International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 18s 2025
https://www.theaspd.com/ijes.php

Inverse Kinematic Analysis For A 5 DOF Robotic Arm Using Deep Neural Network

Sherko A. Ibrahim [a*], Karim H. Ali [b]

[a] Department of Mechanical Engineering/College of Engineering /University of Baghdad/ Baghdad/ Iraq Email: sherko.abd2303@coeng.uobaghdad.edu.iq

[b] Mechanical Department/ Engineering College/University of Baghdad /Baghdad, Iraq Email: dr.karimhaliabood@coeng.uobaghdad.edu.iq

ABSTRACT

In this study, the kinematic analysis including the forward and inverse kinematic developed for a 5 degree of freedom robotic arm. The forward kinematic is elaborated using Denavit-Hartenberg (DH) convention. Inverse Kinematic is established using Deep Neural Network (DNN) model with five hidden layer each contain 50 neurons fully connected using ReLu activation. A data of inputs and outputs are created and trained. The inputs are the end-effector position and orientation. The outputs are the joint angles of the manipulator. The data is generated by the forward kinematics, where a set of joint angles that limited by their corresponding ranges are inserted to the forward kinematic equations to result the end-effector positions and orientations. The results of the DNN are tested and show an accuracy of the approach with a Mean Square Error of 0.0002569 which means the approach is highly accurate.

Keywords: *Neural Network, Inverse Kinematic, Forward Kinematics, Robotic arm.*

INTRODUCTION

The Robotic arm is one of the most significant of automation systems like industrial manufacturing, biomedical applications, and service robotics. Through various types of manipulators, the 5 DOF robotic arm is characterized by its simplicity and functional diversity, making it typical for executing some tasks like pick and place operations, assembly, and sorting. The design of control system for robotic arm starts with the kinematic and kinetics analysis of the robotic arm which they will lead to the dynamic model which is the fundamental of the control designing process. The kinematic analysis includes the Forward Kinematics (FK) and Inverse Kinematics (IK) which they are related to determining the relationship between the joint angles and the end-effector position and trajectory planning. The FK determines the end- effecter position and orientation depending on the joint angles of the manipulator by a method called Denavit-Hartenberg (DH) convention. While the IK determines the joint angles depending on the end- effector position and orientation. Since the relationship of the joint angle with the end-effector position and orientation are coupled nonlinearly in FK, it is difficult to find the joint angles depending on the end-effector position and orientation [1]. Accordingly, different methods are used for solving the IK problem. In [2, 3], metaheuristic optimization techniques used for solving IK, these techniques are slow in real time application and need a fine initial guess also it some time stuck in local minimum which will not be the best solution. In [4, 5], geometric approach is used for inverse kinematic, this method offers an accurate result, but it depend on a sequence of equations which will increase the elapsed time of the computation, this will effect in the real-time application. In [6], the IK for a 3 DOF robotic arm is determined using multilayer feedforward neural network (MLFFNN), the data is generated for a sinusoidal of the joint angles and inserted to FK to result the end-effecter position and orientation, however, the result of the proposed model shows a high reliability and robustness, the author recommend to apply the proposed method to a higher

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 18s 2025
https://www.theaspd.com/ijes.php

degree of freedom in which will be in this study. In [7], IK for a 2 DOF was implemented using Closed- form solution which based on robot geometry which result a polynomial where used if the DOF is less than 4. In [8], iterative method was presented for IK of a 2 DOF and 3 DOF planer robotic arm the proposed method is simple and accurate, but it take several iteration to reach an accurate result which will effect in real- time application. In [9], Artificial Neural Network (ANN) used for IK for a 5 DOF robotic arm but the inputs for the proposed model are the end-effector position only while it is better to insert the orientation of the end-effector to the model.

In this study, the IK for a 5 DOF robotic arm is accomplished using Neural Network (NN) approach and validated practically. First, a data is generated from the FK then the generated date inserted to the NN to be trained where the inputs are the end-effecter position and orientations while the outputs are the corresponding joint angles. Once the training process finished, an NN solver created which will be ready to be used for simulation or in real-time application. The NN method overcome any traditional methods by its faster computation with high accuracy.

FORWARD KINEMATIC

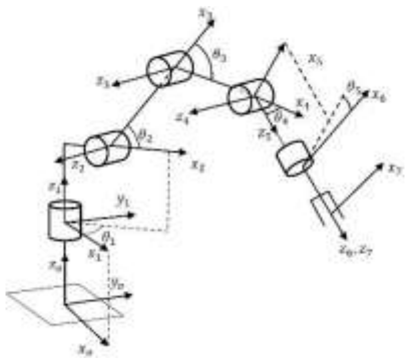
The forward kinematic means find the relationship between the joint angles which are inputs and the end-effector position and orientation which are the output [10]. In this study, the model of the robotic arm is shown in Figure 1, consisting of 5 revolute joints and a gripper which represent the end-effector of the robot. This type of robotic arm is usually used for light–industrial applications and as an educational tool. The first step in deriving the kinematic analysis is the Denavit-Hartenberg (DH) Conventions, which systematically describe the robotic arm's geometry. It [5]requires attaching a local coordinate frame to each joint and four parameters, which they abbreviated as (L_i, a_i, θ_i, and α_i) giving the location of link i relative to link (i -1) by the following transformation matrix:

$$T_i^{i-1} = \begin{bmatrix} cos\theta_i & -sin\theta_i cos\alpha_i & sin\theta_i sin\alpha_i & a_i cos\theta_i \\ sin\theta_i & cos\theta_i cos\alpha_i & -cos\theta_i sin\alpha_i & a_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & L_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Where the first three rows of the last coulomb represent the x, y, z coordinate of frame coordinate i relative to $i - 1$. The coordinate frames are attached to each joint as shown in Figure 1.c. Despite the robotic arm has 5 revolute joint, seven coordinate frames are assigned. At joint 4, two coordinate frame are assigned, this is because the rotation of joint 5 are orthogonal to those of joint 4, so to align the frame coordinate of joint 4 to the joint 5, the frame coordinate 4 is rotated about z_4 by an angle (90 + θ_4) which will make x_4 aligned to $x5$ while $z4$ still perpendicular to $z5$, then frame 4 is rotated about $x5$ which will make $z4$ aligned to z_5, after this, the frame coordinate 5 moves along z_5 to the frame coordinate 6 which it is belong to joint 5, this make the frame coordinates suitable for the DH Conventions. However, the parameters of the DH Conventions are listed in table (1). By this table, the transformation matrix of each link will be found.

(a) (b)

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 18s 2025
https://www.theaspd.com/ijes.php



(c)

Figure (1) The model of the robotic arm (a) 3D view (b) Side view (c) Schematic diagram

Table(1) DH parameters of the robotic arm

Frame	d_i(mm)	$a_i(mm)$	θ_i(degree)	α_i(degree)
1	L_o=75	0	0	0
2	A=14	B=12	θ_1	90
3	0	L_1 =102	θ_2	0
4	0	L_2 =130	θ_3	0
5	0	0	90+θ_4	90
6	L_3 =68	0	θ_5	0
7	L_4 =105	0	0	0

To find the end-effector position relative to the base coordinate, the transfer matrices of the whole links are multiplied as follows:

$T^0_7 = T^0_1 T^1_2 T^2_3 T^3_4 T^4_5 T^5_6 T^6_7$	(2)
$T^0_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 80 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	(3)
$T^1_2 = \begin{bmatrix} cos\theta_1 & 0 & sin\theta_1 & 12 * cos\theta_1 \\ sin\theta_1 & 0 & -cos\theta_1 & 12 * sin\theta_1 \\ 0 & 1 & 0 & 14 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	(4)
$T^2_3 = \begin{bmatrix} cos\theta_2 & -sin\theta_2 & 0 & 105 * cos\theta_2 \\ sin\theta_2 & cos\theta_2 & 0 & 105 * sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	(5)
$T^3_4 = \begin{bmatrix} cos\theta_3 & -sin\theta_3 & 0 & 145 * cos\theta_3 \\ sin\theta_3 & cos\theta_3 & 0 & 145 * sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	(6)
$T^4_5 = \begin{bmatrix} \cos(90+\theta_4) & 0 & \sin(90+\theta_1) & 0 \\ sin(90+\theta_4) & 0 & -cos(90+\theta_4) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	(7)

$T_6^5 = \begin{bmatrix} cos\theta_5 & -sin\theta_5 & 0 & 0 \\ sin\theta_5 & cos\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 75 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	(8)
$T_7^6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 100 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	(9)

The resulted transformation matrix in equation(2) will give the position coordinate and orientation of the end-effector relative to the base coordinate as the following form:

$T_7^0 = \begin{bmatrix} R_{xx} & R_{yx} & R_{zx} & P_x \\ R_{xy} & R_{yy} & R_{zy} & P_y \\ R_{xz} & R_{yz} & R_{zz} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$	(10)

R_{xx}, R_{xy}, R_{xz} represent the orientation of the local x axis of the end-effector to the global x_0, y_0, z_0 axis respectively, R_{yx}, R_{yy}, R_{zz} represent the orientation of the local y axis of the end-effector to the global x, y, z axis respectively, R_{zx}, R_{zy}, R_{zz} represent the orientation of the local z axis of the end-effector to the global x, y, z axis respectively, P_x represent the coordinate of the global x axis of the end-effector, , P_y represent the coordinate of the global y axis of the end-effector and , P_z represent the coordinate of the global z axis of the end-effector. In the result the position of the end-effector represent in P_x, P_y, P_z while its orientation represent in $R_{xx}, R_{xy}, R_{xz}, R_{yx}$, R_{yy},
R_{zz}, R_{zx}, R_{zy}, and R_{zz}. The orientation can be transformed to Euler angles γ, β, ϕ which stand for the rotation about global x_0, y_0, z_0.

INVERSE KINEMATIC

The next subject of the kinematic analysis of the robotic arm is the Inverse Kinematics, which is the opposite of the forward kinematics. It computes the joint angles at a desired position and orientation of the end- effector. The challenging matter in IK is the difficulty of extracting joint angles from the FK and formulate equations of joint angles as a function of the end-effector position and orientation. Some methods are utilized such as geometric method, but it will lead to a set of equations that will be solved numerically and iteratively, which will take time and will be unsuitable for real-time applications. So, for more effective and faster execution, a Neural Network is used to create a solver that takes care of the inverse kinematic problem. A data is generated from the forward kinematics in which a large set of joint angles $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$ are inserted into the forward kinematics to result in the end-effector position coordinate (P_x, P_y, P_z) and orientation $(\gamma, , \beta, \phi)$, but the resulted data contain duplicate solution to visualize this a sinusoidal data of the joint angles are inserted to the FK and plot the results as shown in figure (2).

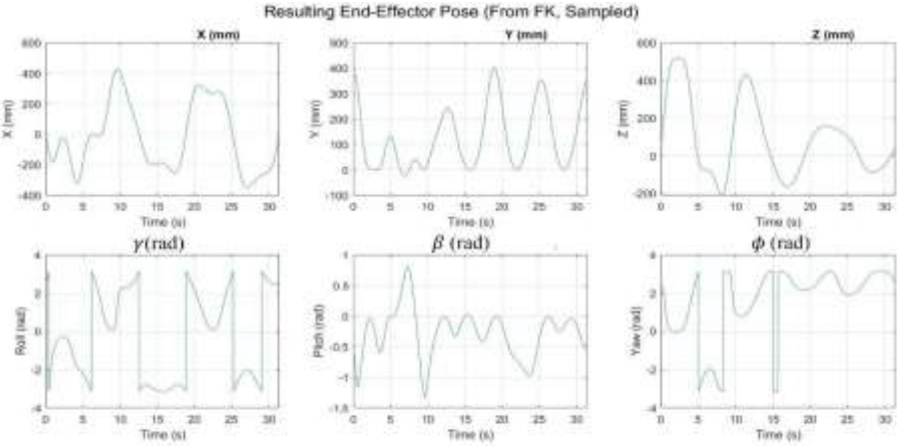


Figure (2) end-effector position and orientation (γ, β, ϕ) resulted from FK by sinusoidal joint angles

Repeated or duplicated values appear in γ and ϕ for example at time 15 seconds ϕ has two values 2.5 rad and -2.5 rad this will affect on the training process in the DNN and will result in inaccurate model because DNN will take the average for the two values. To overcome this problem, it is suggested to generate a data of joint angles ($\theta_1, \theta_2, \theta_3, \theta_4$) while maintaining θ_5 constant as it will not affect on the end-effector position then insert them to FK to result end-effecter position while the orientation is considered as follows:

$$\boldsymbol{\psi} = \theta_2 + \theta_3 + \theta_4 \tag{10}$$

Where $\boldsymbol{\psi}$ refer to the angle of the last link with the horizon. A sinusoidal joint angle are generated and inserted to FK to visualize the results and check weather it will show repeated solution or not, the results shown in figure (3). It is shown that the end-effector position and orientation has a smooth behavior and does not contain repeated values or discontinuity. This make sure that the training process in the DNN will run smoothly without any estimation of the repeated values.

Then the resulting data is inserted into a Neural Network model, the input data of the model is the end- effector position coordinate and the orientation, and the output data is the joint angles. After the training was completed, the model is ready for any input of end-effector coordinates and orientation to result in the joint angles within a very short time compared to other methods.

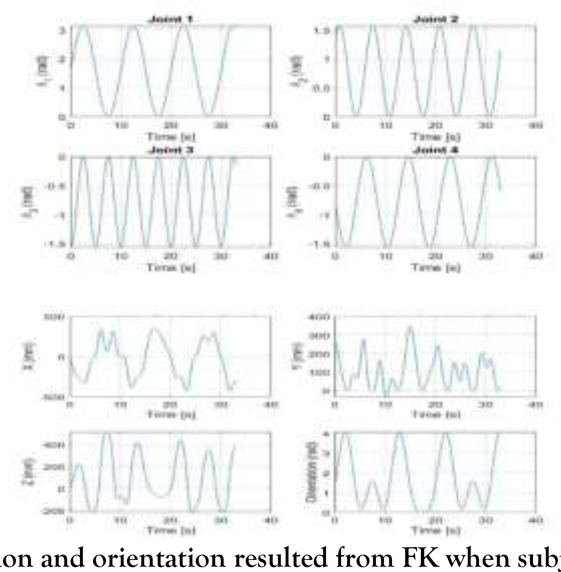


Figure (3) end-effector position and orientation resulted from FK when subjected to sinusoidal joint angle and keeping θ_5 constant

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 18s 2025
https://www.theaspd.com/ijes.php

NEURAL NETWORK (NN)

The basic idea of NN is to train a model on a data of inputs and outputs, then the model will create a solver that able to map any inputs to result outputs. There are several types of NN, in this study the Feedforward NN (FNN) with a Multilayer Perceptron (MLP) is utilized. First a data of inputs which are joint angles with the corresponding end-effector position and orientation are generated by FK and validated practically by real-time robotic arm. Then they will be inserted to the FNN training model which it consists input layer, hidden layer, output layer. Each layer consists neurons where the input layer has 6 neurons of $P_x, P_y, P_z,$
γ, ψ, ϕ, the hidden layer has 5 layers where each layer has 50 neurons and the output layer has 4 neurons stands for $\theta_1, \theta_2, \theta_3$ and θ_4 as shown in figure (2)

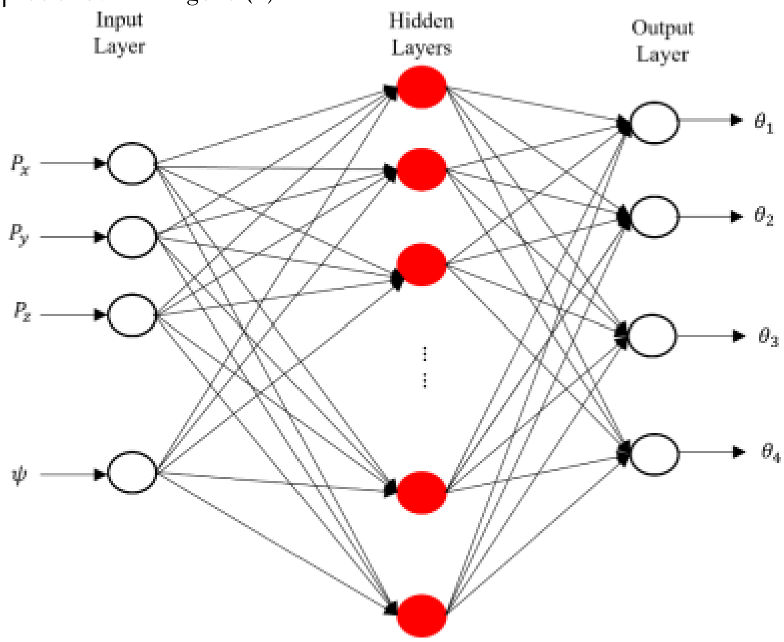
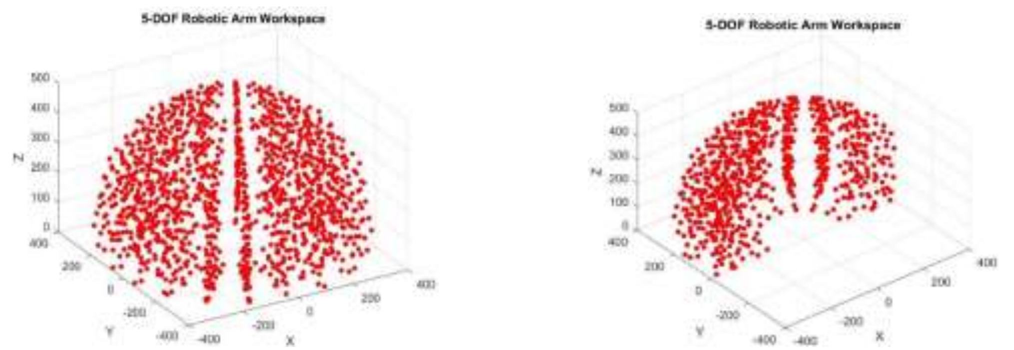


Figure (4) Deep Neural Network (DNN) Architecture

RESULT AND DISCUSSION

A set of joint angles are applied to FK to result the end-effector position and orientation. Two sets of data are generated as shown in Figure (5, a,b) the difference between them is in the joint angle1 (θ_1), that in figure (5,a) the range of θ_1 is from 0^o to 360^o while in figure (5,b) the range is from from 0^o to 180^o. This difference result in a similarity about the y axis this properity is useful for the training DNN where
only half of the data are trained and when it is demanded the other half a simple trick is used illustrated in figure (6). A data of 351520 is generated for joint1, joint2, joint3, joint4, orientation and the end-effector position. And for simplcity joint5 is considered fixed because its movement will not effect on the end- effector position. Since the data collected is very lage, only 10 samples are tabulated in this article as shown in table (2)



(a) (b)

Figure (5) work space of the robotic arm (a) when θ_1 is from 0^o to 360^o (b) when θ_1 is from 0^o to 360^o.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 18s 2025
https://www.theaspd.com/ijes.php

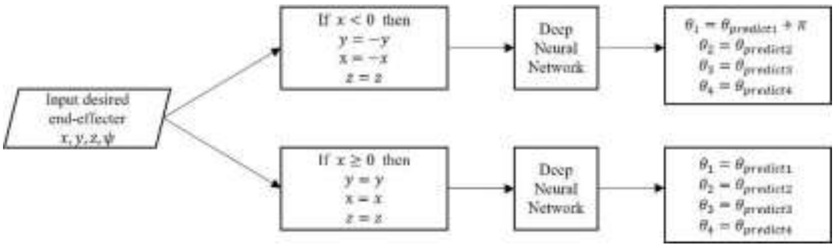


Figure (6) Flow chart of the IK solver

Table(3) Sample of the results of model2

No.	Sample of joint angles	End-effector position and orientation using FK	Predicted Joint angles	End-effector position and orientation for Joint angles from NN using FK	Absolute error of the joint angles	Absolute error of end-effector position and orientation
1	$\theta1 = 0$ $\theta2 = 0$ $\theta3 = 0$ $\theta4 = 0$	$Px = 437$ $Py = 0$ $Pz = 94$ $\psi = 0$	$\theta pred1 = -0.0071$ $\theta pred2 = 0.0870$ $\theta pred3 = -0.0673$ $\theta pred4 = -0.0882$	$Px = 428.0773$ $Py = -6.5568$ $Pz = 64.0969$ $\psi = 0$	$\Delta\theta1$ =0.0071 $\Delta\theta2$ =0.0870 $\Delta\theta3$ =0.0673 $\Delta\theta4$ =0.0882	ΔPx =0.8459 ΔPy =3.0908 ΔPz =0.0318 $\Delta\psi$ = 0.0127
2	$\theta1 = \pi/6$ $\theta2 = \pi/12$ $\theta3 = -\pi/12$ $\theta4 = -\pi/6$	$Px = 355.0502$ $Py = 204.9883$ $Pz = 33.6760$ $\psi = 0.5236$	$\theta pred1 = 0.5061$ $\theta pred2 = 0.2376$ $\theta pred3 = -0.2199$ $\theta pred4 = -0.5256$	$Px = 360.3042$ $Py = 199.6821$ $Pz = 36.1805$ $\psi = 0.5238$	$\Delta\theta1$ = 0.0175 $\Delta\theta2 = 0.0242$ $\Delta\theta3 = 0.041$ $\Delta\theta4$ =0.0020	ΔPx =5.2540 ΔPy =5.3062 ΔPz =2.5045 $\Delta\psi$ = 0.0002
3	$\theta1 = \pi/3$ $\theta2 = \pi/6$ $\theta3 = -\pi/6$ $\theta4 = -\pi/6$	Px =199.7436 $Py = 345.966$ $Pz = 59.0000$ $\psi = 1.0472$	$\theta pred1 = 1.0505$ $\theta pred2 = 0.4956$ $\theta pred3 = -0.4842$ $\theta pred4 = -0.5502$	$Px = 198.6320$ $Py = 346.6905$ $Pz = 55.7923$ $\psi = 1.0619$	$\Delta\theta1$ =0.0033 $\Delta\theta2 = 0.0280$ $\Delta\theta3 = 0.039$ $\Delta\theta4$ =0.0266	ΔPx =1.1116 ΔPy =0.7245 ΔPz =3.2077 $\Delta\psi$ = 0.0147
4	$\theta1 = \pi/2$ $\theta2 = \pi/4$ $\theta3 = -\pi/4$ $\theta4 = -\pi/6$	$Px = 0$ $Py = 382.800$ $Pz = 80.7462$ $\psi = 1.5708$	$\theta pred1 = 1.5613$ $\theta pred2 = 0.7532$ $\theta pred3 = -0.7389$ $\theta pred4 = -0.5431$	$Px = 3.6402$ $Py = 384.6595$ $Pz = 79.5894$ $\psi = 1.5756$	$\Delta\theta1$ =0.0095 $\Delta\theta2 = 0.0322$ $\Delta\theta3 = 0.046$ $\Delta\theta4$ =0.0195	ΔPx =3.6402 ΔPy =1.8589 ΔPz =1.1568 $\Delta\psi$ = 0.0048
5	$\theta1 = 2\pi/3$ $\theta2 = \pi/3$ $\theta3 = -\pi/4$ $\theta4 = -\pi/3$	$Px = -164.1515$ $Py = 284.3187$ $Pz = 98.7177$ $\psi = 2.3562$	$\theta pred1 = 2.0861$ $\theta pred2 = 1.0244$ $\theta pred3 = -0.7500$ $\theta pred4 = -1.0581$	$Px = -162.668$ $Py = 287.2219$ $Pz = 99.4789$ $\psi = 2.3606$	$\Delta\theta1 = 0.0083$ $\Delta\theta2 = 0.0228$ $\Delta\theta3 = 0.0354$ $\Delta\theta4 = 0.0109$	ΔPx =1.4835 ΔPy =2.9032 ΔPz =0.7612 $\Delta\psi$ = 0.0044
6	$\theta1 = \pi$ $\theta2 = 0$ $\theta3 = 0$ $\theta4 = 0$	$Px = 437$ $Py = 0$ $Pz = 94$ $\psi = 0$	$\theta pred1 = -0.0071$ $\theta pred2 = 0.0870$ $\theta pred3 = -0.0673$ $\theta pred4 = -0.0882$	$Px = 428.0773$ $Py = -6.5568$ $Pz = 64.0969$ $\psi = 0$	$\Delta\theta1$ =0.0071 $\Delta\theta2$ =0.0870 $\Delta\theta3$ =0.0673 $\Delta\theta4$ =0.0882	ΔPx =0.8459 ΔPy =3.0908 ΔPz =0.0318 $\Delta\psi$ = 0.0127
7	$\theta1 = \pi + \pi/6$ $\theta2 = \pi/12$ $\theta3 = -\pi/12$ $\theta4 = -\pi/6$	$Px = -355.050$ $Py = -204.9883$ $Pz = 33.6760$ $\psi = 0.5236$	$\theta pred1 = 0.5061$ $\theta pred2 = 0.2376$ $\theta pred3 = -0.2199$ $\theta pred4 = -0.5256$	$Px = -360.304$ $Py = -199.682$ $Pz = 36.1805$ $\psi = 0.5238$	$\Delta\theta1$ = 0.0175 $\Delta\theta2 = 0.0242$ $\Delta\theta3 = 0.041$ $\Delta\theta4$ =0.0020	ΔPx =5.2540 ΔPy =5.3062 ΔPz =2.5045 $\Delta\psi$ = 0.0002

The result of the DNN is tested also on prescribed path as shown in figure (5) where the differences between the prescribed path i.e. True, and the predicted is very small this proves the accuracy of the model.

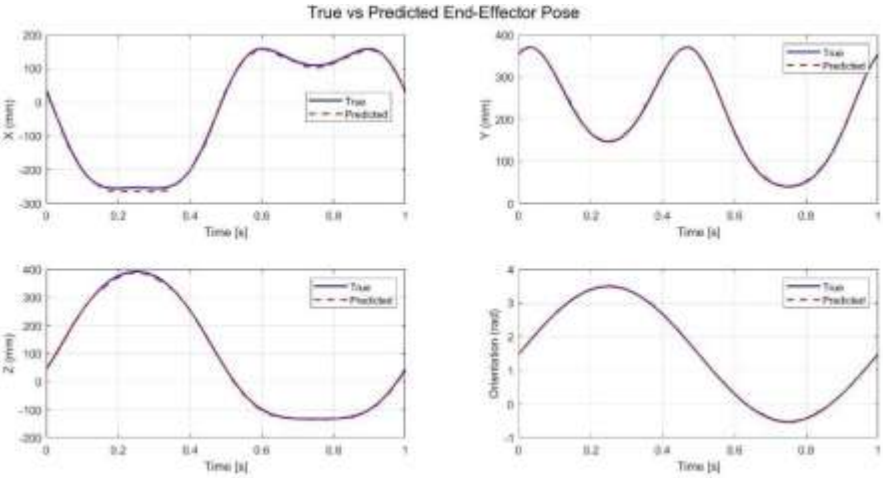


Figure (5) True vs. predicted end effector position and orientation

REFERENCES

1. Hajar Abd Al-Sattar Ali and Hatem H. Obeid, "Experimental and simulation study of a four-degree of freedom robot arm moving through space planner path," International Journal of Computational Vision and Robotics, vol. 14, no. 6, pp. 631 - 653, 2024.
2. Manpreet Kaur, Venkata Karteek Yanumula & Swati Sondhi , "Trajectory planning and inverse kinematics solution of Kuka robot using COA along with pick and place application," Intelligent Service Robotics , vol. 17, p. 289–302, 2024.
3. Guojun Zhao, Ying Sun, Du Jiang, Xin Liu, Bo Tao, Guozhang Jiang, Jianyi Kong, Juntong Yun, Ying Liu, and Gongfa Li, "A 7DOF redundant manipulator inverse kinematic solution algorithm based on bald eagle search optimization algorithm," Soft Computing, vol. 28, p. 13681–13699, 2024.
4. Angad Alnoor Saad Bader Alawad and Ahmed Abdul Hussein Al, "Simulation and The Theoretical of Forward Kinematics and Inverse," Journal of Mechanical Engineering Research and Developments, vol. 44, no. 8, pp. 457-466, 2021.
5. Tahseen F. Abaas, Ali A. Khleif, and Mohanad Q. Abbood, "Kinematics Analysis of 5 DOF Robotic Arm," Tahseen F. Abaas a, Ali A. Khleif b , Mohanad Q. Abboodc*, vol. 38, no. 3, pp. 412-422, 2020.
6. Abdel-Nasser Sharkawy and Shawkat Sabah 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, 2023.
7. Ya'nan Lou, Pengkun Quan, Haoyu Lin, Dongbo Wei and Shichun , "A closed-form solution for the inverse kinematics of the 2n-dof hyper-redundant manipulator based on general spherical joint," Applies sciences, vol. 11, no. 3, p. 1–19, 2021.
8. A. O. Elnady, "Iterative Technique for Solving the Inverse Kinematics Problem of Serial Manipulator," Journal of Mechanical Engineering and Automation, vol. 10, no. 1, pp. 12-18, 2021.
9. A. E. TIRYAKI, "ARTIFICIAL NEURAL NETWORK APPLICATION FOR INVERSE KINEMATIC SOLUTION OF A 5-DOF ROBOTIC ARM," EJONS International Journal on Mathematic, Engineering and Natural Sciences, vol. 4, no. 16, p. 965–977, 2020.
10. R. Singh, V. Kukshal, and V. S. Yadav, "A review on forward and inverse kinematics of classical serial manipulators," Lecture Notes in Mechanical Engineering, p. 417–428, 2021.
11. Mohd Riduwan Ghazali, Mohd Ashraf Ahmad, Noor Asma' Shahirah Yuseri and Mohd Helmi Suid, "Development and Control of a 5-DOF Robotic Arm Using Forward Kinematics and IoT Integration," Proceedings of the 2024 and International Conference on Robotics, Control and Vision Engineering, pp. 18-23, 2024.