ADAPTIVE ROBOTIC ARM CONTROL USING ARTIFICIAL NEURAL NETWORK

Pre- Project (EEP401) report submitted in partial fulfilment of the requirements for the award of the degree of

## Bachelor of Technology in Electrical Engineering

By

SHUBHAM SAIN (2020BELE079)

KAJAL KUMARI (2020BELE113)

RISHIB BAJGOTRA (2020BELE013)

Under the guidance

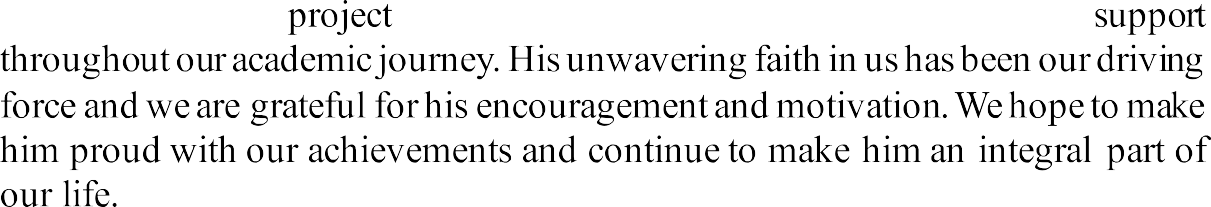
of

Prof. M. A. Bazaz

## Department of Electrical Engineering National Institute of Technology Srinagar

## Srinagar, Jammu and Kashmir - 190006 (India)

**Dedication**



**Declaration**

We hereby certify that the work which is being presented in the thesis entitled " **Adaptive Robotic Arm Control using Artificial Neural Network**" submitted by us in partial fulfilment of the project to Department of Electrical Engineering at National Institute of Technology Srinagar, J&K is an authentic record of our own work carried out during a period from August 2023 to January 2024 under the supervision of **Prof. M. A. Bazaz**, Department of Electrical Engineering at National Institute of Technology Srinagar. We declare that this written submission represents our ideas in our own words and where in other ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea, data, fact or source in our submission. It is further certified that the work presented in this dissertation has not been submitted elsewhere for the award of any degree.

2020BELE079 Shubham Sain

2020BELE113 Kajal Kumari

2020BCELE013 Rishib Bajgotra

Date: 26th Feb, 2024

Place: NIT Srinagar

This is to certify that the above statement made by the candidates are correct to the best of my knowledge.

Signature of Supervisor

**ABSTRACT**

This study presents the design and implementation of an artificial neural network (ANN) controller for a 4-degree-of-freedom (4-DOF) robotic arm, integrating kinematic modeling with the Denavit-Hartenberg (DH) convention and forward kinematics. The Denavit-Hartenberg parameters are utilized to accurately represent the arm's geometry and kinematics, facilitating precise end-effector positioning through successive joint transformations. Forward kinematics is employed to determine the arm's spatial configuration from joint angles. The ANN controller optimizes motion planning by mapping desired end-effector positions to corresponding joint angles, enhancing precision and adaptability. This integration of DH convention, forward kinematics, and neural networks empowers the robotic arm to execute complex tasks with improved accuracy and flexibility. Experimental results demonstrate the efficacy of the proposed approach, showcasing its potential for advancing automation capabilities in various fields.

**Acknowledgement**

**First of all, we would be thankful to God for our health and happiness, which made us capable to partial complete this dissertation work. We'd like to express our sincere gratitude to one and all who helped us with our project. We would like to say thanks to our supervisor,** Prof. M. A. Bazaz**, to give suggestions, lavish encouragement, brilliant ideas and instructions. Without him, completing this work would be difficult. We counted ourself really lucky person that we have a supervisor who was genuinely interested to complete this project work. I wholeheartedly acknowledge Dr. J. I. Bhatt, Prof. Head of electrical Engineering Department, NIT Srinagar for providing me all the facilities. We believe no words can describe the amount of patience, understanding, and sacrifices made by our loving parents through this work as such. We also want thank our friends who constantly gave me support and helped us whenever needed during this work. At last, we want to convey our appreciation to one and all who helped us to achieve life milestone.**

**Shubham Sain**

**Kajal Kumari**

**Rishib Bjagotra**

**CONTENTS**

1. Introduction

1.1 Motivation

2. LITERATURE REVIEW

2.1 OBJECTIVES

3.1 SETUP AND CONFIGURATION OF THE ROBOTIC ARM

3.2 KINEMATICS MODELLING OF A 4-DOF ROBOTIC ARM

3.2.1 DH Convention

3.2.2 Forward KINEMATICS

3.3 THE ROBOTIC ARM CONTROLLER

3.3.1 Input and Output Parameters

3.3.2 The Approximated Jacobian Matrix

3.3.3 Training of Artificial Neural Network (ANN)

4. Result

5. CONCLUSION AND FUTURE WORKS

6. REFERENCES

**1. INTRODUCTION**

Robots have found diverse applications and one of the most important is protecting human lives. Robotic arms play a pivotal role in modern automation, facilitating tasks ranging from industrial manufacturing to medical surgeries. Robotic arms are necessary for remote manipulation of objects and so accurate control is important. Robotic arms are highly nonlinear systems hence analytical and conventional control methods may not be applicable in these cases. This problem is even enhanced if robotic arm have to carry varying payload weights. Using adaptive control methods such as artificial neural networks we can solve the problem. Achieving precise control and adaptability in robotic arm manipulation is essential for enhancing efficiency and expanding application domains. So we are presenting the design and implementation of an artificial neural network (ANN) controller for a 4-degree-of-freedom (4-DOF) robotic arm. This controller integrates kinematic modeling techniques, including the Denavit-Hartenberg (DH) convention and forward kinematics. By leveraging DH parameters, the geometric and kinematic properties of the robotic arm are accurately represented, enabling precise end-effector positioning through sequential joint transformations. Additionally, forward kinematics is employed to determine the spatial configuration of the arm from joint angles. The ANN controller enhances motion planning by mapping desired end-effector positions to corresponding joint angles, thereby optimizing control and adaptability. This report details the methodology, implementation, and experimental results of the proposed approach, highlighting its potential contributions to advancing automation capabilities in diverse fields.

**1.1 MOTIVATION**

The complexity of real-world robotic systems often introduces nonlinear-behavior that pose challenges for precise control and manipulation. Traditional methods for controlling robotic arms, while effective in some scenarios, may struggle to handle these nonlinearities adequately. As such, there is a pressing need for innovative control strategies that can effectively address the nonlinear behavior of robotic arms, ensuring accurate and adaptable performance across various tasks.

The utilization of kinematic modeling techniques, such as the Denavit-Hartenberg (DH) convention and forward kinematics, provides a solid foundation for understanding the geometric and kinematic properties of robotic arms. However, while these methods offer valuable insights into the arm's configuration, they may not fully address the complexities introduced by nonlinear behaviors.

The integration of artificial neural networks (ANNs) presents a promising approach to mitigate the challenges posed by nonlinearities in robotic arm control. ANNs have demonstrated remarkable capabilities in approximating complex nonlinear mappings, making them well-suited for modeling and controlling systems with intricate dynamics.

In this context, the motivation behind our work lies in the development of an artificial neural network controller for a 4-degree-of-freedom (4-DOF) robotic arm. By combining kinematic modeling with the DH convention and forward kinematics, alongside the power of neural networks, we aim to provide a solution that effectively addresses the nonlinear behavior of robotic arms. This approach holds the potential to enhance the arm's precision, adaptability, and overall performance, thereby advancing automation capabilities in various industries and research domains. **2. LITERATURE REVIEW**

There are two methods employed for kinematic modelling of a 4-DOF Robotic Arm:

1. DH Convention
2. Forward Kinematics
3. DH Convention:

The Denavit-Hartenberg (DH) convention is a method to model the kinematics of robotic arms. It uses four parameters: link length (ai), link offset (di), link twist (αi), and joint angle (θi). These parameters define the transformation between adjacent coordinate frames, enabling precise representation of the arm's geometry and motion capabilities.

1. Forward Kinematics:

Forward kinematics calculates end-effector position and orientation based on joint angles, using geometric and trigonometric equations for each link.

*Problem Statement:* Robotic arms are highly nonlinear systems hence analytical and conventional control methods may not be applicable in these cases. This problem is even enhanced if robotic arm have to carry varying payload weights.

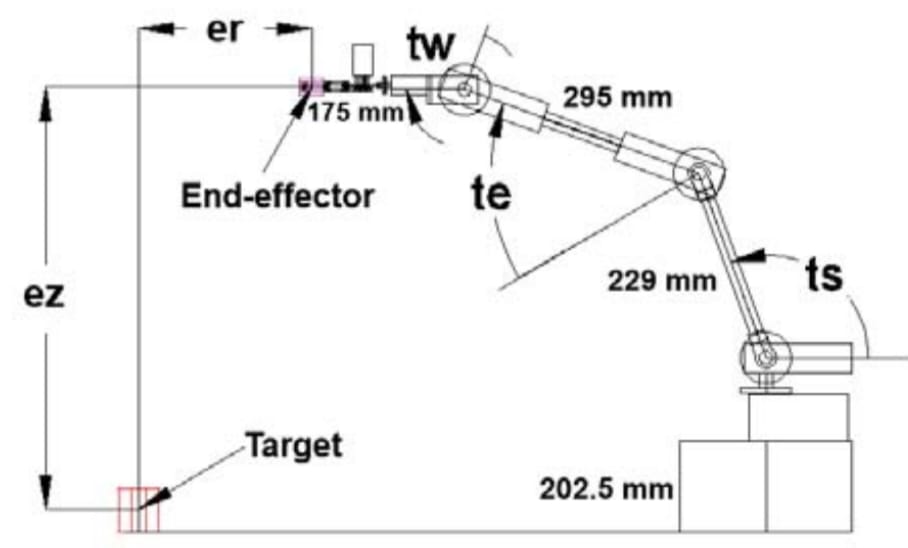
*Solution:* Using adaptive control methods such as artificial neural networks we can solve the problem. Achieving precise control and adaptability in robotic arm manipulation is essential for enhancing efficiency and expanding application domains. So we are presenting the design and implementation of an artificial neural network (ANN) controller for a 4-degree-of-freedom (4-DOF) robotic arm.

**2.1 Objective**

* Develop a comprehensive kinematic model of a 4-degree-of-freedom (4-DOF) robotic arm using the Denavit-Hartenberg (DH) convention and forward kinematics, accurately representing its geometric and kinematic properties.
* Design and implement an artificial neural network (ANN) controller capable of effectively mitigating the nonlinear behavior exhibited by the robotic arm during manipulation tasks.
* Train the ANN controller using appropriate datasets to learn the mapping between desired end-effector positions and corresponding joint angles, optimizing motion planning and control.
* Evaluate the performance of the developed neural network controller through simulation and experimental validation, assessing its ability to accurately predict and control the arm's movements under various conditions.
* Compare the performance of the ANN controller with traditional control methods to highlight its effectiveness in addressing nonlinearities and improving the overall precision and adaptability of the robotic arm.

**3.1 SETUP AND CONFIGURATION OF THE ROBOTIC ARM**

For this study we are using a 4-DOF robotic arm that uses a 2-DOF gripper as end-effector. The robotic arm will be mounted to a bomb disposal robot and should be able to lift varying weights and adapt to different mobile platform orientations. Fig shows the link dimensions as well as the relative position of the joints. There are four base angles that needs to be controlled, namely: base(, shoulder(, elbow( and wrist(. Altogether, the values of this joint angles will influence the coordinates of the end-effector relative to the base. The angular arrowhead for each joint indicates the direction of rotation as the pulse width modulation (PWM) value is increased and the line indicates the reference angle. This study need not be concerned on the forward kinematics equations of the robotic arm but rather on the data that describes its response under different joint angle values.

****

**3.2 KINEMATICS MODELLING OF A 4-DOF ROBOTIC ARM**

A 4-degree-of-freedom (4-DOF) robotic arm is a mechanical manipulator with four joints that allow movement in four independent directions. Each joint provides a degree of freedom, enabling rotation or pivoting along a specific axis. These robotic arms are commonly used in various applications such as assembly lines, material handling, pick-and-place tasks, and research laboratories. The 4-DOF configuration provides a balance between simplicity and versatility, allowing the arm to perform a wide range of tasks while remaining relatively straightforward to control and program.

**3.2.1 DH Convention**

Denavit-Hartenberg (DH) convention is commonly used in the kinematics analysis of the robotic manipulator. It is based on attaching a coordinate frame at each joint and specifying four parameters known as DH parameters for each link, and utilizing these parameters to construct a DH table. Finally, a transformation matrix between different coordinate frames is obtained. The major objective is to control both the position and orientation of the EE or the gripper in its work space. We will first derive the relationship between the joint variables and the position, and orientation of the gripper, using the DH method.

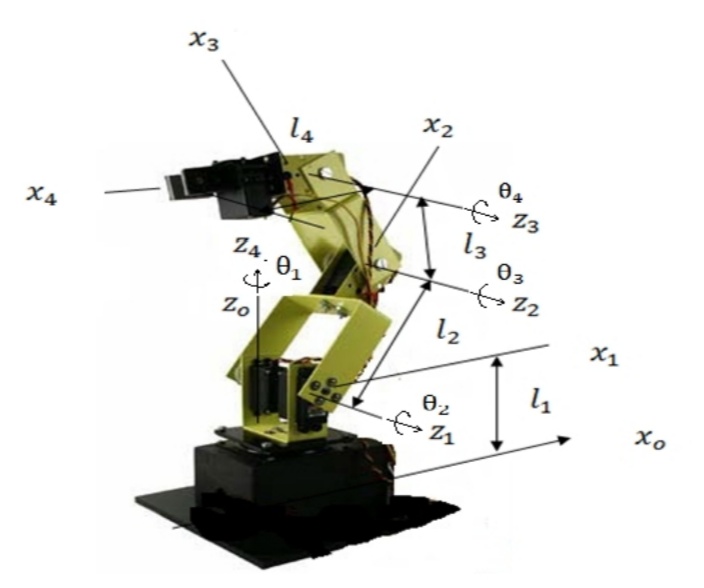


Fig: DH Parameters

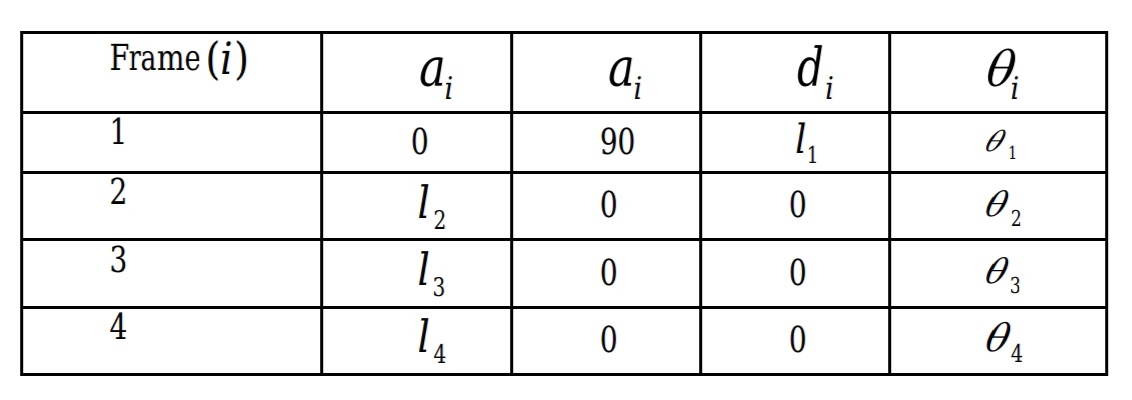


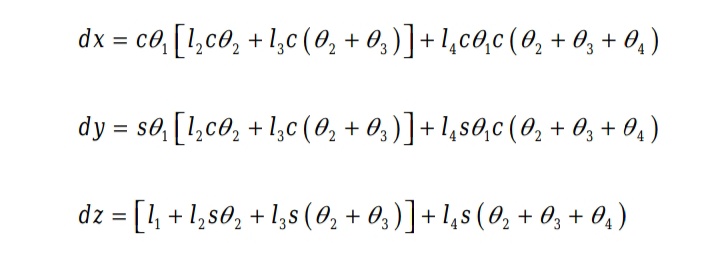
Table I: DH Parameters

**3.2.2 Forward Kinematics**

In the context of kinematic modeling of a 4-degree-of-freedom (4-DOF) robotic arm, forward kinematics is a method used to determine the position and orientation of the end-effector (e.g., tool or gripper) based on the joint angles of the robotic arm.

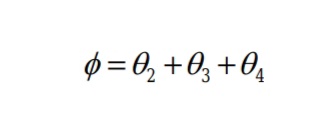
For a 4-DOF robotic arm, forward kinematics involves establishing a sequence of transformations from the base to the end-effector, typically using homogeneous transformation matrices. Each joint's transformation matrix represents the translation and rotation caused by that joint's motion. By multiplying these transformation matrices together in sequence, the overall transformation from the base to the end-effector is obtained, providing the position and orientation of the end-effector relative to the robot's base frame.

In summary, forward kinematics in the kinematic modeling of a 4-DOF robotic arm allows us to compute the end-effector's pose (position and orientation) based on the joint angles, essential for motion planning, control, and task execution.



Where, s stands for sin and c stands for cos. dx, dy, dz are the global end effector coordinates.

End effect orientation will be,



**3.3 THE ROBOTIC ARM CONTROLLER**

A robotic arm controller governs the movement and operation of robotic arms. It typically consists of sensors to gather data about the arm's position and environment, along with actuators to drive the arm's joints. Control algorithms interpret sensor data to determine the arm's current state and calculate the desired trajectory or action. Here the robotic arm control uses an artificial neural network (ANN) that is trained upon the data obtained using the simulation. As opposed to utilizing the known forward and inverse kinematics of the robotic arm, the controller is based upon the Jacobian transpose method us, except that the Jacobian matrix cannot be obtained analytically since it is presumed that the forward kinematics is unknown. Thus, the ANN shall be used to infer the Jacobian matrix based from the data gathered from the robotic arm response. The process can be understood with the following sections:

1. Input and Output parameters
2. The Approximated Jacobian Matrix
3. Training of Artificial Neural Network (ANN)

**3.3.1 Input and Output parameters**

The robotic arm functions as a mapping between joint angles, its inputs, and end-effector coordinates, its outputs. The joint space, representing all possible combinations of joint angles, is discretized by sampling at regular intervals. These sampled configurations serve as inputs to the robotic arm, determining the positions and orientations of its end effector. However, this process is subject to the specified limits for each joint, ensuring that the arm operates within its mechanical constraints. By adjusting the joint angles within these limits, the robotic arm achieves desired end-effector positions and orientations, enabling precise manipulation tasks. This method allows for systematic exploration of the arm's workspace and facilitates trajectory planning and control strategies, vital for various applications ranging from manufacturing to medical surgery.

Input Parameter: Joint angles

Output Parameter: Coordinates

The following Table II shows few input and output parameters:

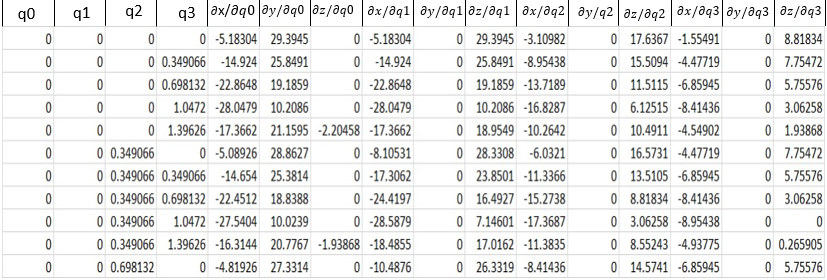
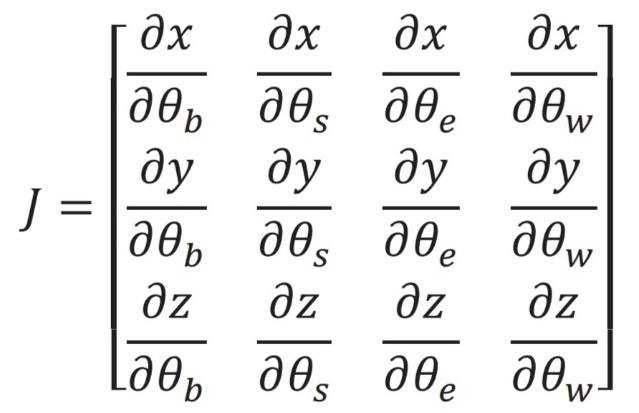


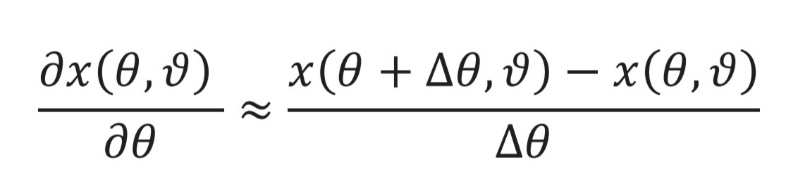
Table II: Data sample from approximated Jacobian matrix

**3.3.2 The Approximated Jacobian Matrix**

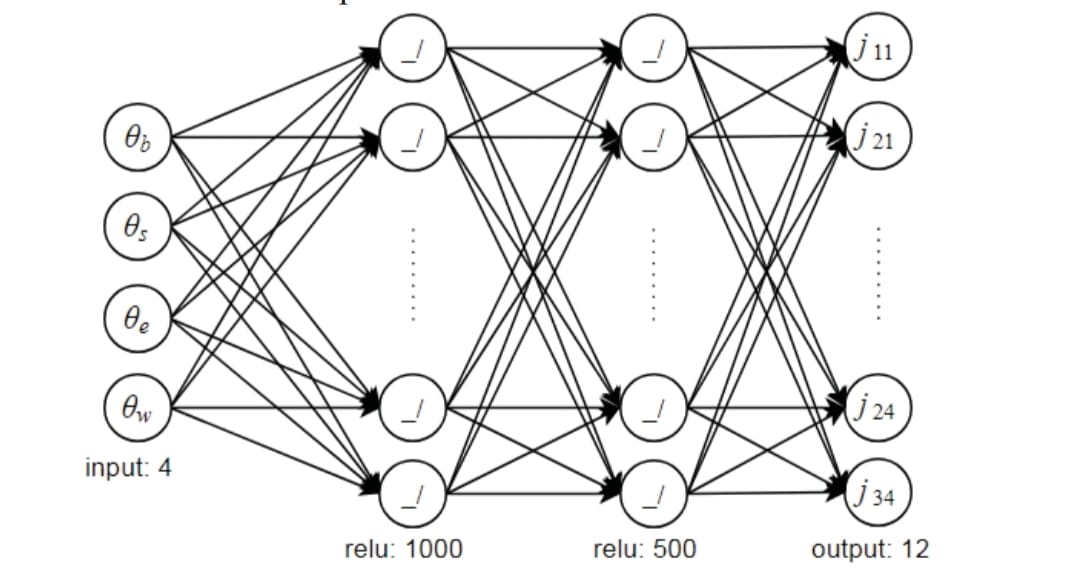
Based from the input and output parameters of the robotic arm, we do know that the Jacobian is a 3x4 matrix:



Given the presumed lack of knowledge about the robotic arm's forward kinematics, obtaining its Jacobian matrix analytically becomes impractical. Consequently, an alternative approach is required. Tabulated input-output pairs, representing sampled joint configurations and corresponding end-effector coordinates, are utilized to approximate the Jacobian matrix of the robotic arm. Through numerical methods or machine learning techniques, these data points are leveraged to infer the mapping between joint angles and end-effector positions. This approximation allows for the estimation of the Jacobian matrix, which describes the relationship between joint velocities and end-effector velocities. While this method introduces some level of error due to discretization and interpolation, it provides a practical means to enable velocity control and inverse kinematics solutions, crucial for real-time motion planning and trajectory tracking in robotic applications where analytical solutions are unavailable or computationally prohibitive. In approximating the partial derivatives, the forward difference quotient was used, i.e.,



where x( is the end-effector coordinate as a function of joint angles and . In performing the partial derivatives, the joint angle values are expressed in radians. From each sampled point from the joint space, the partial derivatives for x,y and z are calculated and tabulated.

**3.3.3 Training of Artificial Neural Network (ANN)**

This figure shows a multilayer feedforward neural network (MLFN) architecture with two hidden layers. This type of neural network is commonly used for supervised learning tasks, such as classification and regression.The network in the image has four input nodes, two hidden layers with 1000 and 500 nodes, respectively, and one output node. Each node in the hidden layers uses a ReLU activation function, while the output node uses a linear activation function.

Here is a breakdown of how the network works:

The input layer takes in four values, which are represented by the four nodes at the bottom of the diagram.These values are then multiplied by weights and passed to the first hidden layer of 1000 nodes. Each node in this layer applies a ReLU activation function, which means that it outputs the value zero if the input is less than zero, and otherwise outputs the input value.The outputs of the first hidden layer are then multiplied by weights and passed to the second hidden layer of 500 nodes. Each node in this layer also applies a ReLU activation function.Finally, the outputs of the second hidden layer are multiplied by weights and passed to the output layer, which has one node. This node applies a linear activation function, which means that it simply outputs the weighted sum of its inputs.The weights in the network are learned during the training process. The goal of training is to find a set of weights that allows the network to make accurate predictions on new data.

Overall, the architecture of the relu+relu+linear ANN is a common and effective way to build neural networks for supervised learning tasks.

**4. RESULT**

We have used 10 input hidden layers and 12 output layers to design 4 input and 12 output neurones ANN, which can be seen from below figure.

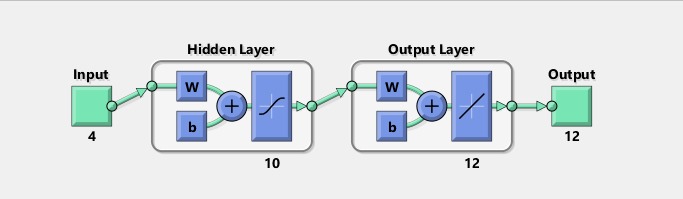


Table III represents,

Samples: This shows the number of samples in each dataset. All three datasets have 94 samples.

MSE: This stands for mean squared error, which is a common metric for measuring the performance of regression models. Lower MSE values indicate better performance. The MSE for the training dataset is the lowest (2.10571e-05), followed by the validation dataset (1.83874e-0) and the testing dataset (1.96712e-0). This suggests that the model generalizes well to unseen data, as the performance on the testing dataset is close to the performance on the training dataset.

R: This value represents the R-squared coefficient, which is another metric for measuring the performance of regression models. It ranges from 0 to 1, with higher values indicating better performance. In this case, all R values are very close to 1, which suggests that the model explains a large proportion of the variance in the target variable.

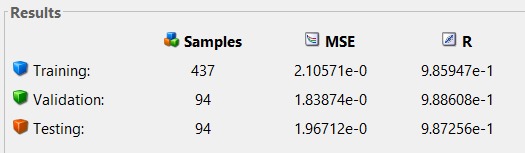
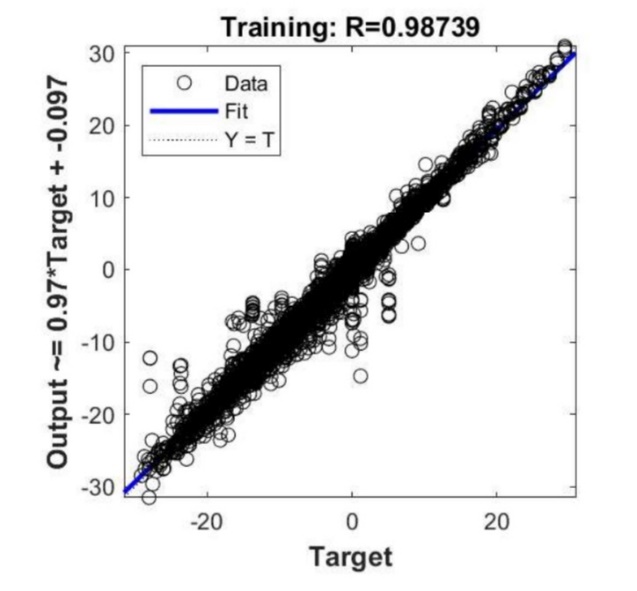
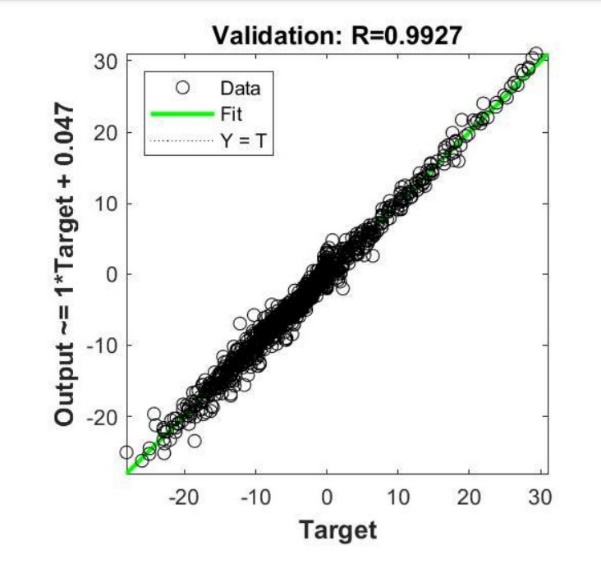
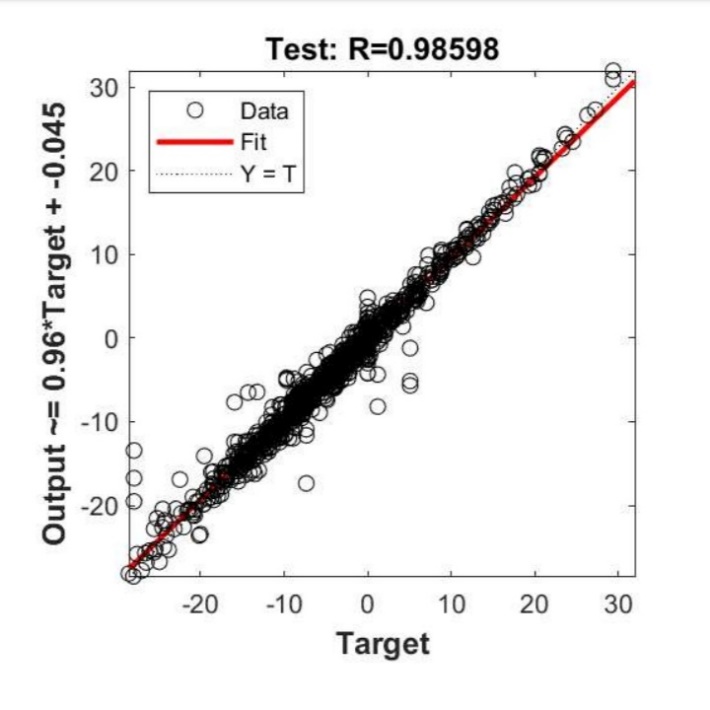
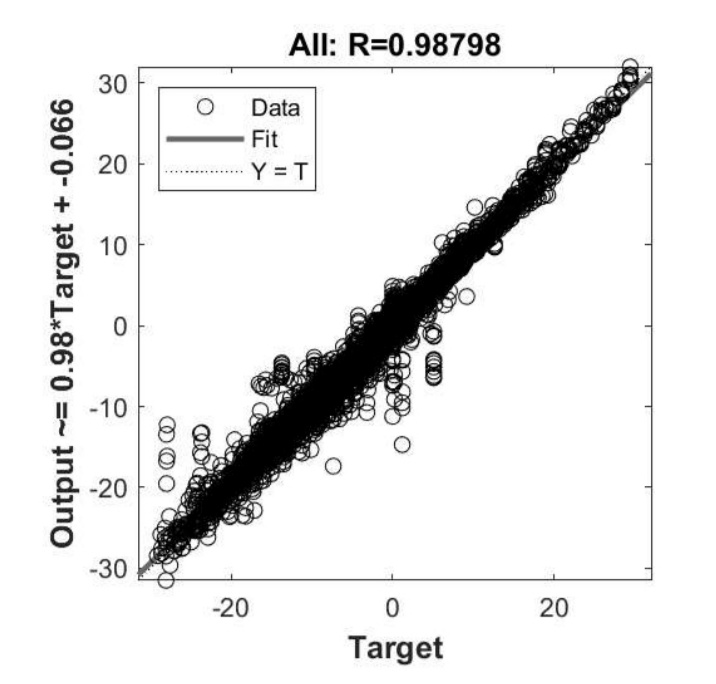


Table III

The MATLAB Neural Networks obtained are as follows:





**5. CONCLUSION AND FUTURE WORKS:**

**5.1 Conclusion**

The designed artificial neural network controller for the 4-degree-of-freedom (4-DOF) robotic arm utilizes kinematic modelling with the Denavit-Hartenberg (DH) convention and forward kinematics to optimize motion. With this approach we accurately determined the arm’s end-effector position by sequentially calculating joint transformations. The neural network optimizes

motion planning by mapping desired end-effector positions to corresponding joint angles, enhancing

precision and adaptability. With the help of Jacobian transpose method stability and convergence of

solution is achieved. Part of future studies includes incorporating a force control mechanism for the robotic arm.

**5.2FUTURE EXTENSION**

**Controller Architecture Design:**



Where,

= vector of required change in joint angles

α= appropriate scalar value

=transpose of the Jacobian matrix

vector of error in coordinates between the end-effector and the target

