KINEMATIC ANALYSIS AND OPTIMIZATION OF A 6DOF ROBOT ARM

BAYO Ogunmola[1, a] and ABDULLAHI Idris[2,b*]

[1]University of Lagos, Lagos Nigeria. ORCID: 0000-0001-5820-4020

[2] University of Lagos, Lagos Nigeria. ORCID: 0009-0007-4183-2451

[a]oogunmola@unilag.edu.ng, [b]abdullahidris.dev@gmail.com

Abstract.

Precise kinematic control is fundamental to the reliable operation of robotic manipulators in dynamic, real-world environments. This paper presents a comprehensive kinematic analysis and optimization of a 6-degree-of-freedom (6DOF) UR5 robotic arm, with the objective of improving operational efficiency across variable manipulation scenarios. A product-of-exponentials forward kinematics model is developed to provide an exact mathematical description of end-effector motion, while gradient descent is employed to solve the inverse kinematics problem numerically. Although a closed-form analytical solution exists for the UR5, its sensitivity to kinematic singularities and joint-limit violations limits its reliability in complex configurations. To address this, a hybrid approach is proposed, wherein the analytical solution serves as an initial estimate for the numerical solver where it converges, and the gradient descent method operates independently in near-singular or infeasible cases. Comparative evaluation of all three methods across ten test scenarios assesses accuracy, feasibility, and computational efficiency. Simulation results show that the hybrid method reduces convergence iterations by up to 78% relative to the purely numerical approach, while maintaining comparable end-effector accuracy, making it well-suited for real-time robotic applications.

Keywords: Forward kinematics; inverse kinematics; 6DOF robotic arm; gradient descent; hybrid optimization; computational efficiency

1. Introduction

In practical situations, a robot arm faces challenges like moving objects, different shapes, and varying placements. To overcome these challenges effectively, it is essential to thoroughly understand how the robot arm moves. Performing a Kinematic analysis of a Robot involves various steps starting with the formulation of the forward kinematics model, followed by solving the inverse kinematics as stated by Kucuk and Bingul (2006), in this research, three different solution methods will be compared which are the Analytic, Numerical using gradient descent algorithm and then an optimized approach.

Forward and inverse kinematics are the two primary subproblems in kinematics. Forward kinematics (FK) calculates the end-effector's position and orientation based on the joint variables. On the other hand, inverse kinematics (IK) determines the necessary joint variables to achieve a desired end-effector position and orientation. Refaai (2022) concurs that while solving forward kinematics is comparatively straightforward, tackling inverse kinematics poses a formidable challenge due to its nonlinearity and the presence of multiple potential solutions.

Numerous studies currently focus on solving the Inverse Kinematics problem purely analytically or numerically, for example, Jiang, Liu, Liu and Huo (2013) proposed a reverse coordinates method for solving inverse kinematics by inverting the base and the end-effector of the manipulator, while Tong, Liu, Liu and Yuan (2021), used an analytical inverse kinematic parameterized method for redundant sliding manipulators. This method solves the inverse kinematics problem analytically and then employs the Newton-Raphson algorithm for secondary adjustments. The method also leverages the unique structural properties of rail-type

redundant sliding manipulators. In further development, Brandstötter, Angerer, and Hofbaur (2014), presented a simplified method for computing the inverse kinematics of common serial manipulator arms with up to 6 degrees of freedom. Their approach focused on the use of essential design dimensions readily available in manufacturing data sheets, eliminating the need for complex Denavit-Hartenberg parameter derivation. Aristidou and Lasenby (2009) reviewed and compared the most popular IK methods based on reliability, computational cost and conversion criteria, with a novel heuristic and iterative method, called Forward and Backward Reaching Inverse Kinematics (FABRIK). On inverse Kinematics, Singh, Baranwal and Nandi (2017), used the gradient descent method to solve the Inverse Kinematics problem of the arms of a 5DOF NAO robot. Eimei Oyama et al (2005), employed a modified Gauss-Newton method to determine the optimal joint angle vector trajectory that minimizes the least-squares error between the initial arm posture and the desired end-effector position and orientation. This approach resulted in an improved modular neural network system capable of accurately modelling the inverse kinematics of a general robotic arm.

Buss (2009), adopted the damped least squares method which is also known as the Levenberg-Marquardt method to avoid many of the problems with singularities when using the pseudo-inverse method's, giving a numerically stable method of selecting $\Delta\theta$ (Joint changes). Lopez-Franco Hernandez-Barragan, Alanis and Arana-Daniel (2018), mentioned several soft computing algorithms that optimizes the numerical approach to solving inverse kinematics problems, the most common of them are Artificial Bee Colony (ABC), Bat Algorithm, Covariance Matrix Adaptation Evolution Strategy, Cuckoo Search, Differential Evolution, Differential Search, Genetic Algorithm and Particle Swarm Optimization. Courty and Arnaud (2008) compared the results of using Artificial Bee Colony (ABC) algorithm with the results of Genetic Algorithm (GA), Particle Swarm Algorithm (PSO) and Particle Swarm Inspired Evolutionary Algorithm (PS-EA) and showed that ABC outperforms the other algorithms. Courty and Arnaud(2008) approached the problem of Inverse Kinematics utilizing the Sequential Monte Carlo methods which had an advantage in avoiding common downsides of numerical inversion methods since only direct calculations were required.

Zhao et al. (2022) introduced an enhanced particle swarm algorithm for solving robot inverse kinematics. Their approach incorporates an adaptive weight adjustment strategy and condition-based settings, considering joint limits and position coefficients in the velocity factor. Evaluating their method on a PUMA560 robotic arm, they compared it with traditional Particle Swarm algorithm (PSO) and quantum particle swarm algorithm (QPSO). The proposed method exhibited superior accuracy in position and orientation solving, along with faster operation time and better algorithm convergence.

Polyzos, Groumpos and Dermatas (2019), assessed a novel approach to solving the inverse kinematics problem based on autoencoders, with the goal of determining the solution that minimizes the deviation of the robot's end-effector from its target position while also limiting joint movement. Liu, Zhou, Lai and Zhu (2013), explored an innovative Inverse Kinematic (IK) algorithm tailored for a PUMA560-structured robot manipulator with the aim of overcoming the limitations of traditional IK algorithms by proposing a novel approach grounded in the orthogonality of rotation submatrix and block matrix multiplication properties.

In choosing the robot model for this study, Liu, Zhou, Lai and Zhu (2013) highlighted that it is imperative for a harvesting robot to possess a minimum of 6 Degrees of Freedom (DOF) to guarantee flexibility in its movement.

2. Methodology

2.1 Configuration, joint and task space of the UR5

The joints of the UR5 are arranged as R-R-R-R-R-R also known as a 6R configuration with S_1, S_2, S_3, S_4, S_5 & S_6 as the screw axis of joints 1-6 corresponding to the z-axis of each joint. The initial configuration is shown in Figure 1 below. The joint space in table 1 below.

UR5 Dimensions

W1 = 109 mm
L1 = 425 mm
H1 = 89 mm

W2 = 82 mm
L2 = 392 mm
H2 = 95 mm

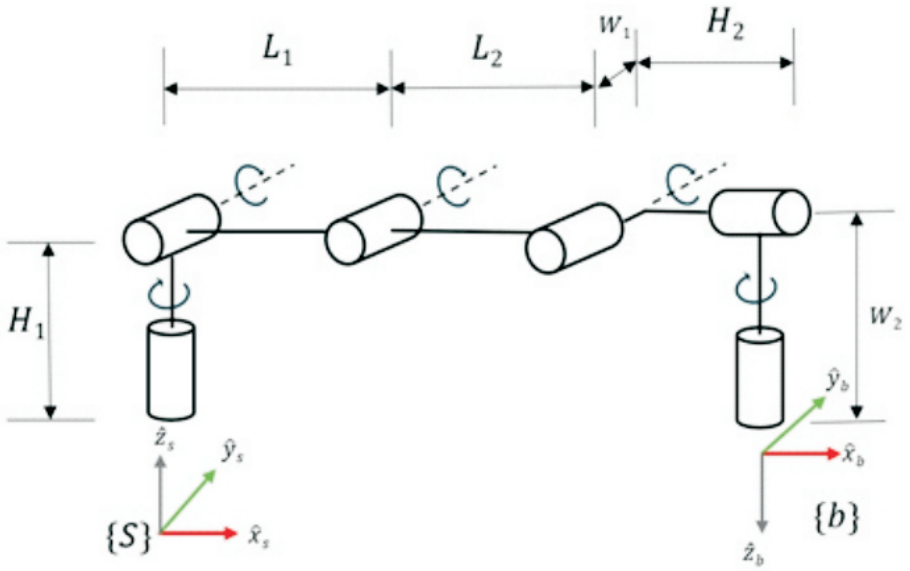

Figure 1. Initial configuration of the UR5 robot arm

Table 1: Joint space & Task Space

Joint Space			Task Space	
Joint	Range (degrees)	Description	Axis	Range
Joint 1	-360° to 360°	Base rotation	X-axis	-0.912m to 0.912m
Joint 2	-90° to 90°	Shoulder rotation	Y-axis	-1.002m to 1.002m
Joint 3	-160° to 160°	Elbow rotation	Z-axis	0.089m to 1.002m
Joint 4	-360° to 360°	Wrist 1 rotation		
Joint 5	-360° to 360°	Wrist 2 rotation		
Joint 6	-360° to 360°	Wrist 3 rotation		

2.2 Forward Kinematics

Let M be the transformation matrix of the end-effector frame {b}with respect to Space frame {S}in the initial configuration, expressed as a combined matrix of R, a 3x3 rotation matrix and P, a 3x1 matrix.

$$M = \begin{bmatrix} [R] & P \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 912 \\ 0 & 1 & 0 & 109 \\ 0 & 0 & -1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Park and Okamura (1994), used the forward kinematics of a spatial 6-DOF open chain which can be expressed using product of exponentials as:

$$T(\theta) = e^{[S]\theta} e^{[S_{...}]\theta_{...}} \ldots e^{[S_i]\theta_i} M \text{ where } i = 1,2,\ldots, 6 \quad (2)$$

Where; $e^{[S]\theta}$ is the matrix exponential of screw axis 'S' at joint 'i' in the space frame {S},

$$e^{[S]\theta} = \begin{bmatrix} e^{[\omega]\theta} & G(\theta)v \\ 0 & 1 \end{bmatrix} \quad (3)$$

the Rotation matrix at joint i is given as:

$$e^{[\omega]\theta} = R = I + \sin\theta[\omega] + (1 - \cos\theta)[\omega]^2 \quad (4)$$

$$G(\theta) = I\theta + (1-\cos\theta)[\omega] + [\theta - \sin\theta][\omega]^2 \quad (5)$$

$$[\omega] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \text{ where } \omega \in R^3 \quad (6)$$

The rotation and linear velocity i.e the screw axis for joints 1-6 s given in Table 3 below

Table 3: Screw axis rotation for each joint

i	S_{wi} (ω_1, ω_2, ω_3)	S_{vi}
1	(0, 0, 1)	(0, 0, 0)
2	(0, 1, 0)	(-89, 0, 0)
3	(0, 1, 0)	(-89, 0, 425)
4	(0, 1, 0)	(-89, 0, 817)
5	(1, 0, 0)	(0, 89, 109)
6	(0, 0, -1)	(-109, 912, 0)

2.3 Inverse Kinematics

Given a homogeneous transform $X \in SE(3)$, i.e. the orientation and x,y,z location of the desired end effector position, we need to find $\theta_1, \theta_2, \ldots, \theta_6$ such that $T(\theta) = X$, this is the inverse kinematics problem.

2.3.1 Analytic Inverse Kinematics solution

The analytic solution to the 6-R UR5 robot can be simplified if the condition for a spherical wrist is met. A spherical wrist is achieved when the axis of rotation of three consecutive joints intersects.

Assumptions:

1. The wrist position depends on the first three joints.
2. The wrist orientation depends on the last three joints.

This approach splits the problem into two; a cartesian problem that determines the position of the wrist on the XYZ plane and a spherical problem that determines the orientation of the wrist and end

effector.

The intersection of the last three joints that meets the spherical wrist condition is shown in Figure 3.4a and Figure 3.4b below.

Let X be the transformation matrix of the desired end effector position and orientation

$$X = \begin{bmatrix} r_{11} & r_{12} & r_{13} & Q_x \\ r_{21} & r_{22} & r_{23} & Q_y \\ r_{31} & r_{32} & r_{33} & Q_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

As stated earlier, the homogeneous transform $X \in SE(3)$ (Special Euclidean Group of R^3) is given and will be represented by T_{sb} also T_6^0 which represents the transformation matrix from the base to the end effector joint (joint 6).

Let the position vector of the end effector joint be *P* and the position of the wrist center be *W*.

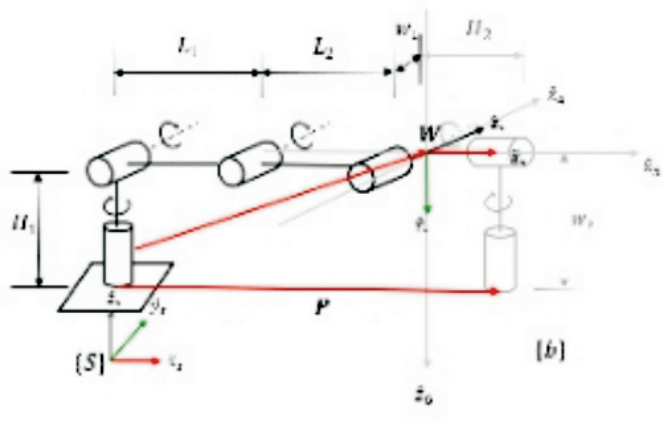

Figure 2. Wrist Centre of the UR5 robot

$$W = P - H_2Q_x - w_2Q_z = \begin{bmatrix} 817 \\ 109 \\ 89 \end{bmatrix}$$

To find θ_1, θ_2 & θ_3 i.e the cartesian problem, a section of the arm from the wrist center to the base will be examined.

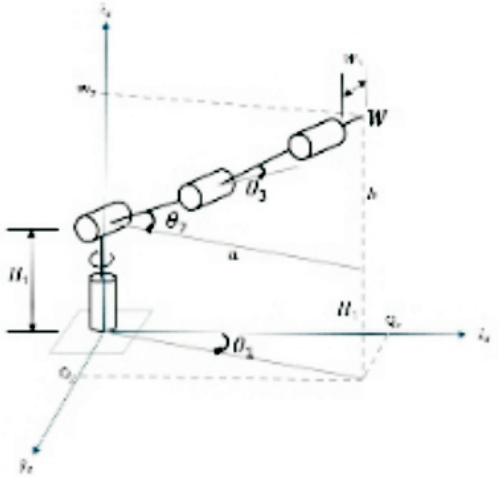

Figure 3a. Wrist center to base section

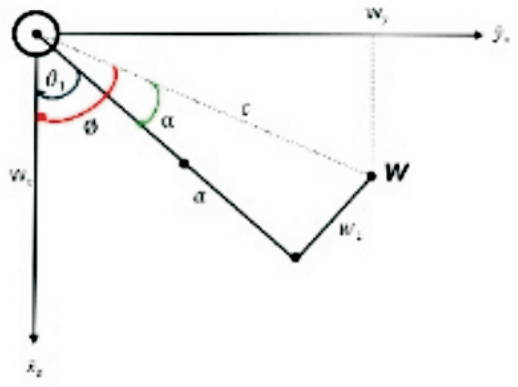

Figure 3b. X-Y axis of the base to wrist center

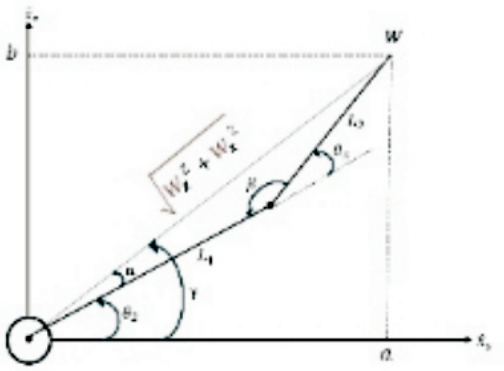

Figure 3c. Z-X axis of the base to wrist center

From Figure 3b,

$\theta_1 = \phi - \alpha$,

$\phi = \text{atan2}(W_x, W_y)$ and $\alpha = \text{atan2}(a, W_1)$

Therefore;

$$\theta_1 = \text{atan2}(W_y, W_x) - \text{atan2}\left(W_1, \sqrt{W_y^2+W_x^2-W_1^2}\right) \quad (8)$$

From Figure 3c.

$\theta_2 = \gamma - \alpha$

$\theta_3 = \pi - \beta$

Using the law of cosines:

$$\alpha = \cos^{-1}\left(\frac{L_1^2-L_2^2+W_x^2+W_z^2}{2L_1\sqrt{W_z^2+W_x^2}}\right) \qquad \beta = \cos^{-1}\left(\frac{L_1^2+L_2^2-W_x^2-W_z^2}{2L_1L_2}\right)$$

$$\gamma = \text{atan2}(b,a) = \text{atan2}\left(Q_z-H_1, \sqrt{W_x^2+W_y^2}\right) = \tan^{-1}\left(\frac{Q_z-H_1}{\sqrt{W_x^2+W_y^2}}\right)$$

$$\theta_2 = \text{atan2}\left(Q_z-H_1, \sqrt{Q_x^2+Q_y^2}\right) - \cos^{-1}\left(\frac{L_1^2-L_2^2+W_x^2+W_z^2}{2L_1\sqrt{W_z^2+W_x^2}}\right) \quad (9)$$

$$\theta_3 = \pi - \cos^{-1}\left(\frac{L_1^2+L_2^2-W_x^2-W_z^2}{2L_1L_2}\right) \quad (10)$$

From this point where θ_1, θ_2 & θ_3 is known, we can manipulate the forward kinematics solution, from *Eq(2)*

$$X = e^{[S_1]\theta_1}e^{[S_2]\theta_2}e^{[S_3]\theta_3}e^{[S_4]\theta_4}e^{[S_5]\theta_5}e^{[S_6]\theta_6}M$$

It can be rearranged to give:

$$e^{[S_4]\theta_4}e^{[S_5]\theta_5}e^{[S_6]\theta_6} = e^{-[S_3]\theta_3}e^{-[S_2]\theta_2}e^{-[S_1]\theta_1}M^{-1}X \quad (11)$$

$$\text{Let, } T = e^{-[S_3]\theta_3}e^{-[S_2]\theta_2}e^{-[S_1]\theta_1}M^{-1}X \quad (12)$$

$$T = \begin{bmatrix} [R_T] & P \\ 0 & 1 \end{bmatrix} \tag{13}$$

We can extract $M_R X_R e^{[\omega_1]\theta_1}$, $e^{[\omega_2]\theta_2}$ and $e^{[\omega_3]\theta_3}$ and *Eq. 11 & Eq. 12* becomes:

$$e^{[\omega_1]\theta_1} e^{[\omega_2]\theta_2} e^{[\omega_3]\theta_3} = R_T = e^{-[\omega_1]\theta_1} e^{-[\omega_2]\theta_2} e^{-[\omega_3]\theta_3} M_R^{-1} X_R \tag{14}$$

$$R_T = \begin{bmatrix} r_{11}c_1c_{23}+r_{21}s_1c_{23}+r_{31}s_{23} & r_{12}c_1c_{23}+r_{22}s_1c_{23}+r_{32}s_{23} & r_{13}c_1c_{23}+r_{23}s_1c_{23}+r_{33}s_{23} \\ r_{21}c_1-r_{11}s_1 & r_{22}c_1-r_{12}s_1 & r_{23}c_1-r_{13}s_1 \\ r_{11}c_1s_{23}+r_{21}s_1s_{23}+r_{31}c_{23} & r_{12}c_1s_{23}+r_{22}s_1s_{23}+r_{32}c_{23} & r_{13}c_1s_{23}+r_{23}s_1s_{23}+r_{33}c_{23} \end{bmatrix}$$

Where; s_1 represents sin θ_1, c_1 represents cos θ_1, s_{23} represents sin $(\theta_2+\theta_3)$... etc.

A wrist mechanism can be represented with the ZYX Euler angles relation as follows:

$$R(\alpha,\beta,\gamma) = \mathbf{Rot}(z,\alpha)\mathbf{Rot}(y,\beta)\mathbf{Rot}(x,\gamma)$$

Then;

$$R(\alpha,\beta,\gamma) = \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\beta \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix}$$

Therefore

$$\mathbf{R}(\theta_4,\theta_5,\theta_6) = \begin{bmatrix} c_4c_5 & c_4s_5s_6 - s_4c_6 & c_4s_5c_6 + s_4s_6 \\ s_4c_5 & s_4s_5s_6 + c_4c_6 & s_4s_5c_6 - c_4s_6 \\ -s_5 & c_5s_6 & c_5c_6 \end{bmatrix} \tag{15}$$

$$\mathbf{R}(\theta_4,\theta_5,\theta_6) = \mathbf{Rot}(z,\theta_4)\mathbf{Rot}(y,\theta_5)\mathbf{Rot}(x,\theta_6) = R_T \tag{16}$$

$\mathbf{Rot}(z,\theta_4)$, $\mathbf{Rot}(y,\theta_5)$ **and** $\mathbf{Rot}(x,\theta_6)$ is the SO(3) component of the matrix exponentials $e^{[S_1]\theta_1}$, $e^{[S_2]\theta_2}$ and $e^{[S_3]\theta_3}$ respectively.

Equating *Eq(14)* and *Eq(15)*

$$\begin{bmatrix} c_4c_5 & c_4s_5s_6 - s_4c_6 & c_4s_5c_6 + s_4s_6 \\ s_4c_5 & s_4s_5s_6 + c_4c_6 & s_4s_5c_6 - c_4s_6 \\ -s_5 & c_5s_6 & c_5c_6 \end{bmatrix}$$

$$= \begin{bmatrix} r_{11}c_1c_{23}+r_{21}s_1c_{23}+r_{31}s_{23} & r_{12}c_1c_{23}+r_{22}s_1c_{23}+r_{32}s_{23} & r_{13}c_1c_{23}+r_{23}s_1c_{23}+r_{33}s_{23} \\ r_{21}c_1-r_{11}s_1 & r_{22}c_1-r_{12}s_1 & r_{23}c_1-r_{13}s_1 \\ r_{11}c_1s_{23}+r_{21}s_1s_{23}+r_{31}c_{23} & r_{12}c_1s_{23}+r_{22}s_1s_{23}+r_{32}c_{23} & r_{13}c_1s_{23}+r_{23}s_1s_{23}+r_{33}c_{23} \end{bmatrix}$$

$\theta_4 = \mathbf{atan2}(r_{21}c_1 - r_{11}s_1, r_{11}c_1c_{23} + r_{21}s_1c_{23} + r_{31}s_{23})$

$\theta_5 = \mathbf{atan2}(r_{31}c_{23} - r_{11}c_1s_{23} - r_{21}s_1s_{23}, \sqrt{1-(r_{31}c_{23} - r_{11}c_1s_{23} - r_{21}s_1s_{23})^2})$

$\theta_6 = \mathbf{atan2}(r_{12}c_1s_{23} + r_{22}s_1s_{23} - r_{32}c_{23}, r_{13}c_1s_{23} + r_{23}s_1s_{23} - r_{33}c_{23})$

2.3 Numerical Inverse Kinematics

The numerical solution method to be used for solving the inverse kinematics problem for the robot in study "UR5" will be the Gradient Descent (GD) Algorithm. The solution is achieved by refining an initial guess of joint angles to achieve a desired end effector position and orientation.

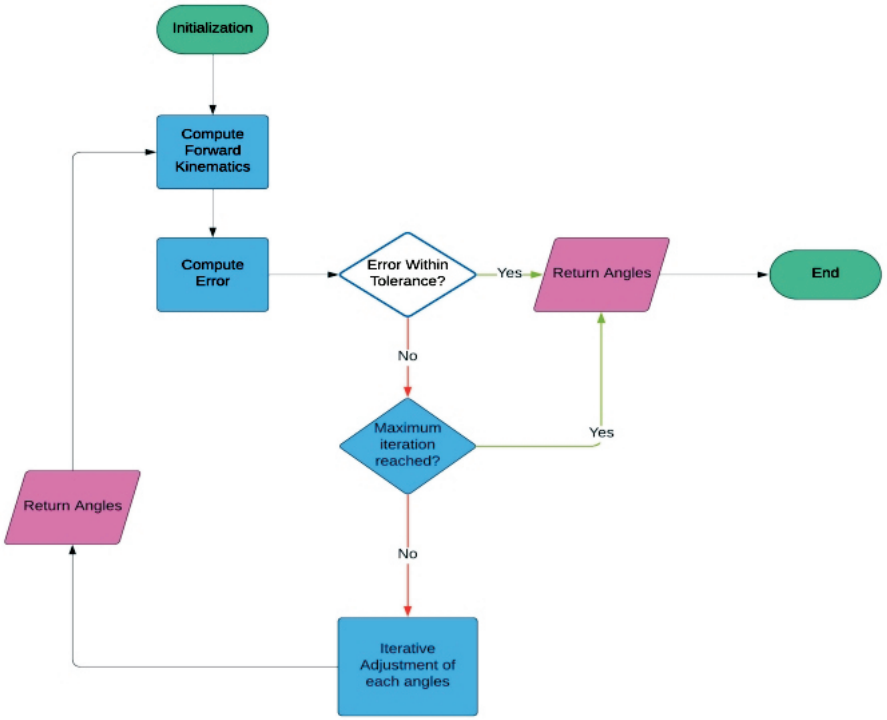

Figure 4. Flow chart of the Gradient-Descent Algorithm

Let $[\theta_0] = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6] = [0, 0, 0, 0, 0, 0]$ and the desired end effector position,

$$\mathbf{X} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & Q_x \\ r_{21} & r_{22} & r_{23} & Q_y \\ r_{31} & r_{32} & r_{33} & Q_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

And the rotation axis of each joint from Table 3:

$$S_{w1} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, S_{w2} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, S_{w3} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, S_{w4} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, S_{w5} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, S_{w6} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \tag{18}$$

$$X_k = \begin{bmatrix} [R_k] & P_k \\ 0 & 1 \end{bmatrix} \tag{19}$$

$P_k \in R^3$ is the current position of the end effector and
$[R_k] \in R^{3\times3}$ is the current orientation of the end effector.

The current position P_k and orientation $[R_k]$ can be computed using forward kinematics from eq(14) as written below:

$$X_k = T(\theta) = e^{[S_1]\theta} e^{[S_2]\theta} e^{[S_3]\theta} e^{[S_4]\theta} e^{[S_5]\theta} e^{[S_6]\theta} M \tag{20}$$

The error vector is given as $e_k(\theta) = \begin{bmatrix} P_k - P_{Target} \\ vec([R_k] - [R_{Target}]) \end{bmatrix}$ (21)

$e_k(\theta) \in R^{12}$ is a combined vector of all the values from the position and orientation errors

2.3.1 Jacobian Calculation

For each angle θ_i in $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$, we find θ_{rot} the angle increment required to get the robot closer to it's end effector target. This involves computing the error function with respect to *each* joint angle

$$J_k(\theta) = \left.\frac{\partial e_k(\theta)}{\partial \theta}\right|_{\theta=\theta_k} \quad (22)$$

The Jacobian is solved using forward difference method:

$$J_i = \frac{e_i\left(\theta_k + \Delta\theta \cdot \hat{e}_{unit}\right) - e_k(\theta)}{\Delta\theta} \quad (23)$$

Where:

a. $J_k(\theta) \in R^{6\times n}$ is the Jacobian matrix, with each element representing the partial derivative of an error component with respect to a joint angle.

b. J_{ki} is the *i-th* row of the Jacobian matrix corresponding to an error component *"i"* in the error matrix i = **1**, **2**, **3**, ..., **12**.

c. $\Delta\theta$ is the increment for the forward difference iteration.

d. e_{ki} is the *i-th* component of the error vector $e_k(\theta)$.

e. $\hat{e}_{unit}$ is the unit vector that multiplies the scalar $\Delta\theta$, converting it to a vector.

2.3.2 Gradient Vector

The gradient of the error function as shown below indicates the direction and magnitude of the change required in each joint angle to minimize the error

$$g_k(\theta) = \alpha \cdot \Delta\theta \cdot J_k(\theta)^T \cdot e_k(\theta) \quad (24)$$

The joint angles are updated using:

$$\theta_{k+1} = \theta_k - g_k(\theta) \quad (25)$$

Where:

- $g_k(\theta)$ R^6 is the gradient vector
- θ_{k+1} is the updated joint angles vector to be used in the next iteration

2.4 Optimization

While IK solutions are traditionally categorized as Analytical and Numerical Solutions, they both have their distinct advantages and limitations. *Analytical methods* offer exact solutions and

computational efficiency but can be challenging to derive for complex systems and are sensitive to singularities and joint limits. *Numerical methods*, on the other hand, provide flexibility and robustness but at the cost of increased computational load and potential convergence issues.

The *Hybrid* approach leverages on the strengths of both approaches while reducing their weaknesses. This hybrid approach will be applied in solving the Inverse Kinematics solution of the UR5 robot being studied.

3. Analysis & Verificaiton

Figures and tables ought to be placed next to the relevant text in the article. All figures and tables should be discussed and explicitly cited in the text.

Table 4: Results for each scenario tests for Analytic, Numerical and Hybrid IK methods

Table 5. Comparative performance of analytic, numerical, and hybrid IK methods across ten test scenarios

Scenario	Analytic		Numerical			Hybrid		
	Error (%)	Feasible	Error (%)	Time (s)	Feasible	Error (%)	Time (s)	Feasible
1	15.707	No	0.219	1.6900	Yes	0.219	1.4588	Yes
2	No sol.	No	1.635	0.4455	No	1.635	0.4296	No
3	98.000	No	0.274	0.3481	No	0.274	0.3318	Yes
4	136.077	No	0.314	0.3512	Yes	0.314	0.2345	Yes
5	No sol.	No	0.222	1.4515	Yes	0.222	1.4436	Yes
6	No sol.	No	0.632	0.6066	No	0.632	0.5801	No
7	No sol.	No	0.236	1.0451	Yes	0.236	1.0619	Yes
8 *	92.403	No	0.282	2.5212	Yes	0.282	0.5627	No
9	141.382	No	0.234	1.1863	Yes	0.234	0.4436	Yes
10	No sol.	No	0.331	0.2709	Yes	0.331	0.2726	Yes

Analytic convergence times are omitted — the closed-form method resolves instantaneously (< 0.0001 s) where a solution exists. * Scenario 8 hybrid: accurate but infeasible due to positional bias

introduced by the analytic initial estimate. Full iteration counts are reported in Appendix Table A1.

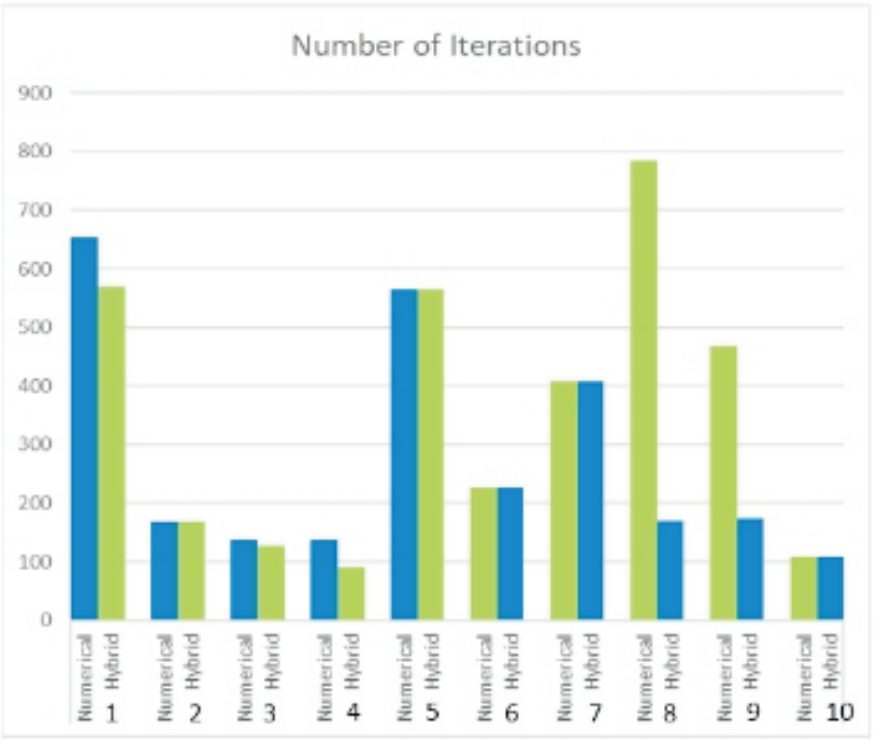

Figure 5. Iterations (Y-axis) for each scenario (X-axis)

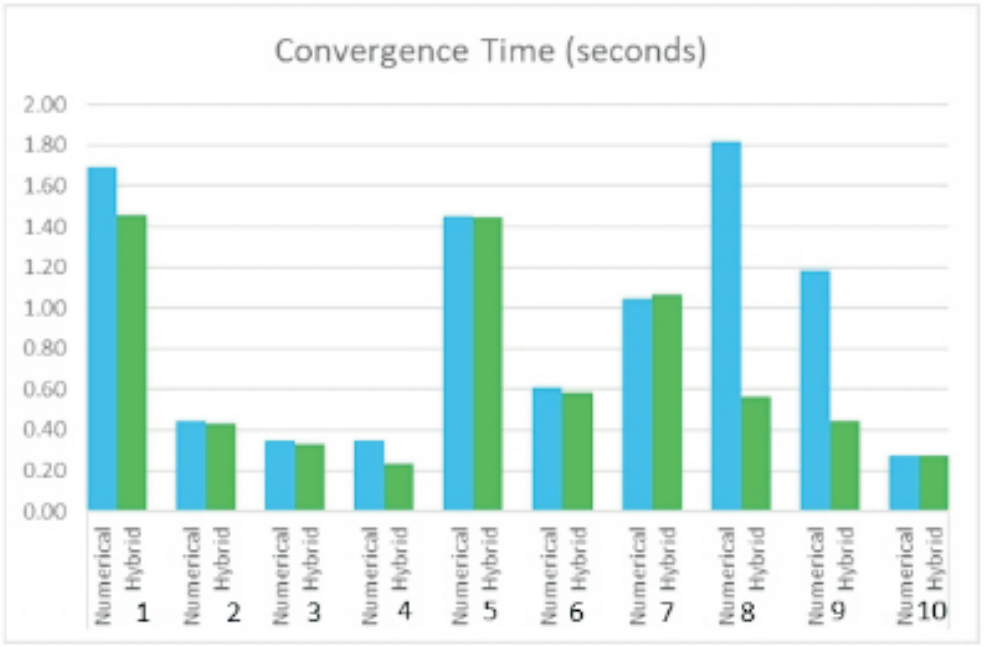

Figure 6. Convergence time (Y-axis) for each scenario (X-axis)

The analytic method is extremely fast, with convergence times in microseconds, but it either finds a solution immediately or fails entirely, making it computationally efficient yet unreliable. The numerical method, while significantly slower (0.27 to 2.52 seconds), provides more consistent performance with a balance of accuracy and computation time. The hybrid method improves on the numerical approach by reducing convergence times and iterations in most scenarios, demonstrating greater efficiency while maintaining accuracy. However, in cases where the analytic method fails to provide an initial estimate, the hybrid method performs similarly to the numerical method.

Figure 7. Error magnitude for each scenario

On accuracy, the analytic method performs poorly, with high error magnitudes in several scenarios and frequent failures in handling complex configurations. The numerical method, on the other hand, demonstrates high accuracy, with error magnitudes ranging from 0.22% to 1.63%, making it reliable for precision tasks. The hybrid method closely matches the numerical method in accuracy, benefiting from its strengths. However, in Scenario 8, it fails to achieve feasibility due to inaccuracies inherited from the analytic method's initial guess.

The analytic method is the least feasible, often failing to find solutions and producing high error magnitudes when it does. The numerical method is generally reliable, successfully solving most scenarios with low errors, though it struggles with complex or ill-conditioned problems. The hybrid method improves feasibility over the analytic method and enhances the numerical method by reducing convergence time and increasing efficiency. However, in Scenario 8, it achieves accuracy but fails feasibility, suggesting limitations in certain complex cases.

Visual Analysis in CoppeliaSim

With the hybrid and numerical methods resulting to the same accuracy figures across board, visualisation of the achieved and target end-effector positions are required. This was achieved using CoppeliaSim with simulations scripted in the “Lua” programming language to provide visual comparisons.

Scenario 1 (Max Reach X-Direction)

The analytic method showed a small offset, while the numerical and hybrid methods aligned perfectly with the target.

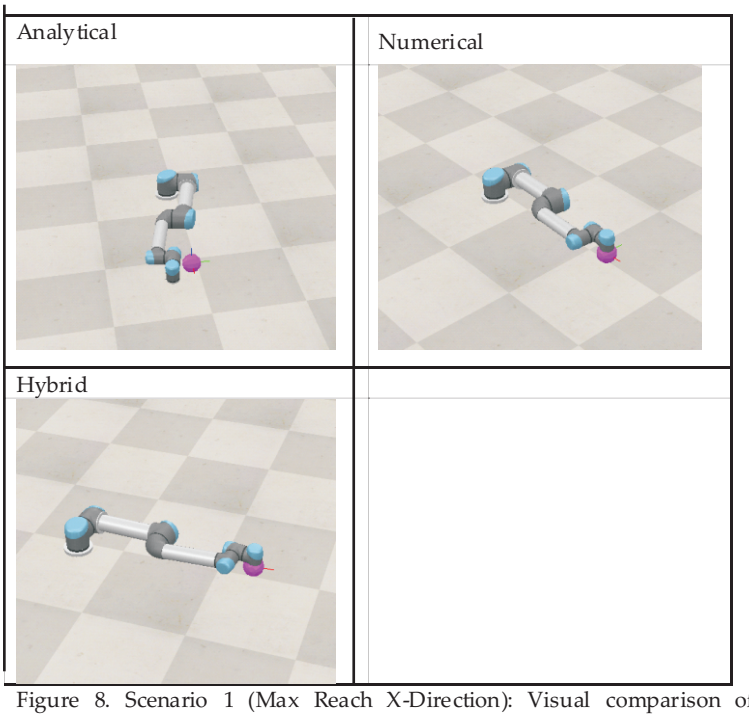

Figure 8. Scenario 1 (Max Reach X-Direction): Visual comparison of (a) Analytical, (b) Numerical, and (c) Hybrid methods in CoppeliaSim.

Scenario 4 (Centre Of Task Space)

Same position requirement as scenario 3, but with different orientation targets (90° Yaw). The analytical solution did not achieve feasibility, but it was able to provide the required orientation.

The numerical and hybrid solution reaching close to the target with accurate positioning, the numerical solution did not align the wrist properly with the target. But with the help of analytic solution the hybrid solution was able to achieve the right position and orientation, hence the most accurate solution. This also shows that if the orientation requirement is not feasible (Scenario 3), all solutions will find it hard to reach the required targets

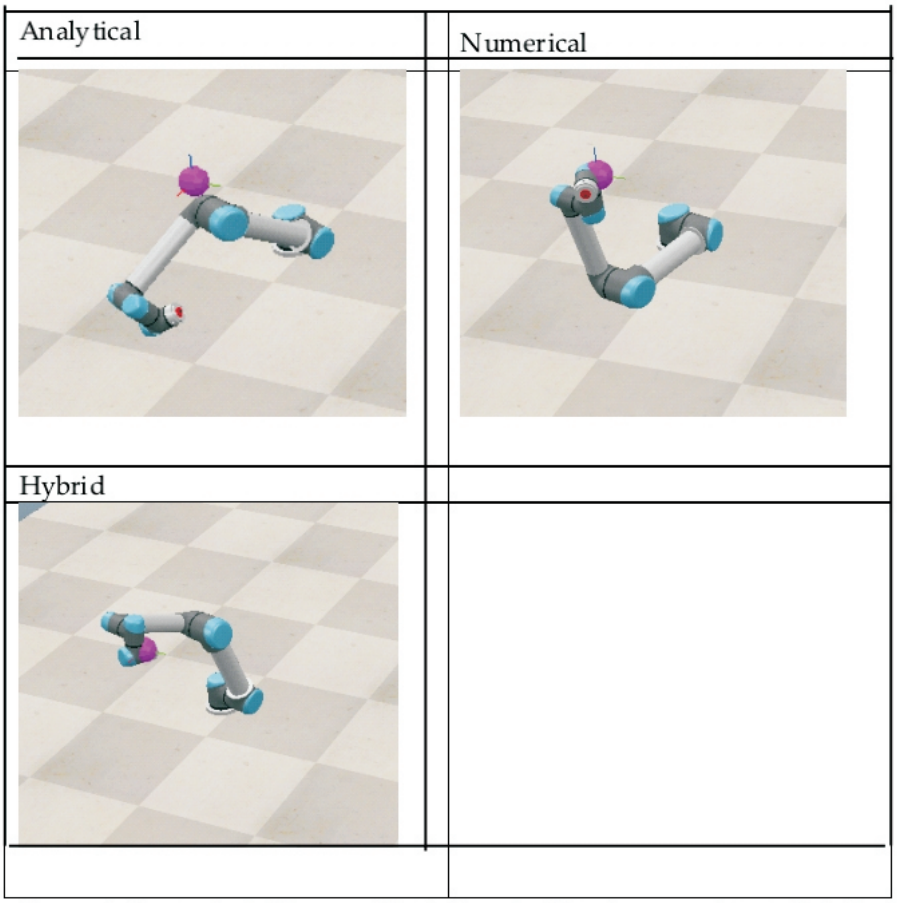

Figure 9. Scenario 4 (Centre of Task Space): Visual comparison of (a) Analytical, (b) Numerical, and (c) Hybrid methods in CoppeliaSim.

Scenario 6 (Near Singularity)

The analytic did not provide a solution leading to the hybrid and numerical method having the same results. Both methods achieved the required position but did not reach the required wrist configuration (orientation), but very close to the target pose.

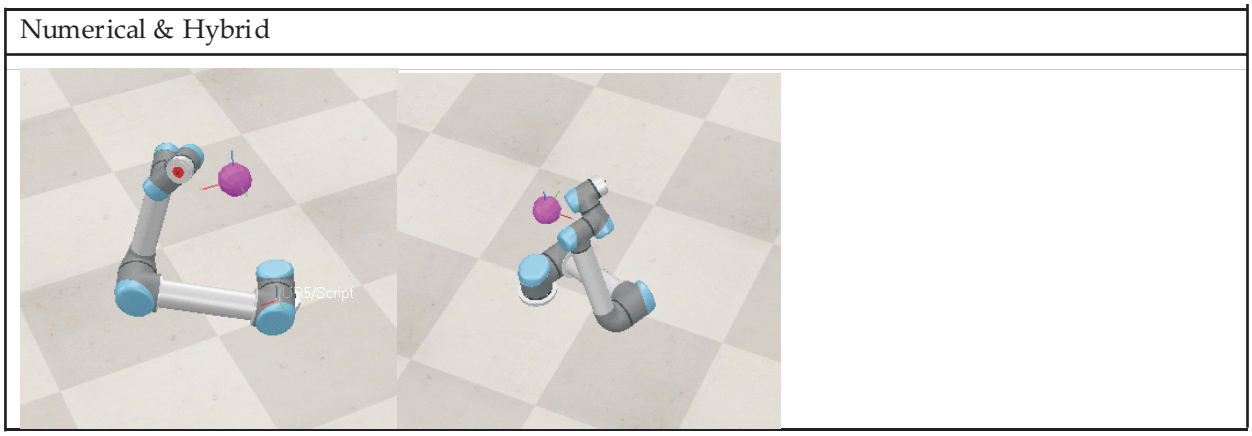

Figure 10. Scenario 6 (Near Singularity): Visual comparison of (a) Analytical, (b) Numerical, and (c) Hybrid methods in CoppeliaSim.

Scenario 7 (Task-Specific Position)
The analytic did not provide a solution leading to the hybrid and numerical method having the same results. Both methods achieved the required position and aligned perfectly with the target.

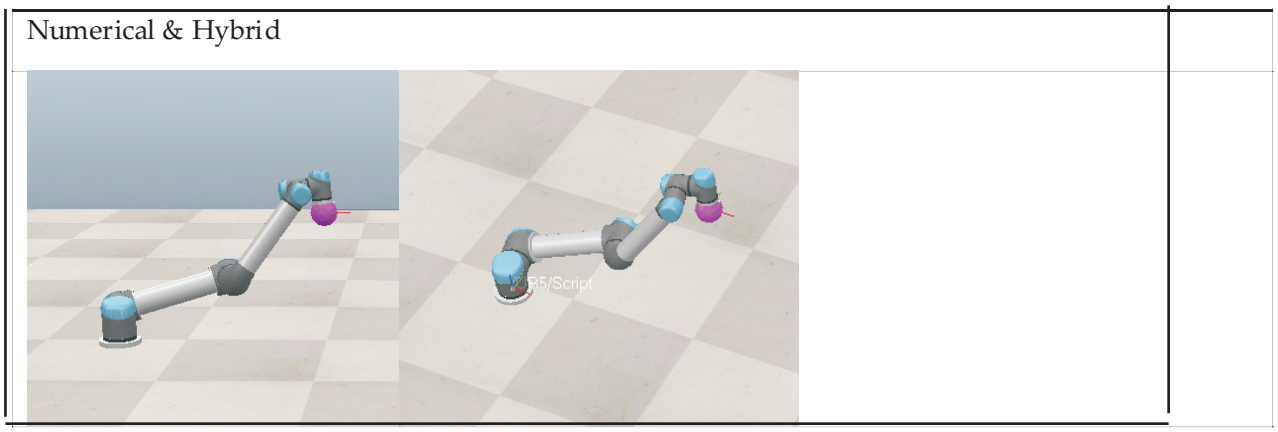

Figure 11. Scenario 7 (Task-Specific Position): Visual comparison of Numerical, and Hybrid methods in CoppeliaSim.

Scenario 8 (Edge Of Task Space: Y-Direction)
The analytic method did not achieve feasibility, but it was able to provide the required orientation. The Numerical method aligned perfectly with the target while the Hybrid method was just centimetres shy of the target, This is a case of the Numerical method achieving feasibility while hybrid method did not due to the initial approximation provided by the analytical solution it depends on.

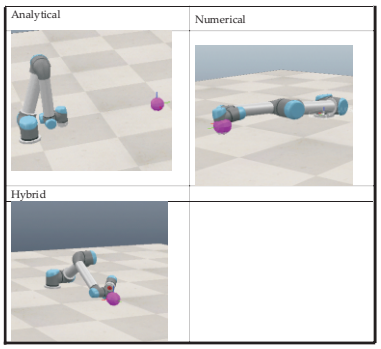

Figure 12. Scenario 8 (Edge of Task Space: Y-Direction): Visual comparison of (a) Analytical, (b) Numerical, and (c) Hybrid methods in CoppeliaSim.

4. Conclusion

This research explored the inverse kinematics in 6-DOF robotic systems through analytic, numerical, and hybrid methods with a particular focus on the hybrid approach. The results demonstrate that the hybrid method combines the computational efficiency of analytic methods with the versatility of numerical methods, mitigating the limitations of both. While analytic methods are computationally inexpensive but struggle with complex or near-singularity scenarios, and numerical methods offer accuracy but can be computationally intensive, the hybrid method strikes a balance that enhances both precision and performance.

In a variety of test scenarios, the hybrid method consistently matched or exceeded the accuracy and efficiency of the numerical method, particularly in achieving precise end-effector positioning and orientation. It demonstrated faster convergence times and reduced computational demands, making it suitable for real-time applications where rapid response is critical. Simulations conducted in CoppeliaSim further validated the hybrid method's superior performance in scenarios that require high precision, such as diagonal reach and task-specific positioning.

These findings suggest that the hybrid method is not just a theoretical improvement, but a practical solution for robotic systems requiring high precision and reliability. The research highlights the potential for further advancements, such as integrating adaptive algorithms that can dynamically select between analytic and numerical methods based on task complexity. Additionally, the hybrid approach could be extended to robots with higher degrees of freedom or multi-robot systems, and combined with emerging technologies like artificial intelligence and machine learning.

In conclusion, this study presents the hybrid method as a significant advancement in solving inverse kinematics problems, offering improved accuracy, efficiency, and real-time applicability. The method's success opens new avenues for future research in robotics, particularly in high-precision and complex task environments, where it can enhance the capabilities of modern robotic systems.

Authors' Contributions

First Author's Contributions[1]

Conceptualization of the problem; problem formulation; results discussion; overall perusal and packaging of the article content. Also I supervised the whole Undergraduate's project from which this paper was drafted.

Second Author's Contributions[2]

Problem formulation and solving; formal investigation; results simulation and discussions; drafting of the technical paper from his Undergraduate's project supervised by me.

Acknowledgement

The authors wish to thank the Department of Mechanical Engineering, University of Lagos.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

COMPETING INTERESTS: The authors declare that there are no competing conflict of interests between the authors regarding the publication of this paper.

1. References

Aristidou, A., & Lasenby, J. (2009).*Inverse kinematics: A review of existing techniques and introduction of a new fast iterative solver* (Technical Report No. CUED/F-INFENG/TR-632).University of Cambridge, Department of Engineering.

Brandstötter, M., Angerer, A., & Hofbaur, M. (2014).An analytical solution of the inverse kinematics problem of industrial serial manipulators with an ortho-parallel basis and a spherical wrist. In *Proceedings of ISR/Robotik 2014: 41st International Symposium on Robotics* (pp. 1–6). VDE.

Buss, S. R. (2009).*Introduction to inverse kinematics with Jacobian transpose, pseudoinverse and damped least squares methods* [Unpublished manuscript]. University of California, San Diego.

Courty, N., & Arnaud, E. (2008).Inverse kinematics using sequential Monte Carlo methods. In F. J. Perales & R. B. Fisher (Eds.), *Articulated motion and deformable objects: AMDO 2008* (Lecture Notes in Computer Science, Vol. 5098, pp. 1–10). Springer.https://doi.org/10.1007/978-3-540-70517-8_1

Jiang, L., Huo, X., Liu, Y., & Liu, H. (2013). An analytical inverse kinematic solution with the reverse coordinates for 6-DOF manipulators. In *Proceedings of the 2013 IEEE International Conference on Mechatronics and Automation (ICMA)* (pp. 1552–1558).IEEE.https://doi.org/10.1109/ICMA.2013.6618145

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*(3), 459–471. https://doi.org/10.1007/s10898-007-9149-x

Kucuk, S., & Bingul, Z. (2006). Robot kinematics: Forward and inverse kinematics. In D. Cubero (Ed.), *Industrial robotics: Theory, modelling and control* (pp. 117–148). InTech.https://doi.org/10.5772/5015

Liu, H., Zhou, W., Lai, X., & Zhu, S. (2013).An efficient inverse kinematic algorithm for a PUMA560-structured robot manipulator.*International Journal of Advanced Robotic Systems, 10*(5), 1–8. https://doi.org/10.5772/56403

Lopez-Franco, C., Hernandez-Barragan, J., Alanis, A. Y., & Arana-Daniel, N. (2018).A soft computing approach for inverse kinematics of robot manipulators.*Engineering Applications of Artificial Intelligence, 74*, 104–120.https://doi.org/10.1016/j.engappai.2018.06.007

Oyama, E., Maeda, T., MacDorman, K. F., Tachi, S., Agah, A., Gan, J. Q., & Rosales, E. M. (2005). Inverse kinematics learning for robotic arms with fewer degrees of freedom by modular neural network systems.In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 2006–2011).IEEE.https://doi.org/10.1109/IROS.2005.1545087

Park, F. C., & Okamura, K. (1994).Kinematic calibration and the product of exponentials formula.In J. Lenarčič & B. Ravani (Eds.), *Advances in robot kinematics and computational geometry* (pp. 119–128).Springer.https://doi.org/10.1007/978-94-015-8348-0_12

Polyzos, K. D., Groumpos, P. P., & Dermatas, E. (2019).Solving the inverse kinematics of robotic arm using autoencoders. In A. Kravets, P. Groumpos, M. Shcherbakov, & M. Kultsova (Eds.), *Creativity in intelligent technologies and data science: CIT&DS 2019* (Communications in Computer and Information Science, Vol. 1083). Springer.https://doi.org/10.1007/978-3-030-29743-5

Refaai, M. R. A. (2022). An improved inverse kinematics solution for a robot arm trajectory using multiple adaptive neuro-fuzzy inference systems.*Advances in Materials Science and Engineering, 2022*, Article 1413952. https://doi.org/10.1155/2022/1413952

Singh, A., Baranwal, N., & Nandi, G. (2017).Development of a self-reliant humanoid robot for sketch drawing.*Multimedia Tools and Applications, 76*, 26763–26784. https://doi.org/10.1007/s11042-017-4358-x

Tang, Y., Chen, M., Wang, C., Luo, L., Li, J., Lian, G., & Zou, X. (2020). Recognition and localization methods for vision-based fruit picking robots: A review. *Frontiers in Plant Science, 11*, Article 510. https://doi.org/10.3389/fpls.2020.00510

Tong, Y., Liu, J., Liu, Y., & Yuan, Y. (2021).Analytical inverse kinematic computation for 7-DOF redundant sliding manipulators.*Mechanism and Machine Theory, 155*, Article 104006. https://doi.org/10.1016/j.mechmachtheory.2020.104006

Zhao, G., Jiang, D., Liu, X., Tong, X., Sun, Y., Tao, B., Kong, J., Yun, J., Liu, Y., & Fang, Z. (2022).A tandem robotic arm inverse kinematic solution based on an improved particle swarm algorithm.*Frontiers in Bioengineering and Biotechnology, 10*, Article 832829. https://doi.org/10.3389/fbioe.2022.832829

Appendix

Table A1. Iteration counts by method and scenario

Scenario	Analytic	Numerical	Hybrid	Notes
1	—	655	569	- 13% vs numerical
2	—	167	167	No gain — analytic failed
3	—	137	128	- 7% vs numerical
4	—	137	90	- 34% vs numerical
5	—	566	566	No gain — analytic failed
6	—	226	226	No gain — analytic failed
7	—	409	409	No gain — analytic failed
8 *	—	785	169	- 78% vs numerical; hybrid infeasible
9	—	469	173	- 63% vs numerical
10	—	107	107	No gain — analytic failed

Analytic iteration count is not applicable (—) — the closed-form method either resolves in a single computation or fails entirely. 'No gain' indicates the analytic method did not converge, so the hybrid fell back to pure gradient descent with no warm-start benefit. * Scenario 8 hybrid achieved the largest iteration reduction (- 78%) but was infeasible.