ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, Vol 9(21), DOI: 10.17485/ijst/2016/v9i21/90570, June 2016 ISSN (Online) : 0974-5645

Solution for a Five Link Industrial Robot Manipulator Inverse Kinematics Using Intelligent Prediction Response Method

A. Siddique Ahmed Ghias[1], M. Dev Anand[2], I. Jacob Raglend[3] and F. Shamila[3]

[1]Department of Mechanical Engineering Agni College of Technology, Thalambur, Off OMR, Chennai – 603103, Tamil Nadu, India; ghias.mec@act.edu.in

[2]Department of Mechanical Engineering, Noorul Islam Centre for Higher Education, Kumaracoil - 629 180, Thuckalay, Kanyakumari, Tamil Nadu, India; anandpmt@hotmail.com

[3]Department of EEE, Noorul Islam Centre for Higher Education, Kumaracoil - 629 180, Thuckalay, Kanyakumari, Tamil Nadu, India; jacobraglend@rediffmail.com, shamisalim.60@gmail.com

Abstract

Background: Robot kinematics suggests and interprets the relationship exists between the kinematic position connectivity and acceleration of each link. In any of the robot manipulator the kinematics solution may be forward kinematics or inverse kinematics. To determine the joint values for a provided desired end effector orientation and position, the inverse kinematics principle is applied. Inverse kinematics is the usage of kinematics equations of a robot to find out the joint parameters that gives a targeted position of the end-effector. **Methods/Statistical Analysis:** In this paper, inverse kinematic solution for a five joint robot involving intelligent prediction response method has been developed and the result will be analyzed based on the performance. The intelligent prediction response method will give the performance based result, which shows the five various angles of an industrial robot. The single variation in the joint angle will be analyzed for every joint angles. This method based inverse kinematics solution is much more useful in real-time adaptive robot control where shorter calculation times are required. **Findings:** The MATLAB 13.0 is used to find the solution for a set of joint parameters. The actual reading and MATLAB program was found acceptable level. **Applications/Improvements:** To solution provided for the inverse kinematics problem with number of joint angles using the intelligent prediction response method such as artificial neural network is used.

Keywords: ANN, Inverse Kinematics, Position, Robot Control

1. Introduction

Robotic system possess high complexity and highly nonlinearity characteristics. This complexity makes the mathematical model to be arise for robot kinematics. For functional trajectory tracking and offline and online program generation requires this kinematic model. In robotics one of the important issue in solving is the inverse kinematics issue. Robot kinematics suggests and interprets the relationship exists between the kinematic position connectivity and acceleration of each link. In any of the robot manipulator the kinematics solution may be forward kinematics or inverse kinematics. In forward kinematics, mathematical equations are derived for computing the end-effector location of the robotic manipulator from a particular values for the joint parameters. The reverse process will give the inverse kinematics solution. That means inverse kinematics specifies the location and associated joint angles.

For solving the inverse kinematics issue the intelligent prediction response method such as artificial neural network has been utilized. Five joint robot's analytical solution for using inverse kinematics is learnt on assessing with the output of intelligent prediction response method.

In earlier research work, multiple robots having repetitive motion was formulated using recurrent NN

* Author for correspondence

and it is illustrated. To determine the status of repetitive motion planning of robot manipulator which is redundant online a dual neural network is used[1].

For path problem which is shortest solving a neural network coupled with Self-Adaptive Auto Wave Pulse (SAAWP) are utilized. In SAAWP coupled neural network auto waves are generated as per the current network state that propagates adaptively. To determine the least feasible path, neural network coupled with self-adaptive auto wave pulse guarantees as it spreads more effectively.

Inverse kinematics solution for a five DOF industrial robot manipulator is found by using a intelligent prediction response method. First of all by using cubic trajectory planning in the volume of work of the robotic manipulator, generated numerous initial and final points. Then based on the real world coordinates (x, y, z), all of the angles are recorded in a file. And the file is named as set to train neural network. Designed neural network will give the correct angles based on t the given Cartesian coordinates[2].

Scan matching is implemented for robot localization using harmony search algorithm which gives better accuracy compared with genetic algorithm. Orientation and positioning is automatically identified using the hybrid algorithm. This technique has the drawback that it has slow convergence speed[3–10].

2. Problem Solving

Inverse kinematics solution for a m degree of freedom manipulator requires mathematical modeling. A five joint robot manipulator used in this study is given in Figure 1, which shows the links and joints.

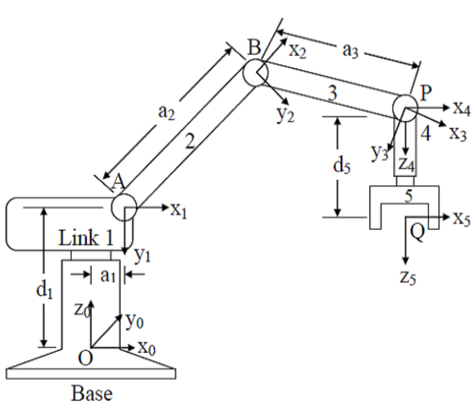

Figure 1. Five joint manipulator arm links and joints.

In the problem solving, in inverse kinematics various joint angles must be calculated at different position for making the data set. The Denavit Hartenberg convention and transformation matrix is used to find out the mathematical equations for finding the different values of angle (θ). The Denavit Hartenberg transformation matrix from joint *i* to *i+1* is given as:

$$\begin{bmatrix} C\theta i & -S\theta i & 0 & ai-1 \\ S\theta iC\alpha i-1 & C\theta iC\alpha i-1 & -S\alpha i-1 & -S\alpha i-1di \\ S\theta iS\alpha i-1 & C\theta iS\alpha i-1 & C\alpha i-1 & C\alpha i-1di \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The inverse of a transformation matrix can be given as:

$$\begin{bmatrix} {}_{23} & -{}_{23} & 0 & {}_1+{}_2{}_2+{}_3{}_{23} \\ 0 & 0 & 1 & d_2 \\ -S_{23} & -C_{23} & 0 & -a_1S_2+a_3S_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

In order to determine θ_1 element (2, 4) in the matrix equation (1) can be used.

$$d2 = -\sin(\theta_1)Px + \cos(\theta_1) \quad (2)$$

Letting,

$$Px = \rho\cos(\varnothing) \quad (3)$$

$$Py = \rho\sin(\varnothing) \quad (3)$$

Where,

$$\rho = \sqrt{p_x^2 + p_y^2} \quad (5)$$

$$\varnothing = \text{Arctan } 2(Px, Py) \quad (6)$$

We know that,

$$\frac{d_2}{\rho} = -\sin(\theta_1)\cos(\phi) + \cos(\theta_1)\sin(\phi) \quad (7)$$

Therefore,

$$\sin(\phi - \theta_1) = \frac{d_2}{\rho} \quad (8)$$

$$\cos(\phi - \theta_1) = \sqrt{1 - \frac{d_2^2}{\rho^2}} \quad (9)$$

$$\phi - \theta_1 = arc\ tan2\left(\frac{d_2}{\rho}, \pm\sqrt{1 - \frac{d_2^2}{\rho^2}}\right) \quad (10)$$

This leaves the solution for θ_1 as shown in,

$$\theta_1 = arc\tan 2(py, px) - \text{arc}\tan 2\left(d_2, \pm\sqrt{p_2^2 + p_y^2 - d_2^2}\right) \tag{11}$$

In order to determine joint angle three we must evaluate the elements from (1, 4) and (3, 4) of equation (1). First

$$C_1px\ S_1\ py = a_3C_{23} + a_2C_2 + a_1 \tag{12}$$

$$pz - d1 = a_3S_{23} - a_2S_2 \tag{13}$$

Next by squaring equations 12 and 13 followed by addition θ_3 can be determined as;

$$\theta_3 = \pm arc\cos\left(\frac{\left(C_1p_x + S_1p_y - a_1\right)^2 + \left(d_1 - p_z\right)^2 - a_2^2 - a_3^2}{2a_2a_3}\right) \tag{14}$$

By placing the $\frac{1}{2}T$ matrix on the left side of the equation (1) gives the transformation matrix as:

$$\begin{bmatrix} C_3 & -S_3 & 0 & a_3C_3 + a_2 \\ S_3 & C_3 & 0 & a_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

Evaluating elements (1, 4) of equation (15) the following formula are left;

$$C_1C_2px + S_1C_2py - S_2pz + d_1S_2 - C_2a_1 = a_3C_3 + a_2 \tag{16}$$

$$C_2(C_1px + S_1py - a_1) + S_2(d_1 - pz) = a_3C_3 + a_2 \tag{17}$$

Assigning, $A = (C_1px + S_1py - a_1)$, $B = (d_1\text{-}pz)$ and $C = a_3C_3 + a_2$ it is easy to solve θ_2 via reduction to a polynomial. Equation (17) becomes

$$C_2A + S_2B = C \tag{18}$$

Where

$$C_2 = \frac{1 - U^2}{1 + U^2} \tag{19}$$

$$S_2 = \frac{2U^2}{1 + U^2} \tag{20}$$

Substituting equation (19) and equation (20) in equation (18), equation (21) is obtained

$$(C+A)U^2 - 2UB + (C - A) \tag{21}$$

Using the quadratic formula rendering equation (21) can be solved as;

$$U = \frac{B \pm \sqrt{B^2 + A^2 - C^2}}{A + C} \tag{22}$$

Where,

$$\theta_2 = 2\ \text{arc}\tan(U) \tag{23}$$

The fourth joint angle, θ_4 , can also be determined simply based on θ_2 and θ_3 .

$$\theta_4 = 90 - \theta_2 - \theta_3 \tag{24}$$

The angle θ_5 remain unchanged in this case.

3. Methodology

Five joint robot's inverse kinematic solution can be studied utilizing the intelligent prediction response method.

3.1 Artificial Neural Network

Artificial Neural Networks (ANN) are basically a system processing processing with highly interconnected processing elements that simulates the behavioual aspects of a biological neural network with certain assumptions. ANN has highly interconnected processing elements which are organized into layers. ANN is a network with many units interconnected by communication channels which channels which carry information encoded by various means. This units operate with the inputs received from the interconnections and the local information is modified durinfg the learning phase.The operation of ANN involves learning of data through a training rule where the strength of the connections called weights are adjusted. The weights and threshold of the processing elements can automatically adjusted by repeatedly applying numerical data and integrating the algorithm for learning. Network is said to be trained when the minimal difference among the neural network output and the targeted output once got adjusted.

3.2 Feed Forward Back Propagation Algorithm

In the feed forward neural network every level's output is provided as the next level's input. The input layer's input signal is provided to the hidden layer and the output layer provides the output. Figure 2 illustrates feed forward neural network model that possess 3 inputs, 20 hidden units and 5 output units. Here the corresponding activation function in the hidden and output layer must be chosen.

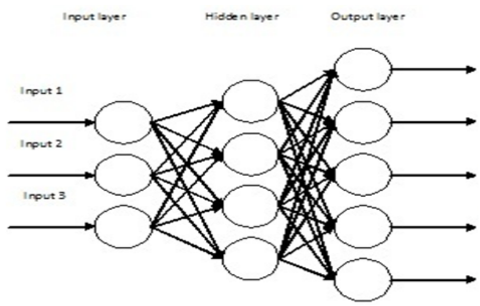

Figure 2. Feed Forward Neural Network Model.

The standard back propagation algorithm is given as follows:

- Generate the weights randomly to small random values to make sure that the network is not saturated by large values of weights and choose the number of input, hidden and output layers.
- Initially, a single training pair is chosen at random.
- Input pattern of the system is transferred to the input layer.
- Calculate the network output.
- Determine the error, the difference among the network output activation and the desired output.
- Adjusts the network weight for minimizing the error.
- Repeat the above steps for each input-output pair in the training set till the error for the whole system is within the tolerable acceptance limit.

The output is typically achieved from the system's output layer. The system error is the deviation among the value of target and output and isfedback to the connection weights.

First,determine the updated weights from output layer to the hidden layer;

$$\Delta Wkj\text{˘ň˘}+ \quad = \eta\delta kOUTk + \alpha\ \Delta Wkj\ n \tag{25}$$

$$WNEWkj = WOLDkj + \Delta Wkj(n+1)] \tag{26}$$

Where, TARGET_k refers desired output value, the learning rate is η, the coefficient of momentumis α, activation function is f‘(NETk), iteration number is n and related weight change is W.

Next from hidden layer to the input layer;

$$\delta j = f`(NETj)\Sigma\delta kWkj \tag{27}$$

$$\delta j = OUTj(1-OUTj)\Sigma\delta kWkj \tag{28}$$

$$\Delta Wji\text{˘ň˘}+ \quad = \eta\delta jOUTj + \alpha\ \Delta Wkji\ n \tag{29}$$

$$WNEWji = WOLDji + \Delta Wji(n+1)] \tag{30}$$

To initiate the learning process, the bias influences the functionof activation whichincreases the speed of the process. The output layer biases is;

$$\Delta WBk\text{˘ň˘}+ \quad = \eta\delta k + \alpha\ \Delta WBk\ n \tag{31}$$

$$WNEWBj = WOLDBj + \Delta WBk(n+1)] \tag{32}$$

The hidden layer biases is:

$$\Delta WBj\text{˘ň˘}+ \quad = \eta\delta j + \alpha\ \Delta WBj\ n \tag{33}$$

$$WNEWBj = WOLDBj + \Delta WBk(n+1)] \tag{34}$$

The iterative process is repeated until the deviation among TARGET and OUT is diminished.

4. Results and Discussion

The joint angles depend on the Cartesian coordinate system was analyzed using intelligent prediction response method such as ANN by using a various set of joint variables. The network were trained utilizing variousparameter values and could be simulated involving MATLAB 13.0.

The neural network was deigned involving fixed parameters like weights, activation function, bias, hidden neurons, network size, output neurons, momentum term, input neurons, learning rate. Choice of every parameters is performed by trial and error approach. Here all these parameters have been fixed so for achieving global minima. Weights initialization has been chosen such that no trapping into local minima is observed. Table 1 provides the mean square value of each joint angles.

Table 1. Mean Square Values of θ

Angles	Mean Square Error
θ_1	1.4336e-05
θ_2	0.0015
θ_3	0.0010
θ_4	0.0010
θ_5	0.0029

Table 2 provides the neural network training parameters and Figure 3 provides the performance polt of the trained neural network at epoch 14 with a performance value of $4.887e^{-10}$.

Table 2. Artificial Neural Network Training Parameters

Number of Iterations	Time	Performance	Gradient
14	0:00:04	$4.89e^{-10}$	$4.98e^{-06}$

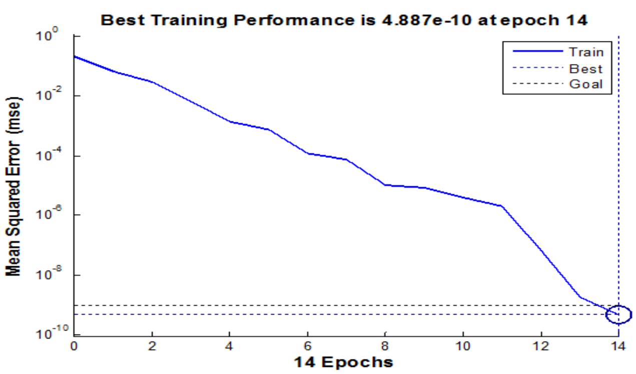

Figure 3. Performance plot of trained neural network.

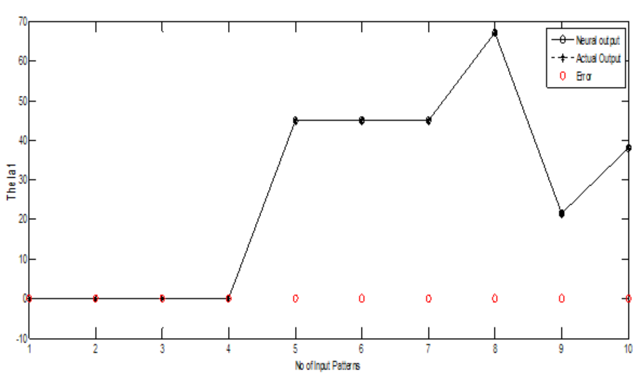

Figure 4. Angle Value for Link 1 (Theta 1).

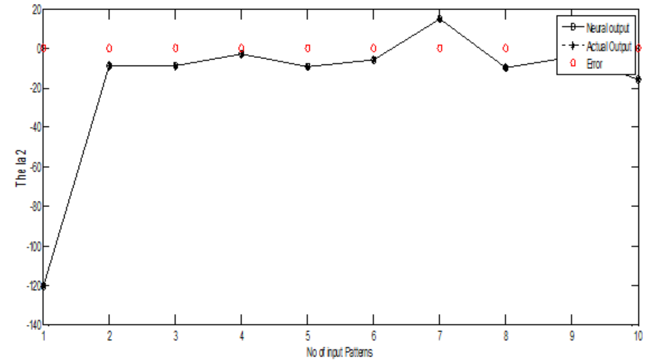

Figure 5. Angle Value for Link 2 (Theta 2).

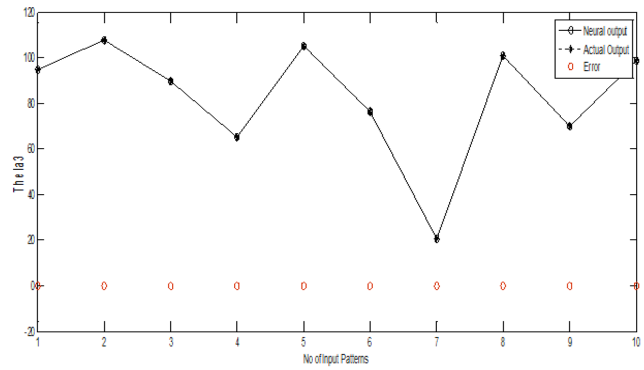

Figure 6. Angle Value for Link 3 (Theta 3).

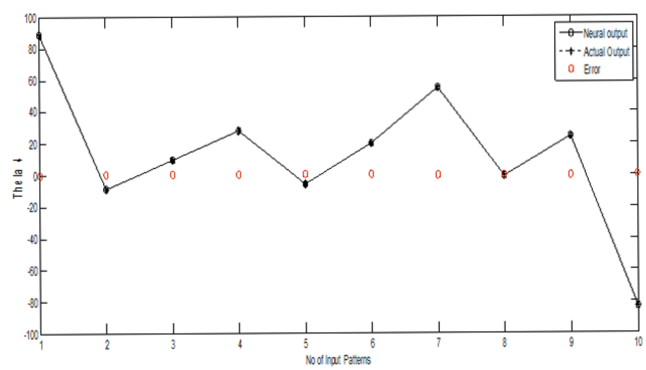

Figure 7. Angle Value for Link 4 (Theta 4).

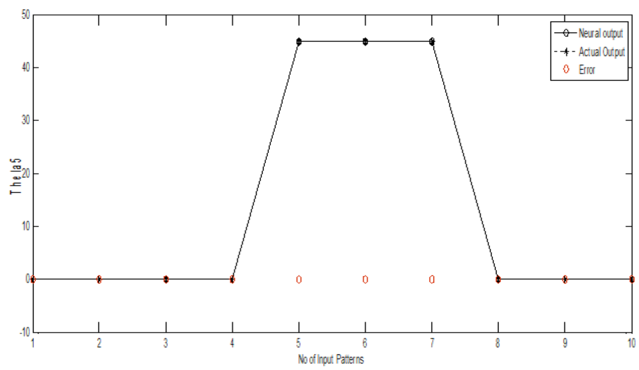

Figure 8. Angle Value for Link 5 (Theta 5).

5. Conclusion

In this paper the solution for a five joint robot manipulator is identified utilizing the intelligent prediction response method applying the inverse kinematics. The potential of intelligent prediction response method for the inverse kinematics problem solution for analyzing a five joint robot manipulator the functioning capability of the neural network. For enhancing the predicted joint accuracy and link angle values, we can increase the input patterns number of numerous coordinate sets with the joint and link values to be trained. Numerous Joint and link angles for different set of coordinates values has been achieved on involving the mathematical model and DH convention. These various set of coordinates values were provided to the network to train involving intelligent prediction response method with feed forward back propagation and the associated link and joint angle values are achieved with minimal error.

6. References

1. Zang Y, Tan Z, Chen K, Yang Z, Xuanijiao L. Repetitive motion of redundant robots planned by three kinds of recurrent NN and illustrated with a four-link planar manipulator's straight line example. Robotics and Autonomous Systems. 2009; 57(6–7):645–51.
2. Koker R, Oz C, Cakar T, Ekiz H. A study of neural network based inverse kinematics solution for a three joint robot. Robotics and Autonomous Systems. 2004; 49(3–4):227–34.
3. Mirkhani M, Forsati R, Mohammed Shari A, Moayedikia A. A novel efficient algorithm for mobile robot localization. Robotics and Autonomous Systems. 2013; 61(9):920–31.
4. Chen PCY, Mills JK, Smith KC. Performance improvement of robot continuous path operation through iterative learning using neural networks. Journal of Machine Learning. 1996; 23(2):191–20.
5. Clark CM, Mills JK. Robotic system sensitivity to neural network learning rate: Theory, simulation, and experiments. International Journal Robotics Research. 2000; 19(10):955–68.
6. Arras KO, Castellanos JA, Siegwart R. Feature based multi hypothesis localization and tracking for mobile robots using geometric constraints. Proceeding of the International Conference on Robotics and Automation (ICRA-02). Washington DC, USA. 2002. p. 1371–77.
7. Duffy J. Analysis of mechanism and robot manipulators. New York: Wiley; 1980.
8. Featherstone R. Position and velocity transformation between robot end effector coordinate and joint angle. International Journal Robotics Research. 1983; 2(2):35–45.
9. Galicki M. Path constrained control of a redundant manipulator in a task space. Robotics and Autonomous Systems. 2006; 54(3):234–43.
10. Padhy NP, Simon SP. Soft computing with MATLAB Programming. New Delhi: Oxford University Press; 2015.