

Inverse Kinematics of a Two-Link 2D Planar Robot Arm for Industrial Assembly

Anthony Tan Velasquez, Ronnie Concepcion II, and Ira Valenzuela

Abstract — In assembly industries, robots play a critical role in executing diverse tasks, with their primary function in positioning end effectors accurately. A significant challenge in robotics is the inverse kinematics problem, as it has significant implications for controlling a robot hand's position and orientation. This study derived formulae for inverse kinematic movement by using geometric and trigonometric concepts. The study then coded and used MATLAB to simulate inverse kinematics for a two-link planar robot arm control system. The robot arm can position its end-effector in a two-dimensional plane using its two revolute joints. MATLAB determines the necessary joint angles for moving the end-effector to the required location after receiving input from the user specifying the link lengths and the intended end-effector position. It then creates a graphic representation of the robot arm showing the joints' positions. The outcomes of this investigation effectively showcased the utilization of inverse kinematics to achieve motion control in a two-dimensional, two-linked robot arm, employing the established mathematical formulas. Given the critical importance of precisely positioning the end-effector in assembly robotics, the practical application of this study pertains to industrial assembly robots. The inverse kinematics simulation introduced here serves as a valuable tool for attaining the level of precision in such contexts.

Index Terms — control system, coordinate geometry, inverse kinematics, robotics, trigonometry

I. INTRODUCTION

ROBOTS have become crucial in modern industries as they are used to accomplish various tasks, with their main objective to move end effectors to a specified location. The application of robotics is widespread and encompasses areas such as healthcare, mining, domestic applications, and manufacturing [1]. An example of a robotic arm that is

widely used is the Universal Robots UR10 [2]. The UR10 is a flexible industrial robot designed to work alongside humans. The highlight of the UR10 robot is its versatility, safety features, and ease of use [2]. This study investigates a two-link 2d planar robot arm. Two revolute joints on the robotic arm allow it to move its end-effector and position it precisely inside a two-dimensional plane. A robot arm is made up of numerous joints and connections, with the end-effector typically attached to the second link. Robot arms are commonly constructed using servos that are linked together with the arm's various segments. The revolute joint was the type of joint used in this study. A revolute joint (Fig. 1), is a type of joint in robotics that enables rotational movement between two connected links around a fixed axis [3].

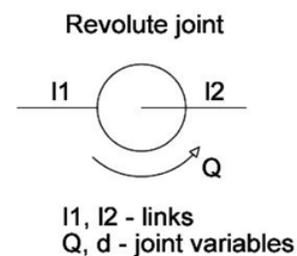


Fig. 1. Revolute Joint [3]

The development of the fundamental concepts of robot motion planning and control can be traced back to their origins in the field of robotics, where they were initially developed with a primary focus on industrial applications. These applications typically involved the use of robotic arms and manipulators in manufacturing and assembly processes, where precision and accuracy were critical for achieving optimal production efficiency [4]. This leads to the inverse kinematic problem, a significant issue in robotics. Controlling the position and orientation of the robot hand requires the use of inverse kinematics (IK). Therefore, it is crucial for robotics to develop answers to the inverse kinematic problem [5].

By using kinematic equations to determine a manipulator’s joint settings, inverse kinematics (IK) enables the movement of an end effector to a predetermined position [6]. As depicted in Fig. 2, the aim of inverse kinematics goal. For the end effector to be positioned and oriented as required, it is crucial to establish the right joint variables [7]. The goal of this study is to derive equations for joint values using inverse kinematics concepts.

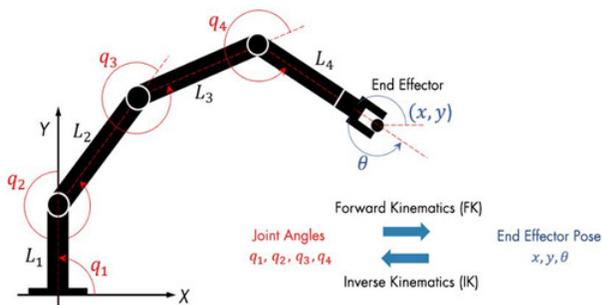


Fig. 2. Configuration of forward or inverse kinematics to set up a robot’s joint positions [8]

An essential part of manufacturing, inverse kinematics is used to train and regulate a robot’s mobility to ensure process accuracy and quality. Calculations are used in manufacturing to determine the necessary joint angles that allow the end-effector to reach a particular geographical position. To create the required form, the movements of the robot arm and material deposition are coordinated simultaneously [9]. Inverse kinematics can be used to design the robot’s motion and ensure that it can carry out the required tasks. The robot’s movements can be planned using inverse kinematics, ensuring that it can do the necessary tasks.

A study on designing a robot arm proposed a planar two link manipulator. This is because of the accessibility of two link planar manipulators as well as being able to resemble human arms [10]. Single-arm industrial robots are limited in their ability to perform certain tasks that require the use of two arms, which has led to an increase in demand for dual-arm robot manipulators in the industrial assembly sector. While operators can complete tasks that require the use of both arms, single-arm robots are unable to replicate this level of dexterity. As a result, industries are turning to dual-arm robots to meet their assembly needs, as they offer greater flexibility and efficiency in performing tasks that require the use of both arms [11]. The availability of two-linked robots is widespread thus, the creation of a precise control system is important because it can improve the robot’s performance in various industrial applications. In pick-and-place operations, which entail taking an object out of one place and putting it in another, it can be carried out by a 2-linked robot arm. To automate repetitive tasks

like sorting, packaging, and assembly, this type of robot arm is often employed in production and assembly lines [12]. With precise and efficient movements, and easy to control for human operators, a 2-linked robot arm can significantly increase productivity and reduce errors in these processes.

Researchers calculated the joint angles required to achieve the target end-effector position to determine the motion and location of the robot arm in a study on 3D-printed robotic arms [13]. Another study employs a geometric method to resolve inverse kinematics issues for arm manipulators while applying inverse kinematics. It entails calculating joint angles with trigonometry to obtain appropriate end-effector location and orientation [14]. This demonstrates how geometry and trigonometry can be applied to inverse dynamics. As a result, by using geometric and trigonometric principles to derive inverse kinematic formulae, a control system can be built.

Fig. 3 illustrates a study that used trigonometry in solving inverse kinematics problems. The study demonstrates that the joint angles may be calculated from the link lengths using the law of cosines and the law of sines [15]. It is possible to establish the location and direction of a robot arm’s end-effector based on the intended location in a coordinate space by utilizing trigonometric functions like sine, cosine, and tangent [16].

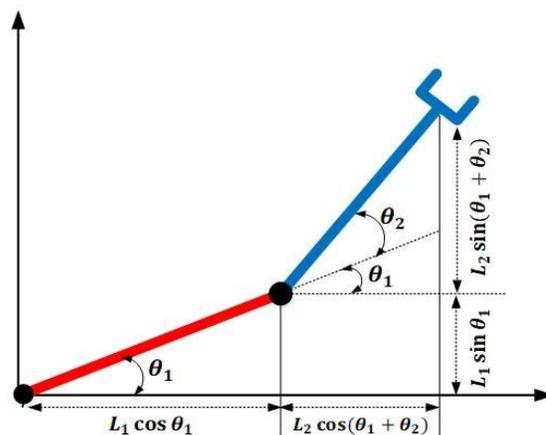


Fig. 3. Inverse kinematics using Trigonometry [15]

A control system for a 2D two-linked robotic arm was developed using the concept of inverse kinematics then simulated. The objective is to first derive equations to obtain the joint values of the arm using geometry and trigonometry. Next, the derived equations will be translated into code to be integrated into a user-interface using Matrix Laboratory (MATLAB). The goal is to simulate the control system by using user-supplied length connections and a specified end-effector location. The outcome will be a plot depicting the computed angles of the two-linked robotic arm.

In this study, joint angles needed to move the end-effector of a two-linked planar robotic arm to a desired coordinate are calculated and visualized using MATLAB to replicate the idea of inverse kinematics in robotics. The main objective is to demonstrate how inverse kinematics can be applied to precisely control the arm's motion in industrial assembly applications. The primary contribution of this study lies in the development and simulation of a control system for a 2D two-linked robotic arm. This involves deriving essential equations, integrating them into the MATLAB interface, and utilizing inverse kinematics to calculate and visualize the joint angles. The study showcase the practical application of inverse kinematics in robotics, specifically in achieving precise control over the arm's movement for industrial assembly tasks.

II. METHODOLOGY

The flowchart of this study's methodology is shown in Fig 4. The study aims to develop an inverse kinematics control system in MATLAB by connecting the concept of inverse kinematics with mathematical equations, specifically using trigonometry and geometry. The developed control system was simulated by providing the length of the links and the preferred end-effector coordinates as inputs. The control system's output will be a plot with the computed angles for the links.



Fig. 4. Flowchart of the Study

The main concept of this study is to employ geometric and trigonometric mathematical formulae to derive equations that will calculate the joint values. The derived formula will

then be coded in MATLAB to create a control system. The control system will be able to calculate the joint values for the inverse kinematic movement to the desired end-effector coordinates. Automating industrial assembly processes requires the use of inverse kinematics, which is a critical approach for computing the joint angles required to place a robot arm's end-effector at a certain location in 2D space [17].

A. Deriving inverse kinematic equations

Deriving the inverse kinematic equation starts by obtaining all the necessary geometric and trigonometric formulas to obtain an equation suitable for a two-link 2d planar robotic arm. Utilizing the relationship between the distance of links aids in determining the values of the joints [18].

Equation 1 displays the Pythagorean theorem-derived Euclidean distance formula. Given that the points are known, it is used to determine the separation between two points in a two-dimensional coordinate plane [19]. In this study, the Euclidean distance formula was used to determine the separation between the end effector and the base of the robot arm.

$$d = \sqrt{(x^2 + y^2)} \quad (1)$$

Equation 2 are the basic trigonometric functions, sine (sin), cosine (cos), and tangent (tan), and their inverse functions. Inverse trigonometric functions are used to determine the angles of their counterpart trigonometric functions, while basic trigonometric functions are used to compute the values of sides and angles of a right triangle. In this study, the functions cos and sin were used to obtain the coordinates of the second joint position by translating polar coordinates to cartesian coordinates. Meanwhile, the robot arm's joint angles were calculated at the same time using inverse trigonometric functions [20]. The arccos and arctan functions were used to calculate the angles of the connections required to establish the locations of the intended end-effector position.

$$\begin{aligned} & \sin \theta \\ & \cos \theta \\ & \tan \theta \\ & \sin^{-1}(x) = \arcsin(x) \\ & \cos^{-1}(x) = \arccos(x) \\ & \tan^{-1}(x) = \arctan(x) \end{aligned} \quad (2)$$

Equation 3 shows the law of cosine. The law of cosine is a geometric solution that establishes a relationship between the lengths of a triangle's sides and the cosine of its angles [21]. In this study, the angle between the robot arm's first link and the line connecting its base to the intended end-effector position was calculated using the cosine law. This

angle was then used to calculate the position of the first joint of the robot arm.

$$\begin{aligned} c^2 &= a^2 + b^2 - 2ab \cdot \cos C \\ b^2 &= a^2 + c^2 - 2ac \cdot \cos B \\ a^2 &= a^2 + c^2 - 2bc \cdot \cos A \end{aligned} \quad (3)$$

Equation 4 shows the derived equation for the joint angles for the links denoted by θ_1 and θ_2 . Based on the location of the end effector, these two equations are used to determine the joint angles of a two-link robot arm. The initial equation (θ_1) determines the angle for the first link by utilizing the x and y coordinates of the end effector, the lengths of both links (L_1 and L_2), and the distance from the end effector to the origin (d). The subsequent equation (θ_2) computes the angle for the second link based on the lengths of the two links (L_1 and L_2) and the distance from the end effector to the origin (d). These equations, when combined, yield the angles corresponding to the two links of the robot arm. These angles can then be employed to govern the arm's motion and accurately position its end effector at the desired location.

$$\begin{aligned} \theta_1 &= (\arctan(y,x))\left(\frac{180}{\pi}\right) - \left(\frac{\arccos((L_1^2 + d^2 - L_2^2)}{(2)(d)(L_1)}\right)\left(\frac{180}{\pi}\right) \\ \theta_2 &= \frac{\arccos((L_1^2 + L_2^2 - d^2)}{(2)(L_1)(L_2)}\right)\left(\frac{180}{\pi}\right) \end{aligned} \quad (4)$$

Equation 5 demonstrates the derivation equation employed to ascertain the second joint's position. The x, y of the two-link robot arm using the first joint's angle (θ_1) and the first link's length (L_1). The origin (0, 0) already specifies the location of the first joint. This equation is crucial for directing the robot arm's motion and placing its end effector where it needs to be.

$$\begin{aligned} x &= (L_1)\left(\cos\left(\theta_1\right)\left(\frac{180}{\pi}\right)\right) \\ y &= (L_1)\left(\sin\left(\theta_1\right)\left(\frac{180}{\pi}\right)\right) \end{aligned} \quad (5)$$

The derived equations are the heart of the control system and were coded in MATLAB. The system can determine the necessary joint values for the inverse kinematics movement using the derived equations.

A. Program Flowchart for the Control System

In Fig. 5, the flowchart visually outlines a systematic procedure for resolving the inverse kinematics problem through MATLAB. It initiates with user input concerning the dimensions of the initial and secondary links, which collectively form the robot arm's components. Subsequently, users are guided to specify their desired X and Y coordinates for the end-effector, the location where the arm interacts with the objects. Once these user inputs are collected, the program proceeds to calculate the separation between the arm's base and the end-effector

by employing the Euclidean distance formula. Following this computation, it ascertains the angle of the initial link, utilizing inverse trigonometric functions. Subsequently, it computes the angle of the second link using the arccosine function and cosine law. These calculations are intricately linked to the initial link's angle and the lengths of both links. The program utilizes these derived link angles to precisely determine the positions of the joints and the coordinates of the end-effector. This level of control enables the robotic arm to achieve the desired placement and orientation within the workspace. Essentially, the flowchart offers a structured and systematic approach to effectively tackle the challenges associated with inverse kinematics.

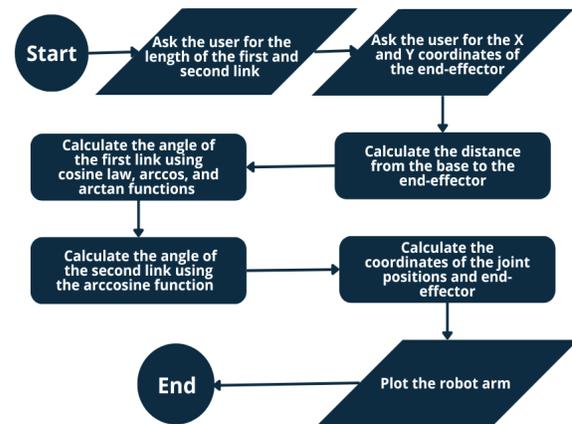


Fig. 5. Flowchart showing the control system program

The program basically asks the user for the link lengths and preferred end-effector position, calculates the angles of the two links using trigonometry and inverse trigonometric functions, and then draws the robot arm based on those angles and link lengths.

C. MATLAB Code

After deriving the equations for inverse kinematics, the derived equations were translated into MATLAB code and organized into a step-by-step program. MATLAB was a suitable choice for this study because it allows for efficient computation and has built-in functions for complex mathematical operations required for inverse kinematics. Additionally, its graphical user interface (GUI) makes it easier for the users to input data and visualize the output [22]. The study employs MATLAB code that engages users to input link lengths and end-effector coordinates. After gathering this data, the program calculates the base-to-end-effector distance using the Euclidean formula. Subsequently, it computes the angles for the initial and second links, employing the cosine law and inverse trigonometric functions. Utilizing trigonometry, the program defines the robot arm's joint positions and end-effector. It then visualizes

the robot arm with black-filled circles representing joints and the end-effector, accomplished through MATLAB's plot and scatter functions. This visualization effectively illustrates the robot arm's movement, offering precise control over its position and motion of the robot arm in a precise and efficient manner.

III. RESULTS AND DISCUSSION

A. Simulation of Control System

Upon running the program, it will prompt the user to input the lengths of the first and second links, as well as the desired coordinates. The ideal ratio for a human's upper arm to forearm is roughly 1 to 0.8837, according to a study for a human-like arm robot system [23]. Therefore, the values 1 and 0.8837 were inputted for the first and second links respectively as shown in Fig. 6. Meanwhile for the desired location, the selected coordinates (0.5,-1.5) were arbitrary and up to the user.

```

Command Window
Enter the length of the first link: 1
Enter the length of the second link: 0.8837
Enter the desired X coordinate of the end-effector: 0.5
fx Enter the desired Y coordinate of the end-effector: -1.5
    
```

Fig. 6. Inputting the X and Y coordinates to the system

Following the user's input of essential data, the MATLAB program will execute the necessary computations to establish the positions of both joints and the robotic arm's end-effector. The ensuing robot arm plot will visually present the link lengths, joint angles, and the end-effector's location, offering a graphical depiction of the arm's arrangement (Fig. 7). Lines shown in red and blue represent the robot arms. The length of those lines is specifically measured at 1.00 and 0.88 respectively. The angle of the first and second joints is also shown, which has the value of -102.27 and 114.01 degrees respectively.

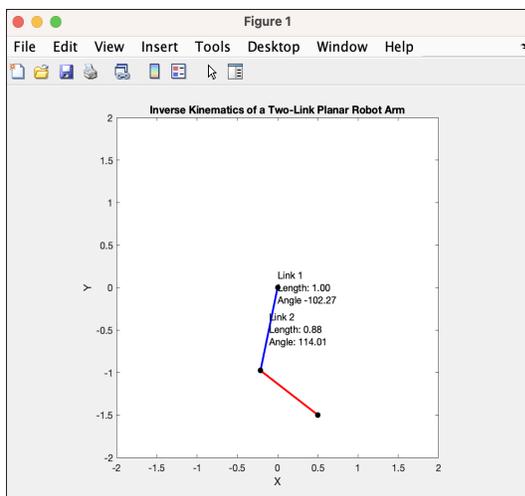


Fig. 7. Plot of the links in 2d cartesian plane

Fig. 8 shows more specific information when the joints in the plot are right-clicked. It is seen that the coordinates of the first joint are (0,0), second joint being at (-0.212445,-0.97713) and the third joint shows the end-effector coordinate which was specified in the user input being (0.5,-1.5). The sizes are pixel sizes for the diameter of the joints.

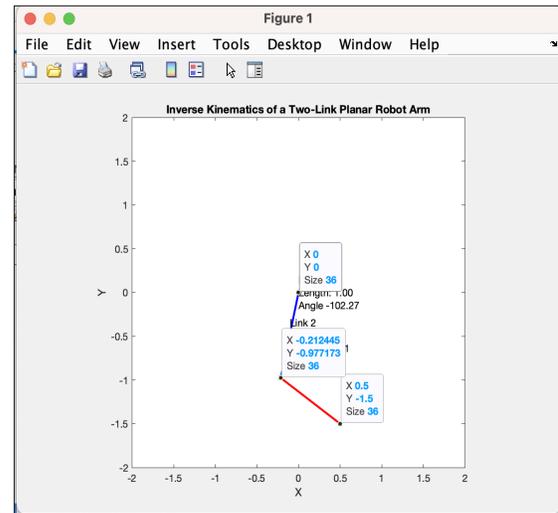


Fig. 8. Additional details provided in the plot

D. Application

This study focuses on the application of inverse kinematics in industrial assembly using robotic arms. In manufacturing, robotic arms are commonly used for precise and repeatable assembly tasks, such as picking up and placing components, tightening screws, or joining components together. To perform these tasks, the robotic arm must be accurately positioned [24]. Due to its critical role in precisely determining the position of the robot arm's end-effector at a preset location, the importance of the inverse kinematics problem is clear in the field of assembly robotics. This capability enables the robot to effectively grip and manipulate components [25]. Within this simulation, users can input the connection lengths and the desired end-effector position for the robot's intended task. The code then determines the joint angles needed to transmit the information to the robot controller. By precisely positioning the robotic arm using inverse kinematics, this control system is crucial for ensuring the dependability of the assembly process.

IV. CONCLUSION

The study successfully illustrated the use of inverse kinematics for manipulating a 2D two-linked robotic arm. The study developed the appropriate equations to solve the inverse kinematic problem using both geometric and trigonometric methods and created a control system using

MATLAB. The program was able to determine the required joint angles to reach a particular end-effector position using the user-provided link lengths. To enhance clarity, the resulting arm configuration was also visually presented. The study's findings underscore the significance of inverse kinematics within the realm of robotics, emphasizing its critical role in facilitating precise movements required across various industrial assembly contexts. In summary, this research represents a substantial advancement in the field of robotics and its practical utility in industrial assembly operations.

REFERENCES

- [1] G. Singh and V. Banga, "Robots and its types for industrial applications," *Materials Today: Proceedings*, vol. 60, pp. 1779-1786, 2022. <https://doi.org/10.1016/j.matpr.2021.12.426>
- [2] Universal UR10," Robots Done Right. [Online]. Available: <https://robotsoneright.com/Universal/Universal-UR10.html>.
- [3] K. Tokarz, "Geometric approach to inverse kinematics for arm manipulator," *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 3, no. 1, pp. 17-22, 2009.
- [4] M. Hägele, K. Nilsson, J.N. Pires, and R. Bischoff, "Industrial Robotics," in B. Siciliano and O. Khatib (eds.), *Springer Handbook of Robotics*, Springer Handbooks, Springer, Cham, 2016, pp. 1295-1322. https://doi.org/10.1007/978-3-319-32552-1_54
- [5] M. Khan, A. Nazir, M. Rafique, S. Nawaz, and S. Saeed, "A Review of Inverse Kinematics Techniques for Robotics Manipulators," *Algorithms*, vol. 15, no. 12, pp. 469-491, 2022.
- [6] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir, "Inverse Kinematics Techniques in Computer Graphics: A Survey," *Computer Graphics Forum*, vol. 37, pp. 35-58, 2018. <https://doi.org/10.1111/cgf.13310>
- [7] A. Toquica, L. Ortiz Martinez, R. Rodriguez, A. Chavarro Chavarro, and T. Cardozo, "Kinematic modelling of a robotic arm manipulator using MATLAB," *Journal of Engineering and Applied Sciences*, vol. 12, 2017.
- [8] K. Waldron and J. Schmiedeler, "Kinematics," in SpringerLink, 01-Jan-1970. [Online]. Available: https://link.springer.com/referenceworkentry/10.1007/978-3-540-30301-5_2.
- [9] K. Castelli and H. Giberti, "Additive Manufacturing as an Essential Element in the Teaching of Robotics," *Robotics*, vol. 8, no. 3, pp. 73, Aug. 2019. <https://doi.org/10.3390/robotics8030073>.
- [10] D. E. Koditschek, M. W. Spong, and R. C. Rosenberg, "Dynamics, Control and Planning for a Two-Link Planar Robotic Arm," in *Proceedings of the IEEE Conference on Robotics and Automation*, 1987, pp. 1294-1301.
- [11] C. Park and K. Park, "Design and kinematics analysis of dual arm robot manipulator for precision assembly," in *2008 6th IEEE International Conference on Industrial Informatics*, Daejeon, Korea (South), 2008, pp. 430-435. <https://doi.org/10.1109/INDIN.2008.4618138>.
- [12] K. Ghadge, S. More, P. Gaikwad, and S. Chillal, "Robotic ARM for pick and place application," *International Journal of Mechanical Engineering and Technology*, vol. 9, pp. 125-133, 2018.
- [13] Petrenko et al., "The method of forming a geometric solution of the inverse kinematics problem for chains with kinematic pairs of rotational type only," *IOP Conference Series: Materials Science and Engineering*, vol. 450, no. 4, p. 042015, 2018. <https://doi.org/10.1088/1757-899X/450/4/042015>.
- [14] "Geometric approach to inverse kinematics for arm manipulator - researchgate." [Online]. Available: https://www.researchgate.net/publication/228412878_Geometric_approach_to_inverse_kinematics_for_arm_manipulator.
- [15] Robot Arm Free Cartesian Space Analysis for Heuristic Path Planning Enhancement - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/The-Inverse-Kinematics-Solutions-Based-on-Trigonometry_fig1_331685972
- [16] Raheem, Firas, Sadiq, Ahmed, and F. Abbas, Noor. "Robot Arm Free Cartesian Space Analysis for Heuristic Path Planning Enhancement." *International Journal of Mechanical & Mechatronics Engineering*, vol. 19, no. 4, 2019, pp. 29-42.
- [17] S. G. Abdallah, S. S. Abdallah, and M. A. Zohdy, "Inverse Kinematics of Humanoid Robots: A Review," *Journal of Computing in Civil Engineering*, vol. 30, no. 5, pp. 04016037-1-04016037-16, 2016.
- [18] Petrescu, R. V., Aversa, R., Akash, B., Bucinell, R., Corchado, J., Apicella, A., & Petrescu, F. I. (2017). Inverse Kinematics at the Anthropomorphic Robots, by a Trigonometric Method. *American Journal of Engineering and Applied Sciences*, 10(2), 394-411. doi: 10.3844/ajeassp.2017.394.411
- [19] J. Chaichawanit and S. Saiyod, "Solving inverse kinematics problem of robot arm based on a-star algorithm," 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE), Khon Kaen, Thailand, 2016, pp. 1-6, doi: 10.1109/JCSSE.2016.7748846.
- [20] P. Srisuk, A. Sento and Y. Kitjaidure, "Inverse kinematics solution using neural networks from forward kinematics equations," 2017 9th International Conference on Knowledge and Smart Technology (KST), Chonburi, Thailand, 2017, pp. 61-65, doi: 10.1109/KST.2017.7886084.
- [21] S. Goto (Ed.), "Robot Arms," Croatia: IntechOpen, 2011, pp. 31.
- [22] E. Dean-Leon, S. Nair and A. Knoll, "User friendly Matlab-toolbox for symbolic robot dynamic modeling used for control design," 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO), Guangzhou, China, 2012, pp. 2181-2188, doi: 10.1109/ROBIO.2012.6491292.
- [23] Chang, Che-Ming & Sanches, Felipe & Gao, Geng & Johnson, Samantha & Liarokapis, Minas. (2022). An Adaptive, Affordable, Humanlike Arm Hand System for Deaf and DeafBlind Communication with the 10.1109/IROS47612.2022.998205 *American Sign Language*.
- [24] K. Dharmaraj, "Automated freeform assembly of threaded fasteners," Ph.D. dissertation, Loughborough Univ., Loughborough, UK, 2015.
- [25] Jha, P. (2017). Inverse kinematic analysis of robot manipulators. *International Journal of Mechanical and Production Engineering Research and Development*, 7(4), 637-644.