ISSN (e): 2250 – 3005 || Volume, 06 || Issue, 6|| June – 2016 ||
International Journal of Computational Engineering Research (IJCER)

Joint Angle Optimization of Industrial Robot using Artificial Neural Network

Kshitish Kumar Dash[1], K S Raghuram[2], Nirmalendu Hota[3]
[1,3]Assistant Professor, Department of Mechanical Engineering, Gandhi Institute For Technology (GIFT), Bhubaneswar
[2] Assistant Professor, Department of Mechanical Engineering, Gandhi Engineering College, Bhubaneswar

Abstract: *A neural network based inverse kinematics solution of a robotic manipulator is presented in this paper. Inverse kinematics problem is generally more complex for robotic manipulators. Many conventional solutions are inadequate, if the joint arrangement of the manipulator is more complicated. In Inverse kinematics problem, the neural computation have to find the coordinates by giving the joint angles for a given Cartesian position and orientation of the end effectors. Many initial and final points are generated in the work volume of the robotic manipulator by solving the neural network. The entire real-world coordinates (x, y, z) as per the angles recorded in a file named as training set of neural network model. The designed neural model has specified the correct coordinates according to the certain angles of Cartesian coordinate. The inverse kinematics problems of industrial robot manipulator are solved by using Artificial Neural Network (ANN) algorithm. The 6R robot manipulator is selected as industrial robot manipulator because geometric feature of this robot does not allow for solving inverse kinematics problems analytically.*
Keywords: *Inverse kinematics, Artificial Neural Network, 6-D.O.F, Industrial Robot*

I. INTRODUCTION

A neural network based inverse kinematics solution method yields multiple and precise solutions with an acceptable error and it is suitable for real time adaptive control of robotic manipulators. Neural networks are competent of learning and building of complex functions, which led to their use in different applications such as recognition, approximation, fitting of data, and dynamic systems to be checked. The Nonlinear dynamic systems included with the robots for executing tasks repetitively, are shown to advantage from the use of neural network controllers. The results are reported in the literature demonstrated that neural network is valuable in many problems in of inverse kinematics problem. In this work inverse kinematics solution using neural networks is presented with theoretical background of neural mosel. A neural network is the preliminary data processing structure of interest in neuron computing. A parallel, distributed information processing structure composed of a number of simple, processing elements to be interconnected similar to neurons in the human nervous system is defined. The processing elements interact locally through a set of unidirectional weighted connectors. The neural network trains itself to generalize a mapping or functional relationship from example sets of input vectors and consequent output. The neural network then stores the connection strengths (weights) between processing units. The weights correspond to the strength between neurons and are adjusted during the learning process. The information of the model is internally represented by the values of the weights and the topology of the connections. The network is able to solve for unknown output when new input is displayed. The neural network in contrast with established methods where the specific relationships between input and output must be supplied by user defined algorithms. Self-organization, fault tolerance, optimization, association, generalization etc. are allowed in the neural network characteristics.

1.1 Related work

Nan et al [1] presents a new method for the inverse kinematics of 6-DOF manipulator based on dual quaternion which uses dual quaternion to respect both rotation and translation vector. The simulation is provided to verify the feasibility and precision of the proposed new method. Vasilyev et al [2] presents analytical solution to Inverse Kinematic Problem for 6-DOF Robot-Manipulator. In this paper the solution technique based on deriving the system of nonlinear equations and solving it in the general view. Mashhadany et al [3] proposed a cognitive architecture for solution of inverse kinematics problem (IKP) of 6- DOF elbow manipulator with spherical wrist by Locally Recurrent Neural Networks (LRNNs) and simulated the solution by using MATLAB/Simulink This design is aimed to allow the manipulator system to perform complex movement operations by solving the Inverse Kinematic Problem (IKP) with LRNNs by using the position and orientation of end-effector which represent by wrist with 3-DOF.Suell et al [4] works on new technique for inverse kinematics problem using simulated annealing. This paper present to find the accurate values for each one of actuators in a robotics system knowing the ultimate location .Rasit Koker [5] developed an inverse kinematics problem of redundant manipulators with the determination of joint angles for a desired Cartesian position of the end effectors. For this solution, many established solutions such as geometric, iterative and algebraic can insufficient if the joint structure of the manipulator was more complicated. Martin et. al [6] presents the general method to learn the inverse kinematic of multi-link robots by means of neuro-controllers. These solutions are definite to a particular robot configuration and are not generally applicable to other robot. Hasan et. al [7] proposed the first globally convergent adaptive tracking controller was derived for robots with dynamic uncertainties. Though, not until newly has the problem of concurrent adaptation to both the kinematic and dynamic uncertainties found the solution.

1.2 Inverse Kinematics for Robotic Joints

Manipulator kinematics is a study of the geometry of manipulator arm movement and the performance of specific task is achieved through the movement of the arm links, kinematics is an elementary tool in manipulator design and control. It can describe the inverse kinematics problem as the task of calculating all of the joint angles that would result in a specific position of an end-effector of a robot manipulator arm. A robotic kinematic equation can provide the relationship between the joint displacements and the resultant end-effector position with their orientation. The problem of determine the end-effector position an orientation for a given set of joint displacements is referred to as the direct kinematic problem. The direct kinematic problem allows one to specify in a unique manner the relationship between the (nx1) joint vector q and the (mx1) Cartesian vector x as

$$x = f(q) \qquad (1)$$

Where *f* is a continuous non-linear function whose structure and parameters are known; it associates with each *q* a unique *x* and the inverse mapping is may have many *q*'s associated with each *x*. usually, there are two classes of solution method to solve the inverse kinematic problem: closed form solutions and numerical solutions. Closed forms solution can be obtained by the spatial geometry of the manipulator, or by solving the matrix algebraic equation (1).Because the complexity of equation (1), there are cases wherein a closed form solution does not exist. For non-redundant manipulators (m = n) which do not have a closed form solution, or for those manipulators which have redundant degrees of freedom (m < n), numerical method are commonly used to derive the desired joint displacements [2]. Numerical methods are iterative algorithms such as Levenberg-Marquardt (LM). Because of their iterative nature, the numerical solutions are generally much slower than the corresponding closed form solution. It is important to mention, that with equations (1) and (2), we can obtain only the angles of joint vector or the values of the Cartesian vector. But if we want to know the Cartesian velocity and acceleration, those can be obtained from the equation (1), if f(.) at least once differentiable, then

$$q = f^{-}(x) \qquad (2)$$

$$\dot{x} = j\dot{q} \qquad (3)$$

$$\ddot{x} = j\ddot{q} + j\dot{q} \qquad (4)$$

Where $j = {df}/{dq}$ is the non-square Jacobin matrix. Given a trajectory of end-effector positioning Cartesian space, the inverse problem is to compute corresponding joint trajectories: position q_d, velocity $\dot{q}_d$, and acceleration $\ddot{q}_{\dot{d}}$. Classical approach consists of computing the solution by introducing the pseudo inverse J^+ so that:

$$\dot{q} = j^+\dot{x} + \left(I - j^+ j\right)v \qquad (5)$$

$$\ddot{q} = j^+\left(\dot{x} - j\dot{q}\right) + \left(I - J^+ J\right)\delta \qquad (6)$$

Where *I* is the identity matrix, ($I-J^+J$) is the null space projection matrix, *v* is an arbitrary joint velocity vector and δ an arbitrary joint acceleration vector.

II. NEURAL NETWORKS STRUCTURES

It is well known that neural networks can model very efficiently complex no linear systems. For robotics applications, neural networks offer several advantages compared to conventional computational schemes, for example, neural networks learn to map a two-dimensional robot transformation between joint space and Cartesian space (forward and inverse kinematics equations) using a back propagation algorithm.

2.1 Neural Network Architecture

ANN takes their name from the network of nerve cells in the brain. Recently, ANN has been found to be an important technique for classification and optimization problem. ANN has emerged as a powerful learning technique to perform complex tasks in highly nonlinear dynamic environments. Some of the prime advantages of using ANN models are their ability to learn based on optimization of an appropriate error function and their excellent performance for approximation of nonlinear function. The ANN is capable of performing nonlinear mapping between the input and output space due to its large parallel interconnection between different layers and the nonlinear processing characteristics. An artificial neuron basically consists of a computing element that performs the weighted sum of the input signal and the connecting weight. The sum is added with the bias or threshold and the resultant signal is then passed through a nonlinear function of sigmoid or hyperbolic tangent type.

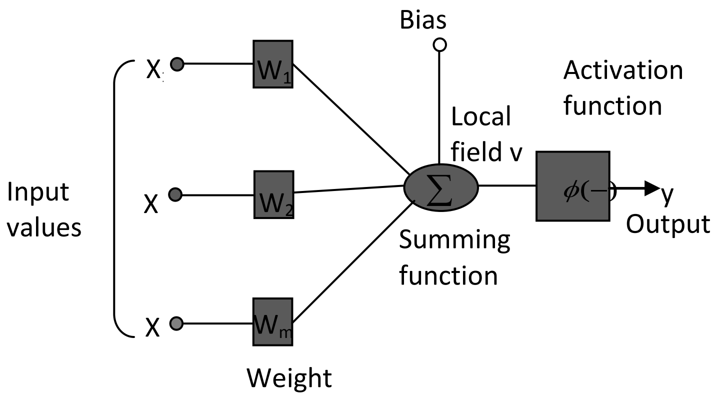

Fig.1. The Neuron Model

2.2 ANN approaches to the inverse kinematics solution

Studying the inverse kinematics of manipulator by using an ANN has two problems. One of these is the selection of the appropriate type of neural network, and the other is the generation of suitable training dataset, when the kinematic parameters of a manipulator are known, then a kinematic model can be made. If the joint parameters are not known, then the experimental results obtained for the manipulator are used to prepare the data set. This is very difficult in practice. However, most of the well-known industrial robots' joint parameters are known. The success of the ANN approach is measured according to the training error rate.

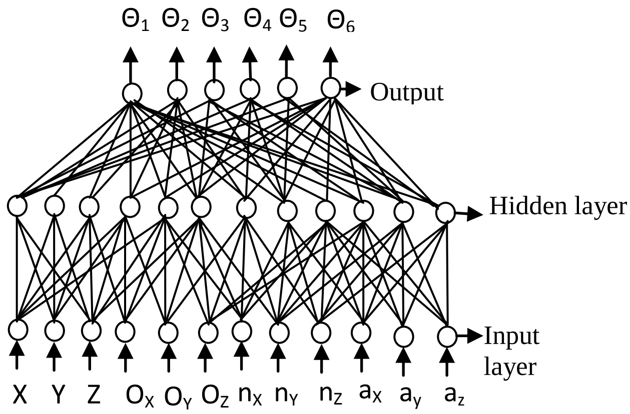

Fig.2. Three Layer Perception with six outputs

$$AHi(t) = \sum_{k=1}^{i} WT_{ik} I_k + \sum_{k=1}^{m} WH_{ik} \, f(AH_k(t-1)); \qquad (7)$$

$i = 1,2,3 \ldots\ldots\ldots\ldots\ldots\ldots m$

The activation function which is used in hidden layer and the output of the network is a weighted sum of the hidden unit o/p.

$$O_i(t) = \sum_{k=1}^{m} WO_{ik}\, f\,(AH_k(t));\ i = 1,2.........\ldots..n \tag{8}$$

$$E(t) = \sum_{p=1}^{pp}\left(\frac{1}{2}\sum_{k=1}^{n} e_{kp}(t)^2\right) \tag{9}$$

$$E(t) = \sum_{p=1}^{pp}\left(\frac{1}{2}\sum_{k=1}^{n}\left(T_{kp}(t) - O_{kp}(t)^2)\right)\right) \tag{10}$$

III. SIMULATION RESULTS

To validate the performance of inverse kinematics problem, simulation studies are carried out by using MATLAB. A set of 130 data sets are first generated and the input parameter angles in degrees. Using these data sets is the basis for the training and evaluation or testing the models. Out of the sets of 130 data points, 100 are used as training data and is used for training the network and for updating the desired weights. For this problem, the capacity models of the joint angles used here are shown in Table 1. The distance between adjacent links with their ranges are mentioned in Table 2. Table 3 shows the normal orientation vector in all coordinates.

Table-.1: Specification of joint

Angle (Degree)	Capacity	Range	Joint
θ_1	320	(-160 to 160)	Waist Joint
θ_2	220	(-110 to 110)	Shoulder Joint
θ_3	270	(-135 to 135)	Elbow Joint
θ_4	532	(-266 to 266)	Wrist Roll
θ_5	200	(-100 to 100)	Wrist Bend
θ_6	532	(-266 to 266)	Wrist Swivel

Table-.2: Distance between adjacent links

d_1	d_2	d_3	d_4	d_5	d_6
0	140	0	420	0	0
0	140	0	420.01	0	50.25
0	149.9	0	433.007	0	56.25
0	145	0	420	0	55
0	160	0	435	0	60

Table -3: Normal Orientation vector in all coordinates

n_x	n_y	n_z
1	0	0
0.17	0.173	-.968
0	0	1
-0.334	0.17	-0.92
0	0.5	-0.866

IV. SIMULATION CURVE

The performance of neural network is discussed and 16 nodes are selected in the hidden layer. The MSE (mean square error) of the model for 1000 epochs for θ_1 to θ_6 are represented in Fig.3. In this figure the best validation performance is analysed properly. The Graph for matching desired and predicted values are shown in Fig.4.

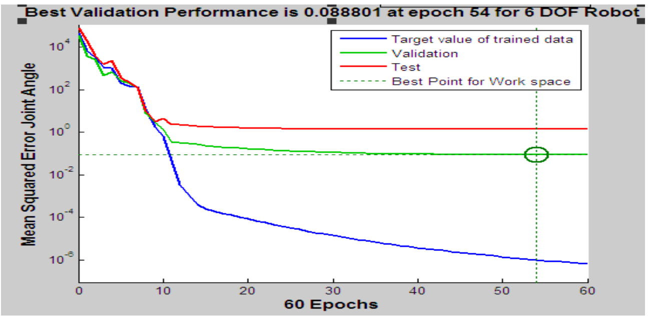

Fig.3. Mean Square Error

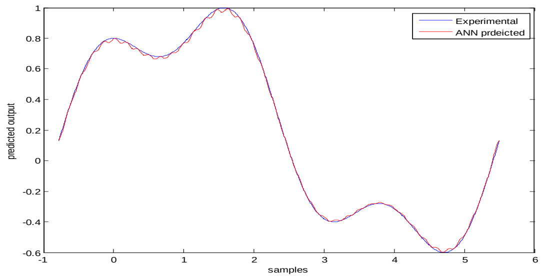

Fig.4. Graph for matching desired and predicted values

Figure-4 Nx, Ny, Nz shows the training convergence graphs. From the Figures it is clearly visible that convergence is achieved using very few epochs. In our case 3 layer Neural Network structure has been used. Log sigmoid function has been used in the hidden layer and output layer user uses a linear activation function. 15 numbers of hidden neurons have been used for simulation. The eerror histograms with 20 bins are presented in Fig.5. Fig.6 shows the performance curve of epochs.

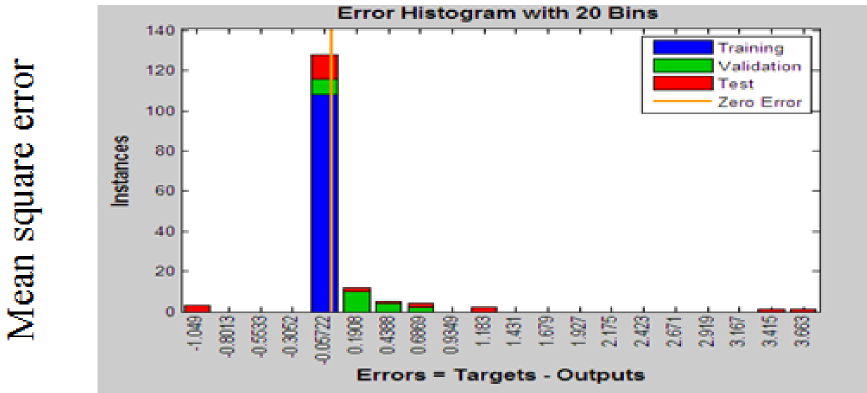

Fig.5. Error Histogram

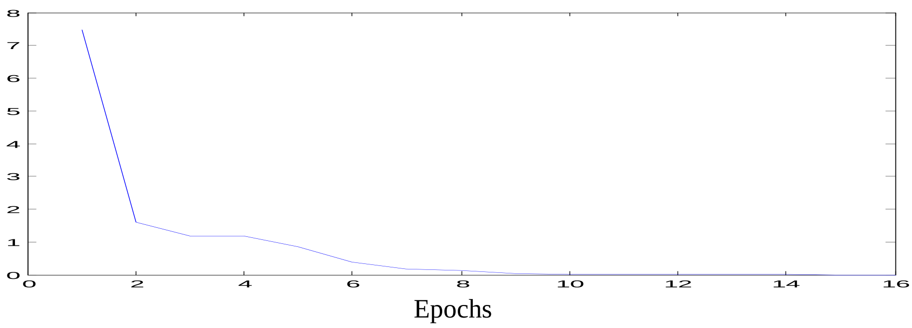

Fig.6. Performance curve

The number of epochs used in every graph and learning of the proposed network is done using Levenberg Marqdt (LM) algorithm which is very fast.

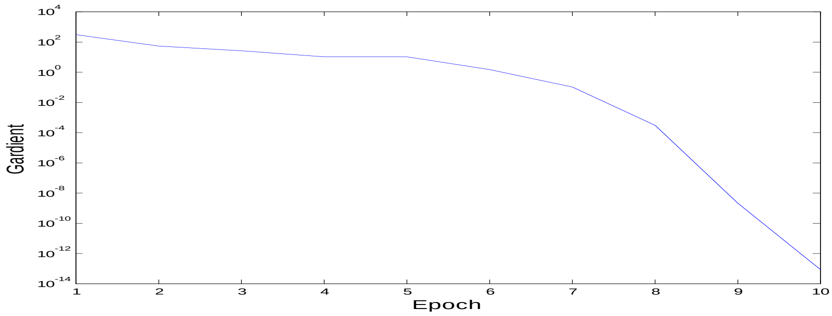

Fig.7. Gradient curve

The figure 7 shows how the error gradient decreases during learning period. The learning rate of the training is adapted for different epochs. The figure 8 shows the change of the learning rate with the epochs.

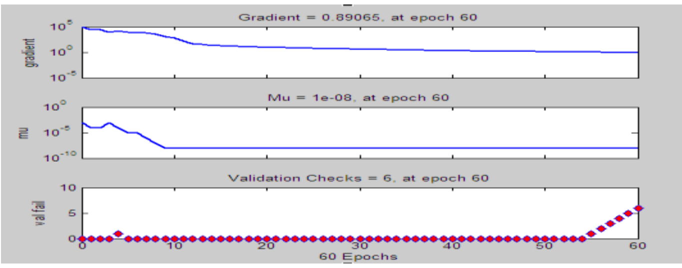

Fig.8. Testing data

V. RESULTS AND DISCUSSIONS

The main drawback of using neural networks to approximate the inverse kinematics of robot arms is the high number of training samples required to attain an acceptable precision. After training 60 % of the input output vector pairs of the data set, the network was tested on the 40 % of the pairs of data set that are not included in the training set. To obtain a better idea for the performance of the neural network prediction, the errors of joint angles for 15 test points were find out. Since error values fluctuate around the origin, effective error, root mean square error, is calculated for each angle as $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$ *and* θ_6 respectively. Note that, the biggest effective error is found for θ_3 .In fact, when the analysis is examined, the worst prediction performance and bigger error amplitudes are observed for third joint angle.

REFERENCES

[1]. Max Suell Dutra "New technique for inverse kinematics problem using Simulated Annealling" International conference on Engineering Optimization , 2008.

[2]. Rasit Ko ker " Reliability-based approach to the inverse kinematics solution of robots using Elman's networks" Engineering Applications of Artificial Intelligence , 18 , pp-685–693, 2005.

[3]. Jose Antonio, Martin H, Javier de Lope b, Matilde Santos "A method to learn the inverse kinematics of multi-link robots by evolving neuro-controllers" Neurocomputing , 72, pp-2806–2814, 2009.

[4]. Hanlei Wang and Yongchun Xie "Adaptive inverse dynamics control of robots with uncertain kinematics and dynamics" Journal of Automatica 45, pp-2114-2119, 2009.

[5]. Ali T. Hasan, A.M.S. Hamouda , N. Ismail, H.M.A.A. Al-Assadi "An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 D.O.F serial robot manipulator" Advances in Engineering Software 37 ,pp-432–438,2006.

[6]. Krzysztof Tchon and Janusz Jakubiak, " Extended Jacobian inverse kinematics algorithm

[7]. for nonholonomic mobile robots" International Journal of Control , Vol. 79, No. 8, pp-895–909, August 2006.

[8]. Rasit Koker, Cemil Oz, Tarık Cakar, Hueyin Ekiz, "A study of neural network based inverse kinematics solution for a three-joint robot", Robotics and Autonomous Systems 49 ,pp- 227–234, 2004.

[9]. P. Kalra , P.B. Mahapatra,D.K. Aggarwal "An evolutionary approach for solving the multimodal inverse kinematics problem of industrial robots, Mechanism and Machine Theory 41 ,pp- 1213–1229,2006.

[10]. K.K. Dash, B.B.Choudhury, A.K. Khuntia and B.B.Biswal "A Neural Network Based Inverse Kinematic Problem" IEEE, pp-471-476,2011.