

Comparison of Inverse Kinematics Approaches for 3-Joint 4-Link Robots in 2D Space

Zhuohao Cao^{1,a,*}

¹*University of Newcastle upon Tyne, Newcastle upon Tyne, United Kingdom*

a. 2957122609@qq.com

**corresponding author*

Abstract: Nowadays, the real world widely uses robotic arms, leading to an increasing amount of research in this field. This work is mainly about the Inverse Kinematics about 3 joint 4 link robots in 2D space. The primary challenge lies in calculating joint angles using known final coordinates. This research will present two distinct approaches, each capable of resolving the problem. By using cosine function, the first way, geometrical, could solve the problem in 2-D space. Despite appearing more challenging than the first method, the algebraic approach, the second method, can solve the problem in both 2-D and 3-D space by modifying the size of the rotation matrix, also known as the DH matrix. Due to geometric limitations, error analysis will be presented exclusively when employing algebraic methods. We calculate the derivative on the axis to depict the changing trend. Meanwhile, further study is also provided.

Keywords: 3 joint 4 link robots, 2D Space, joint angles, error analyze, Inverse Kinematics

1. Introduction

Kinematics is the process of calculating the spatial position of the ends of a linked structure, such as a human body joint, with all joint angles given. This process is relatively simple and has a unique solution. Meanwhile, Inverse kinematics is the opposite. The purpose of inverse kinematics is to solve the angle of the joint for a given spatial position at the end. This process is more difficult, and in general, there may be infinite solutions to inverse kinematic problems.

The process of inverse dynamics is very useful in some scenarios. People use robotic arms, for example, to grasp objects placed in specific spatial locations. In that case, it is necessary to use inverse dynamics to calculate the rotation angle of each joint of the robotic arm and then drive the robotic arm to grasp the object [1]. Additionally, inverse kinematics is a very significant component in the generation of CG animation and game programming. Furthermore, the study of 2-D reverse tracking problems plays a crucial role in solving specific cases. Consider issues such as painting, plane splicing, and PCB board assembly.

However, calculating the joint angles of the robotic arm is the first step in solving the above problems. In that case, it is necessary to do research about inverse kinematics and analyze the rotation error.

The main ways to solve the problem are the geometrical approach and algebraic approach, both of which are based on the finding of the final destination, no matter what kind of dimension the object is in, 2D or 3D. Both approaches are applicable to various types of robots, and this paper demonstrates

their application using a 3-joint, 4-link robot as an example. These two methods, in contrast to others, possess a more intuitive expression and simpler principle.

This paper focuses on a 2D robotic arm due to its straightforward structure and the ease of standardizing hardware manufacturing, which enhances its suitability for large-scale applications.

2. Literature Review

The method of the algebraic approach is based on DH Matrix. Provided by Denavit and Hartenber in 1995, in this method, a coordinate system is fixed on each link of the robot, and then a homogeneous transformation matrix of 4×4 is used to describe the spatial relationship between the two adjacent links [3]. Sequential transformation finally deduces the end effector's pose relative to the base coordinate system, establishing the robot's kinematic equation [4].

However, when the question is considered in 2-D space, the situation changes. Regardless of the complexity of the problems, we can perfectly apply a geometrical approach. Eliminating the need for space calculations simplifies the problem. For example, after shifting in 3-D space, the latest link changes the angle between itself and plane XOY, as well as plane YOZ. It has a final destination coordinate point P(x,y,z). So the angle formed by the line (OP) and surface could be denoted in the following form by using the definition of angle between line and plane:

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{y}{\sqrt{x^2+z^2}}\right) \\ \arctan\left(\frac{z}{\sqrt{x^2+y^2}}\right) \\ \arctan\left(\frac{x}{\sqrt{y^2+z^2}}\right) \end{bmatrix} \quad (1)$$

When α is the angle formed by plane XOZ and line OP, β is the angle formed by plane XOY and line OP, γ is the angle formed by plane YOZ and line OP [5].

It seems complicated. Because when α and β change, γ may change. It may also remain unchanged. This is the problem, because each link may point in different directions in 3D space. No matter how the link changes, the angle between the final destination point P(x,y) has a fixed relation between the angle XOP (θ) and angle YOP (φ) :

$$\begin{bmatrix} \theta \\ \varphi \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{y}{x}\right) \\ \arctan\left(\frac{x}{y}\right) \end{bmatrix} \quad (2)$$

At the same time, the sum of the two different angles remains constant:

$$\theta + \varphi = \frac{\pi}{2} \quad (3)$$

It is evident that tasks are simplified in 2-D space. However, when the number of links increases, the computation of both algebraic and geometric methods will become significantly more challenging. The two methods can be employed when the robotic arm possesses a limited number of linkages.

3. Methodology

In this part, it divides the approach into two parts: the Geometrical Approach and the Algebraic Approach. The principles will be discussed in the following parts.

3.1. Geometrical Approach

Figure 1 illustrates the inverse kinematics in a 2D space, which can be solved using cosine theory.

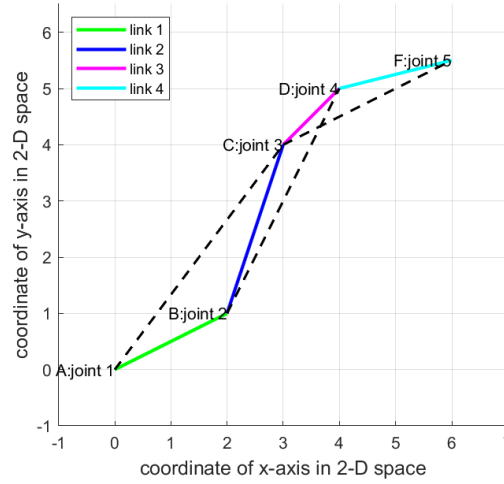


Figure 1: An example of Inverse Kinematics

As shown in Fig. 1, The final destination point F is known (x_f, y_f) , the length of each link is known. First define the angles formed by two relative links, which is also known as joint angle:

$$\begin{bmatrix} \text{joint angle 1} \\ \text{joint angle 2} \\ \text{joint angle 3} \\ \text{joint angle 4} \end{bmatrix} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} \quad (4)$$

Then define the length of links of robotic arm:

$$\begin{bmatrix} \text{link 1} \\ \text{link 2} \\ \text{link 3} \\ \text{link 4} \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} \quad (5)$$

As well as a limitation equation about the four angles :

$$f(\theta_1, \theta_2, \theta_3, \theta_4) = \delta \quad (6)$$

To simplify the problem, we could define the sum of all joint angles γ , then consider the coordinate of point D by calculating the projection of a line segment on the coordinate axes:

$$\sum_{i=1}^4 \theta_i = \gamma \quad (7)$$

$$(x_D, y_D) = [(x_f - \alpha_4 \cdot \cos(\gamma)), ((x_f - \alpha_4 \cdot \cos(\gamma)))] \quad (8)$$

The same way to process coordinate of point C:

$$(x_C, y_C) = [(x_D - \alpha_3 \cdot \cos(\gamma - \theta_4)), ((x_D - \alpha_3 \cdot \cos(\gamma - \theta_4)))] \quad (9)$$

For point B and A, inverse trigonometric functions and cosine function could be used to solve θ_1 and θ_2 :

$$\theta_2 = \pi - \angle ABC = \arccos\left(\frac{\alpha_1^2 + a_2^2 - (x_C^2 + y_C^2)}{2\alpha_1 a_2}\right) \quad (10)$$

$$\theta_1 = \angle CAX - \angle BAX = \arctan\left(\frac{y_C}{x_C}\right) - \arccos\left(\frac{\alpha_1^2 - a_2^2 + (x_C^2 + y_C^2)}{2\alpha_1 \sqrt{x_C^2 + y_C^2}}\right) \quad (11)$$

Based on the above analyze, equations (5) to (10) are the series solutions to the questions. The joint angles could be calculated.

3.2. Algebraic Approach

Different from the way of geometrical, the algebra approach provides a new perspective for solving the model with more links and joints. In this approach, the rotation matrix, also known as the DH matrix, is employed to solve the problem in this part.

In 2-D dimensions, the rotation of a line formed by a random point (named as $P(x_P, y_P)$) and $O(0,0)$ could be defined by the following matrix representations [7], [8]:

$$R_T = \begin{bmatrix} \cos\theta & -\sin\theta & dx \\ \sin\theta & \cos\theta & dy \\ 0 & 0 & 1 \end{bmatrix} \quad (12)$$

When θ is the rotation angle, dx and dy are the movement variables related to axes after changing. So if a point $C(1,1)$ rotates $\frac{\pi}{4}$ and $dx=dy=1$, the final destination coordinate would be: $C'(1, \sqrt{2} + 1)$, calculated by the matrix:

$$C' = R_T C = \begin{bmatrix} \cos\theta & -\sin\theta & dx \\ \sin\theta & \cos\theta & dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 1 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ \sqrt{2} + 1 \\ 1 \end{bmatrix} \quad (13)$$

However, in actual Inverse Kinematics scenarios, the length of different links may be different. If we define the distance a between the point to the rotation center, as well as a constant added [9]:

$${}_{n-1}^n T = \frac{a_n}{a_{n-1}} \begin{bmatrix} \cos\theta & -\sin\theta & dx \\ \sin\theta & \cos\theta & dy \\ 0 & 0 & 1 \end{bmatrix} \quad (14)$$

The new version of the rotation matrix incorporates the division of link lengths, providing the final coordinate after accounting for link differences.

In that case, the equation of algebra could be set up like the following form by adjusting equation (14) into equation (13):

$$P^n = {}_{n-1}^n T P^{n-1} \quad (15)$$

In order to establish a relation between P^n and P^1 , we conclude all the points and try to get the final destination coordinate:

$$P^n = \prod_{i=2}^n {}^i T {}^{i-1} P^{i-1} \quad (16)$$

What's more, in each multiple iteration, the θ is the rotation angle formed by the original point(0,0), point in previous step $P^{i-1}(x_{pi-1}, y_{pi-1})$, and point after rotation $P^i(x_{pi}, y_{pi})$. The rotation angle is defined by the cosine function as:

$$\theta = \arccos \frac{(x_{pi-1})^2 + (y_{pi-1})^2 + (x_{pi})^2 + (y_{pi})^2 - \alpha_i^2}{2\sqrt{(x_{pi-1})^2 + (y_{pi-1})^2}\sqrt{(x_{pi})^2 + (y_{pi})^2}} \quad (17)$$

In the previous part, the final coordinate could be solved successfully by using both ways. However, when talking about the robotic arm, each tiny bias could make a big difference. For example, the rotation angle and movement may be a little bit different to the theoretical calculation, the error is set to δ . In the following analysis, δ is always larger than 0, and δ is approach to 0 in the positive direction; the rotation angle θ is always larger than 0 and less than $\frac{\pi}{4}$. So the equation which tells the final coordinate becomes:

$$(P^n)' = ({}^{n-1}T)' (P^{n-1})' = \frac{a_n}{a_{n-1}} \begin{bmatrix} \cos(\theta + \delta) & -\sin(\theta + \delta) & dx + \mu \\ \sin(\theta + \delta) & \cos(\theta + \delta) & dy + \mu \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{pi-1} \\ y_{pi-1} \\ 1 \end{bmatrix} \quad (18)$$

In order to find the error, we could use the difference of P^n to show the result:

$$\begin{aligned} \Delta P^n = (P^n)' - P^n &= \frac{a_n}{a_{n-1}} \left\{ \begin{bmatrix} x_{pi-1} \cos \theta - y_{pi-1} \sin \theta + dx \\ x_{pi-1} \sin \theta + y_{pi-1} \cos \theta + dy \\ 1 \end{bmatrix} \right. \\ &\quad \left. - \begin{bmatrix} x_{pi-1} \cos(\theta + \delta) - y_{pi-1} \sin(\theta + \delta) + dx + \mu \\ x_{pi-1} \sin(\theta + \delta) + y_{pi-1} \cos(\theta + \delta) + dy + \mu \\ 1 \end{bmatrix} \right\} \quad (19) \end{aligned}$$

Partial derivative could show the trend of change, by using the rule of matrix derivative, the partial derivative of the error ΔP^n to each coordinate could be calculated [10]:

$$\frac{\partial \Delta P^n}{\partial C} = \begin{bmatrix} \frac{\partial \Delta P^n}{\partial x_{pi-1}} \\ \frac{\partial \Delta P^n}{\partial y_{pi-1}} \\ \frac{\partial \Delta P^n}{\partial z_{pi-1}} \end{bmatrix} = \frac{a_n}{a_{n-1}} \begin{bmatrix} \cos \theta - \cos(\theta + \delta) + \frac{\partial \mu}{\partial x_{pi-1}} \\ \cos \theta + \cos(\theta + \delta) + \frac{\partial \mu}{\partial y_{pi-1}} \\ 0 \end{bmatrix} \quad (20)$$

The error μ is a human-defined variable, to make the result more Intuitive, μ is set to have no relation on each coordinate as well as δ , and it increase the same percentage during each iteration. Based on the above fact, the equation becomes:

$$\frac{\partial \Delta P^n}{\partial C} = k\% \cdot \begin{bmatrix} \cos\theta - \cos(\theta + \delta) \\ \cos\theta + \cos(\theta + \delta) \\ 0 \end{bmatrix} \quad (21)$$

However, as the variable δ defined above:

$$\lim_{\delta \sim 0^+} \cos\theta - \cos(\theta + \delta) = 0, \lim_{\delta \sim 0^+} \cos\theta + \cos(\theta + \delta) = 2\cos\theta > 0 \quad (22)$$

So under the following conditions:

1. θ is always larger than 0 and less than $\frac{\pi}{4}$, δ is always larger than 0, and approach to 0 in the positive direction
2. The derivative between ΔP^n and x-axis coordinate nearly keeps the same
3. An increase the same percentage during each iteration

The derivative between ΔP^n and y-axis is always larger than 0, The derivative between ΔP^n and x-axis is always close to 0.

Based on the analysis above, the geometrical approach primarily employs cosine functions and inverse trigonometric definitions, emphasizing the significant impact of dimensionality on computational complexity. While the geometrical method is notably simple in 2-D space, its complexity increases substantially in 3-D space. In contrast, although the algebraic approach may seem computationally in 2-D space, it becomes more efficient than the geometrical method when applied to 3-D calculation.

4. Conclusion

In this research, two distinct solutions are provided to solve the problem, both demonstrating effective performance in 2-D space. The geometrical approach involves setting a limitation equation for all joint angles, enabling the problem to be solved using cosine functions. The algebraic approach utilizes a rotation matrix, known as the DH matrix, to set up the equations for each iteration. Additionally, error analysis is conducted under specific conditions to further validate the methods.

In all, the geometrical way is suitable for robotic arms with fewer links in 2-D space, while the algebra way is effective for robotic arms with fewer links in both 2-D and 3-D space. However, for solving inverse kinematics problems involving robotic arms with fewer links specifically in 2-D space, the geometrical approach demonstrates superior performance.

References

- [1] M. S. Allahkaram, M. Javad Ahmadi and H. D. Taghirad, "Closed-form Inverse kinematics Equations of a Robotic Finger Mechanism," 2021 9th RSI International Conference on Robotics and Mechatronics (ICRoM), Tehran, Iran, Islamic Republic of, 2021, pp. 157-162
- [2] Y. Cao, W. Wang, L. Ma and X. Wang, "Inverse Kinematics Solution of Redundant Degree of Freedom Robot Based on Improved Quantum Particle Swarm Optimization," 2021 IEEE 7th International Conference on Control Science and Systems Engineering (ICCSSE), Qingdao, China, 2021, pp. 68-72
- [3] J. Xu, W. Liu and L. Li, "An improved GA-based inverse kinematics solution algorithm for underwater manipulators," 2022 13th Asian Control Conference (ASCC), Jeju, Korea, Republic of, 2022, pp. 2091-2096
- [4] V. A. Karapetyan and V. N. Miryanova, "Solving the Inverse Kinematics Problem for a Seven-Link Robot-Manipulator by the Particle Swarm Optimization," 2023 International Russian Smart Industry Conference (SmartIndustryCon), Sochi, Russian Federation, 2023, pp. 186-190
- [5] Y. Ning, R. Hu, Y. Chen, X. Chen, S. Ren and B. Li, "POE-Based Inverse Kinematics Analysis for a Six-DOF Robotic Manipulator," 2022 7th International Conference on Automation, Control and Robotics Engineering (CACRE), Xi'an, China, 2022, pp. 71-75

- [6] J. Zhang, W. Zhang and M. Zhang, "Solvable Criteria and Sequential Solution Method for the General 6R Inverse Kinematics Problem," 2021 IEEE International Conference on Mechatronics and Automation (ICMA), Takamatsu, Japan, 2021, pp. 17-21
- [7] O. V. Zakharov and L. V. Seliverstova, "Positional Control of 6-DoF Robot Based on an Optimal Inverse Kinematics," 2024 International Russian Smart Industry Conference (SmartIndustryCon), Sochi, Russian Federation, 2024, pp. 866-871
- [8] X. Yang, Z. Jiang, J. Hu, C. Wang and F. Yang, "Inverse Kinematics Algorithm for 7-DOF Manipulator with Offset and its Application," 2024 8th International Conference on Robotics, Control and Automation (ICRCA), Shanghai, China, 2024, pp. 53-57
- [9] M. Elseiagy, A. El-Sayed and M. Elsamanty, "A Convergent Approach of Mathematical Modelling and Deep Learning for Inverse Kinematics Solutions in Stewart Platform Parallel Robots," 2024 International Conference on Machine Intelligence and Smart Innovation (ICMISI), Alexandria, Egypt, 2024, pp. 194-199
- [10] Y. Yang, X. Chen and Y. Han, "JANM-IK: Jacobian Argumented Nelder-Mead Algorithm for Inverse Kinematics and its Hardware Acceleration," in IEEE Computer Architecture Letters, vol. 23, no. 1, pp. 45-48, Jan.-June 2024
- [11] A. Yonezawa, H. Yonezawa and I. Kajiwaru, "Simple Inverse Kinematics Computation Considering Joint Motion Efficiency," in IEEE Transactions on Cybernetics, vol. 54, no. 9, pp. 4903-4914, Sept. 2024