

Mathematical and numeric analysis of a 4-DOF manipulator kinematics




Análisis matemático y numérico de la cinemática de un manipulador de 4 GDL

García-Trinidad, Enrique^{*a}, Arcos-Hernández, Emmanuel^b, Torres-Valle, José^c and Hernández-Borja, Carlos^d

^a  Universidad Tecnológica Fidel Velázquez •  9353-2024 •  0000-0003-2875-0500 •  271440

^b  Universidad Tecnológica Fidel Velázquez •  1933-2025 •  0000-0002-3404-1375 •  349835

^c  Universidad Tecnológica Fidel Velázquez •  6499-2025 •  0000-0001-8258-9418 •  626539

^d  Universidad Tecnológica Fidel Velázquez •  1898-2025 •  0000-0002-8138-9016

SECIHTI classification:

Area: Engineering
Field: Engineering
Discipline: Electronic Engineering
Subdiscipline: Robotics

 <https://doi.org/10.35429/JCS.2025.9.20.1.1.11>

Article History:

Received: June 30, 2025

Accepted: December 10, 2025

*  [\[enrique.garcia@utfv.edu.mx\]](mailto:enrique.garcia@utfv.edu.mx)



Abstract

This paper analyzes the kinematics of a 4-DOF robotic manipulator. The Denavit-Hartenberg formulation for geometric modeling, which simplifies the description of the links, is detailed. Direct kinematics maps the joint variables to the end-effector pose in a unique way. Inverse kinematics, more complex, determines the joint angles for a desired pose, facing multiple solutions and nonlinear equations. Analytical (geometric and algebraic) and numerical (Newton-Raphson, Levenberg-Marquardt) methods are explored. Practical implications include simulation and visualization in Python's Robotics Toolbox.

Resumen

Este artículo analiza la cinemática de un manipulador robótico de 4 GDL. Se detalla la formulación de Denavit-Hartenberg para el modelado geométrico, que simplifica la descripción de los eslabones. La cinemática directa mapea las variables articulares a la pose del efector final de forma única. La cinemática inversa, más compleja, determina los ángulos articulares para una pose deseada, enfrentando múltiples soluciones y ecuaciones no lineales. Se exploran métodos analíticos (geométricos y algebraicos) y numéricos (Newton-Raphson, Levenberg-Marquardt). Las implicaciones prácticas incluyen la simulación y visualización en la Robotics Toolbox de Python.

Análisis matemático y numérico de la cinemática de un manipulador de 4 GDL		
Objetivos	Methodology	Contribution
The main objective of this study is to systematically analyze the mathematical and physical principles governing the movement of four-degrees-of-freedom (4-DOF) robotic manipulators. This involves establishing the precise mathematical analysis between the robot's joint configuration and the resulting position and orientation of its end-effector, encompassing both direct and inverse kinematics. A deep understanding of these concepts, based on a systematic modeling framework, is indispensable for the design, simulation, and control of any robotic manipulator.	The methodology is based on the Denavit-Hartenberg (D-H) formulation, a standardized method for describing manipulator geometry using four key parameters. For direct kinematics, the end-effector's pose is determined from joint variables through serial multiplication of homogeneous transformation matrices. Inverse kinematics, a more complex problem due to its nonlinearity and multiple solutions, is addressed with two approaches: analytical (geometric, which decomposes the problem into 2D using trigonometry, and algebraic, which manipulates matrix equations) and numerical (iterative algorithms like Newton-Raphson and Levenberg-Marquardt, which minimize pose error and are applicable to complex geometries).	The kinematic analysis significantly contributes to various practical applications in robotics. In robot design, it enables virtual prototyping to optimize geometry, workspace, singularity locations, and manipulability before physical construction. For trajectory planning, it facilitates the generation of smooth and efficient movements that actively avoid singular configurations and maintain high manipulability. Furthermore, it is fundamental for real-time control, allowing for position-based control (via inverse kinematics), resolved-rate velocity control (using the inverse Jacobian), and force control (through the Jacobian transpose), which is crucial for tasks involving contact with the environment.

Análisis matemático y numérico de la cinemática de un manipulador de 4 GDL		
Objetivos	Methodología	Contribución
El objetivo principal de este estudio es analizar sistemáticamente los principios matemáticos y físicos que rigen el movimiento de los manipuladores robóticos de cuatro grados de libertad (4 GDL). Esto implica establecer el análisis matemático preciso entre la configuración de las articulaciones del robot y la posición y orientación resultantes de su efector final, abarcando tanto la cinemática directa como la inversa. Una comprensión profunda de estos conceptos, basada en un marco de modelado sistemático, es indispensable para el diseño, la simulación y el control efectivo de cualquier manipulador robótico.	La metodología se basa en la formulación de Denavit-Hartenberg (D-H), un método estandarizado para describir la geometría del manipulador mediante cuatro parámetros clave. Para la cinemática directa, se determina la pose del efector final a partir de las variables articulares mediante la multiplicación en serie de matrices de transformación homogéneas. La cinemática inversa, un problema más complejo debido a su no linealidad y múltiples soluciones, se aborda con dos enfoques: analíticos (geométrico, que descompone el problema en 2D con trigonometría, y algebraico, que manipula ecuaciones matriciales) y numéricos (algoritmos iterativos como Newton-Raphson y Levenberg-Marquardt, que minimizan el error de pose y son aplicables a geometrías complejas).	El análisis cinemático contribuye significativamente a diversas aplicaciones prácticas en robótica. En el diseño de robots, permite el prototipo virtual para optimizar la ubicación de singularidades y la manipulabilidad antes de la construcción física. Para la planificación de trayectorias, facilita la generación de movimientos suaves y eficientes que evitan activamente configuraciones singulares y mantienen una alta manipulabilidad. Además, es fundamental para el control en tiempo real, posibilitando el control basado en la posición, el control de velocidad resuelta y el control de fuerza, lo cual es crucial para tareas que implican contacto con el entorno.

Robotics, Kinematics, Simulation

Robótica, Cinemática, Simulación

Area: Development of strategic leading-edge technologies and open innovation for social transformation

Citation: García-Trinidad, Enrique, Arcos-Hernández, Emmanuel, Torres-Valle, José and Hernández-Borja, Carlos. [2025]. Mathematical and numeric analysis of a 4-DOF manipulator kinematics. Journal Computational Simulation. 9[20] 1-11: e10920111.



ISSN 2523-6865/© 2009 The Author[s]. Published by ECORFAN-Mexico, S.C. for its Holding Taiwan on behalf of Journal Computational Simulation. This is an open access article under the CC BY-NC-ND license [<http://creativecommons.org/licenses/by-nc-nd/4.0/>]

Peer Review under the responsibility of the Scientific Committee MARVID®- in contribution to the scientific, technological and innovation Peer Review Process by training Human Resources for the continuity in the Critical Analysis of International Research.



1702902 SECIHTI

Introducción

The study of a robot manipulator's movement, or kinematics, is the scientific foundation upon which the entire analysis, control, and application of robotic manipulators are built. Kinematics establishes the precise mathematical analysis between the robot's joint configuration and the resulting position and orientation of its end-effector (Manseur & Doty, 1988). This analysis is fundamentally divided into two problems: direct kinematics and inverse kinematics. This deep understanding, based on a systematic modeling framework, is indispensable for the design, simulation, and control of any robotic manipulator (Malik, Lischuk, Henderson, & Prazenica, 2022).

In general, a robotic manipulator can be described in two distinct but interconnected domains: the joint space and the Cartesian space. The selection of which domain to use depends on the task at hand, whether it is the direct control of the actuators or the interaction of the manipulator's end-effector with the physical world. The joint space is the n -dimensional space defined by the vector of all joint variables, denoted as q . For a 4-degree-of-freedom (DOF) manipulator with four rotational joints, the vector is $q = [q_0, q_1, q_2, q_3]^T$. Each element of this vector represents a single, directly controllable degree of freedom of the robot, such as the angle of a motor shaft. Control systems fundamentally operate in the joint space, as they send commands to the individual joint actuators.

The Cartesian space, also known as the operational space, is the space in which the robot's end-effector performs its tasks. It is typically a six-dimensional space that describes the pose of the end-effector, with three dimensions for its position (P_x, P_y, P_z) and three dimensions for its orientation, which can be represented by Euler angles (ϕ, θ, ψ). Robot programming and task planning are almost always performed in the Cartesian space because tasks like picking up an object, welding, or following a trajectory are naturally defined by the desired movement of the end-effector in the physical world (Guida, De Simone, Dasic, & Guida, 2019). The core of kinematic analysis is to define the mathematical mapping between these two spaces.

Direct and Inverse Kinematics

Direct Kinematics (DK) is the process of determining the Cartesian position and orientation (pose) of the end-effector from a known set of joint variables. This can be expressed as a function f , shown in Equation 1.

$$X = f(q) \quad [1]$$

where X is the end-effector pose vector $[P_x, P_y, P_z, \phi, \theta, \psi]^T$ and q is the vector of joint variables. The direct kinematics problem is considered straightforward because for any given set of joint angles, there is one and only one resulting pose of the end-effector. The mapping is unique and can be calculated directly through a series of matrix multiplications, as detailed later.

Inverse Kinematics (IK) is the reverse and significantly more challenging problem, as it determines the required set of joint variables q that places the end-effector at a desired Cartesian pose X . This inverse mapping is expressed in Equation 2 as:

$$q = f^{-1}(X) \quad [2]$$

The complexity of inverse kinematics arises from several factors (Bao, Liu, & Zhao, 2017), including the non-existence of solutions, since if the desired pose X is outside the manipulator's physical reach, no solution exists. Similarly, for a single desired end-effector pose within the workspace, there can be multiple, and sometimes even infinite, solutions. For example, a simple planar arm can often reach a point with an elbow-up or elbow-down posture (Mohammed & Sunar, 2015).

The control system must have a logical basis for selecting one solution from the set of possibilities. Another factor to consider is that the equations governing inverse kinematics are highly non-linear and trigonometric, making them difficult to solve.

Although analytical, or closed-form, solutions are preferable for their speed and completeness, they only exist for manipulators with specific geometric structures. For more complex robots, iterative numerical methods must be employed.

Denavit-Hartenberg Formulation

To perform kinematic analysis, a consistent method for describing the manipulator's geometry is required. The Denavit-Hartenberg (D-H) formulation is the most widely adopted method for systematically assigning coordinate systems to the links of a manipulator. Its main advantage is that it describes the spatial relationship between consecutive links using a minimal set of four parameters, which greatly simplifies mathematical modeling (Corke, 2007).

The power of the D-H convention lies in its clever mathematical simplification. A general transformation describing the relative pose of one coordinate system with respect to another in 3D space requires six parameters, but D-H reduces this to just four by imposing strict rules on how the coordinate systems are placed. Specifically, the convention requires that the x_i axis of a coordinate system must be perpendicular to and intersect the z_{i-1} axis of the previous system. These two geometric constraints effectively eliminate two degrees of freedom from the general transformation, leaving exactly four.

The D-H procedure involves rigidly attaching a coordinate frame $\{i\}$ to each link i of the manipulator, from the fixed base (link 0) to the end-effector (link n). The procedure is as follows:

1. Identify the joint axes: The axis of actuation (rotation for a revolute joint, translation for a prismatic joint) for joint i is defined as the z_{i-1} axis. The axes are numbered z_0, z_1, \dots, z_{n-1} .
2. Establish the base coordinate system $\{0\}$: The origin is placed anywhere on the z_0 axis. The x_0 and y_0 axes are chosen to form a right-handed coordinate system. This system is the global or inertial reference frame.
3. Iteratively assign frames $\{1\}$ to $\{n-1\}$: For each link i , the frame $\{i\}$ is established based on its relationship with frame $\{i-1\}$:

The origin O_i is located at the intersection of the z_i y la normal común que se extiende desde el eje z_{i-1} axis. The common normal is the unique line segment that is mutually perpendicular to both z_i and z_{i-1} .

The x_i axis is directed along this common normal, pointing from the z_{i-1} axis toward the z_i axis.

The y_i axis is defined by the right-hand rule to complete the coordinate system ($y_i = z_i \times x_i$).

If the z_{i-1} and z_i axes intersect, the common normal has zero length. The origin O_i is placed at the intersection point. The x_i axis is chosen to be normal to the plane containing z_{i-1} and z_i .

If the z_{i-1} and z_i axes are parallel, there are infinite common normals. In this case, there is freedom to choose the location of the origin O_i along the z_i axis. A common practice is to place it in a convenient location, such as the center of the physical link.

The final frame, $\{n\}$, is attached to the end-effector. Its origin O_n and orientation are typically chosen to align with a functionally important feature, such as the center or tip of the end-effector.

4. Once the coordinate frames are assigned, the geometric relationship between frame $\{i-1\}$ and frame $\{i\}$ can be described by four parameters:

a_i is the distance from the z_{i-1} axis to the z_i axis, measured along the x_i axis.

α_i is the angle between the z_{i-1} axis and the z_i axis, measured by a rotation about the x_i axis.

d_i is the distance from the origin O_{i-1} to the intersection of the x_i axis with the z_{i-1} axis, measured along the z_{i-1} axis. For a prismatic joint, d_i is the joint variable.

θ_i is the angle between the x_{i-1} axis and the x_i axis, measured by a rotation about the z_{i-1} axis. For a revolute joint, θ_i is the joint variable.

For any given link in a kinematic chain, three of these four parameters are constant geometric properties of the link, while the fourth is the variable that describes the joint's motion.

Homogeneous Transformation Matrices

The four D-H parameters elegantly define the transformation from frame $\{i - 1\}$ to frame $\{i\}$. This transformation, denoted by the homogeneous transformation matrix A_i , can be expressed as a product of four elementary rotations (Rot) and translations (Tras), as shown in Equation 3.

$$A_i = \text{Rot}_{x,\theta_i} \text{Tras}_{z,d_i} \text{Tras}_{x,a_i} \text{Rot}_{x,\alpha_i} \quad [3]$$

This sequence corresponds to the following operations to move frame $\{i - 1\}$ to coincide with frame $\{i\}$:

1. Rotate about z_{i-1} by an angle θ_i .
2. Translate along z_{i-1} by a distance d_i .
3. Translate along the new x_i axis by a distance a_i .
4. Rotate about the new x_i axis by an angle α_i .

The resulting 4×4 matrix A_i , is a function of a single joint variable q_i and is given by:

$$A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [4]$$

This matrix, shown in Equation 4, encapsulates both the rotation (the upper-left 3×3 submatrix) and the translation (the upper-right 3×1 vector) that maps the coordinates of a point from frame $\{i\}$ to frame $\{i - 1\}$.

Methodology

Applying the D-H formulation, the direct kinematics for any specific manipulator can be derived. This section details the derivation for a 4-DOF manipulator with revolute joints (Nugroho, Setiawan, & Munadi, 2024).

This is a common configuration for tasks such as pick and place, assembly, and material handling.

Direct Kinematics

The direct kinematic transformation, which describes the pose of the end-effector frame $\{4\}$ with respect to the base frame $\{0\}$, is found by serially multiplying the transformation matrices of each individual link, A_i . The order of multiplication is critical and follows the chain from the base to the end-effector, as shown in Equation 5.

$$T_0^4 = A_1(q_0)A_2(q_1)A_3(q_2)A_4(q_3) \quad [5]$$

The resulting matrix T_0^4 , is a 4×4 homogeneous transformation matrix that is a function of the four joint variables and takes the form shown in Equation 6.

$$T_0^4 = \begin{bmatrix} R & P \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [6]$$

The upper-left 3×3 submatrix is the rotation matrix describing the orientation of the end-effector frame relative to the base frame. The upper-right 3×1 column vector P , gives the Cartesian coordinates (P_x, P_y, P_z) of the end-effector frame's origin with respect to the base frame's origin. Extracting these terms provides the final direct kinematics equations.

To provide a concrete mathematical derivation, a typical 4-DOF articulated arm with all revolute joints is considered. This robot consists of a waist (base) rotation, a shoulder joint, an elbow joint, and a wrist joint at the end-effector. All four joints are revolute. This architecture is characteristic of many educational and light industrial robots. A key feature of this design is that the axes for the shoulder, elbow, and wrist joints (joints 2, 3, and 4) are all parallel, which simplifies the kinematic analysis.

The first step is to translate the robot's physical geometry into a concise mathematical format by creating a table of D-H parameters. This table is the fundamental blueprint for the entire kinematic model.

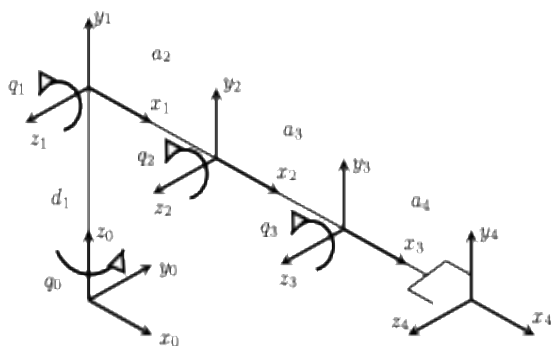
Box 1**Table 1**

D-H Parameters

i	a_i	α_i	d_i	θ_i
1	0	$\pi/2$	d_1	q_0
2	a_2	0	0	q_1
3	a_3	0	0	q_2
4	a_4	0	0	q_3

Source: Own Elaboration

Table 1 is constructed based on a representative geometry shown in Figure 1, where d_1 is the base height, a_2 is the length of the upper arm (shoulder to elbow), a_3 is the length of the forearm (elbow to wrist), and a_4 is the length from the wrist joint to the end-effector tip. The twist angle $\alpha_1 = \pi/2$ is necessary to transition from the vertical rotation axis of the waist, z_0 to the horizontal rotation axis of the shoulder z_1 .

Box 2**Figure 1**

Axis assignment for the manipulator

Source: Own Elaboration

Substituting the parameters from Table 1 into Equation 4 for each link, we obtain the four individual transformation matrices, shown in Equations 7, 8, 9, and 10. Let $c_i = \cos(\theta_i)$ and $s_i = \sin(\theta_i)$.

For $j = 1$:

$$A_1 = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [7]$$

For $j = 2$:

$$A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [8]$$

For $j = 3$:

$$A_3 = \begin{bmatrix} c_3 & -s_3 & 0 & a_3 c_3 \\ s_3 & c_3 & 0 & a_3 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [9]$$

For $j = 4$:

$$A_4 = \begin{bmatrix} c_4 & -s_4 & 0 & a_4 c_4 \\ s_4 & c_4 & 0 & a_4 s_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [10]$$

The total transformation matrix T_0^4 is the product of $A_1 A_2 A_3 A_4$. The multiplication is performed sequentially. First, we find $T_0^2 = A_1 A_2$, then $T_0^3 = T_0^2 A_3$, and finally $T_0^4 = T_0^3 A_4$.

Let $c_{23} = \cos(q_1 + q_2)$, $s_{23} = \sin(q_1 + q_2)$, $c_{234} = \cos(q_1 + q_2 + q_3)$, and $s_{234} = \sin(q_1 + q_2 + q_3)$.

After performing the symbolic matrix multiplication, the resulting matrix T_0^4 is:

$$T_0^4 = \begin{bmatrix} c_1 c_{234} & -c_1 s_{234} & s_1 & c_1 (a_2 c_2 + a_3 c_{23} + a_4 c_{234}) \\ s_1 c_{234} & -s_1 s_{234} & -c_1 & s_1 (a_2 c_2 + a_3 c_{23} + a_4 c_{234}) \\ s_{234} & c_{234} & 0 & d_1 + a_2 s_2 + a_3 s_{23} + a_4 s_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [11]$$

From the last column of this matrix shown in Equation 11, we can extract the direct kinematics equations for the end-effector position (P_x, P_y, P_z):

$$\begin{aligned} P_x &= c_1 (a_2 c_2 + a_3 c_{23} + a_4 c_{234}) \\ P_y &= s_1 (a_2 c_2 + a_3 c_{23} + a_4 c_{234}) \\ P_z &= d_1 + a_2 s_2 + a_3 s_{23} + a_4 s_{234} \end{aligned} \quad [12]$$

This set of terms present in Equation 12 provides the direct and unique mapping from any given set of joint angles (q_0, q_1, q_2, q_3) to the resulting 3D position of the end-effector.

Inverse Kinematics

Inverse kinematics (IK) addresses the problem of finding the necessary joint angles to position the end-effector at a desired location and orientation in Cartesian space. This is a fundamentally more complex task than direct kinematics and is essential for any practical manipulator control application (Kucuk & Bingul, 2004). The difficulty in solving the IK problem stems from the non-linear trigonometric equations derived from direct kinematics. For a general 4-DOF manipulator, several challenges must be addressed.

For example, a solution to the IK problem is not guaranteed, as if the target pose is outside the robot's reachable workspace, no set of real-valued joint angles can satisfy the kinematic equations. Furthermore, a single reachable pose can often be achieved by multiple distinct configurations.

A classic example is the elbow-up and elbow-down ambiguity. For the manipulator in question, additional solutions can arise from the base rotation, such as a left-shoulder versus right-shoulder configuration to reach the same point. The existence of multiple solutions requires the control system to implement a strategy to choose a unique and consistent solution, often based on criteria such as proximity to the current configuration or avoiding joint limits.

IK solutions can be broadly classified into two types: analytical and numerical. Analytical or closed-form solutions provide an exact set of equations to solve for the joint variables. They are computationally very fast and efficient, making them ideal for real-time control. Additionally, they provide a complete characterization of all possible solutions. However, closed-form solutions only exist for manipulators that satisfy certain geometric conditions.

Numerical solutions use iterative algorithms to converge on a solution. They are more general and can be applied to any robot geometry, but they are computationally more expensive and may fail to converge to a solution.

Analytical Solution

A critical aspect of solving the IK for the 4-DOF manipulator is recognizing that it cannot independently achieve an arbitrary 6-DOF pose (3 for position, 3 for orientation). A system with 6 constraints and only 4 controllable variables is overdetermined. Therefore, any solvable IK problem for this manipulator must have an implicit constraint. The problem is usually framed as finding the joint angles for a 3D position (P_x, P_y, P_z) with a single orientation constraint. Two main analytical methods can be used for this solution: the geometric approach and the algebraic approach. The geometric approach is often more intuitive as it relies on decomposing the spatial problem into a series of simpler 2D problems using trigonometry.

The base rotation angle, q_0 , determines the vertical plane in which the rest of the arm operates. By projecting the manipulator onto the base's $x_0 - y_0$ plane, q_0 can be found directly from the desired end-effector coordinates (P_x, P_y) , as shown in Equation 13.

$$q_0 = \text{atan2}(P_y, P_x) \quad [13]$$

The two-argument arctangent function, atan2 is used to correctly determine the quadrant of the angle and handle the case where $P_x = 0$. A second solution for q_0 also exists, which is $q_0' = q_0 + \pi$. This corresponds to a rear-facing solution, which may require the other joints to hyperextend and is often discarded as impractical.

With q_0 known, the 3D problem reduces to a 2D problem within the vertical plane defined by q_0 . If a new coordinate system is defined within this plane, the horizontal distance from the base axis to the end-effector is $r = \sqrt{P_x^2 + P_y^2}$. The position of its center, (P_{ex}, P_{ez}) , relative to the origin of frame $\{1\}$ is given in Equation 14.

$$\begin{aligned} P_{ex} &= r - a_4 \cos(\phi) \\ P_{ez} &= P_z - d_1 - a_4 \sin(\phi) \end{aligned} \quad [14]$$

Here, ϕ is the desired angle of the end-effector with respect to the horizontal, which is a required input for a fully constrained problem. Now that we have a planar manipulator with two revolute joints that must reach the point (P_{ex}, P_{ez}) , the squared distance from the shoulder to the wrist is $D^2 = P_{ex}^2 + P_{ez}^2$. Applying the law of cosines to the triangle formed by links a_2, a_3 , and the line segment D , we get $D^2 = a_2^2 + a_3^2 - 2a_2a_3 \cos(\pi - q_2)$. Since $\cos(\pi - q_2) = -\cos(q_2)$, this simplifies to:

$$\cos(q_2) = \frac{P_{wx}^2 + P_{wz}^2 - a_2^2 - a_3^2}{2a_2a_3} \quad [15]$$

This Equation 15 can be solved for q_2 . Since $\cos(q_2) = \cos(-q_2)$, there are two possible solutions for q_2 , corresponding to the elbow-up and elbow-down configurations:

$$q_2 = \pm \text{acos} \left(\frac{P_{ex}^2 + P_{ez}^2 - a_2^2 - a_3^2}{2a_2a_3} \right) \quad [16]$$

Once q_2 is known, as shown in Equation 16, q_1 can be found using trigonometric identities. Let $\beta = \text{atan2}(P_{wz}, P_{wx})$ and let γ be the angle inside the triangle at the shoulder joint, which can be found with another application of the law of cosines, as shown in Equation 17.

$$\cos(\gamma) = \frac{a_2^2 + D^2 - a_3^2}{2a_2D} \quad [17]$$

Then, q_1 is the sum or difference of these angles, depending on the chosen elbow configuration:

$$q_1 = \beta \mp \gamma \quad [18]$$

The sign in Equation 18 depends on whether the elbow-up ($+q_2$) or elbow-down ($-q_2$) solution was chosen.

The final joint angle, q_3 , is determined by the constraint that the sum of the joint angles must equal the desired absolute angle ϕ , i.e., $\phi = q_1 + q_2 + q_3$.

Therefore, q_3 can be solved directly in Equation 19.

$$q_3 = \phi - q_1 - q_2 \quad [19]$$

This geometric approach yields up to two distinct solutions for the set of joint angles (q_0, q_1, q_2, q_3) for a given reachable pose.

The algebraic approach is more systematic and relies on manipulating the matrix equations of direct kinematics. The goal is to isolate the joint variables one by one.

The process begins with Equation 20, where the desired end-effector pose T_{obj} is known:

$$T_0^4 = T_{obj} = \begin{bmatrix} n_x & s_x & a_x & P_x \\ n_y & s_y & a_y & P_y \\ n_z & s_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [20]$$

Here, we can pre-multiply both sides by the inverse of the first transformation matrix, A_1^{-1} to get Equation 21.

$$\begin{aligned} A_1^{-1}T_0^4 &= A_2A_3A_4 \\ A_1^{-1}T_{obj} &= A_2A_3A_4 \end{aligned} \quad [21]$$

The matrix A_1^{-1} is easy to compute. The product on the right side, $A_2A_3A_4$, results in a matrix whose elements are functions of q_1, q_2, q_3 . The matrix on the left side, $A_1^{-1}T_{obj}$, has elements that are functions of q_0 and the known components of the target pose.

By equating the corresponding elements of the matrices on both sides of the equation, a system of twelve scalar equations is generated. These equations can be solved systematically. For example, in Equation 22, equating the (1,4) and (2,4) elements yields:

$$\begin{aligned} P_x c_1 + P_y s_1 &= a_2 c_2 + a_3 c_{23} + a_4 c_{234} \\ P_z - d_1 &= a_2 s_2 + a_3 s_{23} + a_4 s_{234} \end{aligned} \quad [22]$$

This algebraic method, though less intuitive, is more rigorous and less prone to geometric oversights. It leads to the same set of solutions as the geometric approach, confirming the results through a different mathematical pathway. The full derivation involves extensive use of trigonometric identities to isolate the unknown joint angles.

Numerical Solution

Numerical methods offer an alternative approach to analytical solutions and are characterized by iterative algorithms that progressively refine an initial guess to find the joint angles that minimize the error between the current and desired pose. This effectively transforms the IK problem into an optimization problem, where the objective is to minimize a cost function representing the pose error.

The main advantage of numerical methods lies in their generality and versatility. They are highly applicable to different types of manipulators with complex kinematic structures, such as redundant manipulators. All this without requiring mechanism-specific mathematical derivations beyond direct kinematics. This makes them a good choice when dealing with complex or redundant manipulators where analytical solutions are intractable. The most common numerical methods are:

1. Newton-Raphson, without Jacobian pseudoinverse (NR pse=0) or with Jacobian pseudoinverse (NR pse=1), is a method that iteratively adjusts joint angles by solving for joint velocity changes using the Jacobian pseudoinverse to reduce the error. It is known for its quadratic convergence rate for non-linear functions.
2. Gauss-Newton, without Jacobian pseudoinverse (GN pse=0) or with Jacobian pseudoinverse (GN pse=1), is an extension of Newton's method that approximates the Hessian matrix using the Jacobian matrix.
3. Levenberg-Marquardt (LM), is a hybrid optimization algorithm that combines the strengths of Gauss-Newton and gradient descent. It employs a damping factor to control the step size, which significantly improves robustness, especially when approaching singularities. Some versions include Wampler's (LMW), which allows for dynamic handling of singularities by introducing a damping factor that can be 1×10^{-4} or 1×10^{-6} , considering that a larger damping factor leads to a more stable but slower solution. Another numerical method is Chan's (LMC), which uses a fixed damping factor of 1.0 or 0.1 and provides the method with the stability to resolve singularities and detect targets that the manipulator cannot physically reach. Finally, Sugihara introduces another LM variant (LMS) that allows the LM method to converge under different conditions through a specific factor or bias, which can be 0.001 or 0.0001 .

For the numerical solution of this manipulator, the solutions provided by the Robotics Toolbox (RTB) in Python by Peter Corke are used (Haviland & Corke, 2024).

Results

For the following poses $[P_x, P_y, P_z, \phi, \theta, \psi]$, shown in Table 2, the inverse kinematics of the manipulator is solved using the methods described above. Tables 3, 4, and 5 provide the solutions to the described poses. Figures 2, 3, and 4 show the manipulator's pose for each solution.

Box 3

Table 2

Different poses to be solved

Pose	P_x	P_y	P_z
1	-0.0789	-0.025	1.051
2	0.227	-0.118	0.853
3	-0.333	0.077	0.573
Pose	ϕ	θ	ψ
1	-89.997°	-49.375°	-162.429°
2	-90.000°	-8.911°	152.515°
3	89.977°	-73.462°	166.939°

Source: Own Elaboration

Box 4

Table 3

Solutions to pose 1

Method	q_0	q_1	q_2	q_3
Analítico	0.3067	1.352	0.6457	0.282
NR pse=0	0.3067	-4.418	-0.6478	-5.221
GN pse=0	0.3067	-4.419	-0.6464	-5.221
NR pse=1	0.3067	1.346	0.6605	0.2737
GN pse=1	0.3067	1.352	0.646	0.2818
LMW 1e-4	0.3067	1.869	-0.6589	-5.214
LMW 1e-6	0.3067	1.865	-6.931	1.062
LMC 1.0	0.3067	-4.934	0.6516	0.2785
LMC 0.1	0.3067	1.351	0.6479	0.2807
LMS 0.001	0.3067	1.87	-0.6602	1.07
LMS 0.0001	0.3067	1.349	0.6531	0.2776

Source: Own Elaboration

Box 5

Table 4

Solutions to pose 2

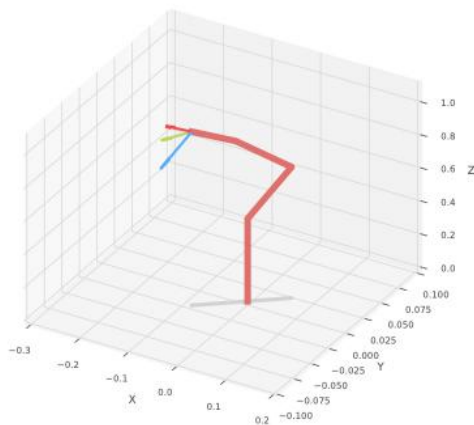
Method	q_0	q_1	q_2	q_3
Analítico	2.913	-0.7324	1.833	0.1816
NR pse=0	-3.37	-0.7327	-4.447	0.1787
GN pse=0	-3.37	-0.7321	1.834	0.1801
NR pse=1	2.913	-5.689	-1.835	-3.761
GN pse=1	2.913	-0.734	-4.449	0.182
LMW 1e-4	-3.37	0.5909	-8.117	2.525
LMW 1e-6	-3.37	-7.016	-4.45	0.1816
LMC 1.0	2.913	-5.693	-1.833	-3.759
LMC 0.1	2.913	0.5905	-1.833	2.525
LMS 0.001	-3.37	-0.7332	-4.447	0.1793
LMS 0.0001	2.913	-0.7329	1.834	0.181

Source: Own Elaboration

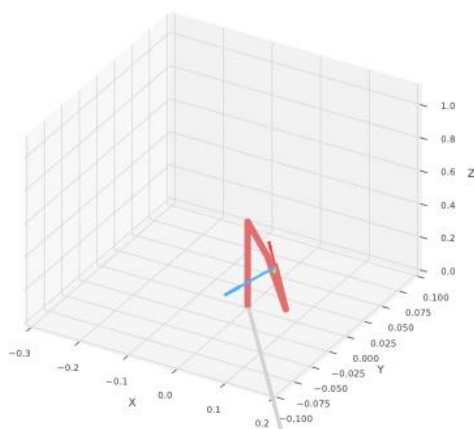
Box 6**Table 5**

Solutions to pose 3

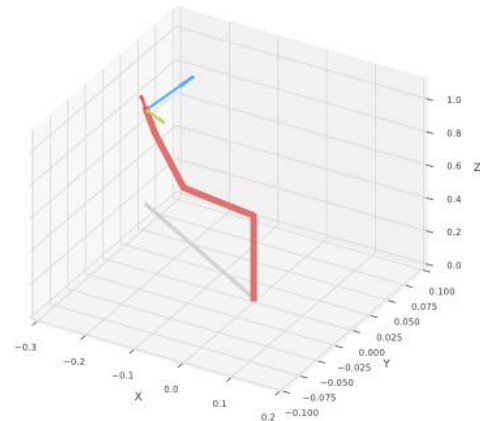
Method	q_0	q_1	q_2	q_3
Analítico	-0.4797	0.9167	-0.3922	2.462
NR pse=0	-0.4797	0.5936	-5.866	1.975
GN pse=0	-0.4797	0.9232	-0.4082	2.471
NR pse=1	-0.4797	-5.682	-5.884	1.986
GN pse=1	-0.4797	-5.688	0.4133	-4.306
LMW 1e-4	-0.4797	0.6025	-5.888	1.988
LMW 1e-6	-0.4797	0.9207	-0.4022	2.468
LMC 1.0	-0.4797	0.9172	-0.3935	-3.821
LMC 0.1	-0.4797	-5.366	-0.3934	2.462
LMS 0.001	-0.4797	-5.359	-0.4115	-3.81

*Source: Own elaboration***Box 7****Figure 2**

Pose 1 of the manipulator

*Source: Own Elaboration***Box 9****Figure 3**

Pose 2 of the manipulator

*Source: Own Elaboration***Box 10****Figure 4**

Pose 1 of the manipulator

*Source: Own Elaboration***Conclusions**

The variability observed in Tables 3, 4, and 5 underscores the problem of multiple solutions and highlights the significant influence of algorithm parameters, such as damping factors and initial conditions, on the convergence path and the resulting configuration. It also demonstrates that some numerical solutions may be less practical in terms of large joint angles or proximity to limits.

Numerical methods transform the IK problem into an optimization problem, representing a fundamental paradigm shift. Instead of seeking an exact closed-form inversion, the problem is reformulated as the minimization of an error function. This approach allows for a general solution that does not depend on the manipulator's specific geometric properties, making it applicable to any robot configuration.

The trade-off is the loss of an exactness guarantee and the introduction of challenges related to convergence and local minima. This shift in approach enables the development of highly versatile robotics software and control systems that can adapt to a wide diversity of robot designs without requiring the derivation of complex analytical solutions. However, the results from numerical methods require careful evaluation in a practical context.

The exhaustive analysis of the 4-DOF manipulator reveals a tightly interconnected chain of mathematical and physical principles. This article has systematically detailed this chain, demonstrating how each concept builds upon the previous one to form a complete model of the robot's behavior. The synthesis of these topics provides an essential understanding for advanced robotics applications.

While the current analysis provides a comprehensive understanding of manipulator kinematics, the next logical step for future work is to extend this analytical framework to the realm of robot dynamics. This extension would involve the systematic incorporation of critical physical factors such as the mass distribution of each link, its moments of inertia, and the influence of external forces acting on the manipulator.

The overall purpose of this dynamic analysis is to accurately model the accelerations and torques required for the robot's motion. In doing so, the bridge from kinematic configuration analysis to a complete understanding and control of dynamic movement would be effectively completed.

Conflict of interest

The authors declare no interest conflict. They have no known competing financial interests or personal relationships that could have appeared to influence the article reported in this article.

Author contribution

García-Trinidad, Enrique: Contributed to the project idea, research method and technique.

Arcos-Hernández, Emmanuel: Contributed to the review and revision of the mathematical analysis.

Torres-Valle, José: Contributed to the derivation of the DH parameters.

Hernández-Borja, Carlos: Contributed to the Linux and ROS2 system configuration.

Availability of data and materials

The data that support the findings of this study are openly available in the main author's GitHub repository at <https://github.com/enriGarcia>.

References

Basics

Bao, G., Liu, S., & Zhao, H. (2017/05). [Kinematics simulation of 4 dof manipulator](#). In Proceedings of the 2017 2nd international conference on materials science, machinery and energy engineering (msmee 2017) (p. 400-408). Atlantis Press. Retrieved from <https://doi.org/10.2991/msmee-17.2017.81> doi: 10.2991/msmee-17.2017.81

Corke, P. I. (2007). [A simple and systematic approach to assigning denavit–hartenberg parameters](#). IEEE Transactions on Robotics, 23 (3), 590-594. doi: 10.1109/TRO.2007.896765

Supports

Guida, R., De Simone, M. C., Dašić, P., & Guida, D. (2019, jul). [Modeling techniques for kinematic analysis of a six-axis robotic arm](#). IOP Conference Series: Materials Science and Engineering, 568 (1), 012115. Retrieved from <https://dx.doi.org/10.1088/1757899X/568/1/012115> doi: 10.1088/1757-899X/568/1/012115

Haviland, J., & Corke, P. (2024). [Manipulator differential kinematics: Part i: Kinematics, velocity, and applications](#). IEEE Robotics and Automation Magazine, 31 (4), 149-158. doi:10.1109/MRA.2023.3270228

KuCuk, S., & Bingul, Z. (2004). [The inverse kinematics solutions of industrial robot manipulators](#). In Proceedings of the iee international conference on mechatronics, 2004. icm'04. (pp.274–279).

Malik, A., Lischuk, Y., Henderson, T., & Prazenica, R. (2022). [A deep reinforcement-learning approach for inverse kinematics solution of a high degree of freedom robotic manipulator](#). Robotics, 11 (2). Retrieved from <https://www.mdpi.com/2218-6581/11/2/44> doi: 10.3390/robotics11020044

Manseur, R., & Doty, K. L. (1988). [A fast algorithm for inverse kinematic analysis of robot manipulators](#). The International Journal of Robotics Research, 7 (3), 52-63. Retrieved from <https://doi.org/10.1177/027836498800700304> doi: 10.1177/027836498800700304

Spong, M. (2022). [Robot modeling and control, second edition](#). IEEE Control Systems, 42 (1), 126-128. doi: 10.1109/MCS.2021.3122271

Differences

Mohammed, A. A., & Sunar, M. (2015). [Kinematics modeling of a 4-dof robotic arm](#). In 2015 international conference on control, automation and robotics (p. 87-91). doi: 10.1109/ICCAR.2015.7166008

Nugroho, E. A., Setiawan, J. D., & Munadi, M. (2024). [Physical modeling of a 4-dof manipulator for learning purposes using cad and gui](#). International Journal of Mechanical Engineering and Robotics Research, 13 (3), 392-401.