Position and orientation analysis of Jupiter robot arm for navigation stability

Omar Shalash[1,2], Adham Sakr[1], Yasser Salem[1], Ahmed Abdelhadi[1], HossamEldin Elsayed[1,2], Ahmed El-Shaer[1]
[1]College of Artificial Intelligence, Arab Academy for Science, Technology and Maritime Transport, Alamein, Egypt
[2]Research and Innovation Center, Arab Academy for Science, Technology and Maritime Transport, Alamein, Egypt

Article Info

Article history:

Received Jul 19, 2024
Revised Sep 7, 2024
Accepted Sep 17, 2024

Keywords:

Analytical solutions
Denavit–Hartenberg
Inverse kinematics
Jupiter robot
Kinematic analysis

ABSTRACT

Jupiter robot has made a great impact in the educational field with its support for autonomous navigation, visual perception, and many other features from its artificial intelligence platform's learning box. This study undertakes a kinematic model design of Jupiter's arm to aid the robot's motion stability. This process involved the determination of a homogeneous transformation matrix, followed by the determination of orientation, position, and Euler angles. Ultimately, the homogeneous transformation matrix was successfully derived, and the simplification of direct kinematic matrices was achieved. Consequently, the kinematic analysis for Jupiter's arm was established using the position Denavit–Hartenberg method, orientation, and Euler angles, proving to be valuable in the context of this research.

Corresponding Author:

Omar Shalash
College of Artificial Intelligence, Arab Academy for Science, Technology and Maritime Transport
Alamein, Egypt
Email: omar.o.shalash@aast.edu

1. INTRODUCTION

Robotics, the interdisciplinary field at the intersection of mechanical engineering, computer science, artificial intelligence, and electronics, has rapidly transformed the way we live and work. In the quest for automating tasks and improving abilities, robotics has become a technology in the 21st century. Originating from the realm of science fiction and finding its way into industries such, as manufacturing and healthcare artificial intelligence has become widely prevalent [1]–[13]. The study of robotics has become a dynamic and critical area of research, offering insights into designs, developments, and deployments of intelligent machines.

The field of robotics has made strides thanks to advancements in modeling techniques. It is crucial to comprehend the positioning and orientations of robots such, as the Jupiter robot in order to effectively control and interact with them in environments. Two key components, forward kinematics and inverse kinematics play a role in this pursuit. Significant research has been devoted to finding solutions for determining the kinematics of humanoid robots. In the past researchers primarily utilized Denavit–Hartenberg (DH) parameters, which offered a standardized approach, for representing the transformations, between neighboring links [14]. In this approach, we assign parameters to each joint allowing us to calculate the positions and orientations of the end effector [15]–[21]. Another used technique in robotics is the homogeneous transformation matrices. They provide a framework, for expressing the position and orientation transformations of robotic systems [22].

These matrices make computations more efficient, by containing all the transformation information within a matrix [23]–[28]. The formula known as the product of exponentials which was introduced by Murray *et al*. [29] is a representation that is derived from Lie algebra. It provides a compact and elegant way to express the end-effector pose as a function of joint variables, allowing for efficient and accurate kinematic computations [30]–[33]. Finding equations, for particular robot configurations is especially valuable when dealing with robots that have clearly defined and relatively uncomplicated shapes [34]. They offer explicit equations for determining end-effector poses [35]–[37]. The flexible method of inverse kinematics and trigonometric methods utilizes inverse trigonometric functions to calculate the angles of joints. In particular, this technique is very straightforward, as the corresponding joints' orientations can easily determine the angles needed to achieve certain target locations of an end-effector [37]–[39]. A representation based on the idea of Geometric interpretation of kinematics clarifies to us how the motions of joints are related with the motion of the end effector. It is a way of conceptualizing robot motion and understanding how everything fits together [40], [41]. Currently, these solutions have been expanded into dealing with kinematically complex structures as well as redundant systems. With this development, analysis solutions have taken on a new dimension; more flexible and complex robotics systems.

Determining the angles of joints to find the suitable position of an end effector, known as kinematics, has been a crucial field of study. Researchers have utilized methods, like techniques and optimization algorithms, to address this complex problem. These methods have shown their effectiveness, in real-time control and planning movements [42]. Iterative Jacobian-based methods are a prevalent approach in solving inverse kinematics [43]. They use the matrix, which helps determine how changes, in velocities, affect the velocities of the end effector [44]. Understanding the relationship between angles and the resulting changes in the position and orientation of the end effector is essential. This matrix serves as a valuable tool, for that purpose [45]. By making modifications these techniques gradually move closer, to the intended position skillfully maneuvering through the range of solutions and steadily moving towards the objective [45], [46]. They perform well in changing environments and situations that demand immediate adaptations. However, their efficiency can be affected by factors such, as the iteration method used the accuracy of calculating the Jacobian, and the occurrence of singularities [44]. Reliable convergence depends on careful parameter tuning. However, the choice between the Iterative Jacobian-based methods and other techniques depends on various considerations which include robot structures, computational resources, and operating environment. In making the right choice about various methods regarding inverse kinematics, it is important to understand each method's strong points as well as its weak points [47]. The other important approach is that the CCD algorithm has been a significant approach for inverse kinematics. Through an iterative adjustment of each separate joint's angle, CCD is developed specifically for the articulated structures possessed by several degrees of freedom robots [48]. The operation begins on its base and improves successive angles on the kinetic chain till it gets to the position it desires for the end effector. This technique takes advantage of the intrinsic organization of many robotic systems and is therefore highly useful for solving complicated articulations like those characterizing humanoid robots as well as multi-limbed robotic arms [49]. Nevertheless, it's important to acknowledge that while CCD excels in many scenarios, it may face challenges converging to a solution in cases involving singularities or highly constrained environments [50]. Additionally, the order of joint adjustments can impact the outcome, warranting thoughtful consideration in its application. Despite these considerations, CCD remains a valuable and pragmatic tool for addressing complex inverse kinematics challenges in robotics as reported in Table 1 [51]. Gradient-based optimization approaches inverse kinematics as an optimization challenge, employing techniques like gradient descent and Newton's method to minimize an objective function, hence converging towards optimal joint angles [52]. The process whereby joint angles are adjusted incrementally moving the system along the path of decreasing the value of the objective function until reaching a local minimum that coincides with the target end-effector positioning is referred to as gradient decent. It has been observed that some complex and non-linear objective functions, could converge faster than first-order derivatives using Newton's methods [53]. It may be strong but you need to consider the choice of an optimization algorithm, a convergence criterion, and the objective function definition. Since singularities and constrained environments require special handling [49]. Knowledge of such subtleties is crucial in the effective utilization of addressing the inverse kinematics problem in robotics. Damped least squares (DLS) is an approach with both analytical and numerical ingredients applied within inverse kinematics [54]. It optimizes its weighted least square to solve the problem of minimizing this difference between the end-effector poses. Therefore, one should include this damping factor that will provide enough stability that make the solution correct under such instabilities or arising inconsistencies [50]. The advantage of Damped Least Squares is inaccurate end-effector poses for the systems that use it. It works out to be a good choice when it comes to robotic applications such as manipulator arms and humanoid robots [55]. Researchers have delved into diverse optimization-based techniques, such as genetic algorithms, simulated annealing, and particle swarm optimization (PSO), as alternative avenues for tackling inverse kinematics challenges [54]. The use of these methods brings about another paradigm because they apply to nature-based search strategies. Genetic algorithms use principles of natural selection and genetics for exploring the solution space and hence are very effective in solving difficult, multidimensional optimization problems [56]. These methods offer diverse strategies for tackling the complex problem of inverse kinematics, each with its own strengths and considerations. Their application depends on factors such as the robot's stability kinematic structure, computational resources, and specific task requirements.

This research contributes by developing a kinematic model for the Jupiter robot, enhancing its arm functionality in several key ways. First, the model allows the robot's arm to extend fully, providing an advantage in competitive settings by enabling faster object-grabbing. Second, the kinematic model improves the stability of the robot's movements, ensuring smooth and controlled operation. These features allow the Jupiter robot to operate at maximum speed without risking tipping over, while swiftly reaching and grasping target objects.

2. JUPITER BACKGROUND

The Jupiter robot as seen in Figure 1, created by Lattel Robotics, is a versatile home assistant weighing 10.3 kg and boasting external dimensions of 352×352×920 mm, with a ground clearance of 15 mm. Lattel Robotics, a company dedicated to promoting AI-focused robotics education, is headquartered in Singapore and Malaysia. The heart of the Jupiter robot is its onboard computer, equipped with an Intel Core i5-10210U processor running at 1.6 GHz, 8 GB of RAM, and a 120 GB SSD for internal storage. It also features a Wi-Fi remote controller with a swift 300 Mbps transmission rate, enabling seamless interaction. Speech interaction capabilities are incorporated, offering a frequency response between 50 Hz and 16 kHz for clear communication. For mobility, the robot utilizes the mobile base Kobuki unit, allowing for a maximum payload of 5 kg, a top speed of 0.5 m/s, and a rapid rotation speed of 160 degrees per second. Powering this remarkable machine are standard 4400mAh Li-Ion batteries, with the option for extended battery life using an additional 4400mAh Li-Ion unit. The robot is equipped with an array of sensors, including 25,700 CPR encoders, a rate gyroscope with a factory calibration of 110 deg/s, and auxiliary sensors (3x forward bump, 3x Cliff, 2x wheel drop). It also features a 3D stereo camera with a resolution of 640px x 480px, recording at 30 fps, and a Slamtec A2m8-R4 RP LiDAR system. With all these features, the Jupiter robot is equipped with various support for autonomous navigation, visual perception, speech interaction, mobile manipulation and AI, machine learning, and cloud computing, hence proven to be essential in the educational field, which enabled the robot to be the core element in RoboCup@Home Education contest [57].

Figure 1. Jupiter robot

Jupiter's arm system as seen in Figure 1 is comprised of 5 servos, including the end effector servo, along with 4 links. Its external dimensions (L×W×H) are 32×50×40 mm, and it offers an impressive accuracy of 0.29 degrees [57]. This robotic wonder seamlessly integrates a blend of features, including robot intelligence, natural interaction, computer vision, mobile platform capabilities, and object manipulation. The robot's ARM is a 4DOF (four degrees of freedom) system. It comprises four joints, each with its unique rotation: Joint 1: Yaw rotation, Joint 2: Pitch rotation, Joint 3: Pitch rotation, Joint 4: Pitch rotation, These four joints collectively endow the robot's effector with four degrees of freedom, enabling motion in the x, y, and z axes, as well as pitch and yaw rotation.

3. METHODOLOGY

The transformation matrix for joint 'i' with respect to an adjacent joint 'i' in three-dimensional space is in (1).

$$T_i = \begin{bmatrix} cos(\theta_i) & -sin(\theta_i) & 0 & \alpha_{i-1} \\ cos(\alpha_{i-1})sin(\theta_i) & cos(\alpha_{i-1})cos(\theta_i) & -sin(\alpha_{i-1}) & -d\ sin(\alpha_{i-1}) \\ sin(\alpha_{i-1})sin(\theta_i) & sin(\alpha_{i-1})cos(\theta_i) & cos(\alpha_{i-1}) & d\ cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The transformation from joint 'i' to joint 'j', where 'i' ranges from 1 to 'N', and 'j' ranges from 'i+1' to 'N', is given by (2).

$$T_i^j = T_i^{i+1} \cdot T_{i+1}^{i+2} \cdot \ldots \cdot T_{j-1}^j \quad (2)$$

For the case of a 5-joint robotic arm as observed in Figure 2, the transformation matrices are filled for each joint (2), and the multiplication is processed sequentially to obtain the transformation from the base to the end-effector frame (3).

$$^0_4T = \begin{bmatrix} ^4_0R_{11} & ^4_0R_{12} & ^4_0R_{13} & ^4_0X \\ ^4_0R_{21} & ^4_0R_{22} & ^4_0R_{23} & ^4_0Y \\ ^4_0R_{31} & ^4_0R_{32} & ^4_0R_{33} & ^4_0Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

In order to prepare the DH model for Jupiter's arm, see Figure 2 for illustration, the model's parameters (the link length (a_i), link offset (d_i), rotation angle (θ_i), and link twist (α_i)) are calculated and presented in Table 1.

Solving the inverse kinematic problem depends on each robot's design. Iterative numerical methods may eventually lead to a solution, but they are prone to encountering singularities, resulting in potential failures even when a valid solution is possible. Furthermore, their performance is generally sub-optimal. On the other side, analytical solutions are swift and highly precise, yet uncovering them necessitates a substantial amount of effort.

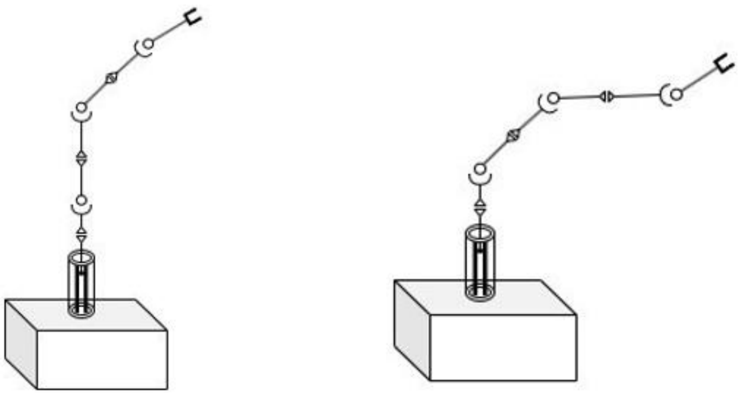


Figure 2. Five-joint robotic arm

Table 1. DH-model parameters

i	αi−1	ai−1	d_i	θ_i
1	0	0	a_1	θ_1
2	−90°	0	0	θ_2
3	0	L_2	0	θ_3
4	0	L_3	0	θ_4
5	0	L_4	0	0

3.1. Geometric solution

The geometric solution for inverse kinematics was chosen according to two factors. First, it is an Efficient solution for simple structures with closed-form solutions, thus greatly reducing many calculations and increasing efficiency. Second, the geometric solution is adaptable, allowing more modifications to the arm without requiring any changes to the inverse kinematics model, as long as the parameters and coefficients stay the same. In comparison to other methods, such as the algebraic track, the geometric solution is valued for its efficiency in less complex structures. While the algebraic method boasts greater versatility and can handle a broader spectrum of robotic systems, it may grapple with challenges, particularly in scenarios that involve highly nonlinear systems [58]. Moreover, when compared to numerical solutions such as neuro-fuzzy models, aside from being known for their adaptability, they introduce complexities related to computational costs, interpretability, and data requirements. The decision to choose one of these models depends on the needs of the robotic system. Necessitating a careful balance between adaptability, computational requirements, and interpretability [59] and comparison with new techniques, such as the use of PSO. combined with POSIX threads for inverse kinematics. While this innovative strategy holds promise by facilitating parallelism and potential speedup, it presents challenges related to thread synchronization, load balancing, and system complexity. Rigorous testing and validation under various conditions is essential to accurately evaluate its performance [59]. After carefully considering these factors, the geometric solution approach emerged as the preferred choice in this study. Its efficiency in handling simple structures and closed form solutions, and the adaptability of the geometric solution allow for dynamic changes in link lengths or the end-effector while maintaining the same configuration. This feature is particularly beneficial in scenarios where alterations to the robot's structure are required. It is the most fit solution for Jupiter's robotic system and its unique requirements. The top view of the robotic arm, the value of r_1, using the Pythagorean theorem in (5).

Given that y_o is the position of the end effector along the y-axis and x_o is the position of the end effector along the x-axis.

$$\theta_4 = \tan^{-1}\left(\frac{y_o}{x_o}\right) \tag{4}$$

θ_1 represents the rotational joint of the arm (base joint). Given that, z_0 is the distance the end effector along the z-axis, and a_0 is the base link length. The side view of the robot's arm:

$$r_1 = \sqrt{x_0^2 + y_o^2}; \quad r_2 = z_0 - a_0; \quad r_3 = \sqrt{r_1^2 + r_2^2} \tag{5}$$

By applying the cosine rule (6) on the r_3 link

$$(r_3^2 = l_1^2 + l_2^2 - 2 \cdot l_1 \cdot l_2 \cdot \cos\alpha) \tag{6}$$

The angle α can be denoted by (7)/

$$\cos\alpha = \frac{l_1^2 + l_2^2 - r_3^2}{2 \cdot l_1 \cdot l_2} \tag{7}$$

θ_2 and α are supplementary and can be obtained by (8).

$$\theta_2 = (\pi - \alpha); \quad \cos\theta_2 = -\cos\alpha; \quad \theta_2 = \cos^{-1}\left(\frac{r_1^2 + r_2^2 - (l_1^2 + l_2^2)}{2 \cdot l_1 \cdot l_2}\right) \tag{8}$$

Then, applying (SOHCAHTOA) trigonometric functions (9) to deduce the angles equation.

$$\theta_1 = \gamma - \beta; \tan\beta = \frac{l_2 \sin\theta_2}{l_1 + l_2\cos\theta_2}; \tan\gamma = \frac{r_2}{r_1}; \beta = \tan^{-1}\left(\frac{l_2 \sin\theta_2}{l_1 + l_2\cos\theta_2}\right); \gamma = \tan^{-1}\left(\frac{r_2}{r_1}\right) \quad (9)$$

This represents the result of substituting previous equations to result in (10).

$$\theta_1 = \tan^{-1}\left(\frac{r_2}{r_1}\right) - \tan^{-1}\left(\frac{l_2 \sin\theta_2}{l_1 + l_2\cos\theta_2}\right) \quad (10)$$

Finally, we calculate the angles of movement in the equations below where θ_{base} represents the base rotation angle, and φ is the Euler angle.

$$\theta_{\text{benc}} = \tan^{-1}\left(\frac{Y}{X}\right); \quad \varphi = \theta_1 + \theta_2 + \theta_3; \quad \theta_3 = \varphi - (\theta_1 + \theta_2) \quad (11)$$

4. RESULTS

A set of tools and libraries have been used to implement and test the robot's arm in a simulation. First, a visualization of the robot's robotic arm using the "Visualize Kinematics" library on Python as seen in

Figure. In this library in Python, the kinematics model is implemented and tested. The robot's end-effector is given a location and orientation, these coordinates are compared with the origin coordinate frame to prove accuracy.

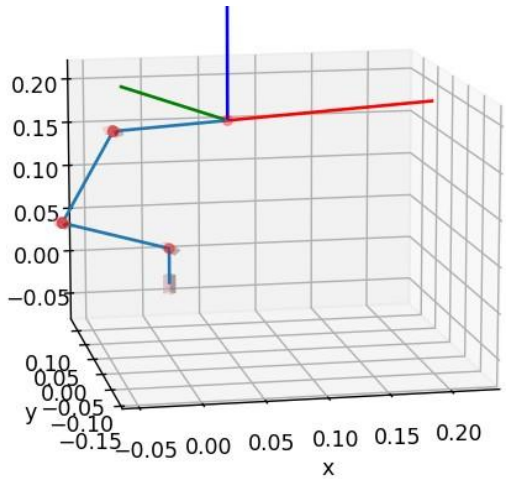


Figure 3. Kinematics visualization

In the proposed study, we employed a versatile robot equipped with an advanced arm and a multi-Astra depth camera. The experiments were conducted in diverse environments, such as kitchen and living room settings, to assess the robot's ability to autonomously navigate, detect different target objects with different dimensions and grasping points and center of mass, and perform precise arm manipulations, for the arm's position see Figure 3. The robot's end effector was designed for adaptability, allowing for the grasping of various objects and their delivery to the user.

Notably, the robot's end effector is a modular component that can be changed or modified to perform various tasks. The adaptation also involves tweaking of the link lengths which is possible but does not pose any issue for inverse kinematics. With respect to this feature, the robot becomes more flexible, performing diverse operations in various settings with high accuracy and performance.

4.1. Final task

Sensors coupled with algorithms enabled the robot to move precisely to the predefined waypoint. At this point, the robot started 3D localization of the ball on the spot by multiple astra-depth cameras with respective x, y, and z coordinates. Subsequently, the robot seamlessly transitioned to arm manipulation, employing inverse kinematics to calculate joint angles for an accurate grasp and relocation of the ball.

4.2. Validation

Validating the accuracy and reliability of the proposed inverse kinematics model is crucial to ensure its applicability and effectiveness in practical scenarios. To initiate the validation, an experimental setup was designed by manually positioning the end effector to a predefined point in space and measuring the join angles θ_1,θ_2 and θ_4, then cross-validating the measured values with the proposed model.

The end effector was placed at point (10,0,14) in space which corresponds to x, y, z coordinates respectively, then the joint angles were measured $\theta_1 = 0°$, $\theta_2 = 90°$ and $\theta_4 = 0°$ and given the link lengths of the robot arm l_1 = 10 cm, l_2 = 10 cm and a_0 = 4 cm. we substitute in (4), (8), (9) to obtain that the result of the joint angles $\theta_1 = 0°$, $\theta_2 = 90°$ and $\theta_4 = 0°$, see Figure 1.

5. CONCLUSION

In order to develop a robust kinematic model for Jupiter's robotic arm, we presented an exact forward and inverse kinematics solution, encompassing a step-by-step derivation of a homogeneous transformation matrix, determination of orientation, position, and Euler angles, culminated in a streamlined direct kinematic solution, followed by the simplification of direct kinematic matrices. This contribution not only improves the robot's motion control but also aligns with its educational objectives by facilitating autonomous navigation and enhancing learning experiences. While this study presents a solid foundation, future work could apply the kinematic model to real-life scenarios and adaptive learning mechanisms to further advance Jupiter's educational contributions.

REFERENCES

[1] M. S. Kaiser, S. Al Mamun, M. Mahmud, and M. H. Tania, "Healthcare robots to combat COVID-19," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 60, Springer Singapore, 2021, pp. 83–97. doi: 10.1007/978-981-15-9682-7_10.

[2] J. Holland *et al.*, "Service robots in the healthcare sector," *Robotics*, vol. 10, no. 1, Mar. 2021, doi: 10.3390/robotics10010047.

[3] I. M. Gaber, O. Shalash, and M. S. Hamad, "Optimized inter-turn short circuit fault diagnosis for induction motors using neural networks with LeLeRU," Feb. 2023. doi: 10.1109/CPERE56564.2023.10119618.

[4] A. Khaled, O. Shalash, and O. Ismaeil, "Multiple objects detection and localization using data fusion," Dec. 2023. doi: 10.1109/ICARCE59252.2024.10492609.

[5] O. Shalash, "Design and development of autonomous robotic machine for knee arthroplasty," Ph.D. dissertation, Department of Biomedical Engineering, University of Strathclyde, 2018.

[6] S. K. Kavoussi, K. M. Kavoussi, and D. I. Lebovic, "Robotic-assisted tubal anastomosis with one-stitch technique," *Journal of Robotic Surgery*, vol. 8, no. 2, pp. 133–136, Dec. 2014, doi: 10.1007/s11701-013-0442-z.

[7] O. Shalash and P. Rowe, "Computer-assisted robotic system for autonomous unicompartmental knee arthroplasty," *Alexandria Engineering Journal*, vol. 70, pp. 441–451, May 2023, doi: 10.1016/j.aej.2023.03.005.

[8] E. Khatab, A. Onsy, M. Varley, and A. Abouelfarag, "A lightweight network for real-time rain streaks and rain accumulation removal from single images captured by AVs," *Applied Sciences (Switzerland)*, vol. 13, no. 1, Dec. 2023, doi: 10.3390/app13010219.

[9] M. Roche, "The MAKO robotic-arm knee arthroplasty system," *Archives of Orthopaedic and Trauma Surgery*, vol. 141, no. 12, pp. 2043–2047, Nov. 2021, doi: 10.1007/s00402-021-04208-0.

[10] B. Yan, P. Fan, X. Lei, Z. Liu, and F. Yang, "A real-time apple targets detection method for picking robot based on improved YOLOv5," *Remote Sensing*, vol. 13, no. 9, Apr. 2021, doi: 10.3390/rs13091619.

[11] A. Abouelfarag, M. A. Elshenawy, and E. A. Khattab, "Accelerating sobel edge detection using compressor cells over FPGAs," in *Smart Technology Applications in Business Environments*, IGI Global, 2017, pp. 1–21. doi: 10.4018/978-1-5225-2492-2.ch001.

[12] H. Said *et al.*, "Forearm intravenous detection and localization for autonomous vein injection using contrast-limited adaptive histogram equalization algorithm," *Applied Sciences (Switzerland)*, vol. 14, no. 16, Aug. 2024, doi: 10.3390/app14167115.

[13] M. Elkholy, O. Shalash, M. S. Hamad, and M. S. Saraya, "Empowering the grid: a comprehensive review of artificial intelligence techniques in smart grids," in *2024 International Telecommunications Conference, ITC-Egypt 2024*, Jul. 2024, vol. 15, pp. 513–518. doi: 10.1109/ITC-Egypt61547.2024.10620543.

[14] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *Journal of Applied Mechanics*, vol. 22, no. 2, pp. 215–221, Jun. 1955, doi: 10.1115/1.4011045.

[15] C. Klug, D. Schmalstieg, T. Gloor, and C. Arth, "A complete workflow for automatic forward kinematics model extraction of robotic total stations using the Denavit-Hartenberg convention," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 95, no. 2, pp. 311–329, Sep. 2019, doi: 10.1007/s10846-018-0931-4.

[16] M. Flanders and R. C. Kavanagh, "Build-a-robot: Using virtual reality to visualize the Denavit-Hartenberg parameters," *Computer Applications in Engineering Education*, vol. 23, no. 6, pp. 846–853, Apr. 2015, doi: 10.1002/cae.21656.

[17] A. C. Reddy, "Difference between Denavit - Hartenberg (D-H) classical and modified conventions for forward kinematics of robots with case study," in *International Conference on Advanced Materials and manufacturing Technologies (AMMT)*, 2014, pp. 267–286.

[18] P. I. Corke, "A simple and systematic approach to assigning Denavit-Hartenberg parameters," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 590–594, Jun. 2007, doi: 10.1109/TRO.2007.896765.

[19] S. Shim, S. Lee, S. Joo, and J. Seo, "Denavit-Hartenberg notation-based kinematic constraint equations for forward kinematics of the 3-6 Stewart platform," *Journal of Mechanisms and Robotics*, vol. 14, no. 5, Mar. 2022, doi: 10.1115/1.4053822.

[20] R. Issa *et al.*, "A data-driven digital twin of electric vehicle Li-Ion battery state-of-charge estimation enabled by driving behavior application programming interfaces," *Batteries*, vol. 9, no. 10, Oct. 2023, doi: 10.3390/batteries9100521.

[21] S. Karmakar and C. J. Turner, "Forward kinematics solution for a general Stewart platform through iteration-based simulation," *International Journal of Advanced Manufacturing Technology*, vol. 126, no. 1–2, pp. 813–825, Feb. 2023, doi: 10.1007/s00170-

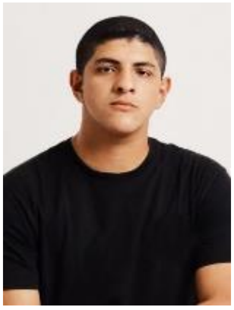
023-11130-9.
[22] S. Kucuk and Z. Bingul, "Robot kinematics: Forward and inverse kinematics," in *Industrial Robotics: Theory, Modelling and Control*, Pro Literatur Verlag, Germany / ARS, Austria, 2006. doi: 10.5772/5015.
[23] O. Martínez and R. Campa, "Comparing methods using homogeneous transformation matrices for kinematics modeling of robot manipulators," in *Mechanisms and Machine Science*, vol. 94, Springer International Publishing, 2021, pp. 110–118. doi: 10.1007/978-3-030-60372-4_13.
[24] T. P. Singh, P. Suresh, and S. Chandan, "Forward and inverse kinematic analysis of robotic manipulators," *International Research Journal of Engineering and Technology(IRJET)*, vol. 4, no. 2, pp. 1459–1469, 2017.
[25] J. R. Ortiz-Zacarias, Y. S. Valenzuela-Lino, J. Asto-Evangelista, and D. Huamanchahua, "Kinematic position and orientation analysis of a 4 DoF orthosis for knee and ankle rehabilitation," in *2021 6th International Conference on Robotics and Automation Engineering, ICRAE 2021*, Nov. 2021, vol. 18, pp. 141–146. doi: 10.1109/ICRAE53653.2021.9657817.
[26] G. Dudek and M. Jenkin, *Computational principles of mobile robotics*. Cambridge University Press, 2010. doi: 10.1017/cbo9780511780929.
[27] P. Curtis, C. S. Yang, and P. Payeur, "An integrated robotic multi-modal range sensing system," in *Conference Record - IEEE Instrumentation and Measurement Technology Conference*, 2005, vol. 3, pp. 1991–1996. doi: 10.1109/imtc.2005.1604520.
[28] M. Yasser, O. Shalash, and O. Ismail, "Optimized decentralized swarm communication algorithms for efficient task allocation and power consumption in swarm robotics," *Robotics*, vol. 13, no. 5, Apr. 2024, doi: 10.3390/robotics13050066.
[29] F. J. Murray, "Perturbation theory and Lie algebras," *Journal of Mathematical Physics*, vol. 3, no. 3, pp. 451–468, May 1962, doi: 10.1063/1.1724245.
[30] F. C. Park, "Computational aspects of the product-of-exponentials formula for robot kinematics," *IEEE Transactions on Automatic Control*, vol. 39, no. 3, pp. 643–647, Mar. 1994, doi: 10.1109/9.280779.
[31] H. E. Elsayed, N. S. Tawfik, O. Shalash, and O. Ismail, "Enhancing human emotion classification in human-robot interaction," in *2024 International Conference on Machine Intelligence and Smart Innovation, ICMISI 2024 - Proceedings*, May 2024, pp. 19–24. doi: 10.1109/ICMISI61517.2024.10580152.
[32] I. Chen and G. Yang, "Kinematic calibration of modular reconfigurable robots using product-of-exponentials formula," *Journal of Robotic Systems*, vol. 14, no. 11, pp. 807–821, Nov. 1997, doi: 10.1002/(sici)1097-4563(199711)14:11<807::aid-rob4>3.3.co;2-3.
[33] C. He, S. Wang, Y. Xing, and X. Wang, "Kinematics analysis of the coupled tendon-driven robot based on the product-of-exponentials formula," *Mechanism and Machine Theory*, vol. 60, pp. 90–111, Feb. 2013, doi: 10.1016/j.mechmachtheory.2012.10.002.
[34] J. Yang and Z. Jason Geng, "Closed form forward kinematics solution to a class of hexapod robots," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 3, pp. 503–508, Jun. 1998, doi: 10.1109/70.678460.
[35] M. A. Ali, H. A. Park, and C. S. G. Lee, "Closed-form inverse kinematic joint solution for humanoid robots," in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, Oct. 2010, pp. 704–709. doi: 10.1109/IROS.2010.5649842.
[36] Y. Liu, M. Kong, N. Wan, and P. Ben-Tzvi, "A geometric approach to obtain the closed-form forward kinematics of H4 parallel robot," *Journal of Mechanisms and Robotics*, vol. 10, no. 5, Jul. 2018, doi: 10.1115/1.4040703.
[37] S. Tejomurtula and S. Kak, "Inverse kinematics in robotics using neural networks," *Information sciences*, vol. 116, no. 2, pp. 147–164, Jan. 1999, doi: 10.1016/S0020-0255(98)10098-1.
[38] J. A. Abdor-Sierra, E. A. Merchán-Cruz, and R. G. Rodríguez-Cañizo, "A comparative analysis of metaheuristic algorithms for solving the inverse kinematics of robot manipulators," *Results in Engineering*, vol. 16, Dec. 2022, doi: 10.1016/j.rineng.2022.100597.
[39] A.-V. Duka, "Neural network based inverse kinematics solution for trajectory tracking of a robotic arm," *Procedia Technology*, vol. 12, pp. 20–27, 2014, doi: 10.1016/j.protcy.2013.12.451.
[40] H. Hadfield, L. Wei, and J. Lasenby, "The forward and inverse kinematics of a delta robot," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12221 LNCS, Springer International Publishing, 2020, pp. 447–458. doi: 10.1007/978-3-030-61864-3_38.
[41] D. R. Parhi, B. B. V. L. Deepak, and A. Amrit, "Forward and inverse kinematic models for an articulated robotic manipulator," *International Journal of Artificial Intelligence and Computational Research*, vol. 4, pp. 103–109, 2013.
[42] A. El-Sherbiny, M. A. Elhosseini, and A. Y. Haikal, "A comparative study of soft computing methods to solve inverse kinematics problem," *Ain Shams Engineering Journal*, vol. 9, no. 4, pp. 2535–2548, Dec. 2018, doi: 10.1016/j.asej.2017.08.001.
[43] M. Meredith and S. Maddock, "Real-time inverse kinematics: The return of the Jacobian," *Tech. Rep. CS-04-06, Department of Computer Science*, pp. 1–15, 2004.
[44] A. Aristidou and J. Lasenby, "Inverse Kinematics: a review of existing techniques and introduction of a new fast iterative solver," *The Lancet*, vol. 158, no. 4071, 2009.
[45] R. C. O. Jesus, L. Molina, E. A. N. Carvalho, and E. O. Freire, "Singularity-free inverse kinematics with joint prioritization for manipulators," *Journal of Control, Automation and Electrical Systems*, vol. 33, no. 3, pp. 1022–1031, Jan. 2022, doi: 10.1007/s40313-021-00860-4.
[46] A. Volinski, Y. Zaidel, A. Shalumov, T. DeWolf, L. Supic, and E. Ezra Tsur, "Data-driven artificial and spiking neural networks for inverse kinematics in neurorobotics," *Patterns*, vol. 3, no. 1, Jan. 2022, doi: 10.1016/j.patter.2021.100391.
[47] B. Kenwright, "Inverse kinematics – cyclic coordinate descent (CCD)," *Journal of Graphics Tools*, vol. 16, no. 4, pp. 177–217, Oct. 2012, doi: 10.1080/2165347x.2013.823362.
[48] G. Tevatia and S. Schaal, "Inverse kinematics for humanoid robots," in *Proceedings-IEEE International Conference on Robotics and Automation*, 2000, vol. 1, pp. 294–299. doi: 10.1109/ROBOT.2000.844073.
[49] S. Ojha, K. Leodler, and T. H. Wu, "Singularity-free inverse kinematics by cyclic coordinate descent of a 6 DOF robotic manipulator," in *2023 20th International Conference on Ubiquitous Robots, UR 2023*, Jun. 2023, pp. 735–740. doi: 10.1109/UR57808.2023.10202406.
[50] H. Wu, J. Yu, J. Pan, G. Li, and X. Pei, "FBCCD: a forward and backward cyclic iterative solver for the inverse kinematics of continuum robot," in *Lecture Notes in Mechanical Engineering*, Springer Nature Singapore, 2023, pp. 329–345. doi: 10.1007/978-981-19-9398-5_18.
[51] J. Te Lin, C. Girerd, J. Yan, J. T. Hwang, and T. K. Morimoto, "A generalized framework for concentric tube robot design using gradient-based optimization," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3774–3791, Dec. 2022, doi: 10.1109/TRO.2022.3180627.
[52] S. Wild, T. Zeng, A. Mohammad, J. Billingham, D. Axinte, and X. Dong, "Efficient and scalable inverse kinematics for continuum robots," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 375–381, Jan. 2024, doi: 10.1109/LRA.2023.3331291.

[53] D. Wu, W. Zhang, M. Qin, and B. Xie, "Interval search genetic algorithm based on trajectory to solve inverse kinematics of redundant manipulators and its application," in *Proceedings - IEEE International Conference on Robotics and Automation*, May 2020, vol. 10, pp. 7088–7094. doi: 10.1109/ICRA40945.2020.9196890.
[54] X. Wang, X. Liu, L. Chen, and H. Hu, "Deep-learning damped least squares method for inverse kinematics of redundant robots," *Measurement: Journal of the International Measurement Confederation*, vol. 171, Feb. 2021, doi: 10.1016/j.measurement.2020.108821.
[55] M. Ahmad, N. Kumar, and R. Kumari, "A hybrid genetic algorithm approach to solve inverse kinematics of a mechanical manipulator," *International Journal of Scientific and Technology Research*, vol. 8, no. 9, pp. 1772–1782, 2019.
[56] C. T. Lee and J. Y. Chang, "A workspace-analysis-based genetic algorithm for solving inverse kinematics of a multi-fingered anthropomorphic hand," *Applied Sciences (Switzerland)*, vol. 11, no. 6, Mar. 2021, doi: 10.3390/app11062668.
[57] "Jupiter2," *Lattel Robotics - Robots*. https://lattelrobotics.com/home/jupiter-robot (accessed Aug. 02, 2024).
[58] E. Lazarevska, "A neuro-fuzzy model of the inverse kinematics of a 4 DOF robotic arm," in *Proceedings - 2012 14th International Conference on Modelling and Simulation, UKSim 2012*, Mar. 2012, pp. 306–311. doi: 10.1109/UKSim.2012.51.
[59] H. Danaci, L. A. Nguyen, T. L. Harman, and M. Pagan, "Inverse kinematics for serial robot manipulators by particle swarm optimization and POSIX threads implementation," *Applied Sciences (Switzerland)*, vol. 13, no. 7, p. 4515, Apr. 2023, doi: 10.3390/app13074515.

BIOGRAPHIES OF AUTHORS

Omar Shalash is an assistant professor at the Arab Academy for Science, Technology, and Maritime Transport at the College of Artificial Intelligence in Egypt and specializes in robotics. He holds a Ph.D. in Biomedical Engineering from the University of Strathclyde, where he developed an autonomous robotic machine for knee arthroplasty. His research, which includes publications on swarm robotics and electric vehicle battery management, demonstrates his expertise in both theoretical and applied aspects of computer engineering, robotics, and AI. Dr. Shalash is committed to advancing artificial intelligence and robotics through both his teaching and numerous supervised research projects. He can be contacted at omar.o.shalash@aast.edu.

Adham Sakr is an AI and robotics engineer with a solid background in autonomous systems, machine learning, and intelligent robotics. Proven experience in creating practical solutions for challenging environments, including autonomous driving and service robotics. Currently exploring how to integrate other technologies with AI technologies to enhance automation and efficiency in real-world scenarios. He committed to advancing the field through innovative projects and impactful applications. He can be contacted at adhamsakr65@student.aast.edu.



Yasser Salem is an AI developer and robotics engineer with expertise in smart systems, autonomous vehicles, and robotics design. Currently pursuing my master's at Breitenberg Technical University (BTU), experienced with hands-on projects in AI-driven perception and automation. He is passionate about advancing robotics and AI to create innovative solutions. He can be contacted at yassersalem@student.aast.edu.

Ahmed Abdelhadi is an AI and robotics engineer with extensive experience in autonomous systems, machine learning, and computer vision. Skilled in developing advanced robotics solutions, including autonomous vehicles and home assistance robots. Currently focused on real-world applications of AI and robotics, including hazardous environment automation. He is passionate about leveraging cutting-edge technologies to drive innovation in AI-driven automation. He can be contacted at Ahmody2892003@student.aast.edu.

HossamEldin Elsayed is a teaching assistant at the College of Artificial Intelligence, Arab Academy for Science, Engineering, Alamein Campus. He completed his bachelor's degree in computer engineering from the Arab Academy for Science, Technology, and Maritime Transport in 2020. Recognizing his passion for the field, HossamEldin continued his academic journey at the same institution, where he is currently pursuing a master's degree in computer engineering. He can be contacted at hossameldeen@aast.edu.

Ahmed El-Shaer is an associate professor at the College of Artificial Intelligence, Arab Academy for Science, Engineering, Alamein Campus. His work mainly with fabrication, supercapacitors, and electronic measurements. He can be contacted at ahshaer1@aast.edu.