

Multi-Layer Feedforward Neural Network Modelling of a Kinematics Solution of A 3-DoF Manipulator Robot

Samara Munaem Naeem

Department of Aeronautical Engineering, College of Engineering, University of Baghdad, Iraq
s.naeem@coeng.uobaghdad.edu.iq (corresponding author)

Received: 10 July 2025 | Revised: 4 August 2025 | Accepted: 22 August 2025

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | DOI: <https://doi.org/10.48084/etasr.13279>

ABSTRACT

Modeling forward kinematics with neural networks allows for efficient handling of nonlinear relationships and realistic error correction in time-critical applications by relying on accurate training data. This paper presents a Multi-Layer Feed-Forward Neural Network (MLFFNN) to solve the forward kinematics of a 3-DOF robot. The proposed MLFFNN consists of 50 hidden neurons and was trained using 628319 samples to find only the position (x, y, z) of the end-effector. Data were generated by MATLAB, assuming an incremental motion of joints. The joint variables (θ_1 , θ_2 , and θ_3) are the inputs of the NN, which outputs the positions of the end effector (x, y, z) calculated using the Denavit-Hartenberg (DH) method. The results demonstrate that the proposed MLFFNN has high performance and is efficient for solving the forward kinematics, with a Mean Squared Error (MSE) between the desired and estimated position of 4.3881×10^{-11} . This performance clearly demonstrates that, despite the large size of the dataset, it can be effectively mastered with only a small number of neurons. The simplicity of the network allows it to learn a compact and efficient representation of the data. This improves the reliability of using the proposed network for similar applications in other robotic systems.

Keywords-multilayer neural network; forward kinematics; 3 DOF robot; serial robot; Denavit-Hartenberg method; incremental motion

I. INTRODUCTION

Due to the developments in modeling robots, it has become necessary to prepare the data of the forward and inverse kinematics according to the type of robot. It is also essential to identify the workspace in which the robot moves, considering its movement mechanism and geometric dimensions, to model kinematics using Artificial Neural Networks (ANNs). Many researchers have addressed the problem of forward kinematics of serial robots using different ANN models to generate an accurate solution for the end position and orientation. In [1], the forward kinematics of an open-chain serial robot with two Degrees of Freedom (2-DoF) was modeled by training a three-layer ANN using backpropagation. In [2], a safety method was developed for a 3-DoF industrial robot by modeling only the joint locations as input to the RNN, trained using the Levenberg-Marquardt (LM) algorithm. In [3], the forward kinematics of a 3-DoF robot was modeled using a radial basis function ANN, tuned by a cooperative search optimization algorithm. In [4], a three-layer ANN used PID input to control the position of a 3-DoF robotic arm. Other studies have focused on the accuracy of modeling forward kinematics. In [5], a plan of the robot's structure and its dimensions was used to develop the Denavit-Hartenberg (DH) method, explaining the forward kinematics of a serial industrial robot to reduce the error in its location resulting from thermal deformation during operation.

Other studies focused on solving and modelling the inverse kinematics of robots to control the location of the manipulator's end. In [6], the Bayesian regularization backpropagation method was used to train an ANN consisting of 3 hidden layers and 25 neurons as an inverse kinematics model for a 3D robot. In [7], three neural networks with two hidden layers were trained to compute the variables of each joint and solve the inverse kinematics of a 5-DoF robot. In [8], a new framework was proposed for controlling kinematics using a reversible ANN with two different architectures for the UR5 robot. In [9], the overall position and rotation of a 7-DoF robot and a soft robot were extracted using two ANN models, the FF net and the Aug net. In [10], an adaptive neuro-fuzzy inference system was used to train the network, using data with two inputs and one output to solve the inverse kinematics of 2- and 3-DoF planar robots. In [11], a comparative study was carried out to solve the inverse kinematics using an FFBP-NN, changing some of its parameters, training it for a 2-DoF robot and comparing the results. In [12], the forward and inverse kinematics of 3-DoF robots were determined by training multilayer ANNs using the LM method [12]. In [13], an ANN was trained using backpropagation from the forward kinematics equations to solve the inverse kinematics of a 4-DoF robot [13].

There is a diversity in the structures of robots whose forward kinematics have been modeled, between parallel and serial robots. The methods used for these two types of robots were determined in [14]. Some researchers focused on modeling the forward kinematics of parallel robots using ANNs because it is a controversial topic with its different structures, such as cable driving. In [15, 16], a multi-layer and a supervised ANN were used for this purpose, respectively. In [17], forward kinematics for the HEXA robot was modeled using a multi-layer ANN. Many researchers have proposed solving the forward kinematics of Stewart's platform with different types of ANN depending on the case studied [18-21]. Due to the complex nonlinear system of the SCARA robot and its closed chain structure, the study in [22] relied on its forward kinematics with a 3-PRR joint and suggested dividing the workspace into multiple sub-spaces, using an ANN to model movements. In [23], an ANN was trained using three different datasets to represent the forward kinematics of the SCARA robot with 4 DoF. In [24], a two-layer ANN was used to solve the forward kinematics of a hydraulic arm structure consisting of three revolute joints that move the arm using its four cylinders, which was assumed to be a parallel robot. In [25], the forward kinematics of a robot with a parallel and serial structure was studied, modeling its kinematics using two ANNs trained for this specific hybrid structure. Accordingly, researchers with this type of robotic structure were interested in finding the forward kinematics of the parallel robot, and their sole interest was in finding a way to model this kinematics. In [26], a new method was proposed to model the forward kinematics of an industrial robot with convergent movement, considering it a closed-chain robot, taking into account passive joints and geometry parameters.

These studies show that it is essential to pay attention to the strategies used to train ANNs. The study in [27] reviewed the algorithms used to train networks, offering a greater understanding of their success through modelling. In [28], the flexible NeuralNet package was used in three types of backpropagation to train the ANN, providing an overview of the hidden layers and the uses of this package. Although some researchers relied on performing non-model calibration of various robot structures based on ANN, several implementation methods have been explored [29-32].

This study investigates the forward kinematics of a 3D serial robot arm consisting of three revolute joints using the Denavit-Hartenberg (DH) method. A total of 628,319 unique movement combinations were generated to map the full reachable workspace and determine the maximum expected position of the end effector. These data points served as the basis for training an ANN model based on the backpropagation algorithm. Two different ANN architectures were examined, using joint variables as inputs and the corresponding end-effector position and orientation as outputs. A Multi-Layer Feedforward Neural Network (MLFFNN) was adopted because of its ability to model complex nonlinear relationships accurately and efficiently. The network was trained using the LM algorithm, with performance measured by the Mean Squared Error (MSE). Training and validation were performed using MATLAB R2021b, ensuring robust learning verification.

What distinguishes this study is its novel implementation of a high-accuracy MLFFNN to solve the forward kinematics problem, achieving a significantly lower MSE than previous approaches. Unlike prior works that depend on analytical models or RNNs with long training times and limited accuracy, the proposed model utilizes a large-scale dataset that allows for excellent generalization across the robot's workspace. The proposed method reduces computational complexity while maintaining high prediction reliability. Comparison with previous studies clearly demonstrates that this work provides a significant advancement in robotic kinematics modeling, offering a practical, scalable, and data-driven solution for 3-DoF manipulators.

II. FORWARD KINEMATIC SOLUTION

A. Forward Kinematic model

This section presents the forward kinematics of a serial manipulator consisting of 3-RRR, as shown in Figure 1, to find the position and orientation of the end effector relative to the base of the robot, using the DH method, which is particularly helpful for serial robots due to its flexibility and suitability for modeling its joints and links, regardless of the complexity involved [30]. This method consists of standard parameters, which represent the relationship between the joint variables and help determine the HTM faster than any other method. Table I presents the DH parameters, outlining the rotational and linear displacement relationships between each pair of frames. Two of these parameters represent the rotations (α , θ), and the other two represent the linear displacement (d , r).

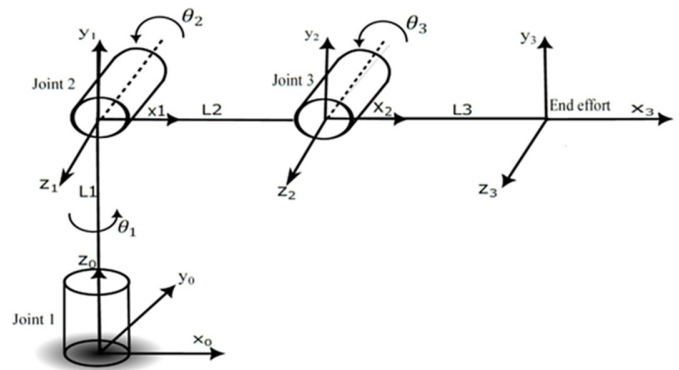


Fig. 1. The 3-DoF framework robotic manipulator.

TABLE I. DH PARAMETERS FOR 3-DOF FRAMEWORK ROBOTIC MANIPULATOR

Link	r	α	d	θ
1	0	90°	L1	θ_1
2	L2	0	0	θ_2
3	L3	0	0	θ_3

From Table I, the total HTM of the kinematic model of the above framework can be obtained, and the position and orientation of the end-effector can be represented with respect to the base of the frame as [31]:

$$HTM_3^0 = \begin{bmatrix} C_1 C_{23} & -C_1 S_{23} & S_1 & L_3 C_1 C_{23} + L_2 C_1 C_2 \\ S_1 C_{23} & -S_1 S_{23} & -C_1 & L_3 S_1 C_{23} + L_2 S_1 C_2 \\ S_{23} & C_{23} & 0 & L_3 S_{23} + L_2 S_2 + L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where the last column of this matrix characterizes the position of the end-effector from joint space to Cartesian space as P_x , P_y and P_z , respectively, which is important in calculating desired positions for robot manipulator design. $C = \cos \theta_i$, $S = \sin \theta_i$, with $\theta_i = 1, 2, 3, \dots, n$, and L_i represents the length of link, with $i = 1, 2, 3, \dots, n$.

This study aimed to build a neural network by modelling the forward kinematics of this robot, using the joint variables as the inputs and the position of the end-effector as the outputs. The outputs can be identified as follows:

$$P_x = L_3 C_1 C_{23} + L_2 C_1 C_2 \quad (2)$$

$$P_y = L_3 S_1 C_{23} + L_2 S_1 C_2 \quad (3)$$

$$P_z = L_3 S_{23} + L_2 S_2 + L_1 \quad (4)$$

These equations are more suitable to give the desired outputs to build the MLP ANN than (1).

B. Data Generated from the Forward Kinematics

Forward kinematics involves calculating the position and orientation of a robot's end effector based on its joint parameters. In this paper, data were generated using MATLAB, assuming that the joint angles will incrementally range from -180° to 180° . Each link of the robot has a length of 0.5 m, and the total number of samples collected was 628,319. In MATLAB, the joint angles θ_1 , θ_2 , and θ_3 are defined as follows:

$$\theta^1 = \theta_2 = \theta_3 = [-\pi:0.00001:\pi]$$

The data utilized for training, validating, and testing the proposed MLFFNN, designed to address forward kinematics, will calculate the position of the end effector (P_x, P_y, P_z) based on (2-4). Figures 2 and 3 show the working area of the proposed robot.

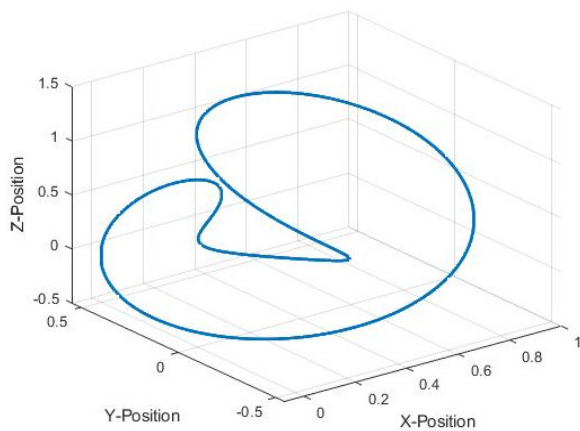


Fig. 2. The working area in 3D.

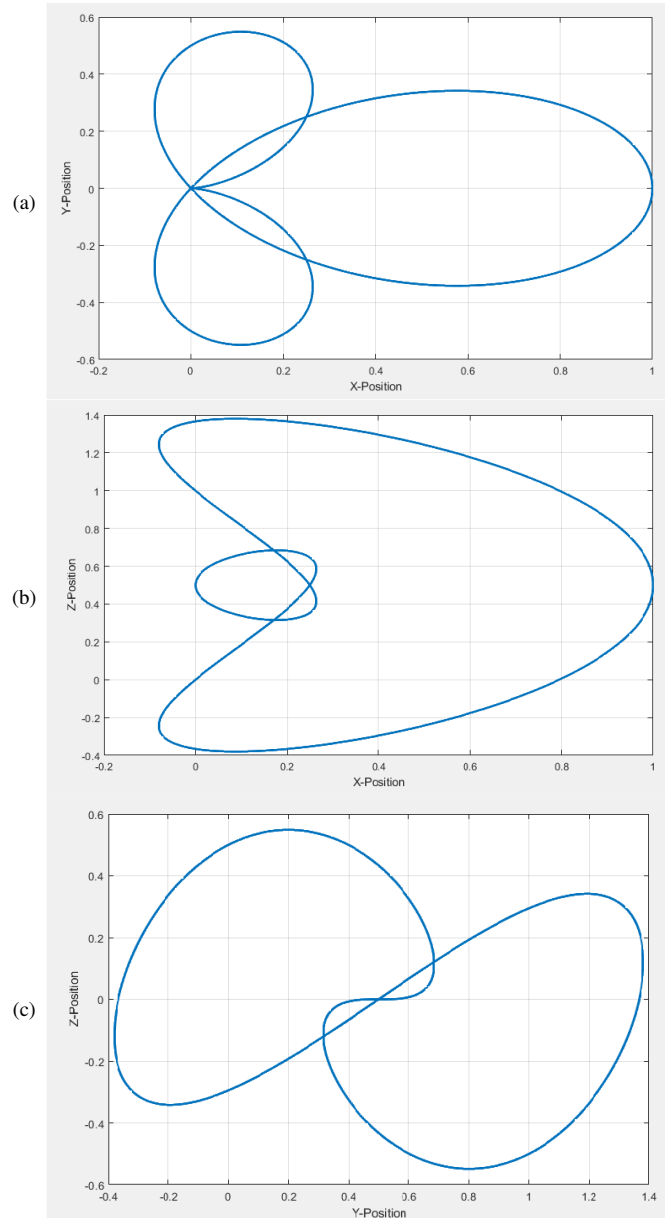


Fig. 3. The working area in 2D directions: (a) x-y, (b) x-z, (c) y-z.

The aim of building the network is to learn all the end-effector positions of the proposed robot in different scenarios. The network is trained using the angle of each joint as input and the position of the end effector of the manipulator as the output.

III. NEURAL NETWORK MODEL

The proposed MLFFNN, illustrated in Figure 4, is designed with three layers. The input layer consists of three inputs, the joint variables θ_1 , θ_2 , and θ_3 . The second layer (hidden) utilizes a sigmoid activation function, allowing the network to approximate complex functions, and contains several hidden neurons. The output layer is linear and estimates the position of the end-effector of the manipulator, denoted as (P_x, P_y, P_z).

These three outputs are compared with the definite values obtained from (2-4), as illustrated in Figures 2 and 3, to assess network reliability when the comparison between the actual and desired value is minimal and near zero.

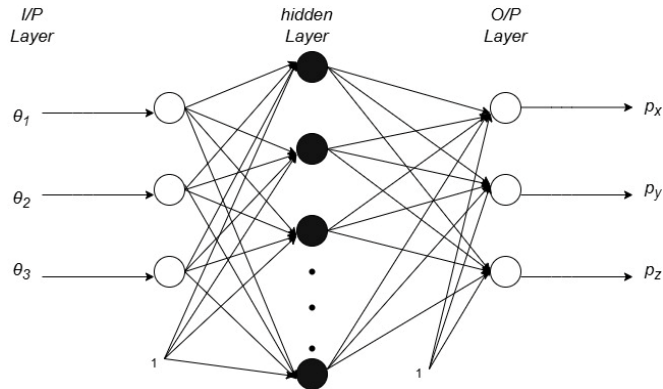


Fig. 4. The proposed MLFFNN for solving the forward kinematics (position of the end-effector of the 3-DoF manipulator).

In designing the ANN, the focus was on achieving higher performance, ensuring an output MSE close to zero. This commitment guarantees the reliability and precision of the results. To accomplish this goal, a comprehensive training strategy was employed, leveraging a vast array of datasets and the powerful LM learning algorithm. This approach ensures optimized performance and accuracy. In addition, the two-layer MLFFN with sigmoid hidden neurons and the linear output neurons is suitable for regression tasks.

A. Training the Network

The data were divided into three subsets: 80% (502655 samples) for training the network, 10% (62832 samples) for validation, and 10% for testing. The MLFFN model was configured with 50 hidden neurons to provide sufficient learning capacity for accurately capturing the nonlinear behavior of the robotic manipulator. The LM algorithm was chosen due to its high efficiency and fast convergence, particularly in medium-scale problems. A *tansig* transfer function was used in the hidden layer to introduce nonlinearity, while a *purelin* function was applied in the output layer to accommodate the continuous nature of the outputs.

The performance of the model was evaluated using MSE. To ensure gradual and stable convergence during training, a small initial value was established for the μ parameter at 1×10^{-9} . This approach aimed to achieve high performance, training the ANN to minimize the MSE to a value close to zero. Through trial and error, it was found that using 50 neurons in the hidden layer produced the best results when employing the LM algorithm as the training method. The training parameters outlined in Table II lead to impressive results, as demonstrated in Figure 5. This clear relationship highlights the effectiveness of the selected training approach.

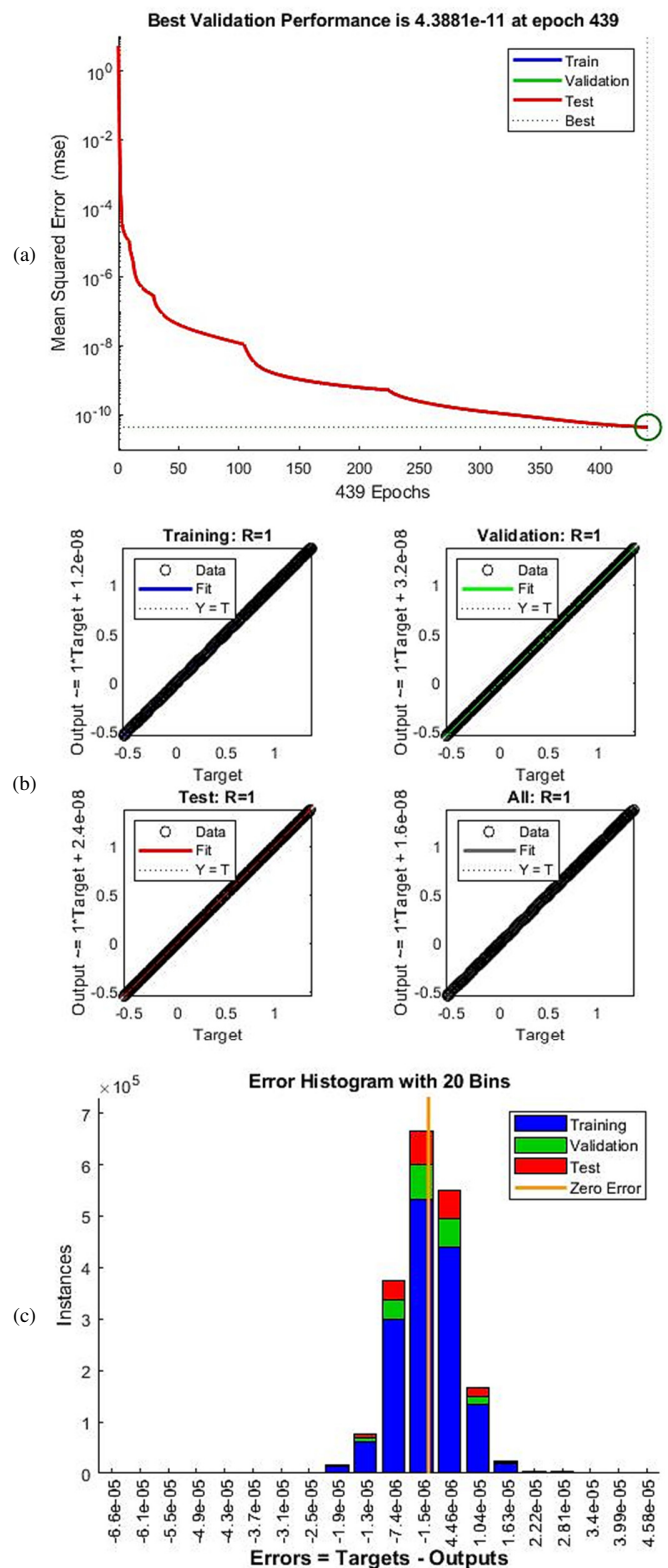


Fig. 5. Results of the proposed MLFFNN: (a) Training performance, (b) Regression results, (c) Error histogram.

TABLE II. TRAINING PARAMETERS OF MLFFNN

Parameter	value
Number of hidden neurons	50
Training algorithm	LM
Number of samples	628319
Data division	Random
Percentage of trained data	80%
Percentage of validation data	10%
Percentage of testing data	10%
Performance	MSE
μ	1×10^{-9}

The proposed MLFFNN demonstrates exceptional performance, as illustrated in Figure 5(a), with an MSE of 4.3881×10^{-11} . This virtually zero error signifies that the prediction output from the ANN aligns closely with the desired outputs for the 3-DoF robot's end-effector position, where the regression obtained from training is equal to 1, as shown in Figure 5(b). The error histogram shown in Figure 5(c) is notably narrow, demonstrating a reduced spread of errors. This

characteristic is a clear indicator of enhanced accuracy, underscoring the reliability of the results.

B. Verification and Testing of the Trained ANN

The proposed model was tested with 62832 samples. The results obtained from training were compared with the desired results for the (x, y, z) -position of the end-effector of the 3-DoF robot, as shown in Figures 6-8. The estimated positions produced by the proposed MLFFNN correspond accurately with the actual positions, exhibiting an exceptionally low error rate, nearly zero. This remarkable accuracy underscores the reliability of the proposed network in effectively solving the forward kinematics of a 3D robot, showcasing the thoroughness of its training and the confidence in its performance. The error between the desired position and the predicted position of the three axes is impressively low, nearing zero, as demonstrated in Figure 9. Table III unequivocally presents the resulting parameters related to this absolute error.

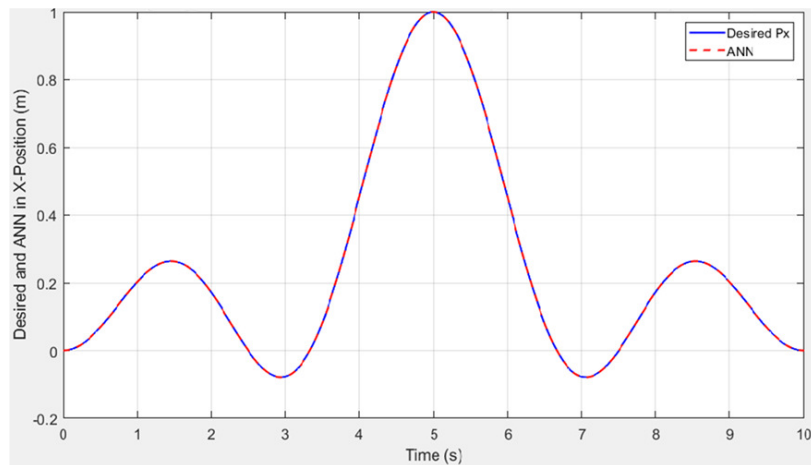


Fig. 6. Comparing the desired and estimated end-effector positions of the robot in the x-direction.

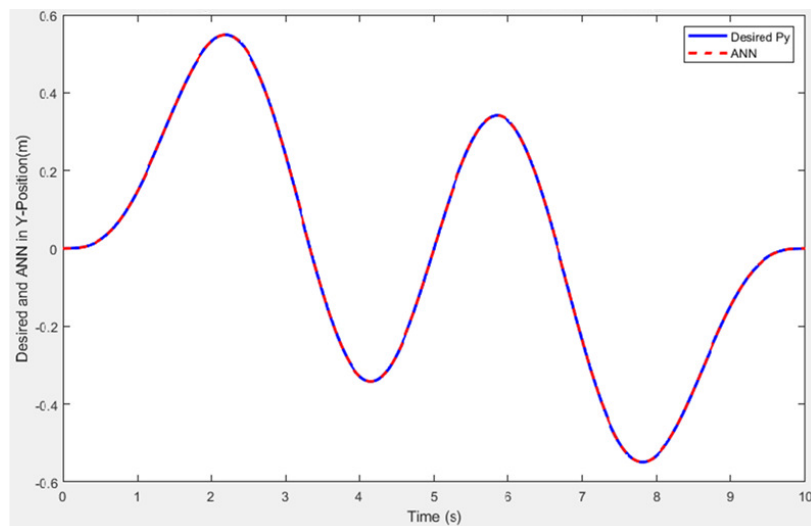


Fig. 7. Comparing the desired and estimated end-effector positions of the robot in the y-direction.

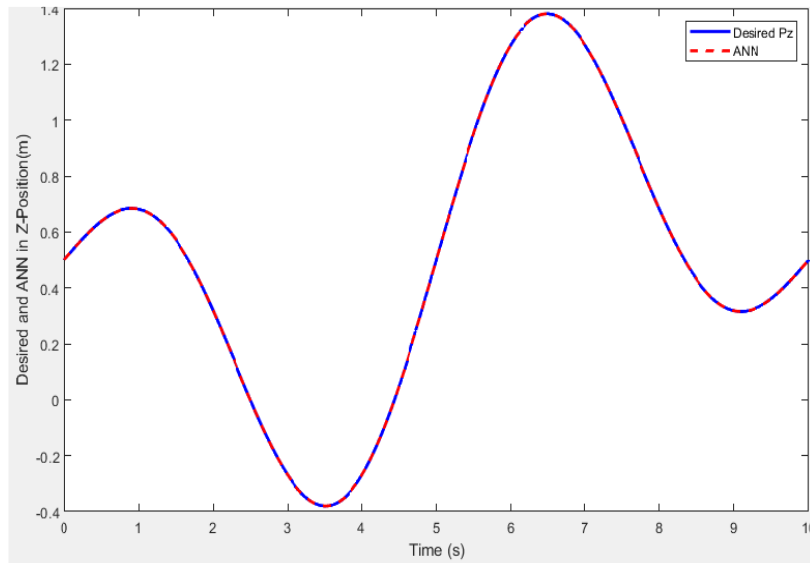


Fig. 8. Comparing the desired and estimated end-effector positions of the robot in the z-direction.

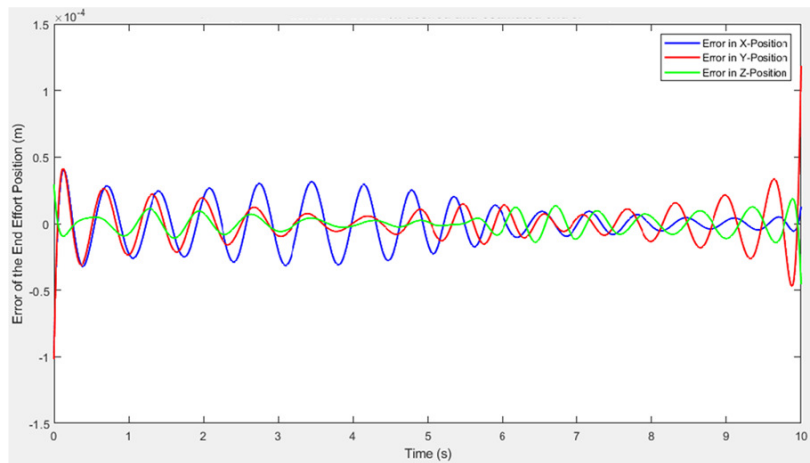


Fig. 9. The approximation error between the desired and estimated end-effector position.

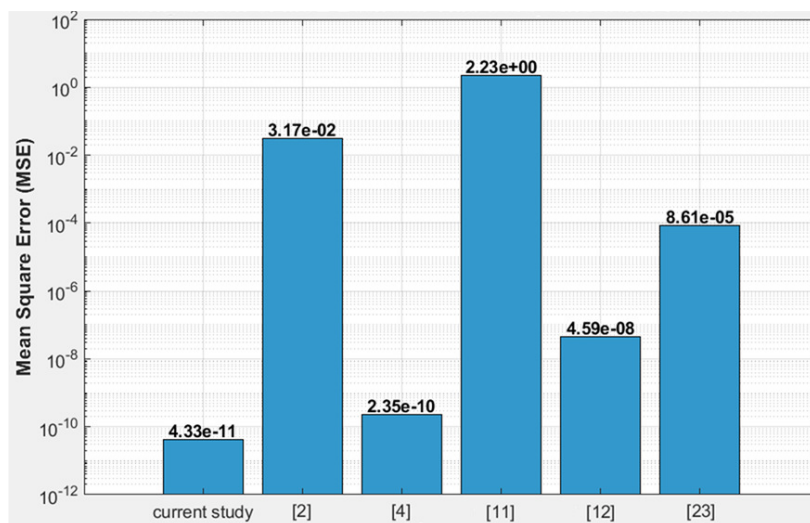


Fig. 10. Comparison of the results of this and previous studies in solving forward kinematics using an ANN.

TABLE III. RESULTS ON THE ABSOLUTE ERROR BETWEEN DESIRED AND ESTIMATED END-EFFECTOR POSITIONS

Position	Min.	Max.	Mean	Std
x	9.726948×10^{-5}	4.049119×10^{-5}	4.526177×10^{-8}	1.5840619×10^{-5}
y	1.014670×10^{-4}	1.188229×10^{-4}	2.2631995×10^{-9}	1.431889×10^{-5}
z	4.5533338×10^{-5}	2.920263×10^{-5}	1.0815529×10^{-8}	6.549562×10^{-6}

IV. DISCUSSION

This study significantly improves the reliability of the trained ANN in solving the forward kinematics of a 3D robot by accurately determining the position of the end effector with superior performance. A thorough comparison with previous studies demonstrates the effectiveness of the proposed method. Figure 10 presents the MSE results of previous studies focused on solving the forward kinematics of serial robots. A lower MSE reflects a higher performance of the trained network. The proposed method achieved the lowest MSE compared to previous studies, demonstrating its reliability and the accuracy of its results, which are very close to the actual outcomes.

The proposed MLFFNN for solving the forward kinematics of a 3-DoF manipulator demonstrates superior accuracy compared to previous ANN methods, including those in [4, 23]. It is important to note that in [2], an RNN required 200 hours of training, but still yielded subpar performance. In addition to [12], an FFNN was strategically employed to enhance the forward motion of a 3-DoF manipulator utilizing three inputs to generate 12 outputs. This complex setup reflects the intricacies of the trained network, which plays a significant role in influencing the error rate. Moreover, in [11], it was revealed that despite using backpropagation to train the network on a 2D robot, the performance results were notably the poorest. This underscores the critical need for further optimization in similar applications. The analysis focused on evaluating the performance of the trained network with an emphasis on achieving the lowest possible MSE. The findings clearly show that this study stands out, achieving the best performance with the lowest MSE, reinforcing the effectiveness of the proposed approach and highlighting its significance in the field.

V. CONCLUSIONS

This study proposed an MLFFNN to address the forward kinematics problem of a serial robot designated 3-DoF. The model takes three inputs, which represent incremental motions of the three joint variables, ranging from -180° to $+180^\circ$. The outputs correspond to the end-effector positions determined using the DH method. The proposed MLFFNN was trained using large sample data, using a few neurons to find only the position of the robot end-effector. The results clearly demonstrate that the proposed MLFFNN is highly capable of efficiently solving the forward kinematics of the manipulator, with an MSE between the desired and estimated position at 4.33×10^{-11} , which is extremely low, nearly reaching zero, ensuring precise outcomes. A huge number of samples helps the model learn accurate data representations, reducing overall error.

This study also concludes that the larger the number of samples, the more stable the statistics used during training. The strong performance confirms that the data is large, but can be learned easily with only a few neurons. The proposed method can be treated as a strategy for any similar future work, especially if the relationship between inputs and outputs is not very complex.

Future work can utilize current modeled data to assess how noise or variations in joint speeds impact forward kinematics, employ ANN modeling to determine the Jacobian matrix for a 3-DoF robot, and apply ANN modeling to solve the forward kinematics of a parallel robot.

NOMENCLATURE

MLFFNN: Multi Layer Feed Forward Neural Network

3-DOF: Three Degrees of Freedom

DH: Denavit-Hartenberg Method

RNN: Recurrent Neural Network

LM: Levenberg-Marquardt Method

MSE: Mean Squared Error

REFERENCES

- [1] L. Nguyen, R. V. Patel, and K. Khorasani, "Neural network architectures for the forward kinematics problem in robotics," in *1990 IJCNN International Joint Conference on Neural Networks*, San Diego, CA, USA, 1990, pp. 393–399 vol.3, <https://doi.org/10.1109/IJCNN.1990.137874>.
- [2] K. H. Mahmoud, A. N. Sharkawy, and G. T. Abdel-Jaber, "Development of safety method for a 3-DOF industrial robot based on recurrent neural network," *Journal of Engineering and Applied Science*, vol. 70, no. 1, Dec. 2023, Art. no. 44, <https://doi.org/10.1186/s44147-023-00214-8>.
- [3] S. K. R. Moosavi, M. H. Zafar, and F. Sanfilippo, "Forward Kinematic Modelling with Radial Basis Function Neural Network Tuned with a Novel Meta-Heuristic Algorithm for Robotic Manipulators," *Robotics*, vol. 11, no. 2, Apr. 2022, Art. no. 43, <https://doi.org/10.3390/robotics11020043>.
- [4] T. Tegegne, "Artificial Neural Network Based Position Control of Three-Degree of Freedom (3-DOF) Serial Robot," *International Journal of Scientific Research*, vol. 6, no. 3, pp. 1415–1424, 2020.
- [5] C. Cristoiu, M. Ivan, I. G. Ghionea, and C. Pupăză, "The Importance of Embedding a General forward Kinematic Model for Industrial Robots with Serial Architecture in Order to Compensate for Positioning Errors," *Mathematics*, vol. 11, no. 10, May 2023, Art. no. 2306, <https://doi.org/10.3390/math11102306>.
- [6] E. Jiménez-López, D. Servín De La Mora-Pulido, L. A. Reyes-Ávila, R. Servín De La Mora-Pulido, J. Melendez-Campos, and A. A. López-Martínez, "Modeling of Inverse Kinematic of 3-DoF Robot, Using Unit Quaternions and Artificial Neural Network," *Robotica*, vol. 39, no. 7, pp. 1230–1250, Jul. 2021, <https://doi.org/10.1017/S0263574720001071>.
- [7] M. Fouz, A. Bayoumy, and S. Rezek, "Neural-Networks-Based Inverse Kinematics for a Robotic Manipulator," *International Conference on Aerospace Sciences and Aviation Technology*, vol. 15, pp. 1–18, May 2013, <https://doi.org/10.21608/asat.2013.22084>.
- [8] M. R. Diprasetya, J. Pöppelbaum, and A. Schwung, "KineNN: Kinematic Neural Network for inverse model policy based on homogeneous transformation matrix and dual quaternion," *Robotics and Computer-Integrated Manufacturing*, vol. 94, Aug. 2025, Art. no. 102945, <https://doi.org/10.1016/j.rcim.2024.102945>.
- [9] F. Cursi, W. Bai, W. Li, E. M. Yeatman, and P. Kormushev, "Augmented Neural Network for Full Robot Kinematic Modelling in

- SE(3)," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7140–7147, Jul. 2022, <https://doi.org/10.1109/LRA.2022.3180428>.
- [10] S. Alavandar and M. J. Nigam, "Inverse Kinematics Solution of 3DOF Planar Robot using ANFIS," *International Journal of Computers, Communications & Control*, vol. 3, pp. 150–155, 2008.
- [11] R. Bouzid, H. Gritli, and J. Narayan, "Investigating Feed-Forward Back-Propagation Neural Network with Different Hyperparameters for Inverse Kinematics of a 2-DoF Robotic Manipulator: A Comparative Study," *Chaos Theory and Applications*, vol. 6, no. 2, pp. 90–110, Jun. 2024, <https://doi.org/10.51537/chaos.1375866>.
- [12] A. N. Sharkawy and S. S. Khairullah, "Forward and Inverse Kinematics Solution of A 3-DOF Articulated Robotic Manipulator Using Artificial Neural Network," *International Journal of Robotics and Control Systems*, vol. 3, no. 2, pp. 330–353, May 2023, <https://doi.org/10.31763/ijrcs.v3i2.1017>.
- [13] P. Srisuk, A. Sento, and Y. Kitjaidure, "Forward kinematic-like neural network for solving the 3D reaching inverse kinematics problems," in *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Phuket, Jun. 2017, pp. 214–217, <https://doi.org/10.1109/ECTICon.2017.8096211>.
- [14] T. P. Valayil and R. S. Augustine, "Methods to solve forward kinematics of parallel and serial manipulators," presented at the 2nd International Conference on Energetics, Civil and Agricultural Engineering 2021 (ICECAE 2021), Tashkent, Uzbekistan, 2022, Art. no. 030003, <https://doi.org/10.1063/5.0115314>.
- [15] A. Ghasemi, M. Eghtesad, and M. Farid, "Neural Network Solution for Forward Kinematics Problem of Cable Robots," *Journal of Intelligent & Robotic Systems*, vol. 60, no. 2, pp. 201–215, Nov. 2010, <https://doi.org/10.1007/s10846-010-9421-z>.
- [16] U. A. Mishra and S. Caro, "Unsupervised Neural Network Based Forward Kinematics for Cable-Driven Parallel Robots with Elastic Cables," in *Cable-Driven Parallel Robots*, vol. 104, M. Gouttefarde, T. Bruckmann, and A. Pott, Eds. Springer International Publishing, 2021, pp. 63–76.
- [17] M. Dehghani, M. Ahmadi, A. Khayatian, M. Eghtesad, and M. Farid, "Neural network solution for forward kinematics problem of HEXA parallel robot," in *2008 American Control Conference*, Seattle, WA, USA, Jun. 2008, pp. 4214–4219, <https://doi.org/10.1109/ACC.2008.4587155>.
- [18] C. S. Yee and K. Lim, "Forward kinematics solution of Stewart platform using neural networks," *Neurocomputing*, vol. 16, no. 4, pp. 333–349, Sep. 1997, [https://doi.org/10.1016/S0925-2312\(97\)00048-9](https://doi.org/10.1016/S0925-2312(97)00048-9).
- [19] Z. Geng and L. Haynes, "Neural network solution for the forward kinematics problem of a Stewart platform," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA, USA, 1991, pp. 2650–2655, <https://doi.org/10.1109/ROBOT.1991.132029>.
- [20] S. N. Yurt, E. Anli, and I. Ozkol, "Forward kinematics analysis of the 6-3 SPM by using neural networks," *Meccanica*, vol. 42, no. 2, pp. 187–196, Mar. 2007, <https://doi.org/10.1007/s11012-006-9037-3>.
- [21] L. H. Sang and M. C. Han, "The estimation for forward kinematic solution of Stewart platform using the neural network," in *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No.99CH36289)*, Kyongju, South Korea, 1999, vol. 1, pp. 501–506, <https://doi.org/10.1109/IROS.1999.813053>.
- [22] F. Tavassolian, H. Khotanlou, and P. Varshovi-Jaghargh, "Forward Kinematics Analysis of a 3-PRR Planer Parallel Robot Using a Combined Method Based on the Neural Network," in *2018 8th International Conference on Computer and Knowledge Engineering (ICCKE)*, Mashhad, Iran, Oct. 2018, pp. 320–325, <https://doi.org/10.1109/ICCKE.2018.8566243>.
- [23] R. Bouzid, J. Narayan, and H. Gritli, "Exploring artificial neural networks for the forward kinematics of a SCARA robotic manipulator using varied datasets and training optimizers," *Engineering Research Express*, vol. 6, no. 4, Dec. 2024, Art. no. 045209, <https://doi.org/10.1088/2631-8695/ad81cc>.
- [24] A. Ghobakhloo and M. Eghtesad, "Neural network solution for the forward kinematics problem of a redundant hydraulic shoulder," in *31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005.*, Raleigh, NC, USA, 2005, Art. no. 6, <https://doi.org/10.1109/IECON.2005.1569211>.
- [25] R. Kang, H. Chanal, T. Bonnemains, S. Pateloup, D. T. Branson, and P. Ray, "Learning the forward kinematics behavior of a hybrid robot employing artificial neural networks," *Robotica*, vol. 30, no. 5, pp. 847–855, Sep. 2012, <https://doi.org/10.1017/S026357471100107X>.
- [26] C. Cristoiu and A. Nicolescu, "New Approach for Forward Kinematic Modeling of Industrial Robots," *Research and Science Today*, vol. 13, 2017, Art. no. 136.
- [27] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring Strategies for Training Deep Neural Networks," *Journal of Machine Learning Research*, 2009.
- [28] F. Günther and S. Fritsch, "neuralnet: Training of Neural Networks," *The R Journal*, Jun. 2010.
- [29] M. Tiboni, G. Legnani, N. Pellegrini, and Università degli Studi di Brescia via Branze 38, Brescia 25123, Italy, "Study of Neural-Kinematics Architectures for Model-Less Calibration of Industrial Robots," *Journal of Robotics and Mechatronics*, vol. 33, no. 1, pp. 158–171, Feb. 2021, <https://doi.org/10.20965/jrm.2021.p0158>.
- [30] A. Widyacandra, A. R. Al Tahtawi, and M. Martin, "Forward and inverse kinematics modeling of 3-DoF AX-12A robotic manipulator: Pemodelan kinematika maju dan terbalik dari manipulator robot 3-DoF AX-12A," *JITEL (Jurnal Ilmiah Telekomunikasi, Elektronika, dan Listrik Tenaga)*, vol. 2, no. 2, pp. 139–150, Sep. 2022, <https://doi.org/10.35313/jitel.v2.i2.2022.139-150>.
- [31] A. Ashagrie, A. O. Salau, and T. Weldcherkos, "Modeling and control of a 3-DOF articulated robotic manipulator using self-tuning fuzzy sliding mode controller," *Cogent Engineering*, vol. 8, no. 1, Jan. 2021, Art. no. 1950105, <https://doi.org/10.1080/23311916.2021.1950105>.
- [32] T. T. K. Ly, N. T. Thanh, H. Thien, and T. Nguyen, "A Neural Network Controller Design for the Mecanum Wheel Mobile Robot," *Engineering, Technology & Applied Science Research*, vol. 13, no. 2, pp. 10541–10547, Apr. 2023, <https://doi.org/10.48084/etasr.5761>.