One Solution for Inverse Kinematics of Robot Based on Artificial Neural Network

Shweta Pandey[1], Vivek Chawla[2]

1Master of Technology Student, Department of Mechanical and Automation Engineering,

Indira Gandhi Delhi Technical University for Women Delhi

2Assistant Professor, Department of Mechanical and Automation Engineering,

Indira Gandhi Delhi Technical University for Women Delhi

ABSTRACT

The most interesting problem of the robot is inverse kinematics. The inverse kinematics problem involves the determination joint angles for the desired Cartesian position in robotics of the end effector.the control a robot arm is to find the joint angles for Cartesian position and orientation of the end effectors. The unique solution does not exist for the inverse kinematics thus there is soft computing domain for appropriate model. Artificial neural network is a technique which can be used to yield the acceptable results. The paper proposed an artificial neural network (ANN) model to find the inverse kinematics solution of a three-link manipulator. The gradient descent type learning is applied here.The multi-layered perceptron artificial neural network (MLPNN) used here. The multilayer perceptron gives a minimum error.

Keywords: *ANN, MLP, BNN.*

INTRODUCTION:

A manipulating arm of a manipulator are consists of links that are connected by joints. Rotational and prismatic there is two types of joints are used, ie. An additional degree of freedom is given by changing their length. The joint angles and joint offset describe the position of the end effector by giving the knowledge of the length of all rigid links. It is necessary to understand the relationship between the actuators to developing control mechanism for the robot arms. If we want to calculate the position and orientation with the help of known link parameters then it is called as forward kinematics whereas finding the position and orientation with the help of know joint variables is known as inverse kinematics.

In terms understanding, the difficulty level of inverse kinematics is greater than the forward kinematics. In forward kinematics, forgiven goal position the system is within constraint, there could be an infinite number of solutions. The same end point could lead to a different configuration. So, the major problem in robotics is inverse kinematics. There are many traditional methods are available such as algebraic, geometric and iterative solutions. These methods are adopted for solving inverse kinematics problem. These problem with these methods are as: time-consuming, high computational cost and suffer from numerical problems. So, the need arises to apply intelligent systems such as Fuzzy Logic, Artificial Neural Network, soft computing etc. for solving inverse kinematics.

The literature review shows that there are many methods available to solve the inverse kinematics by replacing the place of traditional numerical computational methods. The artificial neural network involves the presence of hidden layer where the computation performed whereas in fuzzy logic the number of rule bases involved. For evolutionary methods, the complexity and high computational load involves the complex algorithm. These methods are difficult to implement on hardware because these methods give a better result.

INVERSE KINEMATICS

The inverse kinematics is the analysis or procedure which is used to find the joint coordinates for a given set of the end effector. Basically, this type of method requires solving a set of equations. Equations are, in general, are nonlinear and complex. If a nonlinear equation can be solved is cannot guarantee the uniqueness of system.

We saw that for the *R-P* manipulator, the direct kinematics equations are:

$$x = d_2.\cos\theta_1$$

$$y = d_2.\sin\theta_1 \qquad \ldots\ldots (1)$$

If we restrict the revolute joint to have a joint angle in the interval [0, 2π), there are two solutions for the inverse kinematics:

$$d_2 = \sigma\sqrt{(x^2 + y^2)}$$

$$\theta_1 = a\tan2\left(\frac{y}{d_2}, \frac{x}{d_2}\right) \qquad \ldots\ldots\ldots\ldots (2)$$

$$\sigma = \pm 1$$

The inverse kinematics analysis for a planar 3-*R* manipulator appears to be complicated but we can derive analytical solutions. Recall that the direct kinematics equations (10) are:

$$x = l_1\cos\theta_1 + l_2\cos(\theta_1 + \theta_2) + l_3\cos(\theta_1 + \theta_2 + \theta_3) \qquad (3)$$

$$y = l_1\sin\theta_1 + l_2\sin(\theta_1 + \theta_2) + l_3\sin(\theta_1 + \theta_2 + \theta_3) \qquad (4)$$

$$\varphi = \theta_1 + \theta_2 + \theta_3 \qquad (5)$$

We assume that we are given the Cartesian coordinates, x, y, and φ and we want to find analytical expressions for the joint anglesθ_1, θ2, and θ_3 in terms of the Cartesian coordinates. Substituting 10(c) into 10(a) and 10(b) we can eliminate θ_3 so that we have two equations in θ_1 and θ_2:

$$x - l_3\cos\varphi = l_1\cos\theta_1 + l_2\cos(\theta_1 + \theta_2) \qquad (6)$$

$$y - l_3\sin\varphi = l_1\sin\theta_1 + l_2\sin(\theta_1 + \theta_2) \qquad (7)$$

Where the unknowns have been grouped on the right-hand side; the left-hand side depends only on the end effector or Cartesian coordinates and is therefore known.

Rename the left hand sides, $x' = x - l_3\cos\varphi$, $y' = y - l_3\sin\varphi$ for convenience. We regroup terms in (d) and (e), square both sides in each equation and add them:

$$(x' - l_1\cos\theta_1)^2 + (y' - l_1\sin\theta_1)^2 = (l_2\cos(\theta_1 + \theta_2))^2 + (l_2\sin(\theta_1 + \theta_2))^2$$

After rearranging the terms we get a single nonlinear equation in θ1:

$$(-2l_1 x')\cos\theta_1 + (-2l_1 y')\sin\theta_1 + (x'^2 + y'^2 + l_1^2 - l_2^2) = 0 \qquad (8)$$

Notice that we started with three nonlinear equations in three unknowns. We reduced the problem to solving two nonlinear equations in two unknowns. And now we have simplified it further to solving a single nonlinear equation in one unknown (f). Equation (f) is of the type

$$P\cos\alpha + Q\sin\alpha + R = 0 \qquad (9)$$

There are two solutions for θ1 given by:

$$\theta_1 = \gamma + \sigma\cos^{-1}\left[\frac{-(x'^2 + y'^2 + l_1^2 - l_2^2)}{2l^2\sqrt{x_1^2 + y_1^2}}\right] \qquad (10)$$

$$\text{Where } \gamma = a\tan2\left[\frac{-y'}{\sqrt{x'^2_1 + y'^2_1}}, \frac{-x'}{\sqrt{x'^2_1 + y'^2_1}}\right]$$

$$\text{And } \sigma = \pm 1$$

Note that there are two solutions forθ1, one corresponding to σ=+1, the other corresponding to σ = −1. Substituting any one of these solutions back into Equations gives us:

$$\cos(\theta_1+\theta_2) = \frac{x' - l_1 \cos\theta_1}{l_2}$$

$$\sin(\theta+\theta_2) = \frac{y' - \sin\theta_1}{l_2}$$

And,

$$\theta_2 = a\ tan2\left[\frac{y' - \sin\theta_1}{l_2}, \frac{x' - l_1 \cos\theta_1}{l_2}\right] - \theta_1 \qquad \textbf{(11)}$$

Thus, for each solution for $\theta 1$, there is one (unique) solution for $\theta 2$.

Finally, $\theta 3$ can be easily determined.:

$$\theta 3 = \Phi - \theta 1 - \theta 2 \qquad \textbf{(12)}$$

Equations (h-j) are the inverse kinematics solution for the 3-*R* manipulator. For a given end effectors position and orientation, there are two different ways of reaching it, each corresponding to a different value of σ.

ARTIFICIAL NEURAL NETWORK

A neural network is a machine that is designed to model the way in which the brain performs a particular task or function of interest. To achieve good performance, they employ a massive interconnection of simple computing cells referred to as 'Neurons' or 'processing units'. Hence a neural network viewed as an adaptive machine can be defined as A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experimental knowledge and making it available for use. It resembles the brain in two respects:

1. Knowledge is acquired by the network from its environment through a learning process.
2. Interneuron connection strengths, known as synaptic weights, are us to store the acquire knowledge.

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. We can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements. Commonly neural networks are adjusted, or trained so that a particular input leads to a specific target output. Such a situation in the following figure

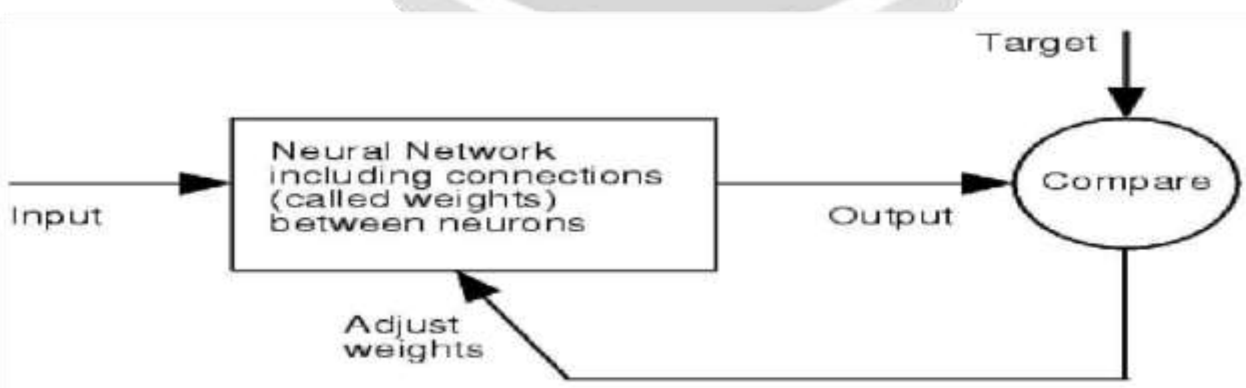

Fig 4.1: Neural Network Model

The true power and advantage of neural networks lie in their ability to represent both linear and non-linear relationships and in their ability to learn these relationships directly from the data being modeled. Traditional linear models are simply inadequate when it comes to modeling data that contains non-linear characteristics. Neural networks are designed to work with patterns - they can be classified as pattern classifiers or pattern associates.

RESULTS

The network is trained to generate the desired result. Here the input-output mapping is used to compare the generated result with the desired result. Approximately one thousand epochs are used to generate the desired output. One thousand maximum fail value is set here. 'Trainum'and backpropagation algorithm is used here.

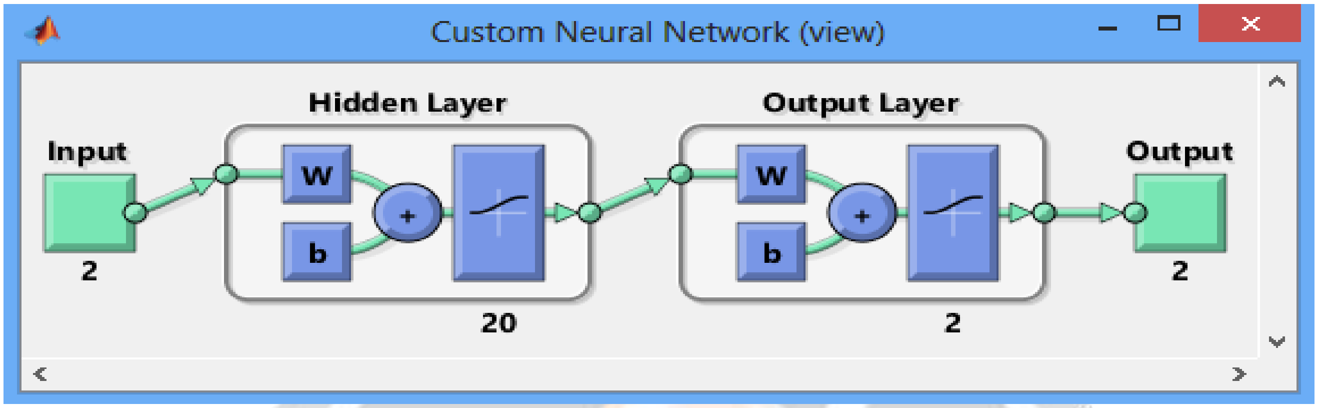

Figure 1 Artificial Neural Network Used for Training

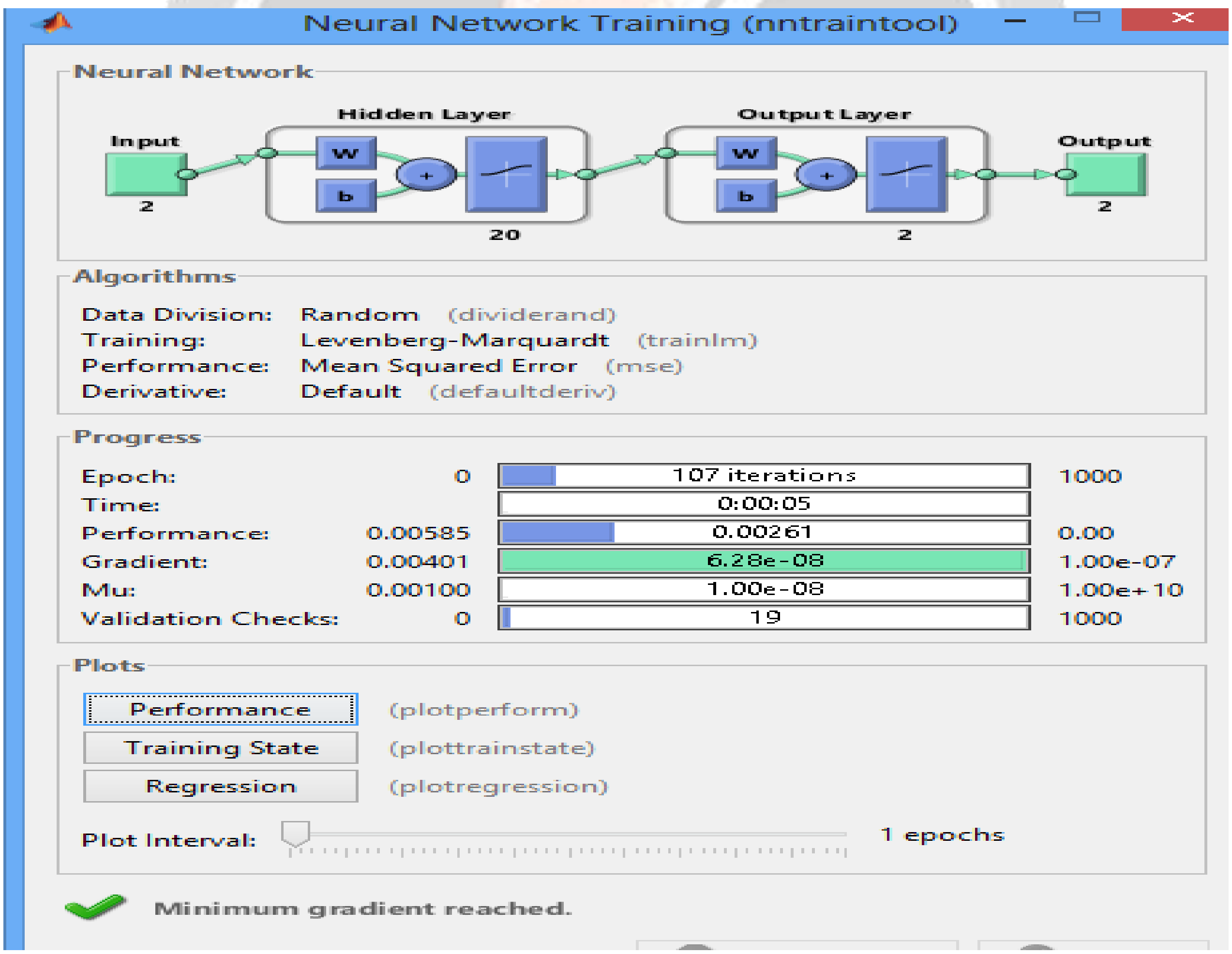

Figure 2 Artificial Neural Network Training

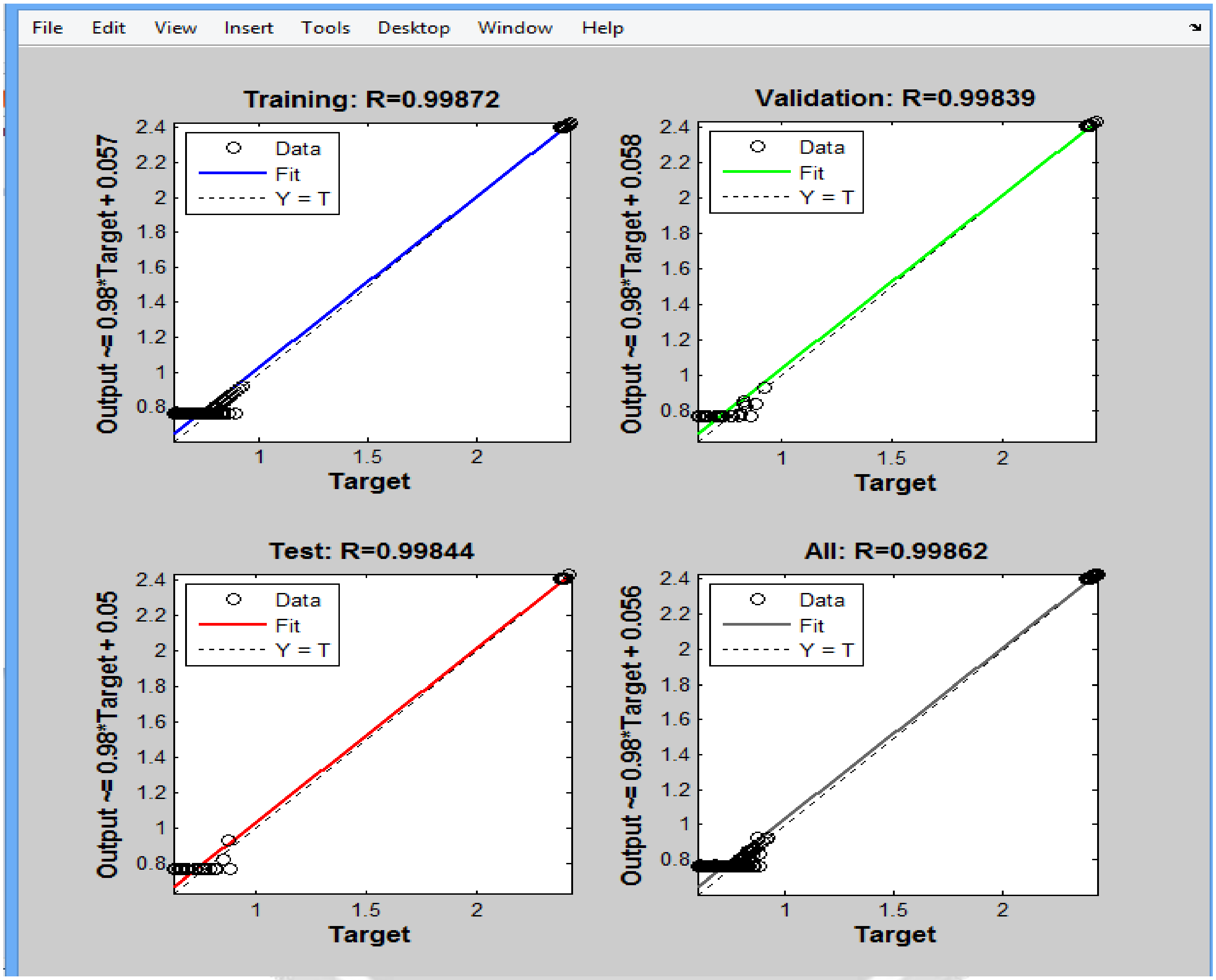

Figure 3 Regression Plot

ACKOKNOWLEDGEMENT

The first author wants to thank her Grandfather Mr. Devi Prasad Mishra for his blessing that are always on her and which enabled her to do this work. The first author also wants to place her sincere thanks to her father Mr. Krishna Kumar Pandey, mother Seema Pandey, her sister Shraddha Pandey, Aditi Pandey, her brother Anant Pandey for their support and resources that they gave for the work that she did. The first author also wants to thank her aunty Mrs. Reeta Pandey and Mrs. Madhuri Pandey for the moral support that they gave her.

REFERENCES:

1) John J. Craig, Introduction to Robotics: Mechanics and Control, Prentice Hall, New Jersey, 2004.
2) Christian Smith, Henrik I. Christensen. Robot Manipulator, IEEE Robot and Automation Magazine, vol. 64, 2009, pp 75-83.
3) Jolly Shah, S.S. Rattan, B.C. Nakra. "Kinematics Analysis of 2-DOF Planar Robot Using Artificial Neural Network", World Academy of Science, Engineering and Technology, vol. 81, 2011, pp. 282-285.
4) V.K. Singh and Shweta Pandey, "Minimum Configuration MLP for Solving XOR Problem", Published in IndiaCom2016 International Conference on Computing for Sustainable Global Development, Publisher IEEE, Bharati Vidhyapeeth sInstitute of Computer Applications and Management (BVICAM), New Delhi, March 17-3-2016.
5) Handelman, A.D., Stephen, H.L. and Gelfand, J.J., "Integrating Neural Networks and Knowledge – Based System for Intelligent Robotic Control", IEEE Control System Magazine, pp. 77-87, April 1990.
6) Antsaklis, P.J. and Passino, K.M., "Towards Intelligent Autonomous Control Systems: Architecture and Fundamental Issues: International Journal of Intelligent and Robotic Systems, pp. 315-342, 1989.
7) Verbose, P.J., "Backpropagation Through Time: What It Does and How to Do It", Proceedings of the IEEE, Vol. 78, No. 10, pp. 1550-1560, October 1990.
8) Vaibhav Kant Singh and Shweta Pandey, "Proposing an Ex-Nor Solution using ANN", International Conference on Information, Communication & Computing Technology, 2016.
9) J. Guo and V. Charkassky, "A Solution to the Inverse Kinematics in Robotics using Neural Network Processing", International Conference on Neural Networks, Vol. 2, pp. 299-304, 1989.
10) Prashant Badoni, "Neural Network Based Inverse Kinematics Solution for 6-R Robot Using Levenberg-Marquardt Algorithm", International Journal of Scientific Engineering and Research, Vol 3, pp. 79-83,2015
11) A. Abreu and S. Olaru, "Optimization of Robots Inverse kinematics Results by using the Neural Network and LabView Simulation", IPCSIT, 2011 pp. 40-45.
12) B. Choi and C. Lawrence "Inverse Kinematics problem in Robotics using Neural Networks", Natural Aeronautics and space Administration, Lewis Research Centre, Cleveland, 1992
13) M. Tokita, T. Fukuda, T. Matsuoka and T. Kurihara, "Force control of Robot Manipulator by Neural Network," JRSJ, vol 7. No. 1, pp.47-51(1989).