Chinese Intelligent Automation Conference*, Fuzhou, China, March 2015.
- [18] Z. Wang, H. Dong, S. Bai et al., "A new approach of kinematic geometry for error identification and compensation of industrial robots," *Proceedings of the Institution of Mechanical Engineers Part C Journal of Mechanical Engineering Science*, vol. 233, no. 5, 2018.
- [19] J. Meneses, C. Castejón, E. Corral, H. Rubio, and J. C. García-Prada, "Kinematics and dynamics of the quasi-passive biped "PASIBOT" Strojniški Vestnik—Journal of Mechanical Engineering, vol. 57, no. 12, p. 879, 2011.
- [20] E. Corral, M. J. G. García, C. Castejon, J. Meneses, and R. Gismeros, "Dynamic modeling of the dissipative contact and friction forces of a passive biped-walking robot," *Applied Sciences*, vol. 10, no. 7, p. 2342, 2020.
- [21] E. C. Abad, J. M. Alonso, M. J. G. García, and J. C. García-Prada, "Methodology for the navigation optimization of a terrain-adaptive unmanned ground vehicle," *International Journal of Advanced Robotic Systems*, vol. 15, no. 1, 2018.