International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:5, No:9, 2011

Kinematic Analysis of 2-DOF Planer Robot Using Artificial Neural Network

Jolly Shah, S.S.Rattan, B.C.Nakra

***Abstract*—**Automatic control of the robotic manipulator involves study of kinematics and dynamics as a major issue. This paper involves the forward and inverse kinematics of 2-DOF robotic manipulator with revolute joints. In this study the Denavit-Hartenberg (D-H) model is used to model robot links and joints. Also forward and inverse kinematics solution has been achieved using Artificial Neural Networks for 2-DOF robotic manipulator. It shows that by using artificial neural network the solution we get is faster, acceptable and has zero error.

***Keywords*—**Artificial Neural Network, Forward Kinematics, Inverse Kinematics, Robotic Manipulator

I. Introduction

MANIPULATOR kinematics is a study of the geometry of manipulator arm motions. Since the performance of specific manipulator tasks is achieved through the movement of the manipulator linkages, kinematics is of fundamental importance in robot design and control. A kinematic equation provides the relationship between the joint displacement and the resulting end-effector position and orientation. The problem of finding the end-effector position and orientation for a given set of joint displacements is referred to as the forward kinematics problem. That is, the forward kinematics problem allows one to specify in a unique manner the relationship between the (n x 1) joint vector θ and the (m x 1) Cartesian vector *x* as:

$$x(t) = f(\theta(t))$$

Where f is the function defining the forward kinematic relation of the manipulator. Normally, the forward Kinematic equation can be obtained from the spatial geometry of the manipulator or by solving certain matrix algebraic equations. As the number of degrees of freedom (n) increases, the kinematic equation becomes more complex. Hence, the amount of computation required to compute the end-effectors position can become quite large.The inverse kinematics problem consists of the determination of the joint variables corresponding to a given end-effectors orientation and position and is given by the equation,

$$\theta(t)=f^{-1}(x(t))$$

As compared to forward kinematics problem, the inverse kinematics problem is much more complex due to the following reasons:

Jolly Shah PhD Scholar, Department of Mechanical Engineering, National Institute of Technology,Kurukshetra, Haryana, India Corresponding author, Phone +919910272582, email: jollyshah80@yahoo.com)
Dr.S.S.Rattan, Department of Mechanical Engineering, National Institute of Technology, Kurukshetra, Haryana, India
Dr. B.C.Nakra, Department of Mechanical Engineering, ITM University, Sector – 23A, Gurgaon, Haryana, India

- In the joint variables, the equations to be solved are in general non-linear form and, thus, mostly, it is not always possible to find a closed form solution.
- Multiple solutions may exist.
- Infinite solutions may also exist, example – in redundant robot manipulators.

In order to find the solutions to the inverse kinematics, problem can be formulated either by finding a closed form solution using algebra or geometry or by finding a numerical solution by some successive approximation algorithm.[1,2,3,5,8]

Artificial Neural networks provide a fast, method of learning to produce a set of output states given a set of input states. The advantages of using an artificial neural network approach in robotics applications such as in the forward kinematics problem are given below and also discussed in [6]:

- Programming required is not complex;
- The time to obtain a solution is independent of the number of manipulate or degrees of freedom;
- Fault tolerance;
- Capability of operating in real-time.

In this study, the Denavit-Hartenberg (D-H) approach has been used for modeling the 2-DOF robotic manipulator. The forward and inverse kinematic solutions have been deduced using kinematic equations for 2-DOF robotic manipulator as shown in fig.1.

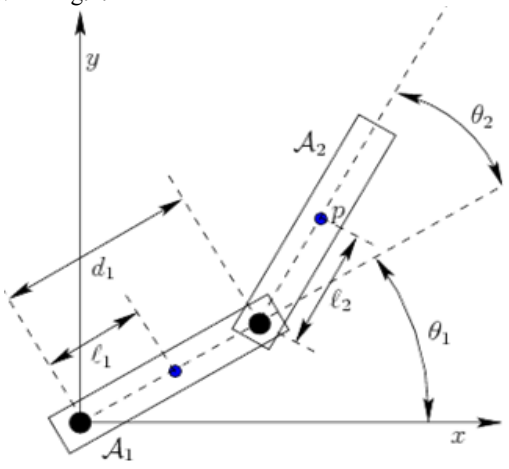

Fig. 1 Representation of 2-DOF robotic manipulator

By MATLAB programming a data set has been generated. Data obtain in form of set of position-orientation of robot is used to train an artificial neural network, feed forward neural network is used and trained by using MATLAB toolbox.

II. Artificial Neural Network

It is an area of computer science, which is basically concerned with *designing intelligent computer systems*. It is mostly inclined towards those systems that exhibit the characteristics we associate with intelligence in human behavior. In other words, *Artificial Intelligence* is a branch of computer science that is concerned with the automation of intelligent behavior. In order to build intelligent systems, it includes various technologies in its domain such as expert systems, neural networks, fuzzy logic, cellular automata and probabilistic reasoning.

"A Neural Network is a massive parallel distributed processor made up of simple processing units has a tendency to acquired the knowledge from its environment through a learning process, store the knowledge and making it available for us."[4]

Artificial neural networks are widely used as an effective approach for handling non-linear and noisy data[7], especially in situations where the physical processes relationships are not fully understood and they are also particularly well suited to modelling complex systems on a real-time basis. Basic structure of any artificial neural network is shown in fig.2. Also basic neural network architecture is shown in fig.3.

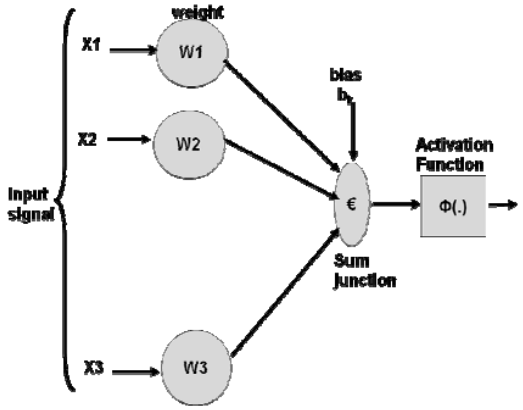

Fig. 2 Representation of ANN Structure

In mathematical terms , we may describe a neuron *K* by writing the following terms:

Uk=(w11x1+w12x2+...wkmxm)

and

Yk=ϕ(uk+bk)

Where*, x1, x2, x m* are the input signals,

Wkj are the synaptic weights of neuron *K*,

Uk is the linear combiner output due to the input signals,

Bk is the bias,

ϕ (.) is the activation function

Here the work focuses on a popular feed forward model of neuralnetworks.

In this model a set of inputs are applied to the network, and multiplied by a set of connection weights.

All of the weighted inputs to the neuron are then summed and an activation function is applied to the summed value. This activation level becomes the neuron's output and can be either an input for other neurons, or an output for the network. Learning in this network is done by adjusting the connection weights based upon training vectors (input and corresponding desired output). When a training vector is presented to a neural net, the connection weights are adjusted to minimize the difference between the desired and actual output.

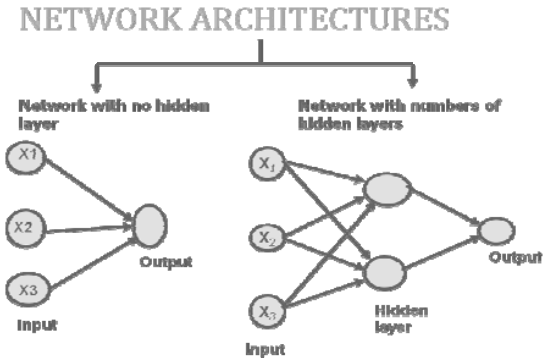

Fig. 3 Artificial Neural Network Architecture

III. Kinematics

Denavit-Hartenberg (D-H) representation is used to model the joints of 2-DOF robotic manipulator, as discussed in [1,2]. The D-H parameters are described as *di* (distance on the *z* axis between two successive normals), *ai* (distance between joints along *x* axis), *θi* (rotation about the *z* axis on *xy* plane) and *αi* (joint twist). These parameters describe the location of a robot link frame *Fi* (a joint) from a preceding link frame *Fi-1* (previous joint) through the sequence of translations and rotations.

The D-H parameters for 2-DOF robotic manipulator are given in Table I.

TABLE I
D-H PARAMETERS FOR 2-DOF ROBOTIC MANIPULATOR

Joint	θ_i (^{0})	α_i (^{0})	a_i (mm)	d_i (mm)
1	90	0	179	0
2	90	0	177	0

In this paper, a 2-DOF robotic manipulator has been considered with link lengths *L1* = 179 mm and *L2* = 177 mm, respectively, as shown in fig.1, where, *L1* and *L2* are link lengths and *θ1* and *θ2* are joint angles, respectively. In forward kinematics solution, if the link lengths and angles are known, then it is possible to calculate the co-ordinates of the end-effectors of the robot at any instant. The MATLAB program has been run to find the co-ordinates (*x, y*) of the 2-DOF robotic manipulator and which has found out to be (177, 179) mm, respectively at the home position.

The forward kinematics equations for 2-DOF robotic manipulator have been deduced as:

$$x = L_1 \times \cos\theta_1 + L_2 \times \cos(\theta_1 \quad + \theta_2) \tag{1}$$

$$y = L_1 \times \sin\theta_1 + L_2 \times \sin(\theta_1 \quad + \theta_2) \quad (2)$$

Using the forward kinematic equations of the robotic manipulator, all the (*x, y*) co-ordinates of the 2-DOF robotic manipulator have been generated and the work-volume has been plotted for all possible *θ1* and *θ2* combinations as represented in fig.4. [9]

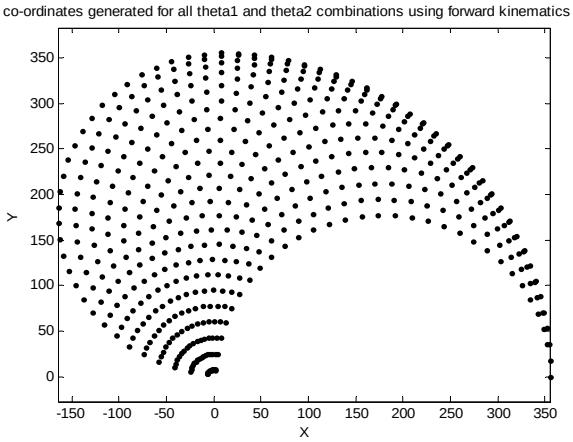

Figure 4. Work-volume of 2-DOF robotic manipulator.

A.Forward Kinematics Using Artificial Neural Network

Total 25 set of data obtain in form of position-orientation of robot using MATLAB programme is used to train neural network. Feed Forward neural network is used and trained by using MATLAB toolbox. Training graph is as shown in fig.5. X-axis and Y-axis of fig.5 represents number of epochs and performance respectively.

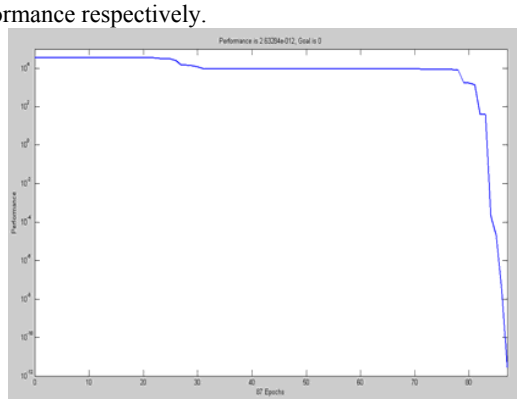

Fig. 5 Training graph for forward kinematics using ANN

Also comparison between actual target points and predicted target points are shown in fig.6, X co-ordinate values and Y-coordinate values are represented by blue and green colours respectively. Also dot represents actual targets and dash represents predicted targets values.

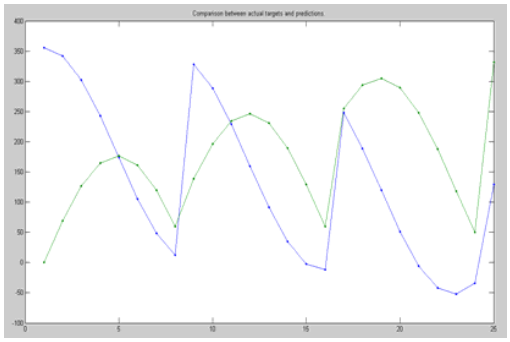

Fig. 6 Comparison between actual and predicted targets for forward kinematic

B. Inverse Kinematics Using Artificial Neural Network

Total 25 set of data obtain in form of orientation-position of robot using MATLAB programme is used to train neural network. Feed Forward neural network is used and trained by using MATLAB toolbox. Training graph is as shown in fig.7. X-axis and Y-axis of fig.7 represents number of epochs and performance respectively.

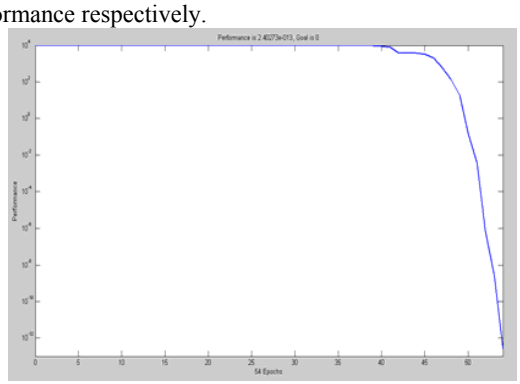

Fig. 7 Training graph for inverse kinematics using ANN

Also comparison between actual target points and predicted target point are shown in fig.8.[10], X co-ordinate values and Y-coordinate values are represented by blue and green colours respectively. Also dot represents actual targets and dash represents predicted targets values.

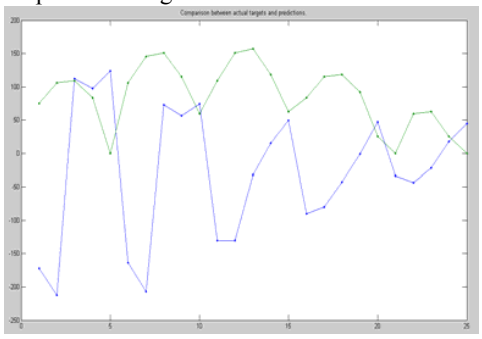

Fig. 8 Comparisons between actual and predicted targets for inverse kinematics

VI. Conclusions

1. The differences in actual and predicted target values using ANN for 2-DOF robotic manipulator, shows that zero error is achieved by the proposed method.
2. ANN can be trained so that fast and acceptable solutions are achieved.
3. ANN can become an alternate method to map the forward and inverse kinematics solutions.

References

[1] Saeed B. Niku, Introduction to Robotics: Analysis, Systems, Applications, Prentice Hall, Jul 2001, p1-349.
[2] R.K. Mittal, I.J. Nagrath, Robotics and Control. New Delhi: Tata Mc Graw Hill Publishing Company Limited, 2003.
[3] S. K. Saha, Introduction to Robotics, Tata McGraw Hill, 2008, p1-425.
[4] M Gopal, Digital Control and State Variable Methods: Conventional and Intelligent Control Systems, Third Edition, 2009.
[5] Vijayant Agarwal, Ph. D. Thesis, Studies of Dynamics and Control of Robotic System using Soft Computing Techniques, University of Delhi, July 2008.
[6] Allon Gurez ,Ziauddin Ahmad, Solution to the inverse kinematics problem in robotics by neural network., ICNN88 March 1988.
[7] L.Acosta,G.N.Marichal, L.Moreno,J.J.Rodrigo, A.Hmilton,J.A.Mendez, A robotic system based on neural network controllers.,artificial intelligence in engineering 13 (1999) 393-398
[8] Vijyant Agarwal, A. P. Mittal, B. C. Nakra, A Study of Fuzzy Logic Based Inverse Kinematics Solution, Proceedings International Conference on Computer Applications in Electrical Engineering, Roorkee, Sept. 2005, p1-4.
[9] Srinivasan Alavander, M.J. Nigam, Inverse Kinematics Solution of 3DOF Planar Robot using ANFIS, International Journal of Computers, Communications and Control, Vol. III(2008), p150-155.
[10] Jose Antonio Martin H. ,Javier de Lope, Matilde Santos.A method to learn the inverse kinematics of multi-link robots by evolving neuro-controllers, Neurocomputing(2009), 2806-2814