INTERNATIONAL JOURNAL OF INTEGRATED ENGINEERING
ISSN: 2229-838X e-ISSN: 2600-7916

IJIE

Vol. 16 No. 5 (2024) 373-384
https://publisher.uthm.edu.my/ojs/index.php/ijie

Artificial Neural Network-based Approach for Inverse Kinematics of PUMA 260 Robot

Israa Rafie Shareef[1]*

[1] *Department of Mechatronics Engineering, Al Khwarizmi College of Engineering, University of Baghdad, IRAQ*

*Corresponding Author: *israarafie@kecbu.uobaghdad.edu.iq*
DOI: https://doi.org/10.30880/ijie.2024.16.05.028

Article Info

Received: 13 January 2024
Accepted: 20 June 2024
Available online: 14 August 2024

Keywords

Artificial neural network, denavit hartenberg, end-effector cartesian position, inverse kinematics, PUMA 260 robot manipulator, robotics vision control

Abstract

This study addressed the use of Artificial Intelligence (AI) techniques to solve the Inverse Kinematics (IK) and analysis issues for robotic manipulators, particularly PUMA 260 models with 6-Degrees of Freedom (6-DOF). A Robotics Vision Control (RVC) toolbox simulation in MATLAB was utilised to evaluate the Denavit Hartenberg (DH) and Forward Kinematics (FK) of the robot. Furthermore, the relationship between the joint angles of rotation and the end-effector Cartesian positions of the robotic manipulator was investigated. In reducing the complexity of the IK analysis, an Artificial Neural Network (ANN) was applied to estimate the IK of the robot using the Neural Network (NNtool) toolbox in MATLAB. This study successfully demonstrated the efficiency and applicability of the proposed ANN method for controlling robot motion by predicting IK parameters with high precision and minimal error. In determining the values of some IK joints, the results of the proposed technique significantly decreased to 1.579%. Thus, integrating the RVC and NNtool programmes considerably improved the accuracy in estimating ANN joint angles as a response to the input end effector positions. Consequently, the suggested ANN controller design was simpler, inexpensive, and more accurate in estimating the joint angles of the robot than standard regulating approaches. This study effectively and practically addressed the IK issues in robotic manipulators based on the results.

1. Introduction

Inverse Kinematics (IK) represents one of the fundamental challenges for robotic manipulators, where the input and output are the position and orientation of the robotic end effector relative to the robot base and the rotation values of the joints, respectively. Meanwhile, Forward Kinematics (FK) is concerned with determining the position of the end effector and joint orientation of the robot (angle or rotation). When the FK and IK challenges are addressed, a robotic manipulator can be efficiently controlled and programmed to accomplish numerous jobs and motions [1, 2]. Hence, several FK and IK solutions employing analytical and deep learning models have been presented in various research [3-5]. Moreover, many frameworks and their executions for robotic movements are controlled to upgrade the performance and activity using Artificial intelligence (AI) techniques [6]. Similarly, a system review study has been conducted on cooperate dual robotic arm manipulators and different classifications of control strategies have been discussed [7].

An Artificial Neural Network (ANN) is typically applied to solve the IK for robot manipulators, which AI techniques are ideally suited for. This AI technique can solve complex, challenging, and time-consuming issues for IK applications, while every stage of robotic technology enables the extensive use of AI [8]. Chiddarwar *et al.* in

2010 proposed an ANN model to determine the IK of the KUKA 6-Degree of Freedom (6-DOF) serial manipulator [9]. Similarly, an ANN model was applied to the FK of the robot, in which the FK equations were demonstrated to be effective in obtaining ANN training data. Thus, this outcome suggested that AI-based techniques could solve IK using a 6-DOF revolute robotic manipulator [10, 11]. Based on the Jacobian methodology (besides ANN), the study insisted on avoiding the singularity phenomenon.Moreover, the hybrid metaheuristic approach was proposed with three Elman ANN techniques, in which separate training sets were utilized to obtain higher precision solutions [12]. Similarly, deep learning and analytical approaches have been presented to solve inverse kinematics solutions with high DOFs. The performance results are compared with computational and analytical methods in generating robot trails[13]. A study has been conducted for an autonomous robot as a line follower using the proposed direction in [14]. Furthermore, optimization methods have been discussed to provide a kinematics solution of a higher degree of robotic movements by calculating the actual and estimated positions of the end effector manipulator [15]. A study showcased the applications of AI for robotic manipulation to solve IK using ANN, whereas the 5-DOF configuration is considered for industrial-need robots [16].

Several studies reported a feed-forward mechanism with multiple layers of perceptron back propagation ANN to solve the IK of PUMA560 and delta model robots [17, 18]. A study by Duka [19] summarised an ANN in controlling robot motion and trajectory tracking. Alternatively, Akos *et al*. adopted supervised learning to solve the IK concern to avoid constructing ideal robot models for calibration [20]. Another study by Mohan *et al*. [21] employed a kinematic technique for PUMA 560 robots to estimate the position and orientation vectors in activating the 6R manipulator. The 2D image was then used to analyze the robotics movement using a MATLAB/Simulink robotic toolbox. Therefore, this study introduced the planar two DOF and spatial three DOF structures. Consequently, a simple solution for solving the IK of a 6-DOF PUMA 260 manipulator robot was obtained. To the authors' knowledge, no previous studies comprised simulations of the building and ANN-controlling robots. In addition, no studies effectively addressed the PUMA 260 model robot when analyzing its IK. The following are the primary contributions of this study as follows:

- A simple method for solving IK for a 6-DOF PUMA 260 manipulator robot.
- An AI technique in simplifying the robot's IK equations and control motion analysis.
- An effective ANN technique in controlling robot motion by precisely estimating IK joint parameters with minimum error.
- Successful comparison of the AI technique to exhibit its simplicity, low cost, and precision with other conventional controlling methods.

2. Kinematic Analysis of PUMA 260

The robotic manipulator kinematics describes the relationship between the end effector (position and orientation) and the robot's joint angles. Each automated analysis requires robot kinematics alongside robot trajectory generation and motion control. Therefore, this study investigated the PUMA 260 model robot system with 6-DOF and six revolute joints. The initial coordinate was fixed at the robot's base, while the remaining coordinates were attached to manipulator links. Meanwhile, Fig.1 portrays the schematic diagram of the PUMA 260 Robot utilised in this study. The homogeneous matrix of transformation represents the orientation of the end effector alongside its position relative to the base coordinate. Thus, the transformation matrix is denoted in Eq. (1), in which i represents the robot link number as follows:

- αi : twist angle from zi−1 (previous link) to zi (present link) on xi
- di : link offsets from xi−1 (previous) to xi (present link) on zi−1
- ai : length of links from zi−1 (previous) to zi (present link) on xi
- θi : joint rotation angle from xi−1 (previous) to xi (present link) on z_{i-1} [22-24]. Each of L and t is mentioned in Fig. 1.

The Denavit Hartenberg (D-H) parameters of the PUMA robot are tabulated in Table 1 as follows:

Table 1 *Summary of the D-H parameters for PUMA 260*

i	□ range/°	□/°	*a*/mm	*d*/mm
1	$\theta 1$	90	0	13
2	$\theta 2$	00	8	00
3	$\theta 3$	90	0	L
4	$\theta 4$	90	0	08
5	$\theta 5$	90	0	00
6	$\theta 6$	00	0	t

When L and t are defined in Fig. 1, the transformation matrix T_{BASE}^{TOOL} is determined. The multiplication result of the following matrices, initiating from base to tool, is as follows:

$$T = \begin{bmatrix} \cos\theta i & -\sin\theta i \cos\alpha i & \sin\theta i \cos\alpha i & ai\ \cos\theta i \\ \sin\theta i & \cos\theta i \cos\alpha i & -\cos\theta i\ \sin\alpha i & \alpha i\ \sin\theta i \\ 0 & \sin\alpha i & \cos\alpha i & di \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$T_{BASE}^{TOOL} = T_0^1 T_1^2 T_2^3 T_4^5 T_5^6 \quad (2)$$

Where,

$$T_0^1 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & s & 0 \\ 0 & -1 & 0 & 13 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_1^2 = \begin{bmatrix} c_2 & -s_2 & 0 & 8c_2 \\ s_2 & c_2 & 0 & 8s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_2^3 = \begin{bmatrix} c_3 & 0 & s_3 & 0 \\ s_3 & 0 & -c_3 & 0 \\ 0 & 1 & 0 & -L \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_3^4 = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c & 0 \\ 0 & -1 & 0 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_4^5 = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{and } T_5^6 = \begin{bmatrix} c_6 & -s_2 & 0 & 0 \\ s_6 & c_2 & 0 & 0 \\ 0 & 0 & 1 & t \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

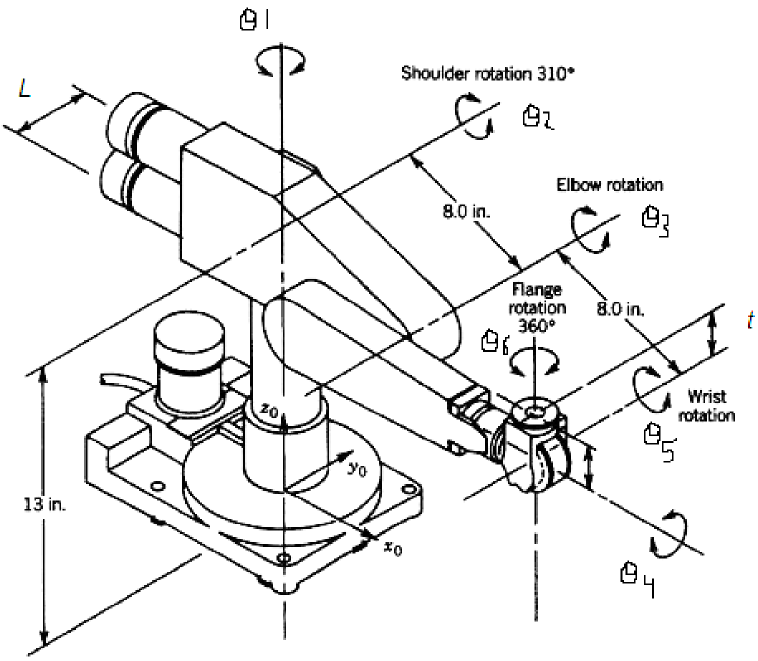

Fig. 1 *Schematic diagram of the PUMA 260 Robot [25]*

The tool coordinates concerning the robot base are as follows:

$$P_x = c_1(t(c_{23}c_4s_5 + s_{23}c_5) + 8s_{23} + 8c_2) - ts_1s_4s_5 \quad (3)$$

$$P_y = s_1(t(c_{23}c_4s_5 + s_{23}c_5) + 8s_{23} + 8c_2) - tc_1s_4s_5 \quad (4)$$

$$P_z = t(c_{23}c_5 + s_{23}c_4c_5) + 8c_{23} - 8s_2 \quad (5)$$

$$= c_1(c_{23}c_4s_5 + s_{23}c_5) - s_1s_4s_5a_x \quad (6)$$

$$= s_1(c_{23}c_4s_5 + s_{23}c_5) + c_1s_4s_5a_y \quad (7)$$

$$= -s_1c_4s_5 + c_{23}c_5a_z \quad (8)$$

Graphically, the PUMA 260 model robot is represented by Robotics Vision Control (RVC) toolbox simulation in MATLAB [25] (see Fig. 2) as follows:

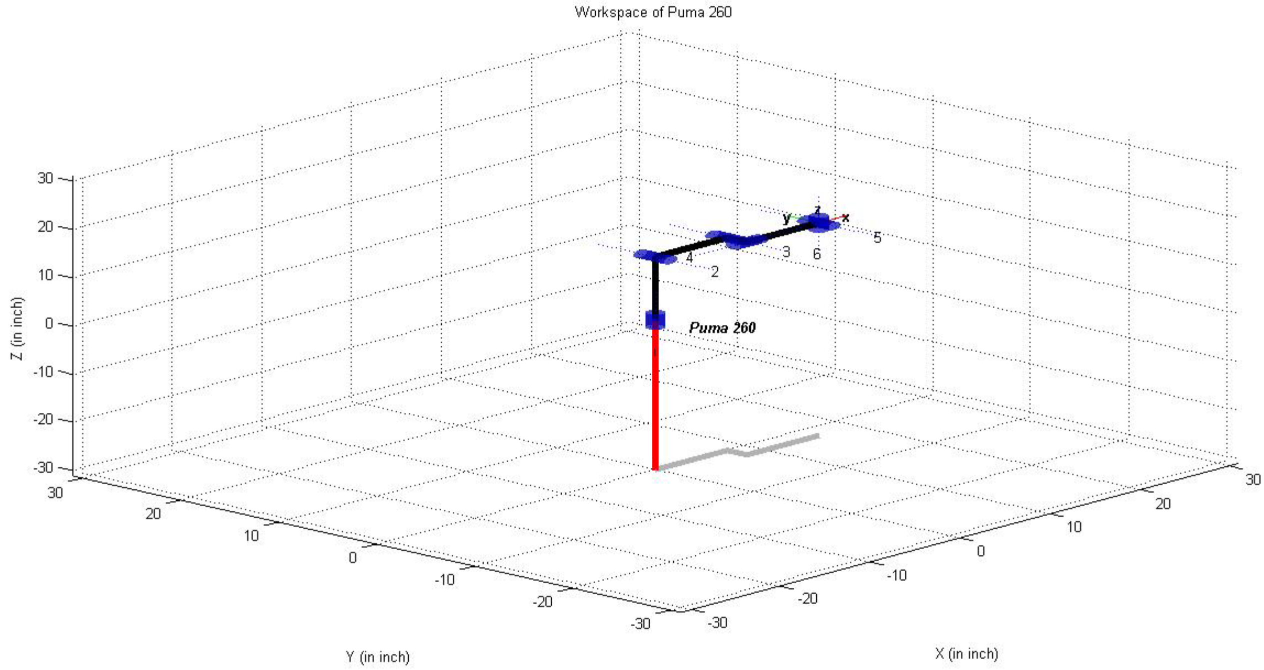

Fig. 2 *The PUMA 260 robot representation in the robot simulation*

Each joint angle of the robot corresponds to multiple configurations, including elbow down, elbow up, wrist down, wrist up, shoulder back, and shoulder forward. The FK determines the following location and orientation of the tool as follows:

$$[p_x, p_y, p_z, \theta_x, \theta_y, \theta_z] = FK\ [q_1, q_2, q_3, q_4, q_5, q_6] \tag{9}$$

Generally, IK is a difficult task for robotic manipulators due to the complexity and nonlinearity of the equations. Additionally, the motion of any robotic arm is determined by the IK equations defining the relationship between the Cartesian coordinate of the end effector and the robotic joints. Numerous solutions are associated with the IK for each joint variable, necessitating an optimal solution based on variables such as robot geometry, workspace, design range of motion, and singularity.

In this study, ANN was utilised to determine the IK of the 6-DOF PUMA robotic arm. The simulation of the RVC software yielded the delivered training data for the proposed network. Therefore, this study eliminated the requirements for a real robot with specific characteristics or reduced the robot device consumption by hundreds of labor-intensive, costly, and time-consuming experiments. In contrast, the ANN predicted the output joints' values based on the end effector's input orientation and position. The IK was a key mechanism for controlling the motion of robotic manipulators. Resultantly, finding a simple solution for IK was a crucial approach for regulating robot motion.

3. The Proposed ANN

This study employed ANN to solve the IK of PUMA robots, in which the input layer comprised six variables. These variables were the 3D positions of p_x, p_y, p_z and 3D orientations of $\theta_x, \theta_y, \theta_z$ of the end-effector. There were 10 hidden layers in this ANN with six elements of joint angles$(q_1, q_2, q_3, q_4, q_5, q_6)$ representing the ANN output layers. The MATLAB neural network toolbox (NNtool) trained, validated, and tested the data. Thus, the flow chart and the block diagram illustrating the role of ANN in optimising PUMA robot motion are exhibited in Figs. 3 and 4, respectively. The inputs were enclosed by a workspace occupying a particular position (see Fig. 5). Therefore, the relevant input and output for the ANN data were retrieved from the FK solution, where each robot position acquired one joint configuration.

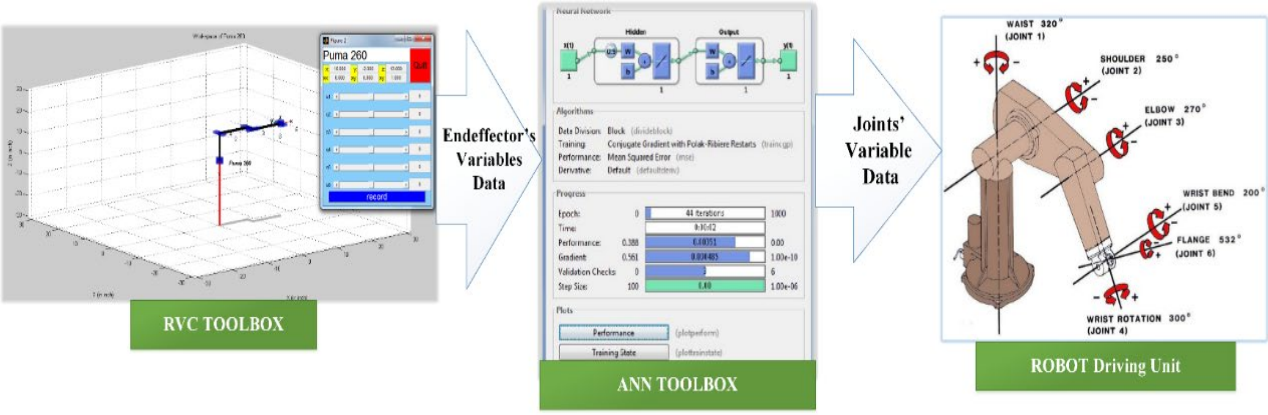

Fig. 3 *Brief block diagram of the IK prediction of the robot with the ANN technique*

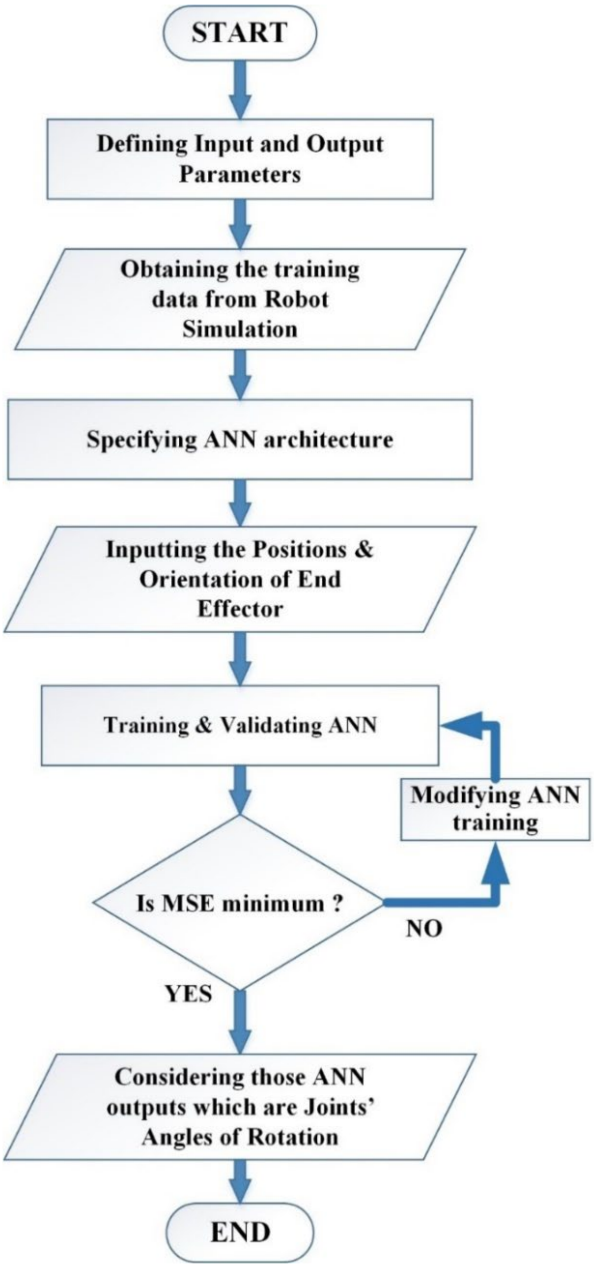

Fig. 4 *Flow chart in obtaining PUMA IK by using the ANN technique*

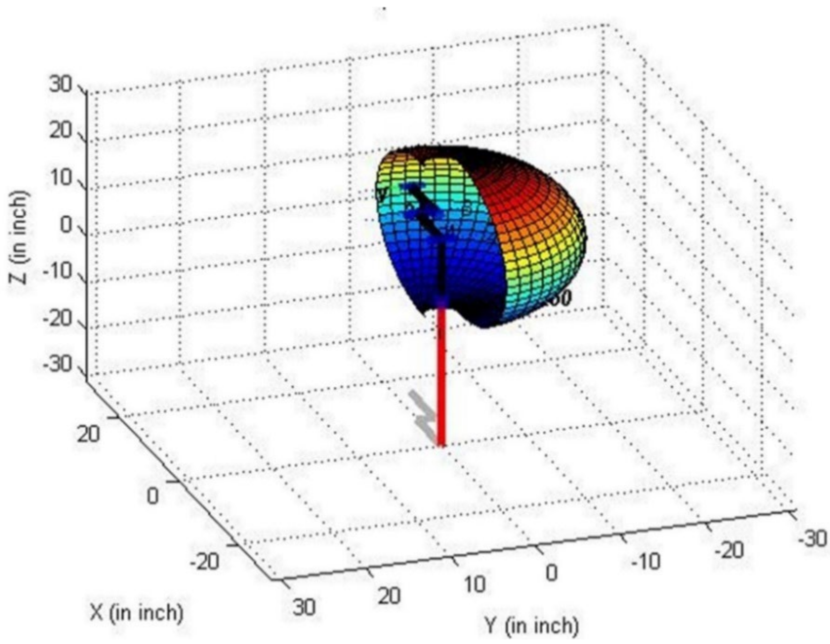

Fig. 5 *Workspace of PUMA260 model robot*

A multilayer perceptron neural network was described in this study as the complex nonlinear correlations between input and output variables were predicted. Moreover, the MLP neural network implemented the feed-forward Back Propagation (BP) technique, which depended on supervised learning and modifying the network output for the following layer. The adjustment of weight values is dependent on Mean Square Error (MSE), which is computed as follows:

$$MSE = \frac{1}{2n}\sum_{i}^{2} E_i^2 \tag{10}$$

Where n represents the number of data that is determined in each batch and E_i^2 represents the error value of input data number i. Furthermore, this neural network acquires six inputs (x, y, z, $a_x, a_{y,}$ and a_z), where $x, y,$ and z are the positions of the end effector in 3D. Subsequently, $a_x, a_{y,}$ and a_z are the orientations of the end effector concerning the three essential axes. Alternatively, six outputs (q_1, q_2, q_3, q_4, q_5, and q_6) represent the joints variables, while the PUMA joints are angular. Hence, these variables are the joint angles of rotation $\theta_{1\ldots6}$.

The sigmoid is a proposed activation function for the 10 hidden network layers, while the hyperbolic tansigmoid is the activation function engaged in the hidden network layers. Meanwhile, the Levenberg Marquardt (LM) learning approach trains the ANN. Consequently, the LM algorithm provides optimal weights by altering the learning rate of the neural network and attaining the smallest error value [20, 21]. In addition, this process causes the activation function of the output layer to be linear. Figure 6 depicts the MATLAB NNtool structure of the suggested ANN to predict the IK values for the robotic manipulator PUMA 260 [22, 23].

The outputs are generated via the supervised batch learning technique. From the investigated configurations in the toolbox of the robot, target values are passed to an ANN, which subsequently predicts the output. The following mathematical model defines the updating of weights alongside the biases of ANN:

$$S = \sum_{i-1}^{n}(w_i x_i) + b \tag{10}$$

where x_i are the six inputs (orientation and position of a tool), w_i is the weight, b is the bias, and n is the number of variables. Thus, the network trains until the MSE is as low as possible.

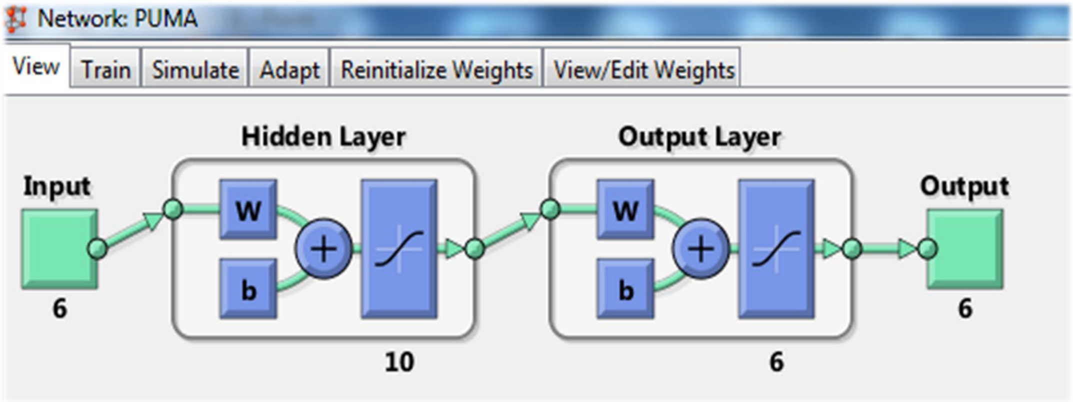

Fig. 6 *The ANN architecture in determining the IK of PUMA*

4. Results and Discussion

This section displayed and discussed the results acquired from the ANN technique proposed for solving the IK issues of the PUMA 260 robot model. The ANN structure successfully predicted the IK values of the robot as depicted in Fig. 4. Alternatively, Fig. 7 elucidates the ANN performance in calculating the MSE of training, in which the percentage error reduces until it converges to zero, which indicates that ANN undergoes training epochs until the difference between actual and predicted IK values are diminished. It indicates a convergence toward accurate predictions. Based on the results, the best validation performance was 9.4441 at epoch 2, indicating a minimal error during validation, reinforcing the reliability and accuracy of the proposed AI approach.

Moreover, the 10 optimal robot configurations (C1–C10) were determined using the RVC toolbox to evaluate the ANN performance. These configuration values were then provided to the ANN for training the data likely to represent the specific combinations of joint angles for the PUMA 260 robot that are considered optimal for testing or validation purposes. Eventually, the trained ANN data successfully predicted the IK (joint angle values) for any given robot tool position and orientation.

Table 2 lists the obtained robotic configurations (C1–C10). The findings revealed that the proposed ANN technique accurately forecasted the joint angle values of the PUMA 260 robot model. Hence, the comparison between the IK values acquired from the robot simulation programme and the ANN predictions suggested the exceptional performance of the AI technique in estimating the joint angles of the robot (average error of only 1.579%).

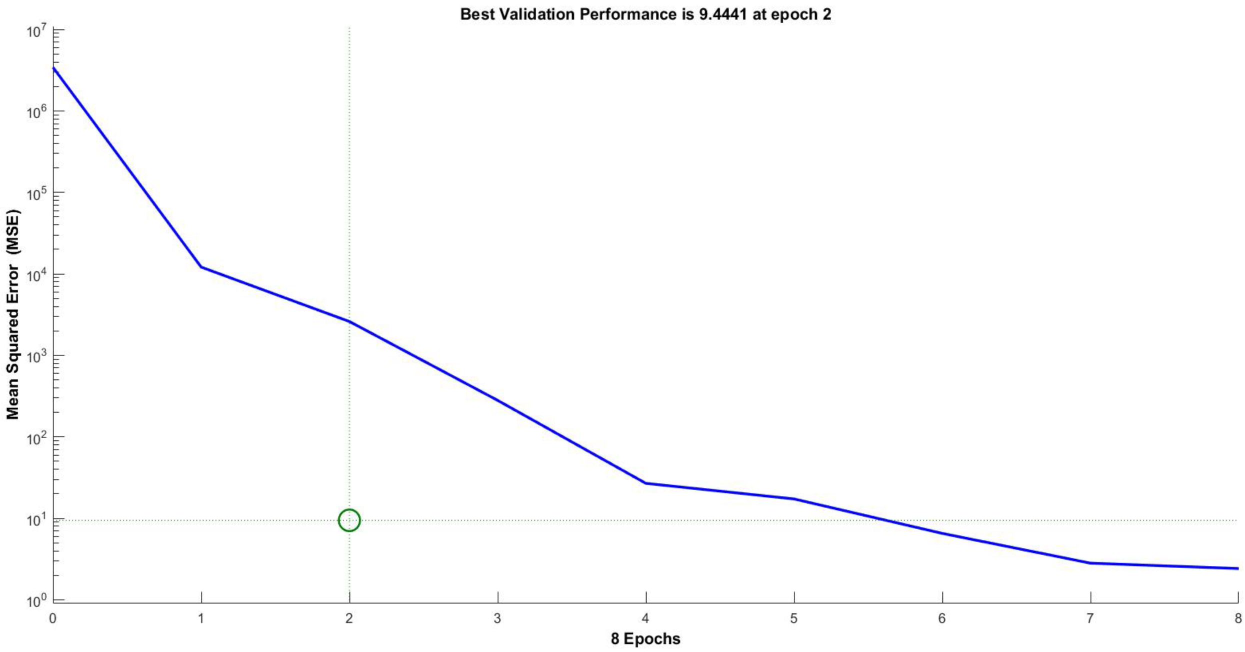

Fig. 7 *The performance of the ANN*

Table 2 *Summary of the studied robotic configurations*

Configuration number (C1–C10)	End effector position			End effector orientation		
	x	y	z	a_x	a_y	a_z
1	-01.321	-5.847	06.265	-0.525	-0.833	-0.174
2	04.868	-2.536	08.362	0.170	-0.804	-0.570
3	-05.709	-2.798	26.990	0.293	-0.860	-0.418
4	06.508	2.000	00.834	0.536	-0.276	0.798
5	15.004	2.000	09.135	-0.747	-0.440	-0.498
6	02.162	0.183	05.591	0.261	-0.764	-0.591
7	-10.536	2.000	06.360	-0.807	-0.451	0.381
8	00.000	2.000	13.000	0.000	0.000	1.000
9	00.000	2.000	13.000	0.000	0.000	1.000
10	14.096	7.769	12.070	0.193	-0.003	0.981

Figs 8A to 8F reveal the convergence results obtained using the robot simulation and the performance of a predicted ANN for each robot joint angle of a 6-DOF PUMA robotic manipulator. The average percentages of error between the target (based on training data from robot simulation) and the output (produced by the ANN) for each joint ($\theta_1$to θ_6). These percentage errors are reported as follows: (9.717, 1.697, 5.560, 7.413, 1.579, and 7.971%) respectively. From the figures, these values predicted by ANN nearly matched the actual values acquired from the robot simulation. In addition, the solid regression value of 0.99981 (see Fig. 9) was a crucial indicator of the effectiveness of the proposed neural network (ANN) in estimating the IK of the 6-DOF PUMA robotic manipulator, which depicts the convergence trajectories or behaviours of the joint angles during simulation. Moreover, the high regression value indicates that the AI model provides a reliable and precise estimation of the IK of robots.

This observation demonstrated that the trained ANN model with the 10 best configurations could predict the IK for every given orientation and position. Therefore, the findings of this study highlighted the efficiency and applicability of the suggested AI technique by lowering the IK analysis complexity of the robot and enhancing its precision. Compared to conventional controlling approaches, the model possessed the advantages of simplicity, low cost, and minimal error in calculating the joint angles of the robot. Thus, this study offered a comprehensive analysis of the PUMA 260 model robot, showcasing the viability of ANN approaches for regulating robot motion. However, some limitations are observed in solving the Inverse Kinetics equations of the PUMA robot using the ANN approach. Training data sometimes gives noisy results, where the end effector's position and orientation require sensing and detecting efforts in computations over fittings during the model development. Each robotic model may require a special network pattern to deal with its Inverse Kinematics issue. Therefore, to generalise this approach and enhance it for unseen data, feedback of the current robotic arm joint angles could be added to the input pattern of the neural network besides each of the desired positions and orientations of the end effector, ultimately increasing the accuracy and performance.

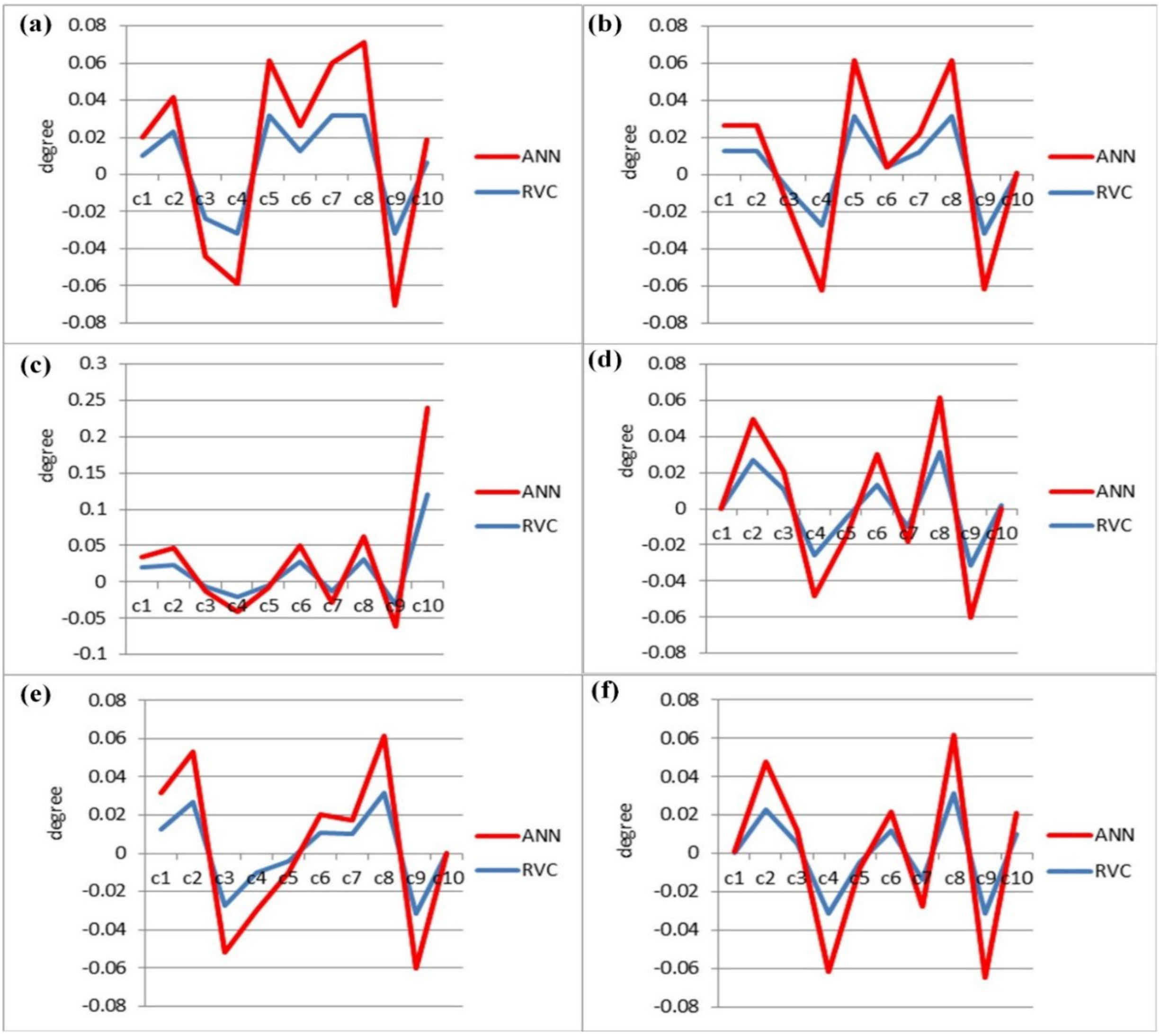

Fig. 8 *Graphs indicating (a) THETA 1-waist joint angle; (b) THETA 2-shoulder joint angle; (c) THETA 3-elbow joint angle; (d) THETA 4-wrist rotation joint angle; (e) THETA 5-wrist bend joint angle; and (f) THETA 6-flange rotation joint angle*

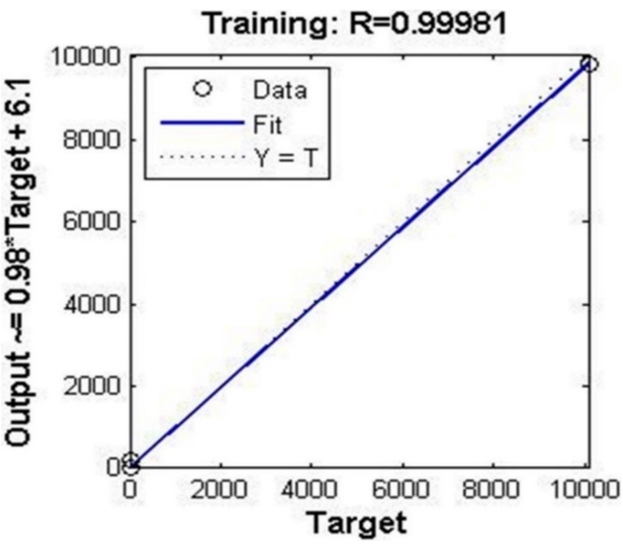

Fig. 9 *Regression value for the ANN training data*

5. Conclusion

This study successfully presented a novel method for solving the IK concerns using the ANN technique for a 6-DOF PUMA 260 model robot. Hence, the results demonstrated that the proposed AI method in estimating the joint angles of the robot was effective and useful. The input parameters of x, y, z, a_x, a_y and a_z were trained using ANN in MATLAB, representing the end-effector position and orientation of the robot. A back-propagation neural network was subsequently introduced to utilize the input data, in which the joint angles were measured. The estimated ANN outputs were compared with the intended outputs by calculating the MSEs. A comparison was also performed between the IK findings from the robot simulation program and the ANN.

The findings indicated that the provided AI method performed exceptionally well in estimating the joint angles of the robot, with a maximum error of 1.579% for some IK joint values. Compared to classical analytic methodologies suited to the most difficult kinematic issues of multi-link robotics, the suggested ANN findings revealed superior effectiveness in calculating the angles of the robot joints with high precision and little error. Moreover, the RVC toolbox integrated with NNtool considerably enhanced the accuracy of estimating the ANN joint angles output in response to the input end-effector position. Thus, this study could advance robotics and control applications, particularly industrial automation. The potential future aspects of the proposed study could include:

1. Accuracy in robotics advancements: The proposed ANN technique for solving IK problems demonstrates the improvement in accuracy and robotics joint precision, which ultimately can be implemented in the precise control of advanced robotics systems containing high DOF.
2. Integration with AI Tools: By integrating the AI methods for robot simulation programs using ANN methods, the pre-operational testing and development of robotic systems in a virtual environment could be significantly improved prior to practical implementation. Which further reduces the reliance on classical analytical techniques because the AI methods provide a potential shift to simulate kinematics issues in multi-link robotics. Therefore, future research may explore alternate ANN-driven methods for solving complex robotics problems, which alleviates the reliance on conventional analytical solutions.
3. General applicability to Other Robotic Systems for Enhanced Accuracy: The proposed AI approach may be successfully applied to various robotic systems from a specific robot type. Integrating the RVC toolbox with NNtool and its accuracy improvement implies that integrating specialised tools may yield better results, performance, and accuracy. Future research may examine the ANN technique's suitability for solving IK problems in a wider range of robotic designs and applications.
4. Continuous Improvements and Optimisation: the research findings indicate a maximum error of 1.579%, which suggests high accuracy. Meanwhile, future research may focus on enhancing performance by improving and optimising the ANN model to reduce errors and achieve high precision.

Acknowledgement

I would like to express my sincere gratitude to the University of Baghdad, Al-Khwarizmi College of Engineering for the academic support during this paper. Besides, special thanks to the department of Mechatronics Engineering staff for their valuable guidance and encouragement.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

References

[1] B. Siciliano, O. Khatib, and T. Kröger, *Springer handbook of robotics*. Springer, 2008.

[2] J. Zhang, Q. Qu, and X.-b. Chen, "Assessing the sustainable safety practices based on human behavior factors: an application to Chinese petrochemical industry," *Environmental Science and Pollution Research,* vol. 29, no. 29, pp. 44618-44637, 2022/06/01 2022, doi: 10.1007/s11356-022-18909-5.

[3] P. Srisuk, A. Sento, and Y. Kitjaidure, "Inverse kinematics solution using neural networks from forward kinematics equations," in *2017 9th International Conference on Knowledge and Smart Technology (KST)*, 1-4 Feb. 2017 2017, pp. 61-65, doi: 10.1109/KST.2017.7886084.

[4] H. Ye, D. Wang, J. Wu, Y. Yue, and Y. Zhou, "Forward and inverse kinematics of a 5-DOF hybrid robot for composite material machining," *Robotics and Computer-Integrated Manufacturing,* vol. 65, p. 101961, 2020/10/01/ 2020, doi: https://doi.org/10.1016/j.rcim.2020.101961.

[5] J. S. Toquica, P. S. Oliveira, W. S. R. Souza, J. M. S. T. Motta, and D. L. Borges, "An analytical and a Deep Learning model for solving the inverse kinematic problem of an industrial parallel robot," *Computers & Industrial Engineering,* vol. 151, p. 106682, 2021/01/01/ 2021, doi: https://doi.org/10.1016/j.cie.2020.106682.

[6] A. K. Abbas, Y. Al Mashhadany, M. J. Hameed, and S. Algburi, "Review of Intelligent Control Systems with Robotics," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI),* vol. 10, no. 4, pp. 734-753, 2022.

[7] M. Abbas, J. Narayan, and S. K. Dwivedy, "A systematic review on cooperative dual-arm manipulators: modeling, planning, control, and vision strategies," *International Journal of Intelligent Robotics and Applications,* vol. 7, no. 4, pp. 683-707, 2023/12/01 2023, doi: 10.1007/s41315-023-00292-0.

[8] S. Dereli and R. Köker, "Simulation based calculation of the inverse kinematics solution of 7-DOF robot manipulator using artificial bee colony algorithm," *SN Applied Sciences,* vol. 2, no. 1, p. 27, 2019/12/05 2019, doi: 10.1007/s42452-019-1791-7.

[9] S. S. Chiddarwar and N. Ramesh Babu, "Comparison of RBF and MLP neural networks to solve inverse kinematic problem for 6R serial robot by a fusion approach," *Engineering Applications of Artificial Intelligence,* vol. 23, no. 7, pp. 1083-1092, 2010/10/01/ 2010, doi: https://doi.org/10.1016/j.engappai.2010.01.028.

[10] A. A. Hassan, M. El-Habrouk, and S. Deghedie, "Inverse Kinematics of Redundant Manipulators Formulated as Quadratic Programming Optimization Problem Solved Using Recurrent Neural Networks: A Review," *Robotica,* vol. 38, no. 8, pp. 1495-1512, 2020, doi: 10.1017/S0263574719001590.

[11] A. T. Hasan, N. Ismail, A. M. S. Hamouda, I. Aris, M. H. Marhaban, and H. Al-Assadi, "Artificial neural network-based kinematics Jacobian solution for serial manipulator passing through singular configurations," *Advances in Engineering Software,* vol. 41, no. 2, pp. 359-367, 2010.

[12] R. Köker, "A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization," *Information Sciences,* vol. 222, pp. 528-543, 2013/02/10/ 2013, doi: https://doi.org/10.1016/j.ins.2012.07.051.

[13] N. Wagaa, H. Kallel, and N. Mellouli, "Analytical and deep learning approaches for solving the inverse kinematic problem of a high degrees of freedom robotic arm," *Engineering Applications of Artificial Intelligence,* vol. 123, p. 106301, 2023/08/01/ 2023, doi: https://doi.org/10.1016/j.engappai.2023.106301.

[14] M. Baballe, "Principle Operation of a Line Follower Robot," vol. 3, pp. 6-10, 06/06 2023, doi: 10.5281/zenodo.8011548.

[15] S. Dereli and R. Köker, "A meta-heuristic proposal for inverse kinematics solution of 7-DOF serial robotic manipulator: quantum behaved particle swarm algorithm," *Artificial Intelligence Review,* vol. 53, no. 2, pp. 949-964, 2020/02/01 2020, doi: 10.1007/s10462-019-09683-x.

[16] S. Aravinddhakshan, S. Apte, and S. M. Akash, "Neural Network Based Inverse Kinematic Solution of a 5 DOF Manipulator for Industrial Application," *Journal of Physics: Conference Series,* vol. 1969, no. 1, p. 012010, 2021/07/01 2021, doi: 10.1088/1742-6596/1969/1/012010.

[17] L. Aggarwal, K. Aggarwal, and R. J. Urbanic, "Use of Artificial Neural Networks for the Development of an Inverse Kinematic Solution and Visual Identification of Singularity Zone(s)," *Procedia CIRP,* vol. 17, pp. 812-817, 2014/01/01/ 2014, doi: https://doi.org/10.1016/j.procir.2014.01.107.

[18] A. Gholami, T. Homayouni, R. Ehsani, and J.-Q. Sun, "Inverse Kinematic Control of a Delta Robot Using Neural Networks in Real-Time," *Robotics,* vol. 10, no. 4, p. 115, 2021. [Online]. Available: https://www.mdpi.com/2218-6581/10/4/115.

[19] A.-V. Duka, "Neural network based inverse kinematics solution for trajectory tracking of a robotic arm," *Procedia Technology,* vol. 12, pp. 20-27, 2014.

[20] A. Csiszar, J. Eilers, and A. Verl, "On solving the inverse kinematics problem using neural networks," in *2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, 21-23 Nov. 2017 2017, pp. 1-6, doi: 10.1109/M2VIP.2017.8211457.

[21] N. Mohan, A. Thomas, P. P. Prasanth, M. S. Midhun, and J. Kurian, "Inverse Kinematic Analysis of PUMA 560 for Vision Systems," in *Intelligent Manufacturing and Energy Sustainability*, Singapore, A. N. R. Reddy, D. Marla, M. N. Favorskaya, and S. C. Satapathy, Eds., 2022// 2022: Springer Singapore, pp. 285-293.

[22] R. Mehta and A. Sahu, "Autonomous robot for inventory management in libraries," in *2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, 2020: IEEE, pp. 1-6.

[23] B. O. Nnaji, "A Review of: "Fundamentals of Robotics Analysis and Control", RJ Schilling, Prentice Hall, 1990," *Iie Transactions,* vol. 24, no. 1, pp. 94-94, 1992.

[24] I. R. Shareef, I. A. Zayer, and I. A. Abd Al Kareem, "Investigating Forward kinematic Analysis of a 5-axes Robotic Manipulator using Denavit-Hartenberg Method and Artificial Neural Network," *Engineering and Technology Journal,* vol. 32, no. 11 Part (A) Engineering, 2014.

[25] B. Goodman. Puma 260 Joint Diagram [Online] Available: https://benegoodman.wordpress.com/mechatronicsembedded-systems/puma-260-joint-diagram/