

GEOMETRIC APPROACH TO SOLVING INVERSE KINEMATICS OF SIX DOF ROBOT WITH SPHERICAL JOINTS

NACER HADIDI^{a,*}, MOHAMED BOUAZIZ^a, CHAWKI MAHFOUDI^b,
MOHAMED ZAHARUDDIN^c

^a Ecole Nationale Polytechnique, Laboratoire de Genie Mecanique et developpement, Department of Mechanical engineering, 10 Avenue Hassan Badi, 16200 El Harrach, Algeria

^b University Larbi Ben Mhidi, Campus of AinBeida, Department of Mechanical engineering, Route de Constantine, B.P. 358, 04000 Oum El Bouaghi, Algeria

^c Universiti Teknologi Malaysia, Faculty of Electrical Engineering, Department of mechatronics, 81310 Johor Bahru, Malaysia

* corresponding author: nacer.hadidi@enp.edu.dz

ABSTRACT. Inverse kinematics is a fundamental concept in robotics that plays a crucial role in a robot's ability to perform tasks. In this contribution, we propose a novel geometric approach based on vector calculus to solve the inverse kinematics problem. The primary advantage of this approach originates from the solutions, which exhibit a linear form and uncoupled equations. To validate the effectiveness and correctness of our proposed method, we constructed a six-degrees-of-freedom robot. This robot is controlled by an Arduino Mega 2650 on which we have implemented the inverse kinematics algorithm. The validation process involved considering various desired trajectories of the end-effector, which were simulated in Matlab and then performed by the physical robot. Importantly, our findings confirm that the end-effector successfully tracks the predefined trajectories. Furthermore, we conducted a comparative analysis between Paul's method and the results obtained from joint angles using our proposed approach. Interestingly, our study reveals a significant similarity between the two sets of results, reaffirming the accuracy and validity of the approach presented in this study.

KEYWORDS: Robotic, forward kinematics, inverse kinematics, geometric modelling, space geometry.

1. INTRODUCTION

Due to the interesting and complex nature of field of robotics, several approaches to the research of robots have been taken. One of the challenging tasks is the kinematic modelling (Forward and Inverse Kinematics). Research into the treatment of the robot kinematics continues to spread over different types of robots. In the literature, some works have focused on the serial manipulator robots [1–5]. While others have dealt with parallel robots and hybrid robots' structure [6, 7]. In addition, some research has focused on specific robot configurations, such as snake robots [8], wheeled robots, and quadruped robots [9, 10].

Forward kinematics (FK) does not pose significant difficulties because it involves a mathematically straightforward multiplication of successive homogeneous matrices that lead the end-effector to a corresponding position and orientation [11–13]. However, inverse kinematics (IK) entails specifying a precise pose for the robot's end-effector within its workspace, defined by its position and orientation in Euclidean space R^3 . In this context, finding a suitable set of joint configurations that enables the end-effector to reach the desired positions is the main challenge.

In the literature, Inverse kinematics (IK) problem has been addressed by several authors using different approaches. Basically, these approaches can be categorised into analytical approaches, numerical approaches, and hybrid approaches, using different methods in order to achieve a smooth solution. Where the primary aim of these methods is to provide flexible mobility for the robot, which is dependent on the robot's geometric structure and the number of degrees of freedom (DOF) [14, 15].

In Euclidian space R^3 , analytical methods provide closed-form solutions for the inverse kinematics under certain conditions:

- (1.) When three successive revolute joint axes intersect at the same point;
- (2.) When three consecutive revolute joint axes are parallel.

Mathematically, analytical methods are represented by the following general matrix equation [16–19]:

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

where ${}^{j-1}T_j$ represents the homogeneous matrix between frames j and $j - 1$, and \mathbf{T} is the matrix representing the desired position of the end-effector. However, solutions to these problems are typically derived through nonlinear and coupled equations that often involve matrix calculations.

Several methods have been developed in the literature to address these challenges. Notable approaches include Pieper's method [16], which is applicable to robots with three articulatory joints featuring concurrent axes or three prismatic joints. Raghavan and Roth [17] have formulated a system of 16 multivariate polynomials for closed-loop solutions, suitable for 6-revolute manipulators (6R). Their solutions are based on successive elimination techniques. Paul's method [18] involves isolating and solving variables one by one, making it suitable for most industrial manipulator robots. Gan et al. [19] have derived an inverse kinematics (IK) solution for a 5-DOF robotic arm (Pioneer 2), which has similarities to Paul's method. Other studies have investigated the IK problem of robots with redundancy, providing analytical solutions based on various assumptions related to the redundancy settings [20–22].

Numerous works have addressed the numerical approach to solve the Inverse Kinematics (IK) problem [1–3, 23–27]. The majority of these solutions have been derived using different methods, including numerical algorithms [2], neural networks [24, 28], genetic algorithms [29], fuzzy logic, and others. Within this context, numerical algorithms have been used to ensure convergence and accuracy. However, it's important to note that many of the currently available methods are plagued by high computational costs and the generation of unrealistic poses [14].

In addition, some robots with 7 degrees of freedom (7-DOF) have been addressed using numerical approaches, aiming to resolve redundancy issues in the Inverse Kinematics (IK) solutions. Among these, Y. Wenbin and S. Lei [30] have studied an iterative approach combining two techniques:

- (a) A technique based on “fixed joint angles”,
- (b) A technique involving a “two-phase computational optimisation algorithm using Weighted Least-Norm”.

In addition, B. Ma et al. [31] have presented a hybrid IK solution for 7-DOF robots, based on two analytical techniques that consider redundant parameters to produce approximate solutions, followed by a numerical algorithm that minimises errors to achieve convergence of the solution.

This paper introduces a novel approach based on the geometric description of a robot to solve the Inverse Kinematics (IK) problem. The approach is based on vector calculus and trigonometric relations and offers a significant advantage by providing solutions through simple and decoupled linear equations. This eliminates the need to solve the complex IK problem using matrices and numerical methods.

To assess the feasibility and accuracy of this approach, we conducted Matlab simulations and an experimental study involving the design and construction of a 6-degree-of-freedom (6-DOF) robot. By instructing the robot's end-effector to follow predefined trajectories, our approach successfully converted Cartesian coordinates into joint angle coordinates in both the simulation and the experiment. As a result, the robot was able to accurately track the predefined trajectories.

We also performed a comparative analysis between our approach and Paul's method for a similar robot arm, as presented in reference [32]. This comparison revealed significant consistency and similarity in the results, confirming the validity and reliability of our proposed approach.

2. PRINCIPLE OF THE APPROACH FOR INVERSE KINEMATICS

An angle between two given vectors \vec{U} and \vec{V} in R^3 is defined by its magnitude and its conventional orientation (rotation sense). Mathematically, the magnitude is given by the cosine law.

$$\vec{U} \cdot \vec{V} = U \cdot V \cos(\vec{U}, \vec{V}), \quad (2)$$

where U and V represent the magnitude of the vectors \vec{U} and \vec{V} , respectively. The rotation sense of the angle is described by applying the right hand rule between three vectors: \vec{U} and \vec{V} and \vec{W} . Where \vec{W} is the cross product of \vec{U} and \vec{V} or any other orthonormal vector to the plane formed by \vec{U} and \vec{V} (Plane-uv), as illustrated in Figure 1.

A helpful relation between the cross product and the determinant can be defined as follows:

$$\vec{U} \wedge \vec{V} = U \cdot V \sin(\vec{U}, \vec{V}) \vec{n}_{\vec{W}}. \quad (3)$$

By multiplying both sides of Equation (3) by the same vector $\vec{n}_{\vec{W}}$ and dividing by the scalar UV , we obtain a new equality in Equation (4):

$$\frac{1}{UV} (\vec{U} \wedge \vec{V}) \cdot \vec{n}_{\vec{W}} = \sin(\vec{U}, \vec{V}) \cdot (\vec{n}_{\vec{W}})^2. \quad (4)$$

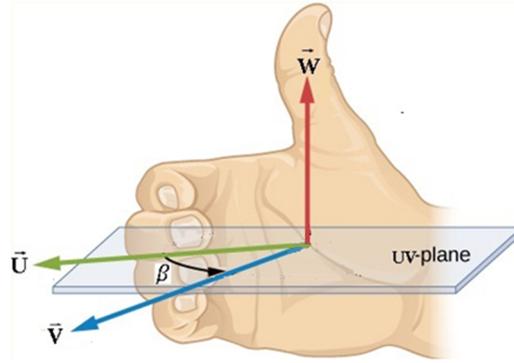


FIGURE 1. Right hand rule and vector order.

Equation (4) leads to the equality below:

$$\sin(\vec{U}, \vec{V}) = (\vec{n}_U \wedge \vec{n}_V) \cdot \vec{n}_W = \det(\vec{n}_U, \vec{n}_V, \vec{n}_W), \tag{5}$$

where:

- $\vec{n}_U = \frac{\vec{U}}{U}$ and $\vec{n}_V = \frac{\vec{V}}{V}$ are two unit vectors of \vec{U} and \vec{V} , respectively, and $(\vec{n}_W)^2 = 1$,
- $\det(\vec{n}_U, \vec{n}_V, \vec{n}_W) = (\vec{n}_U \wedge \vec{n}_V) \cdot \vec{n}_W = \begin{pmatrix} n_{ux} & n_{uy} & n_{uz} \\ n_{vx} & n_{vy} & n_{vz} \end{pmatrix}$ is the determinant of the three vector (or mixed product), which can also be developed as follows:

$$\begin{aligned} \det(\vec{n}_U, \vec{n}_V, \vec{n}_W) &= n_{wx}(n_{uy} \cdot n_{vz} - n_{vy} \cdot n_{uz}) - \\ &\quad n_{wy}(n_{ux} \cdot n_{vz} - n_{vx} \cdot n_{uz}) + \\ &\quad n_{wz}(n_{ux} \cdot n_{vy} - n_{vx} \cdot n_{uy}). \end{aligned}$$

An important property of the determinant regarding its sign is obtained by swapping two columns [33].

$$\det(\vec{W}, \vec{U}, \vec{V}) = (-1) \det(\vec{W}, \vec{V}, \vec{U}).$$

This implies that the sign of the determinant changes with variations in angular orientation.

The magnitude and direction sense of each joint angle can be determined by substituting these two previous parameters of Equation (2) and Equation (5) as arguments into the mathematical function $atan2(x, y)$, see Appendix A.1.

$$\beta = atan2(\det(\vec{n}_U, \vec{n}_V, \vec{n}_W), \cos(\vec{U}, \vec{V})). \tag{6}$$

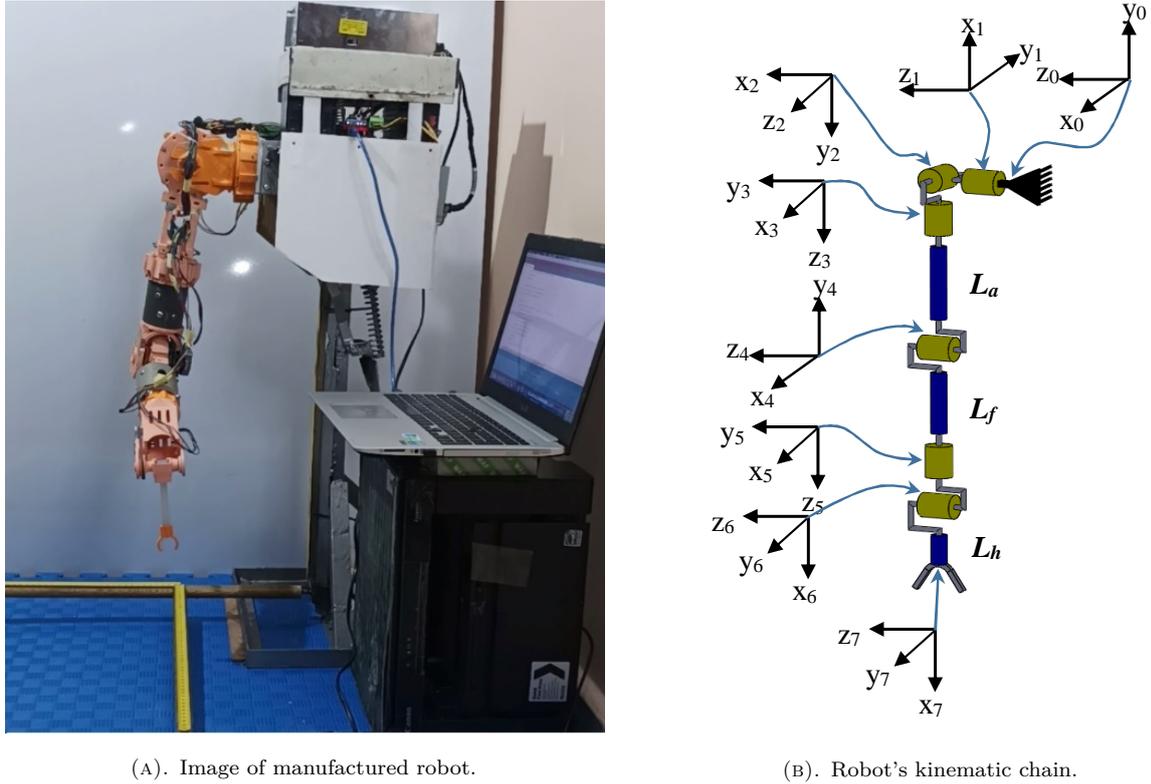
3. ROBOT STRUCTURE AND GEOMETRY

Figure 2 shows a right upper limb with 6-DOF of a biped robot distributed as follows: 3-DOF at the shoulder, 1-DOF at the elbow, and 2-DOF at the wrist. These components are interconnected by three geometric links: the arm (L_a), the forearm (L_f), and the hand (L_h).

Their lengths are summarised in Table 1.

Links	Symbol	Value [m]
Arm	L_a	0.32
Forearm	L_f	0.25
Hand	L_h	0.12

TABLE 1. Robot link lengths.



(A). Image of manufactured robot.

(B). Robot's kinematic chain.

FIGURE 2. Right upper limb of a biped robot and frame assignment according to DH.

4. MATHEMATICAL MODELLING OF THE ROBOT

This section is primarily dedicated to developing the mathematical model that describes the Inverse Kinematics (IK) solutions. The approach outlined below provides the corresponding actuator rotations for each state of the end-effector.

The following assumptions have been used in this work:

- The shoulder, with its 3 degrees of freedom (3 DOF), has its axes of rotation intersecting at the same point. Consequently, the shoulder's rotational movement can be treated as a single geometric point.
- Similarly, the same assumption has been applied to the wrist, which has 2 DOF. Consequently, the wrist is also treated as a single geometric point.

4.1. FORWARD KINEMATICS

The matrix representing the desired position and orientation, denoted as $T = [S \ N \ A \ P]$, of the end-effector, given in Equation (1) and obtained through successive multiplications of homogeneous transformation matrices, can be rewritten as Equation (7).

$$[S \ N \ A \ P] = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 {}^6T_7 = \begin{bmatrix} s_x & n_x & a_x & P_{dx} \\ s_y & n_y & a_y & P_{dy} \\ s_z & n_z & a_z & P_{dz} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (7)$$

where: $\mathbf{S} = (s_x, s_y, s_z)^T$, $\mathbf{N} = (n_x, n_y, n_z)^T$, $\mathbf{A} = (a_x, a_y, a_z)^T$ are the cosine directions of the frame $F_7(x_7, y_7, z_7)$ (orientation of the end-effector). $\mathbf{P} = (P_{dx}, P_{dy}, P_{dz})^T$ is the coordinate of the desired position of the end-effector and ${}^{i-1}T_i$ is the homogeneous transformation between two successive frames ($i = 1, \dots, 7$).

Table 2 shows parameters obtained using the DH convention where α_i is the rotational angle between z_i axis and z_{i-1} axis around x_i axis, $q_i = \theta_i + q_{0i}$ is the rotational angle between x_i axis and x_{i-1} axis around z_i axis where θ_i represents the effective rotation and q_{0i} represents the initial frame rotation corresponding to the home position.

Hence, the homogeneous transformation matrix between two successive frames can be obtained through Equation (8):

Frame	α_i	L_i	q_i	d_i	Joint range
F_0	–	–	–	–	–
F_1	0	0	$\frac{\pi}{2} + \theta_1$	0	$[-\frac{\pi}{2}, \pi]$
F_2	$\frac{\pi}{2}$	0	$\frac{\pi}{2} + \theta_2$	0	$[-\frac{\pi}{2}, -\pi]$
F_3	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2} + \theta_3$	L_a	$[-\pi, \pi]$
F_4	$-\frac{\pi}{2}$	0	θ_4	0	$[0, \frac{3\pi}{4}]$
F_5	$\frac{\pi}{2}$	0	θ_5	L_f	$[0, 2\pi]$
F_6	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2} + \theta_6$	0	$[-\frac{3\pi}{4}, \frac{3\pi}{4}]$
F_7	0	L_h	0	0	–

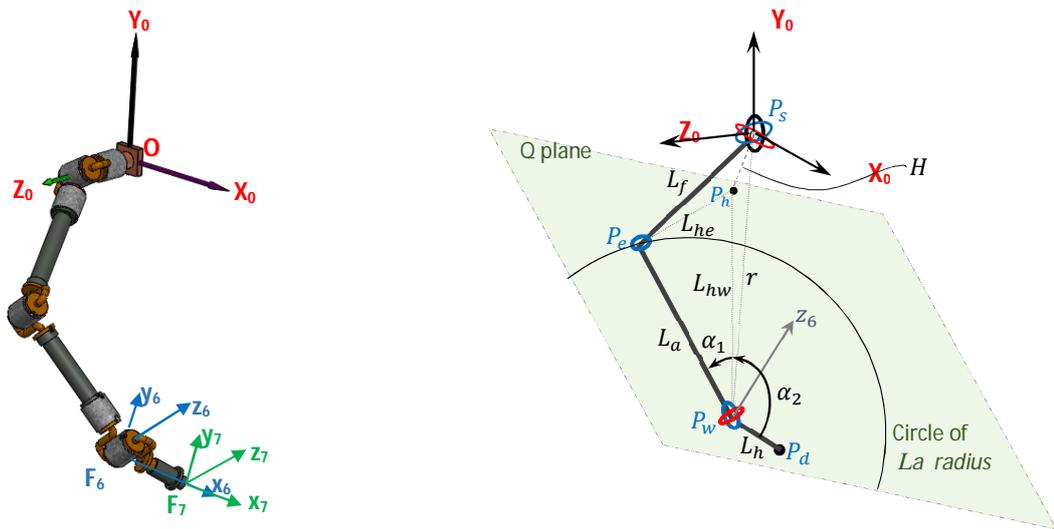
TABLE 2. DH parameters.

$${}^{i-1}T_i = \begin{bmatrix} \cos(q_i) & -\sin(q_i) & 0 & L_i \\ \cos(\alpha_i)\sin(q_i) & \cos(\alpha_i)\cos(q_i) & -\sin(\alpha_i) & -d_i\sin(\alpha_i) \\ \sin(\alpha_i)\sin(q_i) & \sin(\alpha_i)\cos(q_i) & \cos(\alpha_i) & d_i\cos(\alpha_i) \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{8}$$

4.2. INVERSE KINEMATICS SOLUTIONS

Solving the inverse kinematics involves determining the appropriate rotations of each robot’s articulation to bring the end-effector to the desired situation (position and orientation).

Figure 3 represents a geometric parameterisation of the robot (geometric entities), which will be utilised in the inverse kinematics problem solving process.



(A). Robot configuration with assignment of main frames.

(B). Geometric entities.

FIGURE 3. Geometric parameterisation of a right upper limb biped robot.

In this study, the strategy for solving the inverse kinematics problem is divided into two phases: the first phase consists of determining the geometric entities that allow the determination of the vectors defining each joint. The second phase focuses on determining the angles of each joint, which are based on the results obtained in the first phase.

4.2.1. CALCULATE THE GEOMETRIC ENTITIES OF THE ROBOT

Initially, certain geometric entities, including points, vectors, and planes, must be determined. To achieve this, the following strategy is adopted:

- Determine the wrist position.
- Determine the elbow position by the following procedure:
 - ▷ Determine the Q -plane equation through the normal vector at the rotational joint q_6 (z_6 -axis). Geometrically, the forearm link has to belong to the Q -plane because for any position and orientation of the end-effector, the z_6 -axis remains perpendicular to the forearm-link.
 - ▷ Calculate the coordinate of the closest point P_h on Q -plane at the original frame F_0 .
 - ▷ Determine the circle on the Q -plane in which the wrist point P_w and the length of forearm (L_f) are its centre and radius, respectively.
 - ▷ Calculate the coordinate point of elbow (P_e).

According to Figure 3, the frame F_6 of the rotational joint q_6 maintains the same orientation as the frame F_7 ; therefore, the position of the wrist is determined by the following Equation (9), using the given vector, $\vec{S}_6 = (s_{6x}, s_{6y}, s_{6z})^T$, in the matrix (\mathbf{T}) of the end-effector.

$$\overrightarrow{OP_w} = \overrightarrow{OP_d} - L_h \vec{S}_6, \quad (9)$$

where: $\overrightarrow{OP_w} = (P_{wx}, P_{wy}, P_{wz})^T$ and $\overrightarrow{OP_d} = (P_{dx}, P_{dy}, P_{dz})^T$ are the coordinates of the wrist position and the desired position of the end-effector, respectively.

From Figure 3, we derived the tetrahedron form, which consists of two right-angled triangles $\Delta(P_s, P_h, P_e)$ and $\Delta(P_s, P_h, P_w)$, which are defined as shown in Figure 4.

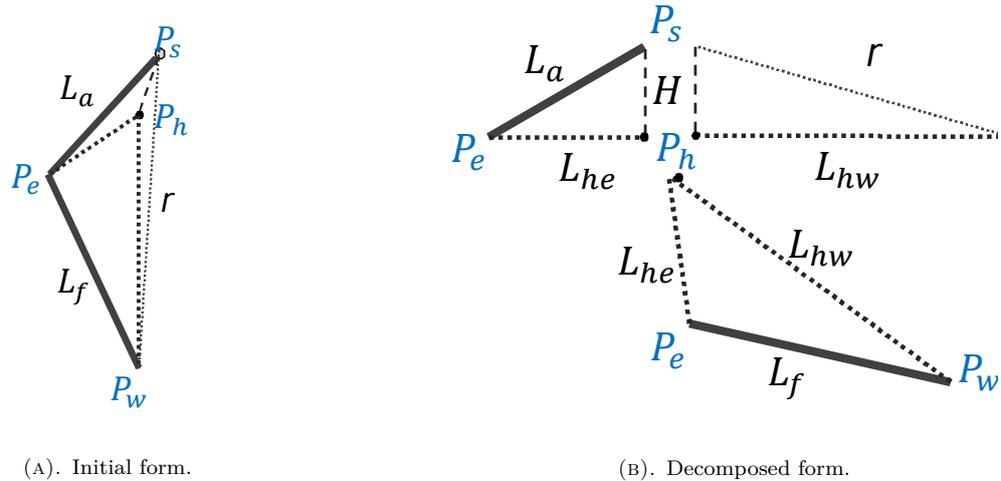


FIGURE 4. Tetrahedron form obtained from the derived geometry space of robot.

P_s , P_e , P_w and P_h are the points designating: shoulder, elbow, wrist, and the closest point on Q -plane to the shoulder frame (F_0), respectively.

From Figure 4, the length between the origin and the wrist can be calculated by Equation (10).

$$r = \sqrt{P_{wx}^2 + P_{wy}^2 + P_{wz}^2}, \quad (10)$$

where P_{wx} , P_{wy} , P_{wz} are the coordinates of the vector ($\overrightarrow{OP_w}$) obtained in Equation (9).

Q -plane equation obtained via the normal vector $\vec{A}_6 = (a_x, a_y, a_z)^T$ of z_6 -axis of the frame F_6 is:

$$a_x x + a_y y + a_z z - D = 0, \quad (11)$$

where $D = (a_x P_{wx} + a_y P_{wy} + a_z P_{wz})$ is the parameter of the plane equation.

From Equation (11), the closest distance, H , between the origin and Q -plane can be calculated as:

$$H = \frac{|a_x P_{wx} + a_y P_{wy} + a_z P_{wz}|}{\sqrt{a_x^2 + a_y^2 + a_z^2}}. \quad (12)$$

And the corresponding coordinates of the closest point on Q -plane from the origin is given by:

$$\begin{aligned} P_{hx} &= \frac{a_x D}{a_x^2 + a_y^2 + a_z^2}, \\ P_{hy} &= \frac{a_y D}{a_x^2 + a_y^2 + a_z^2}, \\ P_{hz} &= \frac{a_z D}{a_x^2 + a_y^2 + a_z^2}. \end{aligned} \tag{13}$$

From Figure 4, the length L_{he} is defined as the distance between P_h (the closest point to the shoulder frame on Q -plane) and P_e (the elbow position). Similarly, L_{hw} is defined between points P_h and P_w (the wrist position). Consequently, using the results obtained in Equations (10) to (13), the two lengths, L_{he} and L_{hw} , can be obtained by the following relationships:

$$L_{he} = \sqrt{L_f^2 - H^2}, \tag{14}$$

$$L_{hw} = \sqrt{r^2 - H^2}. \tag{15}$$

Thus, based on Equations (14) and (15) and from the triangle $\Delta(P_e, P_h, P_w)$ shown in Figure 3, the angle α_1 formed by the points $(P_e, \widehat{P_w}, P_h)$ can be calculated by the cosine law as:

$$\alpha_1 = (P_e, \widehat{P_w}, P_h) = a \cos\left(\frac{L_{hw}^2 + L_f^2 - L_{he}^2}{2L_f L_{he}}\right). \tag{16}$$

In addition, the circle equation of radius L_f and centre point P_w on the Q -plane is defined in Equation (17).

$$\overrightarrow{OP_c} = \overrightarrow{OP_w} + L_f \cos(\phi) \overrightarrow{S_6} + L_f \sin(\phi) \overrightarrow{N_6}, \tag{17}$$

where $\overrightarrow{OP_c} = (P_{cx}, P_{cy}, P_{cz})^T$ and $\overrightarrow{OP_w} = (P_{wx}, P_{wy}, P_{wz})^T$ are the coordinates of circle's circumference and circle's centre respectively. $\overrightarrow{S_6}$ and $\overrightarrow{N_6}$ are the unit vectors of x_6 -axis and y_6 -axis of frame F_6 , respectively, and $\phi = \alpha_1 + \alpha_2$ is the angle between x_6 -axis and $\overrightarrow{V_{we}}$ (the vector between the wrist and the elbow point). Additionally, the value $\alpha_1 = (P_e, \widehat{P_w}, P_h)$ is calculated by Equation (16) and the α_2 magnitude is given by the cosine law:

$$\cos(\alpha_2) = \frac{\overrightarrow{S_6} \cdot \overrightarrow{n_{wh}}}{|\overrightarrow{S_6}| |\overrightarrow{n_{wh}}|}, \tag{18}$$

where $\overrightarrow{n_{wh}}$ and $|\overrightarrow{n_{wh}}|$ are normalised vectors of $\overrightarrow{V_{wh}}$ and its modulus, respectively.

Referring to Figure 3b, where the vectors $\overrightarrow{S_6}$ and $\overrightarrow{V_{wh}}$ (wrist-closest point on the Q -plane to the shoulder frame) are both inside the Q -plane, the orientation of α_2 can be determined by the determinant of these three vectors: $\overrightarrow{A_6}$, $\overrightarrow{n_{wh}}$ and $\overrightarrow{S_6}$ as shown in Equation (19):

$$\det_{\alpha_2} = \det \left[\overrightarrow{A_6} \overrightarrow{n_{wh}} \overrightarrow{S_6} \right]^T. \tag{19}$$

Hence, based on Equations (18) and (19), the angle α_2 is given by the next equation:

$$\alpha_2 = \text{atan2}(\det_{\alpha_2}, \cos(\alpha_2)). \tag{20}$$

Then, the angle ϕ is calculated by the Equations (16) and (20).

$$\phi = \alpha_1 + \alpha_2. \tag{21}$$

Finally substituting Equation (21) in Equation (17), the elbow position is determined by the coordinate's vector (22):

$$\overrightarrow{V_{oe}} = (P_{ex}, P_{ey}, P_{ez})^T. \tag{22}$$

After determining the necessary geometric entities for calculating the joint angles, we proceed to their identifications.

4.2.2. CALCULATE THE JOINT ANGLES OF THE ROBOT

The geometric vectors needed to calculate the joint angles are obtained in Phase 1. Therefore, the joint angle equations are calculable ($\beta_i, i = 1, \dots, 6$).

Geometrically, each joint angle is described by three vectors (see Figure 5a-5f), two mathematical arguments ($\text{arg1}, \text{arg2}$) are expressed which are used in Equation (6) in the form $\text{atan2}(\text{arg1}, \text{arg2})$.

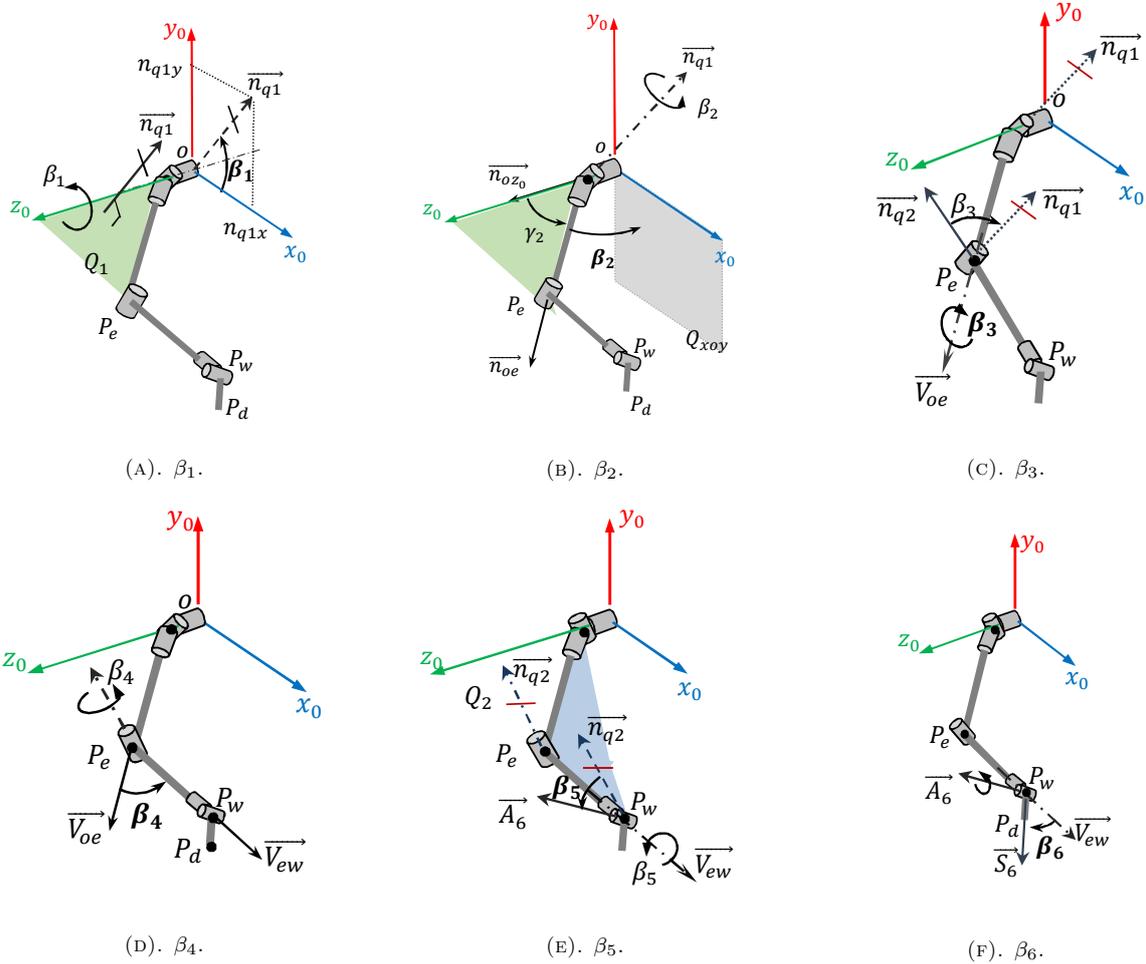


FIGURE 5. Vectorial description of robot's joint angles.

a. Joint angle β_1 : Figure 5a shows the vectorial description of the joint angle β_1 . The angle is defined by the normal vector obtained from the cross product of vectors \vec{V}_{oz_0} and \vec{V}_{oe} .

\vec{n}_{q1} is the axis collinear with the rotational joint q_2 -axis. Furthermore, \vec{n}_{q1} revolves in a plane parallel to plane Q_{xoy} . Therefore, $n_{1x} = \frac{\cos(\vec{V}_{ox_0}, \vec{n}_{q1})}{|\vec{n}_{q1}|}$ and $n_{1y} = \frac{\sin(\vec{V}_{ox_0}, \vec{n}_{q1})}{|\vec{n}_{q1}|}$, where:

- \vec{V}_{ox_0} and \vec{V}_{oz_0} : the unit vectors of x_0 -axis and z_0 -axis, respectively, of frame F_0 ,
- \vec{n}_{q1} : the normalised and normal vector of Q_1 -plan obtained from cross product $\vec{V}_{oz_0} \wedge \vec{V}_{oe}$,
- \vec{V}_{oe} : the vector between the two points (origin-elbow),

Q_{xoy} : the plane defined by two unit vectors of axes, x_0 -axis and y_0 -axis.

As the normal vector \vec{n}_{q1} is normalised to a unit vector ($|\vec{n}_{q1}| = 1$), the equations can be written as: $n_{1x} = \cos(\vec{V}_{ox_0}, \vec{n}_{q1})$ and $n_{1y} = \sin(\vec{V}_{ox_0}, \vec{n}_{q1})$. Therefore, the joint angle is obtained by formula:

$$\beta_1 = \text{atan2}(n_{1y}, n_{1x}). \tag{23}$$

b. Joint angle β_2 : Figure 5b shows the vectorial description of the joint angle β_2 . It is defined by two vectors, \vec{V}_{oz_0} and \vec{V}_{oe} , where, \vec{n}_{oz_0} and \vec{n}_{oe} are their normal vectors, respectively.

From Figure 5b, β_2 can be written as:

$$\beta_2 = \frac{\pi}{2} - \gamma_2, \tag{24}$$

where γ_2 is expressed by the equation $\gamma_2 = \text{atan2}(\det_{\gamma_2}, \cos(\gamma_2))$, where $\cos(\gamma_2) = \vec{V}_{oz_0} \cdot \vec{n}_{oe}$ and $\det_{\gamma_2} = \det [\vec{n}_{q1} \vec{n}_{oz_0} \vec{n}_{oe}]^T$ in which $\vec{n}_{oz_0} = (0\ 0\ 1)^T$, $\vec{n}_{oe} = (n_{ex} \ n_{ey} \ n_{ez})^T$ and $\vec{n}_{q1} = (n_{1x} \ n_{1y} \ 0)^T$.

c. Joint angle β_3 : The joint angle β_3 is the angle between the two planes Q_1 -plane (defined by vectors \vec{V}_{oz_0} and \vec{V}_{oe}) and Q_2 -plane (defined by the three points of the robot's links: shoulder-elbow-wrist) as shown in Figure 5c, where \vec{n}_{q1} and \vec{n}_{q2} are their normal vectors, respectively. Thus, β_3 can be further expressed as follows:

$$\beta_3 = \text{atan2}(\det_{\beta_3}, \cos(\beta_3)), \tag{25}$$

where $\cos(\beta_3) = \vec{n}_{q1} \cdot \vec{n}_{q2}$ and the determinant is: $\det_{\beta_3} = \det [\vec{n}_{oe} \vec{n}_{q1} \vec{n}_{q2}]^T$.

d. Joint angle β_4 : The joint angle β_4 is an angle between \vec{V}_{oe} (arm) and \vec{V}_{ew} (forearm) as shown in Figure 5d. \vec{n}_{oe} and \vec{n}_{ew} are the normalised vectors of \vec{V}_{oe} and \vec{V}_{ew} , respectively. The angle can be calculated by the formula below:

$$\beta_4 = \text{atan2}(\det_{\beta_4}, \cos(\beta_4)), \tag{26}$$

where the $\cos(\beta_4) = \vec{n}_{oe} \cdot \vec{n}_{ew}$ and the determinant of β_4 can be obtained by: $\det_{\beta_4} = \det [\vec{n}_{q1} \vec{n}_{oe} \vec{n}_{ew}]^T$.

e. Joint angle β_5 : The joint angle β_5 is defined as the angle between the vectors \vec{n}_{q2} and $\vec{A}_6 = (a_x, a_y, a_z)^T$ of the z_6 -axis as shown in Figure 5e.

So the arguments of β_5 can be expressed by $\cos(\beta_5) = \vec{n}_{q2} \cdot \vec{A}_6$ and $\det_{\beta_5} = \det [\vec{V}_{we} \vec{n}_{q2} \vec{A}_6]^T$.

Finally, β_5 can be expressed by Equation (27):

$$\beta_5 = \text{atan2}(\det_{\beta_5}, \cos(\beta_5)). \tag{27}$$

f. Joint angle β_6 : The joint angle β_6 is the angle between the forearm and the hand of the robot, which is defined by the vectors \vec{V}_{ew} (elbow-wrist) and \vec{V}_{wd} (wrist-desired position), respectively, see Figure 5f. Therefore, β_6 is the angle between two vectors: $\vec{n}_{ew} = \frac{1}{|\vec{V}_{ew}|} \vec{V}_{ew}$ and $\vec{S}_6 = (s_x, s_y, s_z)^T$.

Thus, the two arguments of β_6 are: $\cos(\beta_6) = \vec{n}_{ew} \cdot \vec{S}_6$ and $\det_{\beta_6} = \det [\vec{A}_6 \vec{n}_{ew} \vec{S}_6]^T$. Consequently, β_6 can be obtained by the following formula:

$$\beta_6 = \text{atan2}(\det_{\beta_6}, \cos(\beta_6)). \tag{28}$$

5. SIMULATIONS AND RESULTS

In this study, simulations have been performed to verify the performance of the proposed approach for solving the Inverse Kinematics (IK) problem. To achieve this, a program was developed using MATLAB based on the IK results presented in Equations (23) to (28) for the simulation case.

Figure 6 illustrates a freely chosen predefined path for the end-effector. The robot should therefore be able to follow these trajectories accurately.

Each part of the end-effector's path has been characterised as an arc of a circle defined by a set of Cartesian coordinates. Only the path's part 9→10 is a circular trajectory defined by its equation (see Appendix A.2).

As mentioned in Table 3, the robot's end-effector will start from the point with coordinates (0,-0.6,0) m, following the trajectories indicated by the following numbered points order:

$$P0 \rightarrow P1 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow P5 \rightarrow P6 \rightarrow P7 \rightarrow P8 \rightarrow P9 \rightarrow P10 \rightarrow P11$$

The orientation of the end-effector is expressed by a given matrix as shown below:

$$[S \ N \ A] = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

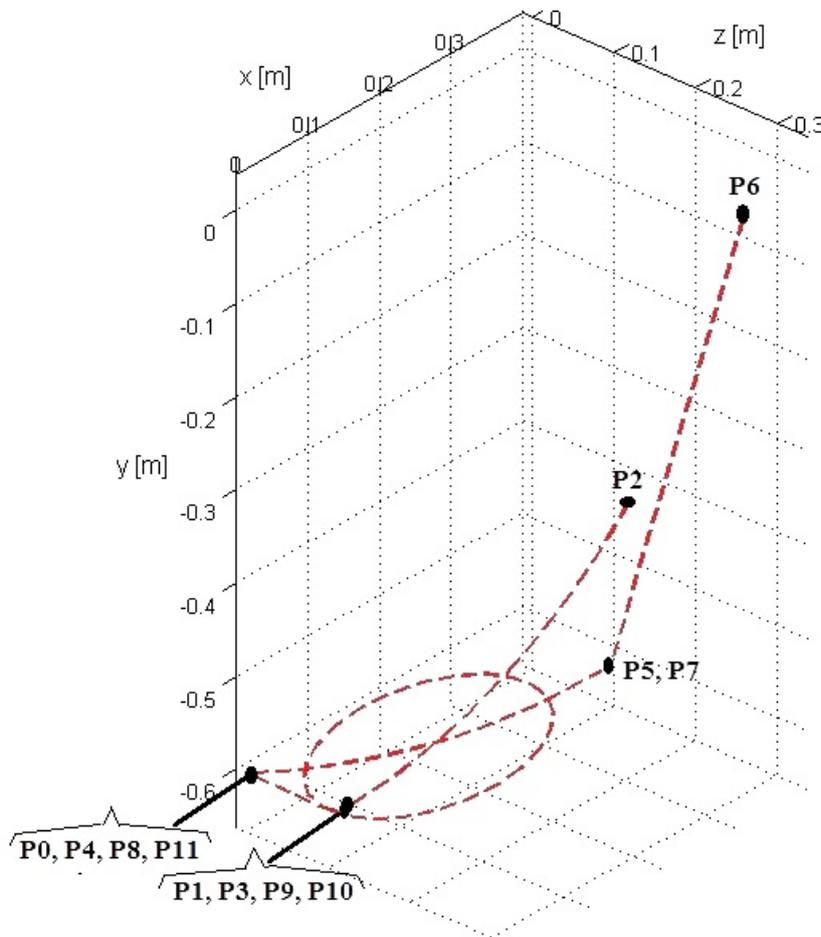


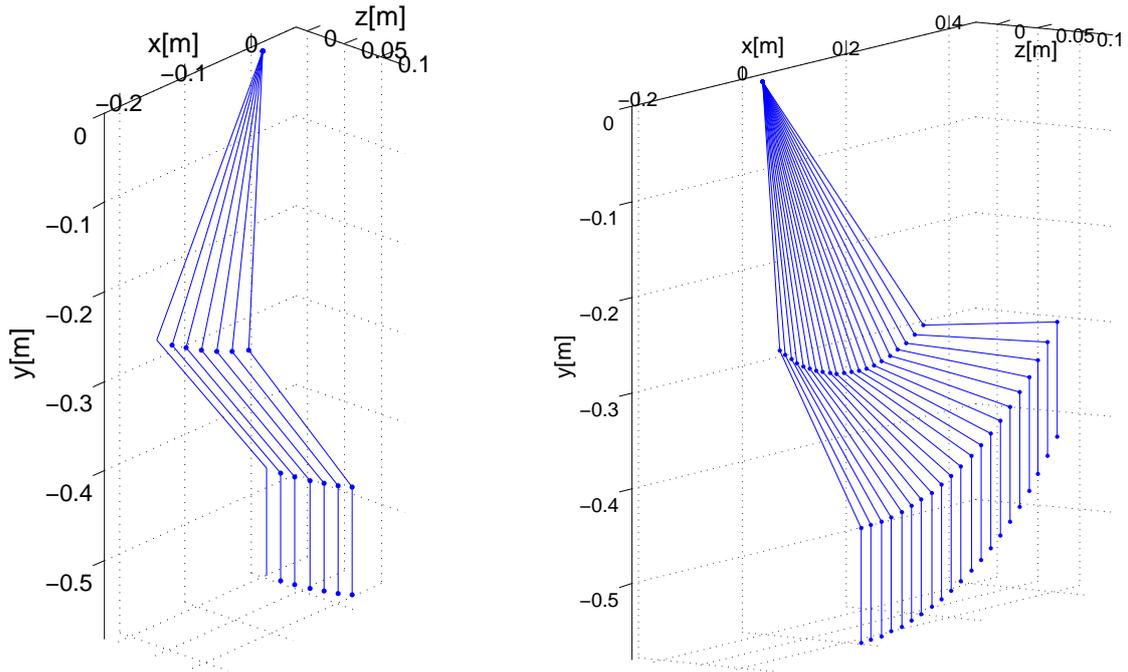
FIGURE 6. Predefined trajectories of end-effector's path.

Points	Coordinates (x, y, z) [m]
P0	(0.00 -0.60 0.00)
P1	(0.00 -0.59 0.12)
P2	(0.40 -0.45 0.12)
P3	(0.00 -0.59 0.12)
P4	(0.00 -0.60 0.00)
P5	(0.23 -0.50 0.23)
P6	(0.32 -0.01 0.32)
P7	(0.23 -0.50 0.23)
P8	(0.00 -0.60 0.00)
P9	(0.00 -0.59 0.12)
P10	(0.00 -0.59 0.12)
P11	(0.00 -0.60 0.00)

TABLE 3. Coordinate points of each limit path section.

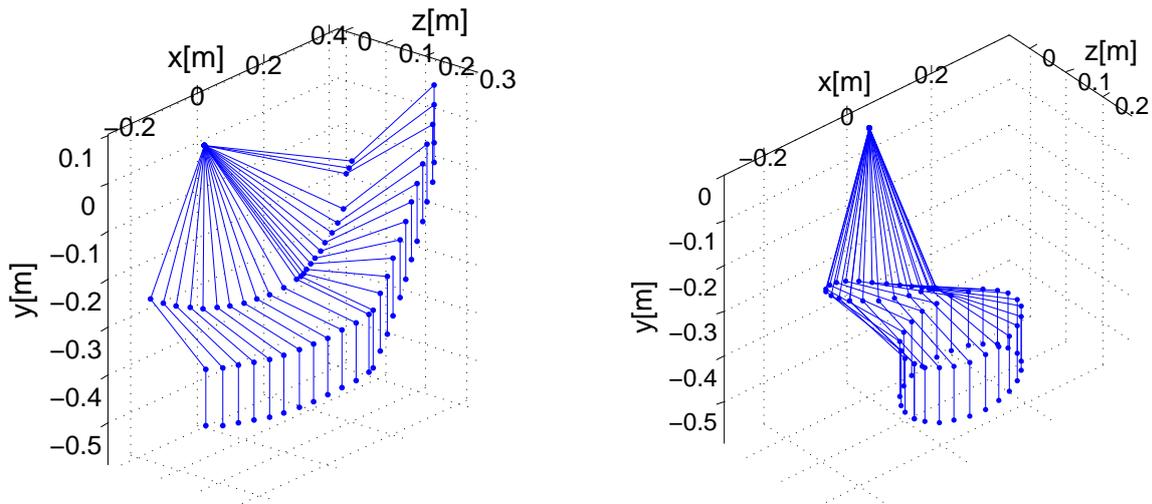
A simulation is used to assess the effectiveness of the current method. Specifically, the angles obtained for the six joint articulations using the current method are simulated using Forward Kinematics (FK) equations within the developed MATLAB program. The robot's end-effector demonstrates the capability to accurately follow the prescribed trajectories, as shown in Figure 7.

From Figure 7, it can be seen that the robot's end-effector has followed exactly the predefined trajectory that is divided into ten paths as numbered above in Figure 6.



(A). Forward P0 → P1, P8 → P9 (Returned P3 → P4, P10 → P11).

(B). Forward P1 → P2 (Returned P2 → P3).



(c). Forward P4 → P5 → P6 (Returned P6 → P7 → P8).

(D). Movement P9 → P10.

FIGURE 7. MATLAB simulations of the robot’s end-effector following predefined trajectories.

6. EXPERIMENTS

In this study, experiments have been conducted to demonstrate the applicability of the proposed approach for solving the Inverse Kinematics (IK) problem. In order to achieve this objective, by exploring the results of IK solutions given in Section 4, an Arduino code has been developed and implemented into an Arduino board 2650. This board serves as an Input/output device between the computer and the robot’s actuators, as shows in the schematic circuit of the control and connection between devices in Figure 8.

Tools and materials for the experiment:

The design of the robot was created using SolidWorks software and its parts were 3D printed using a 3D printer. The robot is equipped with six stepper motors at the joint articulations, each controlled by a dedicated motor driver. The control of these motors is managed by an Arduino board Mega 2650, which is programmed accordingly. The setup also includes a Ramps 1.4 board and power supplies that provides both 12 V and 24 V.

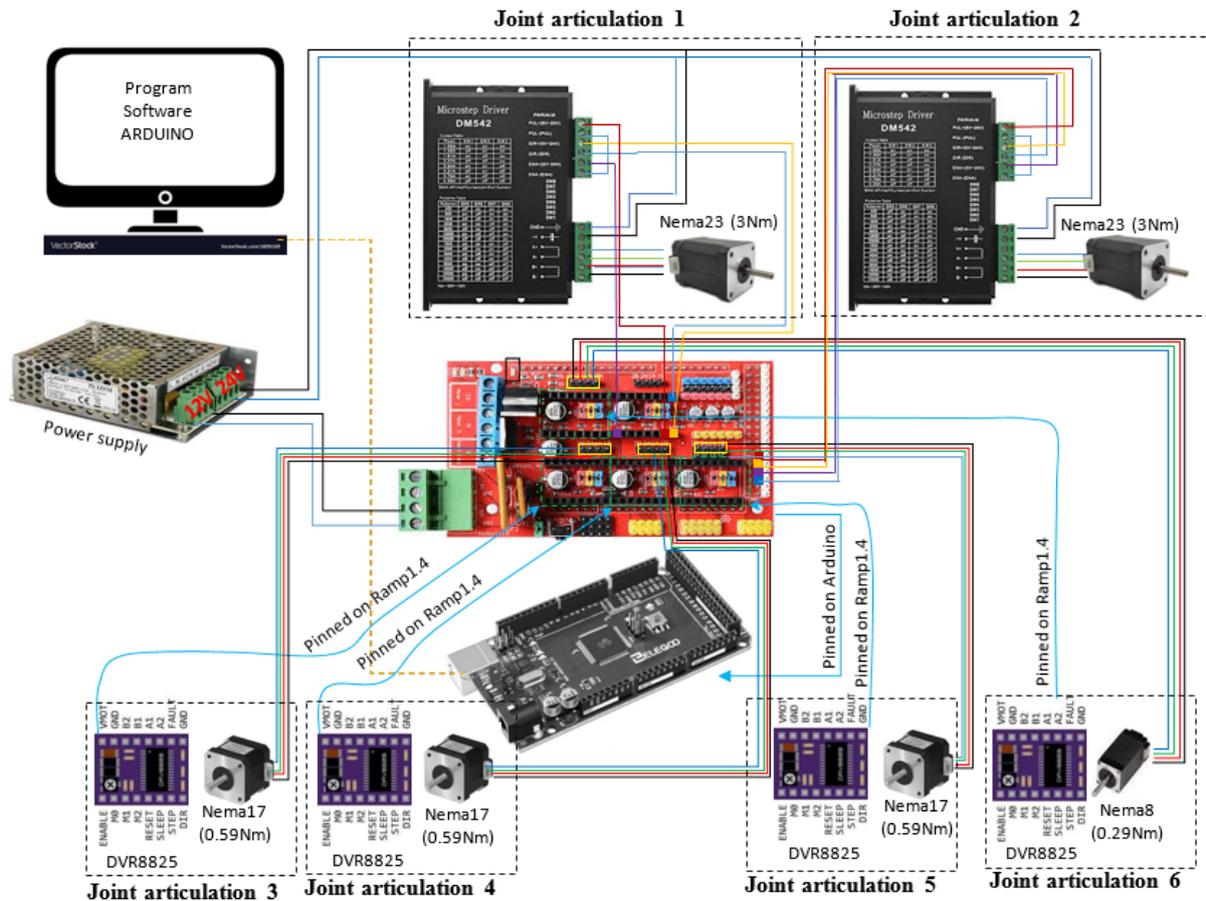


FIGURE 8. Schematic circuit of control and connection between devices.

Figure 8 shows a schematic diagram of the main devices used to control the robots and their corresponding wiring.

As part of further validation, we deliberately instructed the robot to return along the same path in certain cases. This included cases like $P1 \rightarrow P2$ and $P2 \rightarrow P3$, as well as $P5 \rightarrow P6$ and $P6 \rightarrow P7$. In practice, the angle equations of the six joint articulations derived from the current method were directly used in the programmed Arduino code through the Mega2560 control board, after configuring the relevant robot parameters.

Based on the experimental observation, it's evident that the robot can successfully follow the predefined trajectories. Figure 9 displays various poses of the robot as it tracks the trajectories assigned to the end-effector. The six motors of the robot effectively respond to the signals generated by the Inverse Kinematics (IK) solution. Consequently, the IK solutions obtained are validated. However, it's worth noting that in this case, no position control system is implemented on the joint articulations. As a result, slight fluctuations in position and orientation of the end-effector can be observed in some instances.

Figure 10 illustrates the variations in the six articulations of the robot as it moves along the ten path segments.

It can clearly be seen that the variations have regularities, notably in the case where the robot returned along the same path ($P1 \rightarrow P2$ with $P2 \rightarrow P3$, $P0 \rightarrow P1$ with $P3 \rightarrow P4$, ...). Also, in the case of the circular path ($P9 \rightarrow P10$), each of the six motors start and end at the same points.

In fact, there is a compatibility between the angular variations and the parts of the trajectories as suggested in Figure 6, and further confirmation is deduced when the robot comes back on the same path. For example, in the case when the robot followed the path from $P5 \rightarrow P6$ and retraced the same way defined in the graph via points $P6 \rightarrow P7$. It is evident from the graph that the six joint angle variations exhibit perfect symmetry with respect to point P6. This indicates that the robot retraced the path with great precision.

This proves the compatibility between variations of articulations and the predefined trajectories as suggested in Figure 6.

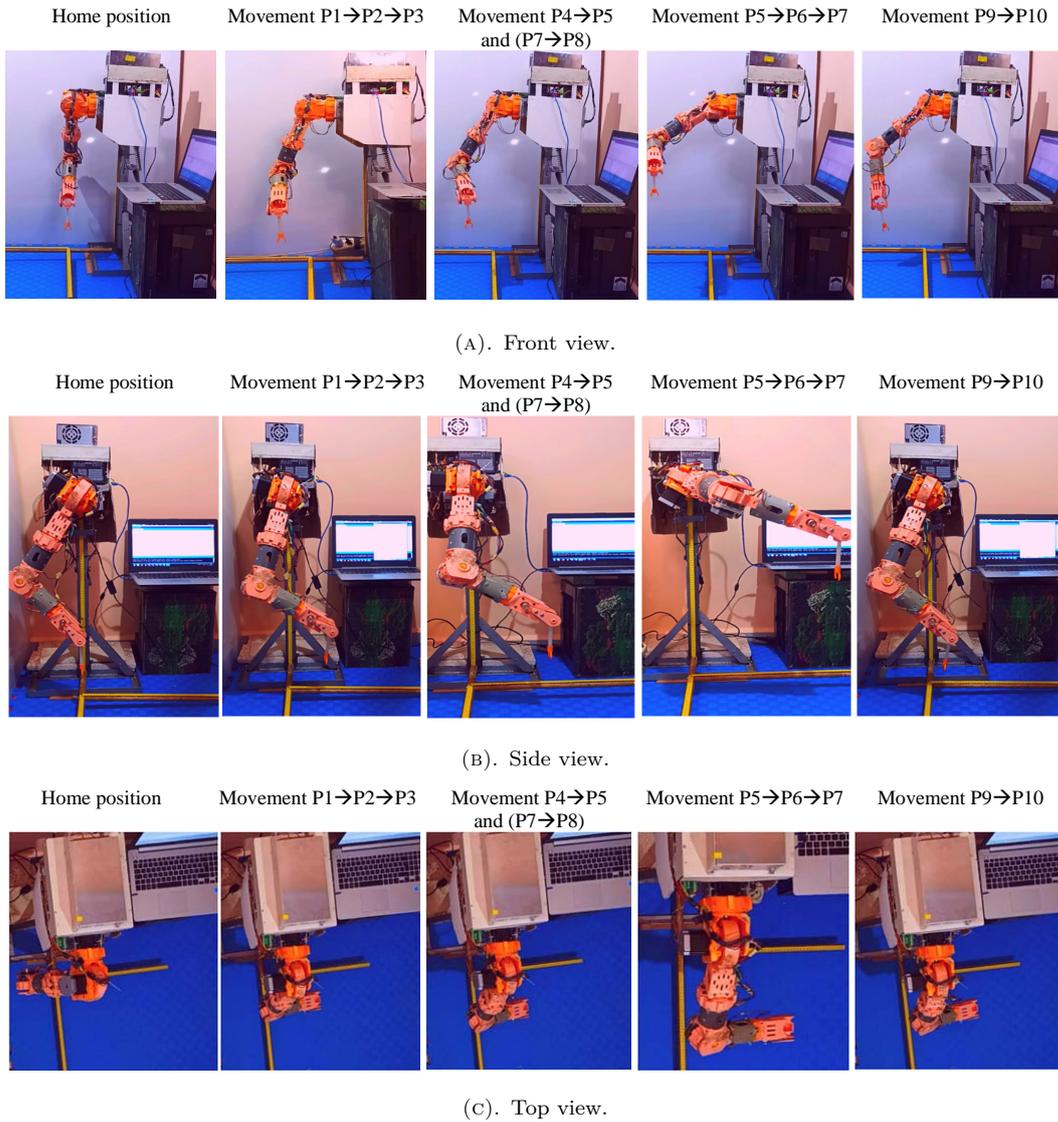


FIGURE 9. Different postures of the robot following predefined trajectories.

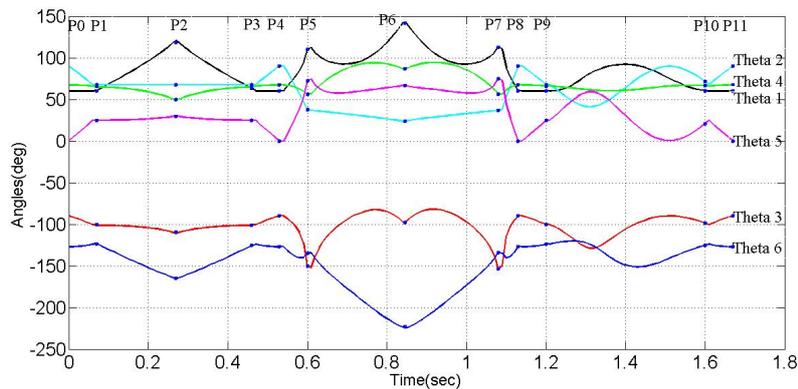


FIGURE 10. Graphs of the variations of the robot's joint angles.

7. COMPARISON AND DISCUSSION

Paul's method is recognised as one of the most effective analytical approaches for solving the Inverse Kinematics (IK) problem in robotics. It starts with the Forward Kinematics (FK) model as given in Equation (7), which leads to a system of nonlinear and coupled equations. For example, the IK problem for a robot with a 6-DOF

configuration similar to ours has been studied and presented in [16] and [32] using Paul's method. Here are the initial key results of the generated system of equations, as shown in Equations (29).

$$\begin{aligned}
& S1.S2.S3.C4.C5.C6 + C1.C3.C4.C5.S6 - S1.C2.S4.C5.S6 + S1.S2.C3.S5.S6 - \\
& C1.S3.S5.S6 + S1.S2.S3.S4.C6 + C1.C3.S4.C6 + S1.C2.C4.C6 = s_x, \\
& S1.S2.S3.C4.C5.C6 + C1.C3.C4.C5.C6 - 21.C2.S4.C5.C6 + S1.S2.C3.S5.C6 - \\
& C1.S3.S5.C6 - S1.S2.S3.S4.S6 - C1.C3.S4.S6 - S1.C2.C4.S6 = n_x, \\
& -S1.S2.S3.C4.S5 - C1.C3.C4.S5 - S1.C2.S4.S5 + S1.S2.C3.C5 - C1.S3.C5 = a_x, \\
& L_a.S1.S2.S3.S4 + L_a.C1.C3.S4 + L_a.S1.C2.S4 + L_f.S1.C2 = s_y, \\
& C1.S2.S3.C4.C5.C6 + S1.C3.C4.C5.S6 + C1.C2.S4.C5.S6 - C1.S2.C3.S5.S6 - \\
& S1.S3.S5.S6 - C1.S2.S3.S4.C6 + S1.C3.S4.C6 - C1.C2.C4.C6 = n_y, \\
& -C1.S2.S3.C4.C5.C6 + S1.C3.C4.C5.C6 + C1.C2.S4.C5.C6 - C1.S2.C3.S5.C6 - \\
& S1.S3.S5.C6 + C1.S2.S3.S4.S6 - S1.C3.S4.S6 + C1.C2.C4.S6 = a_y, \\
& C1.S2.S3.C4.S5 - S1.C3.C4.S5 - C1.C2.S4.S5 - C1.S2.C3.C5 - S1.S3.C5 = s_z, \\
& -L_a.C1.S2.S3.S4 + L_a.S1.C3.S4 - L_a.C1.C2.C4 - L_f.C1.C2 = n_z, \\
& C2.S3.C4.C5.C6 + S2.S4.C5.S6 + C2.C3.S5.S6 + C2.S3.S4.C6 - S2.C4.C6 = a_z, \\
& C2.S3.C4.C5.C6 + S2.S4.C5.C6 + C2.C3.S5.C6 - C2.S3.S4.S6 + S2.C4.S6 = P_{dx}, \\
& -C2.C3.C4.S5 - S2.S4.S5 + C2.C3.C5 = P_{dy}, \\
& -L_a.C2.S3.S4 - L_a.S2.C4 - L_f.S2 = P_{dz}, \tag{29}
\end{aligned}$$

where S_i and C_i are $\sin(\theta_i)$ and $\cos(\theta_i)$ for $i = 1, \dots, 6$, respectively.

To obtain solutions for the articular space variables $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$, Paul's method relies on complex mathematical operations and rearrangements to isolate and identify these variables one by one [16].

How do we obtain the solutions when comparing the two methods and evaluating the Inverse Kinematics (IK) problem? We observe that the solutions of the present approach originate from simple linear equations. In contrast, Paul's method obtains solutions from more complex equations. This distinction becomes evident when considering Equations (23) to (28) for the present approach and Equations (29) for Paul's method.

Figure 11 provides information on the errors in position for each point (defined by its coordinates in the Cartesian coordinate system: x, y, z) of the end-effector. By using the results of the end-effector configurations within the work space that are generated through the FK equations for both methods (Paul's method and the current approach) and by calculating the corresponding errors in the coordinate position of the end-effector ($x_{error}, y_{error}, z_{error}$) during the time for all paths, as shown in Figure 11. We find that the errors are exceptionally negligible, with the largest error value estimated to be in the order of 1×10^{-15} m.

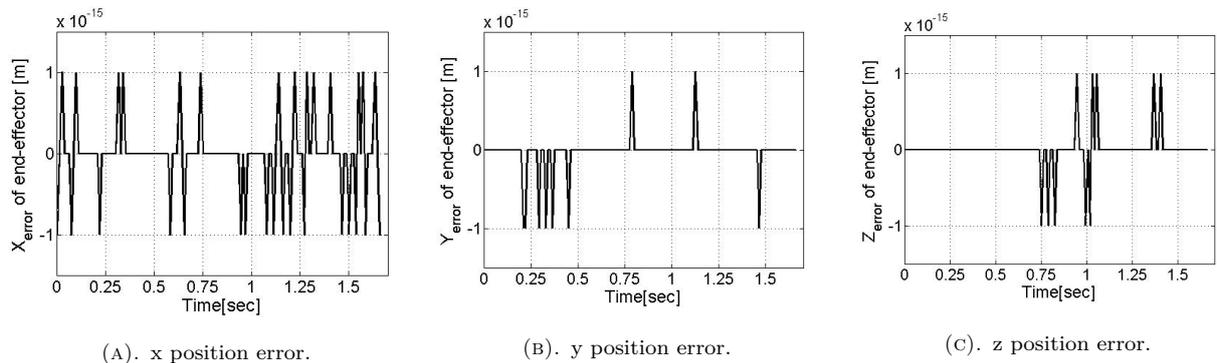


FIGURE 11. Comparison of end-effector position between Paul's method and current approach.

Consequently, it is evident that the current approach has significant accuracy and advantage in obtaining the solutions to the IK problem.

The elapsed time in Matlab was also taken into account for both methods as shown in Figure 12.

From Figure 12, we can see the time elapsed by the two methods separately in each portion of the trajectory, and we can also deduce the total time elapsed during all the robot's movements.

The total elapsed time in Paul's method and the vectorial approach is 1.59 s and 1.86 s, respectively. From these results, it can be concluded that Paul's method is slightly quicker than the vectorial method. This

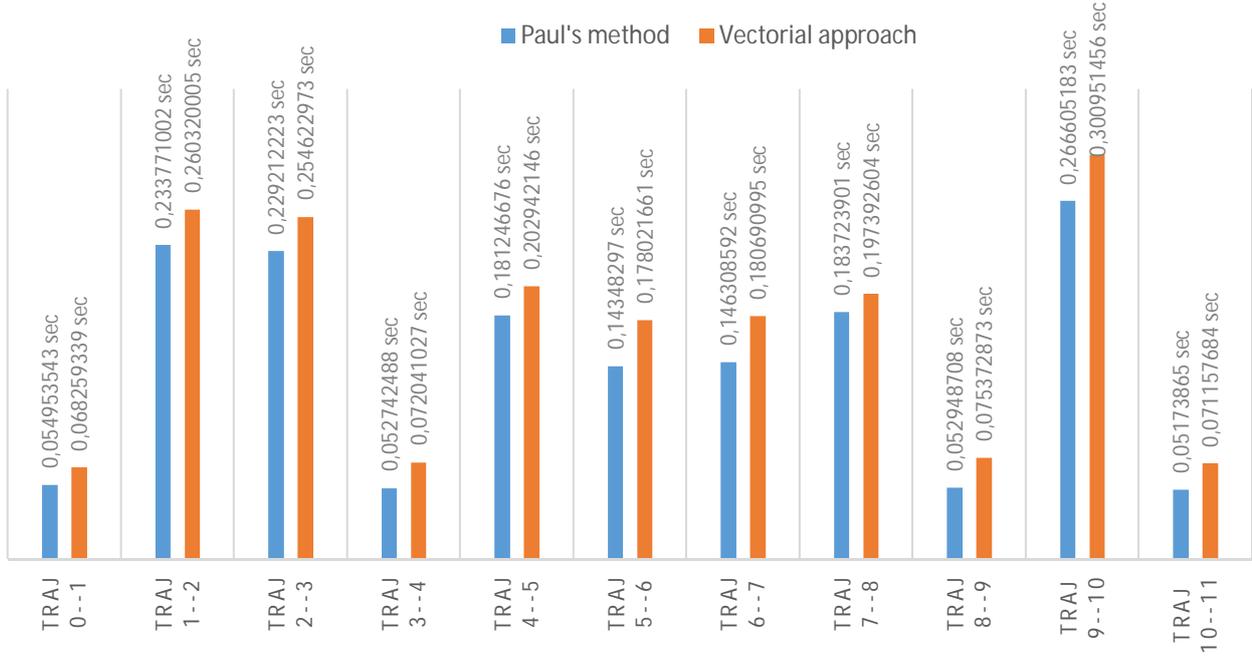


FIGURE 12. Elapsed time [sec] required for the robot's IK solution tracking the path in Matlab.

difference in time comes from the necessity of pre-calculating specific geometric parameters for each situation of the end-effector, which are utilised in the equations that provide the solutions for the joints (angles) of the robot.

From the simulations and the experimental results, we can conclude that the present approach is a powerful asset for a geometric solution of the inverse kinematics problem.

8. CONCLUSION

This study proposes an alternative approach to solving the inverse kinematics problem. The approach is based on vector calculus using spatial geometry and trigonometric relationships. The validity of the approach was confirmed on a 6-DOF robot. By utilising the geometrical lengths of the robot and the given matrix of the desired position along with certain geometrical relationships, all the joint angles were calculated. A major advantage of the proposed approach is that it treats the inverse kinematic problem separately for each joint angle, and the resulting equations have a linear form, which simplifies the mathematical model. Experimental studies were carried out on a designed and manufactured robot. In addition, simulations were performed to test and validate the effectiveness and correctness of the approach. The joint angles were obtained and utilised for various desired trajectories. In both scenarios (experimental and simulation), the end-effector successfully followed the desired trajectories, confirming the accuracy of the suggested approach.

Given these advantages, the approach developed in this work for solving the inverse kinematics problem represents an alternative solution that could find widespread use in various robotic structures.

LIST OF SYMBOLS

- DH* Denavit Hartenberg convention
- FK* Forward Kinematics
- IK* Inverse Kinematics
- DOF* Degrees of freedom

- $L_a, L_f \text{ and } L_h$ Geometrical lengths of the robot, namely: arm, forearm and hand [m]
- $F_i (i = 1, \dots, 7)$ Reference Frame
- $T = [S \ N \ A \ P]$ Matrix (4x4) of the desired orientation and position of the End-effector
- $\vec{OP}_w = (P_{wx}, P_{wy}, P_{wz})^T$ Vector of wrist-position
- $\vec{OP}_d = (P_{dx}, P_{dy}, P_{dz})^T$ Vector of desired position (end-effectors)
- \vec{V}_{se} Vector defined by two points (shoulder-elbow)
- \vec{V}_{wh} Vector defined by two points (wrist-h: the nearest point on the *Q*-plane to origin)
- $\vec{n}_{se} \text{ and } \vec{n}_{wh}$ Normalised vectors of \vec{V}_{se} and \vec{V}_{wh} , respectively

- $\Delta(P_s, P_h, e)$ Triangle of three points
 $Q - plane$ Plane of normal vector \vec{n}_q
 $\det[\vec{A}_6 \vec{n}_{wh} \vec{S}_6]$ Determinant of three vectors
 $(P_e, \widehat{P}_w, P_h)$ Angle by three points in which the midpoint is its vertex
 S_i and C_i for $i = 1, \dots, 6$ $\sin(\theta_i)$ and $\cos(\theta_i)$, respectively
 $\vec{U} \cdot \vec{V}$ Dot product of vectors \vec{U} and \vec{V}
 $\vec{U} \wedge \vec{V}$ Cross product of vectors \vec{U} and \vec{V}

REFERENCES

- [1] H. Ren, P. Ben-Tzvi. Learning inverse kinematics and dynamics of a robotic manipulator using generative adversarial networks. *Robotics and Autonomous Systems* **124**(8):379–386, 2020. <https://doi.org/10.1016/j.robot.2019.103386>
- [2] M. Alebooyeh, R. J. Urbanic. Neural network model for identifying workspace, forward and inverse kinematics of the 7-DOF YuMi 14000 ABB collaborative robot. *IFAC-PapersOnLine* **52**(10):176–181, 2019. <https://doi.org/10.1016/j.ifacol.2019.10.019>
- [3] J. Li, H. Yu, N. Y. Shen, et al. A novel inverse kinematics method for 6-DOF robots with non-spherical wrist. *Mechanism and Machine Theory* **157**:104180, 2021. <https://doi.org/10.1016/j.mechmachtheory.2020.104180>
- [4] M. Toz. Chaos-based Vortex Search algorithm for solving inverse kinematics problem of serial robot manipulators with offset wrist. *Applied Soft Computing* **89**(C):106074, 2020. <https://doi.org/10.1016/j.asoc.2020.106074>
- [5] Z. Liao, G. Jiang, F. Zhao, et al. A novel solution of inverse kinematic for 6R robot manipulator with offset joint based on screw theory. *International Journal of Advance Robotic Systems* **17**(3):1–12, 2020. <https://doi.org/10.1177/1729881420925645>
- [6] J. S. Toquica, P. S. Oliveira, W. S. R. Souza, et al. An analytical and a Deep Learning model for solving the inverse kinematic problem of an industrial parallel robot. *Computers & Industrial Engineering* **151**:106682, 2021. <https://doi.org/10.1016/j.cie.2020.106682>
- [7] H. Ye, D. Wang, J. Wu, et al. Forward and inverse kinematics of a 5-DOF hybrid robot for composite material machining. *Robotics and Computer-Integrated Manufacturing* **65**(1):101961, 2020. <https://doi.org/10.1016/j.rcim.2020.101961>
- [8] O. M. Omisore, S. Han, L. Ren, et al. Deeply-learned damped least-squares (DL-DLS) method for inverse kinematics of snake-like robots. *Neural Networks* **107**(2):34–47, 2020. <https://doi.org/10.1016/j.neunet.2018.06.018>
- [9] M. LiBretto, Y. Qiu, E. Kim, et al. Singularity-free solutions for inverse kinematics of degenerate mobile robots. *Mechanism and Machine Theory* **153**(6):103988, 2020. <https://doi.org/10.1016/j.mechmachtheory.2020.103988>
- [10] M. U. Atique, R. I. Sarker, A. R. Ahad. Development of an 8DOF quadruped robot and implementation of inverse kinematics using Denavit-Hartenberg convention. *Heliyon* **4**(12):e01053, 2018. <https://doi.org/10.1016/j.heliyon.2018.e01053>
- [11] A. El-Sherbiny, M. A. Elhosseini, A. Y. Haikal. A comparative study of soft computing methods to solve inverse kinematics problem. *Ain Shams Engineering Journal* **9**(4):2535–2548, 2018. <https://doi.org/10.1016/j.asej.2017.08.001>
- [12] S. Starke, N. Hendrich, J. Zhang. A forward kinematics data structure for efficient evolutionary inverse kinematics. *Mechanisms and Machine Science* **50**:560–568, 2017. https://doi.org/10.1007/978-3-319-60867-9_64
- [13] Y. Lin, H. Min. Inverse kinematics of modular manipulator robot with shoulder offset based on geometric method mixed with analytical method algorithm. In *IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 1198–1203. 2015. <https://doi.org/10.1109/CYBER.2015.7288114>
- [14] A. Aristidou, J. Lasenby. FABRIK: A fast, iterative solver for the Inverse Kinematics problem. *Graphical Models* **73**(5):243–260, 2011. <https://doi.org/10.1016/j.gmod.2011.05.003>
- [15] W. Suleiman, F. Kanehiro, E. Yoshida. Infeasibility-free inverse kinematics method. In *IEEE/SICE International Symposium on System Integration (SII)*, pp. 307–312. 2015. <https://doi.org/10.1109/SII.2015.7404996>
- [16] W. Khalil, E. Dombre. Transformation matrix between vectors, frames and screws. In *Modeling, identification and control of robots*, pp. 13–34. Butterworth-Heinemann, London, UK, 2002. <https://doi.org/10.1016/B978-190399666-9/50002-6>
- [17] M. Raghavan, B. Roth. Inverse kinematics of the general 6R manipulator and related linkages. *Journal of Mechanical Design* **115**(3):502–508, 1993. <https://doi.org/10.1115/1.2919218>
- [18] Y. Lou, P. Quan, H. Lin, et al. A closed-form solution for the inverse kinematics of the 2n-DOF hyper-redundant manipulator based on general spherical joint. *Applied Sciences* **11**(3):1277–1285, 2021. <https://doi.org/10.3390/app11031277>
- [19] J. Q. Gan, E. Oyama, E. M. Rosales, H. Hu. A complete analytical solution to the inverse kinematics of the Pioneer 2 robotic arm. *Journal Robotica* **23**(1):123–129, 2005. <https://doi.org/10.1017/S0263574704000529>

- [20] B. Tondu. A closed-form inverse kinematic modelling of a 7R anthropomorphic upper limb based on a joint parametrization. In *6th IEEE-RAS International Conference on Humanoid Robots*, pp. 390–397. 2006. <https://doi.org/10.1109/ICHR.2006.321302>
- [21] S. Bertrand, O. Bruneau, F. B. Oueddou, S. Alfayad. Closed-form solutions of inverse kinematic models for the control of a biped robot with 8 active degrees of freedom per leg. *Mechanism and Machine Theory* **49**:117–140, 2012. <https://doi.org/10.1016/j.mechmachtheory.2011.10.014>
- [22] L. Sardana, M. K. Sutar, P. M. Pathak. A geometric approach for inverse kinematics of a 4-link redundant In-Vivo robot for biopsy. *Robotics and Autonomous Systems* **61**(12):1306–1313, 2013. <https://doi.org/10.1016/j.robot.2013.09.001>
- [23] I. Sancaktar, B. Tuna, M. Ulutas. Inverse kinematics application on medical robot using adapted PSO method. *Engineering Science and Technology, an International Journal* **21**(5):1006–1010, 2018. <https://doi.org/10.1016/j.jestch.2018.06.011>
- [24] L. Sciavicco, B. Siciliano. A dynamic solution to the inverse kinematic problem for redundant manipulators. In *IEEE International Conference on Robotics and Automation*, vol. 4, pp. 1081–1087. 1987. <https://doi.org/10.1109/ROBOT.1987.1087921>
- [25] G. Tevatia, S. Schaal. Inverse kinematics for humanoid robots. In *IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, pp. 294–299. 2000. <https://doi.org/10.1109/ROBOT.2000.844073>
- [26] R. Pérez-Rodríguez, A. Marcano-Cedeño, U. Costa, et al. Inverse kinematics of a 6 DoF human upper limb using ANFIS and ANN for anticipatory actuation in ADL-based physical neurorehabilitation. *Expert Systems with Applications* **39**(10):9612–9622, 2012. <https://doi.org/10.1016/j.eswa.2012.02.143>
- [27] F. Chapelle, P. Bidaud. A closed form for inverse kinematics approximation of general 6r manipulators using genetic programming. In *IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 4, pp. 3364–3369. 2001. <https://doi.org/10.1109/ROBOT.2001.933137>
- [28] M. Jin, L. Qiang, B. Wang, H. Liu. An efficient and accurate inverse kinematics for 7-DOF redundant manipulators based on a hybrid of analytical and numerical method. *IEEE Access* **8**:16316–16330, 2020. <https://doi.org/10.1109/ACCESS.2020.2966768>
- [29] W. Yan, H. Lu-Bin, Y. Ting-Li. Inverse kinematics analysis of general 6R serial robot mechanism based on Groebner base. *Frontiers of Mechanical Engineering in China* **1**:115–124, 2006. <https://doi.org/10.1007/s11465-005-0022-7>
- [30] Y. Wenbin, S. Lei. An optimization method for inverse kinematics of a 7-DOF redundant manipulator. In *34th Chinese Control Conference (CCC)*, pp. 4472–4479. 2015. <https://doi.org/10.1109/ChiCC.2015.7260331>
- [31] B. Ma, Z. Xie, Z. Jiang, H. Liu. Precise semi-analytical inverse kinematic solution for 7-DOF offset manipulator with arm angle optimization. *Frontiers of Mechanical Engineering* **16**:435–450, 2021. <https://doi.org/10.1007/s11465-021-0630-x>
- [32] R. O’Flaherty, P. Vieira, M. Grey, et al. Kinematics and inverse kinematics for humanoid robot HUBO2+.GT-GOLEM-001. Technical report, Georgia Institute of Technology, 2013.
- [33] T. Lyche. A short review of linear algebra. In *Numerical linear algebra and matrix factorizations*, vol. 22, pp. 1–24. Springer, Cham, CH, 2020. https://doi.org/10.1007/978-3-030-36468-7_1

A. APPENDICES

A.1. TRIGONOMETRIC FUNCTION $\text{atan2}(y, x)$

$\text{atan2}(y, x)$, it is a helpful trigonometric function utilized especially by the most computer languages. Because, it is summarizing the four quadrants of trigonometric circle and consequently, gives the corresponding angles of any quadrant.

Basically, it works with polar coordinates by converting the cartesian coordinates (x, y) to polar coordinates (r, θ) which mean that $r = \sqrt{x^2 + y^2}$ and $\theta = \text{atan2}(y, x)$, $x = r \cdot \cos \theta$ and $y = r \cdot \sin \theta$.

$$\theta = \text{atan2}(y, x) = \begin{cases} \text{atan}(y/x) & \text{if } x > 0 \\ \text{atan}(y/x) + \pi & \text{if } x < 0 \text{ and } y \geq 0 \\ \text{atan}(y/x) - \pi & \text{if } x < 0 \text{ and } y < 0 \\ \pi/2 & \text{if } x = 0 \text{ and } y > 0 \\ -\pi/2 & \text{if } x = 0 \text{ and } y < 0 \\ \text{undefined} & \text{if } x = 0 \text{ and } y = 0 \end{cases}$$

A.2. PATH OF THE ROBOT’S END-EFFECTOR

Table below contains the ten parts of path that tracked by the robot’s end-effector.

Path	Coordinates	Reverse path	Coordinates	Remakes
0→1	$P_{dx} = 0.0$ $P_{dy} = -\sqrt{0.6^2 - P_{dz}^2}$ $P_{dz} = 0.0 : 0.02 : 0.12$	3→4 10→11 (1→0)	$P_{dx} = 0.0$ $P_{dy} = -\sqrt{0.6^2 - P_{dz}^2}$ $P_{dz} = 0.12 : -0.02 : 0.0$	P_{dz} increments (decrements) by a constant step of 0.02 m
1→2	$P_{dx} = 0.0 : 0.02 : 0.4$ $P_{dy} = -\sqrt{0.59^2 - P_{dx}^2}$ $P_{dz} = 0.12$	2→3 2→1	$P_{dx} = 0.4 : -0.02 : 0.0$ $P_{dy} = -\sqrt{0.59^2 - P_{dx}^2}$ $P_{dz} = 0.12$	The value 0.59 m in P_{dy} represents the last value of path 0→1
4→5	$P_{dx} = 0.6 \sin \alpha + 0.6 \cos \frac{\pi}{4}$ $P_{dy} = -0.6 \cos \alpha$ $P_{dz} = 0.6 \sin \alpha + 0.6 \sin \frac{\pi}{4}$	7→8 5→4	$P_{dx} = 0.6 \sin \alpha + 0.6 \cos \frac{\pi}{4}$ $P_{dy} = -0.6 \cos \alpha$ $P_{dz} = 0.6 \sin \alpha + 0.6 \sin \frac{\pi}{4}$	$\pi \leq \alpha < 3\pi/4$ With a step of: $\Delta\alpha = \frac{\pi}{60}$ rad
5→6	$P_{dx} = P_{dx} + \Delta P_{dx}$ $P_{dy} = -0.6 \cos \alpha$ $P_{dz} = P_{dz} + \Delta P_{dz}$	8→9 (6→5)	$P_{dx} = P_{dx} - \Delta P_{dx}$ $P_{dy} = -0.6 \cos \alpha$ $P_{dz} = P_{dz} - \Delta P_{dz}$	P_{dx} starts from last value of path 4→5 until $P_{dx} = 0.3$ m. same thing to P_{dz} . For P_{dy} : $\frac{3\pi}{4} < \alpha < \pi/2$ and step: $\Delta\alpha = \frac{\pi}{60}$ rad
9→10	$\begin{cases} P_{Cx} = Q_{Ox} + R \cos(\phi)S_{Qx} + R \sin(\phi)N_{Qx} \\ P_{Cy} = Q_{Oy} + R \cos(\phi)S_{Qy} + R \sin(\phi)N_{Qy} \\ P_{Cz} = Q_{Oz} + R \cos(\phi)S_{Qz} + R \sin(\phi)N_{Qz} \end{cases}$ $R = 0.12$ m: raidus of circle $Q_Q = (0.12, -0.58, 0.12)$: center coordinates of circle $\vec{S}_Q = (0.65, 0.00, -0.65)^T$ $\vec{N}_Q = (0.65, 0.23, 0.65)^T$ $\vec{S}_Q = (S_{Qx}, S_{Qy}, S_{Qz})^T$ and $\vec{N}_Q = (N_{Qx}, N_{Qy}, N_{Qz})^T$ are the direction cosines of the frame axes along the C-plane in the reference frame F_0			Circularepath which is located on a plane of equation: $0.19x - 0.94y + 0.19z - 0.365 = 0$ And normal vector: $\vec{n}(0.19, -0.94, 0.19)$ $0 \leq \phi \leq 2\pi$ of step $\Delta\phi = \frac{\pi}{60}$ rad

A.3. INVERSE KINEMATICS SOLUTIONS USING PAUL'S METHOD

Equation (A1) represents the forward kinematics of the robot. It is a successive multiplication of different homogeneous transformation matrices.

$${}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 {}^6T_7 = [\mathbf{S N A P}] = \begin{bmatrix} s_x & n_x & a_x & P_{dx} \\ s_y & n_y & a_y & P_{dy} \\ s_z & n_z & a_z & P_{dz} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (\text{A1})$$

Calculate the inverse of Equation (A1):

$${}^7T_6 {}^6T_5 {}^5T_4 {}^4T_3 {}^3T_2 {}^2T_1 {}^1T_0 = \begin{bmatrix} s_1 & n_1 & a_1 & P_1 \\ s_2 & n_2 & a_2 & P_2 \\ s_3 & n_3 & a_3 & P_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{A2})$$

where:

$$\begin{aligned} s_1 &= \frac{(a_2 \cdot n_3 - a_3 \cdot n_2)}{(a_x \cdot n_y \cdot s_z - a_x \cdot n_z \cdot s_y - a_y \cdot n_x \cdot s_z + a_y \cdot n_z \cdot s_x + a_z \cdot n_x \cdot s_y - a_z \cdot n_y \cdot s_x)}, \\ s_2 &= \frac{-(a_2 \cdot s_3 - a_3 \cdot s_2)}{(a_x \cdot n_y \cdot s_z - a_x \cdot n_z \cdot s_y - a_y \cdot n_x \cdot s_z + a_y \cdot n_z \cdot s_x + a_z \cdot n_x \cdot s_y - a_z \cdot n_y \cdot s_x)}, \\ s_3 &= \frac{(n_2 \cdot s_3 - n_3 \cdot s_2)}{(a_x \cdot n_y \cdot s_z - a_x \cdot n_z \cdot s_y - a_y \cdot n_x \cdot s_z + a_y \cdot n_z \cdot s_x + a_z \cdot n_x \cdot s_y - a_z \cdot n_y \cdot s_x)}, \\ n_1 &= \frac{-(a_x \cdot s_z - a_z \cdot s_x)}{(a_x \cdot n_y \cdot s_z - a_x \cdot n_z \cdot s_y - a_y \cdot n_x \cdot s_z + a_y \cdot n_z \cdot s_x + a_z \cdot n_x \cdot s_y - a_z \cdot n_y \cdot s_x)}, \\ n_2 &= \frac{(a_x \cdot s_z - a_z \cdot s_x)}{(a_x \cdot n_y \cdot s_z - a_x \cdot n_z \cdot s_y - a_y \cdot n_x \cdot s_z + a_y \cdot n_z \cdot s_x + a_z \cdot n_x \cdot s_y - a_z \cdot n_y \cdot s_x)}, \\ n_3 &= \frac{-(n_x \cdot s_z - n_z \cdot s_x)}{(a_x \cdot n_y \cdot s_z - a_x \cdot n_z \cdot s_y - a_y \cdot n_x \cdot s_z + a_y \cdot n_z \cdot s_x + a_z \cdot n_x \cdot s_y - a_z \cdot n_y \cdot s_x)}, \\ a_1 &= \frac{(a_x \cdot n_z - a_z \cdot n_x)}{(a_x \cdot n_y \cdot s_z - a_x \cdot n_z \cdot s_y - a_y \cdot n_x \cdot s_z + a_y \cdot n_z \cdot s_x + a_z \cdot n_x \cdot s_y - a_z \cdot n_y \cdot s_x)}, \\ a_2 &= \frac{-(a_x \cdot s_y - a_y \cdot s_x)}{(a_x \cdot n_y \cdot s_z - a_x \cdot n_z \cdot s_y - a_y \cdot n_x \cdot s_z + a_y \cdot n_z \cdot s_x + a_z \cdot n_x \cdot s_y - a_z \cdot n_y \cdot s_x)}, \\ a_3 &= \frac{(n_x \cdot s_z - n_z \cdot s_x)}{(a_x \cdot n_y \cdot s_z - a_x \cdot n_z \cdot s_y - a_y \cdot n_x \cdot s_z + a_y \cdot n_z \cdot s_x + a_z \cdot n_x \cdot s_y - a_z \cdot n_y \cdot s_x)}, \\ P_1 &= \frac{-(P_{dx} \cdot a_y \cdot n_z - P_{dx} \cdot a_z \cdot n_y - P_{dy} \cdot a_x \cdot n_z + P_{dy} \cdot a_z \cdot n_x + P_{dz} \cdot a_x \cdot n_y - P_{dz} \cdot a_y \cdot n_x)}{(a_x \cdot n_y \cdot s_z - a_x \cdot n_z \cdot s_y - a_y \cdot n_x \cdot s_z + a_y \cdot n_z \cdot s_x + a_z \cdot n_x \cdot s_y - a_z \cdot n_y \cdot s_x)}, \\ P_2 &= \frac{(P_{dx} \cdot a_y \cdot s_z - P_{dx} \cdot a_z \cdot s_y - P_{dy} \cdot a_x \cdot s_z + P_{dy} \cdot a_z \cdot s_x + P_{dz} \cdot a_x \cdot s_y - P_{dz} \cdot a_y \cdot s_x)}{(a_x \cdot n_y \cdot s_z - a_x \cdot n_z \cdot s_y - a_y \cdot n_x \cdot s_z + a_y \cdot n_z \cdot s_x + a_z \cdot n_x \cdot s_y - a_z \cdot n_y \cdot s_x)}, \\ P_3 &= \frac{-(P_{dx} \cdot n_y \cdot s_z - P_{dx} \cdot n_z \cdot s_y - P_{dy} \cdot n_x \cdot s_z + P_{dy} \cdot n_z \cdot s_x + P_{dz} \cdot n_x \cdot s_y - P_{dz} \cdot n_y \cdot s_x)}{(a_x \cdot n_y \cdot s_z - a_x \cdot n_z \cdot s_y - a_y \cdot n_x \cdot s_z + a_y \cdot n_z \cdot s_x + a_z \cdot n_x \cdot s_y - a_z \cdot n_y \cdot s_x)}. \end{aligned}$$

The corresponding solutions of inverse kinematics are given below.

A.3.1. THE JOINT ANGLE θ_4

$$\begin{aligned} \cos \theta_4 &= \frac{(P_1 + L_h)^2 + P_2^2 + P_3^2 - L_a^2 - L_f^2}{2 \cdot L_a \cdot L_f}, \\ \theta_4 &= \text{atan2}(\sqrt{1 - \cos^2 \theta_4}, \cos \theta_4). \end{aligned}$$

A.3.2. THE JOINT ANGLE θ_5

$$\begin{aligned} \cos \theta_5 &= \frac{-P_3}{L_a \cdot \sin \theta_4}, \\ \theta_5 &= \text{atan2}(\sin \theta_5, \sqrt{1 - \cos^2 \theta_5}). \end{aligned}$$

A.3.3. THE JOINT ANGLE θ_6

$$\sin \theta_6 = \frac{L_a \cdot \sin \theta_4 \cdot \cos \theta_5 \cdot (L_h + P_1) + (L_a \cdot \cos \theta_4 + L_f) P_2}{P_2^2 + (P_1 + L_h)^2},$$

$$\cos \theta_6 = \frac{L_a \cdot \sin \theta_4 \cdot \cos \theta_5 \cdot P_2 - (L_a \cdot \cos \theta_4 + L_f)(L_h + P_1)}{P_2^2 + (P_1 + L_h)^2}.$$

A.3.4. THE JOINT ANGLE θ_2

$$\sin \theta_2 = -((- \sin \theta_6 \cdot a_1 - \cos \theta_6 \cdot a_2) \cos \theta_5 + \sin \theta_4 \cdot \sin \theta_5 \cdot a_3 + \cos \theta_4 (\cos \theta_6 \cdot a_1 - \sin \theta_6 \cdot a_2)),$$

$$\theta_2 = \text{atan2}(\sin \theta_2, \sqrt{1 - \sin^2 \theta_2}).$$

A.3.5. THE JOINT ANGLE θ_3

$$\sin \theta_3 = \cos \theta_4 (\cos \theta_5 (\sin \theta_6 \cdot a_1 + \cos \theta_6 \cdot a_2) - \sin \theta_5 \cdot a_3) + \sin \theta_4 (\cos \theta_6 \cdot a_1 - \sin \theta_6 \cdot a_2),$$

$$\cos \theta_3 = \sin \theta_5 (\sin \theta_6 \cdot a_1 + \cos \theta_6 \cdot a_2) + \sin \theta_5 \cdot a_3,$$

$$\theta_3 = \text{atan2}(\sin \theta_3, \cos \theta_3).$$

A.3.6. THE JOINT ANGLE θ_1

$$\sin \theta_1 = \sin \theta_4 (-\cos \theta_5 (\sin \theta_6 \cdot s_1 + \cos \theta_6 \cdot s_2) + \sin \theta_5 \cdot s_3) + \cos \theta_4 (\cos \theta_6 \cdot s_1 - \sin \theta_6 \cdot s_2),$$

$$\cos \theta_1 = -\sin \theta_4 (-\cos \theta_5 (\sin \theta_6 \cdot n_1 + \cos \theta_6 \cdot n_2) + \sin \theta_5 \cdot n_3) + \cos \theta_4 (\cos \theta_6 \cdot n_1 - \sin \theta_6 \cdot n_2),$$

$$\theta_1 = \text{atan2}(\sin \theta_1, \cos \theta_1).$$