

Solution of Inverse Kinematics for SCARA Manipulator Using Adaptive Neuro-Fuzzy Network

Wesam Mohammed Jasim

College of Computer, University of Anbar, Iraq
Wmj_r@yahoo.com

ABSTRAT

Solution of inverse kinematic equations is complex problem, the complexity comes from the nonlinearity of joint space and Cartesian space mapping and having multiple solution. In this work, four adaptive neuro-fuzzy networks ANFIS are implemented to solve the inverse kinematics of 4-DOF SCARA manipulator. The implementation of ANFIS is easy, and the simulation of it shows that it is very fast and give acceptable error.

KEYWORDS

Inverse Kinematics, Adaptive Neuro-Fuzzy, SCARA Manipulator.

1.INTRODUCTION

A robot manipulator is consisted of a set of links connected together by joints. The joints can either be very simple , such as a revolute joint or a prismatic joint, or they can be more complex, such as a ball and socket joint. A revolute joint is similar to a hinge and allows a relative rotation about a single axis, and a prismatic joint permits a linear motion a long a single axis, namely an extension or retraction. The difference between the two situations is that, in the first evidence, the joint has only a single degree-of-freedom of motion; the angle of rotation in the case of a revolute joint, and the value of linear displacement in the case of a prismatic joint. In contrast, a ball and socket joint has two degree-of-freedom.

A robot manipulator with n joints have n+1 links, since each joint connects two links. The joints are numbered from 1 to n, and links from 0 to n, starting from the base. By this convection, joint i connects link i-1 to link i. With the ith joint, a joint variable denoted by q_i . In the case of a revolute joint, q_i is the angle of rotation, and in the case of a prismatic joint, q_i is the joint displacement:

$$q_i = \begin{cases} \theta_i & \text{if joint is revolute} \\ d_i & \text{if joint is prismatic} \end{cases}$$

The problem of kinematic is to demonstrate the motion of the manipulator without consideration of the forces and torques causing the motion. There are two main kinematic problems. First one is forward kinematic problem, which is to determine the position and orientation of the end effector given the values for the joint variables of the robot. The second one is inverse kinematic problem is to determine the amounts of the joint variables given the end effector's position and orientation [1]. At least 6 parameters are needed to describe the motion of the rigid body. Three parameters are for position, three parameters are for orientation. Several methods are used in robot kinematic. The most common method is Denavit and Hartenberg DH notation for definition of special mechanism. This method is based on point transformation approach and it is used 4×4 homogeneous transformation matrix which is introduced by Maxwell. Maxwell used homogeneous coordinate systems to represent points and homogeneous transformation matrices to represent the transformation of points. The coordinate systems are described with respect to previous one. For the base point an arbitrary base coordinate system is used. Hence some singularity problems may occur because of this coordinate systems description. And also in this method 16 parameters are used to represent the transformation of rigid body while just 6 parameters are needed[2].

The inverse kinematics determines the joint variables that would result in a desired position of the end-effector of the manipulator with respect to a reference coordinate system. The inverse kinematics solution is difficult since the mapping between the joint space and Cartesian space is non-linear and involves transcendental equations having multiple solutions. A unique solution may be obtained in such cases if a performance criterion, like total joint displacement minimization, is incorporated in the solution scheme[3].

There are several methods for solving Inverse Kinematics problems, YangshengXu, and Michael C. Nechgba, discusses the automatic generation of the Fuzzy Inverse Kinematic Mapping (FIKM) from specification of the DH parameters, the efficiency of the scheme in comparison to conventional approaches, and the implementation results for both redundant and nonredundant robots [4]. Ramakrishnan M. present a novel single-pass algorithm that is fast and eliminates problems related with improper and large angle rotations [5]. E. Sariyildiz, and H. Temeltas, present a formulation based on screw theory with dual-quaternions for both forward and inverse kinematic equations [2]. P. Kalra and etc. use real-coded genetic algorithms to obtain the inverse kinematics solution of an articulated robotic manipulator [3]. Samuel R. Buss discuss the solution of inverse kinematic with jacobian transpose, pseudoinverse and damped least squares methods [7]. Takehiko Ogawa and Hajime Kanada, propose a network inversion as a method for solving inverse kinematics problem of a robot arm with multiple joints, where the joint angles are conjectured from the given end-effector coordinates [8]. Panchanand J., apply an artificial neural network models to solve both the direct and indirect kinematics of robot manipulator motion [9].

In this paper a Neuro-fuzzy structure is presented to solve the inverse kinematics of the SCARA (Selective Compliant Articulated Robot for Assembly) manipulator, which, as its name suggests, is tailored for assembly operations.

2.SCARA MANIPULATOR

The SCARA manipulator has an (Revolute, Revolute, Prismatic) RRP structure, it has kinematics of four degrees of freedom 4-DOF, the four degrees manipulator is different to others.

SCARA manipulator so different from the spherical manipulator in both appearance and in its range of applications[6].

A 4-DOF SCARA manipulator has parallel shoulder, elbow, and wrist rotary joints, and a linear vertical axis through the center of rotation of the wrist. This type of manipulator is very common in light-duty applications such as electronic assembly, figure 1 shows the shape of SCARA manipulator and its framework, and the DH joint parameters are given in Table (1) [1].

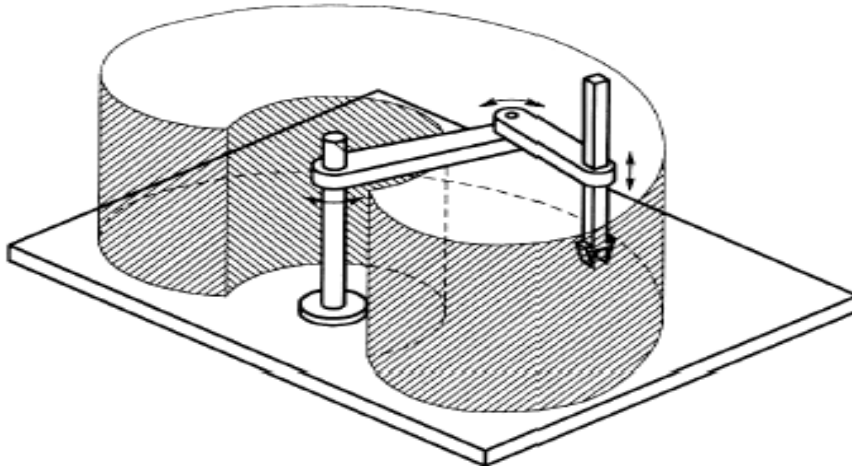


Figure 1. DH coordinate frame assignment for the SCARA manipulator.

Link	a_i	α_i	d_i	θ_i
1	a_1	0	0	θ^*
2	a_2	180	0	θ^*
3	0	0	d^*	0
4	0	0	d_4	θ^*

Table 1. DH Joint parameters for SCARA

And the A-matrices are as follows.

$$A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & a_1 c_1 \\ s_1 & c_1 & 0 & a_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$A_2 = \begin{bmatrix} c_2 & s_2 & 0 & a_2 c_2 \\ s_2 & -c_2 & 0 & a_2 s_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$A_4 = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$T_4^0 = A_1 \dots A_4 = \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix} \quad (5)$$

The inverse kinematic equations are:

$$\theta_1 = \text{atan2}(O_x, O_y) - \text{atan2}(a_1 + a_2 c_2, a_2 s_2) \quad (6)$$

$$\theta_2 = \text{atan2}(c_2, \pm \sqrt{1 - c_2^2}) \quad (7)$$

$$\theta_4 = \theta_1 + \theta_2 - \text{atan2}(r_{11}, r_{12}) \quad (8)$$

$$d_3 = O_z - d_4 \quad (9)$$

$$c_2 = \frac{O_x^2 + O_y^2 - a_1^2 - a_2^2}{2a_1 a_2} \quad (10)$$

Where, $a_1 = 20 \text{ cm}$ (length of link 1), $a_2 = 15 \text{ cm}$ (length of link 2), $c_1 = \cos\theta_1$, $c_2 = \cos\theta_2$, $s_1 = \sin\theta_1$, $s_2 = \sin\theta_2$, and $d_4 = 10 \text{ cm}$.

3. NEURO-FUZZY STRUCTURE

A typical architecture of an ANFIS is shown in figure 2, in which a circle indicates a fixed node, whereas a square indicates an adaptive node. The end-effector position (X, Y, Z) are the inputs and the outputs are the joint variables ($\theta_1, \theta_2, d_3, \theta_4$). Among FIS models, the Sugeno fuzzy model is applied because its high interpretability and computational efficiency. The learning algorithm tunes the membership functions using the training input-output data. implementation of a representative fuzzy inference system using a BP neural network-like structure. The fuzzy if-then rules expressed as:

If x is A and y is B and z is C then $K = px + qy + sz + r$

Where A, B and C are the fuzzy sets in the antecedent, and p, q, s and r are the design parameters that are determined during the training process.

As in figure 2, the ANFIS comprises of five layers. The role of each layer is briefly presented as follows:

Let O_i^l denote the output of node i in layer l , and x_i is the i th input of the ANFIS,

$i = 1; 2; \dots; p$ [10].

Layer 1:

Every node i in employ a node function R given by:

$$O_i^1 = R_i(x_i)$$

Where R_i can adopt any fuzzy membership function (MF).

Layer 2:

Every node calculates the firing strength of a rule via multiplication:

$$O_i^2 = w_i = \min (R_i)$$

Where w_i represent the activation level of a rule.

Layer 3:

Fixed node i in this layer calculate the ratio of the i -th rules activation level to the total of all activation level:

$$O_i^3 = \bar{w}_i = \frac{w_i}{\sum_i w_i}$$

Where \bar{w}_i is referred to as the normalized firing strengths.

Layer 4:

Every node i has the following function:

$$O_i^4 = \bar{w}_i K_i = \bar{w}_i (p_i x + q_i y + s_i z + r_i)$$

Where \bar{w}_i is the output of layer 3, and (p_i, q_i, s_i, r_i) is the parameter set. The parameters in this layer are referred to as the consequence parameters.

Layer 5:

The single node in this layer computes the overall output as the summation of all incoming signals, which is expressed as:

$$O_i^5 = \bar{w}_i K_i = \frac{\sum_i w_i K_i}{\sum_i w_i}$$

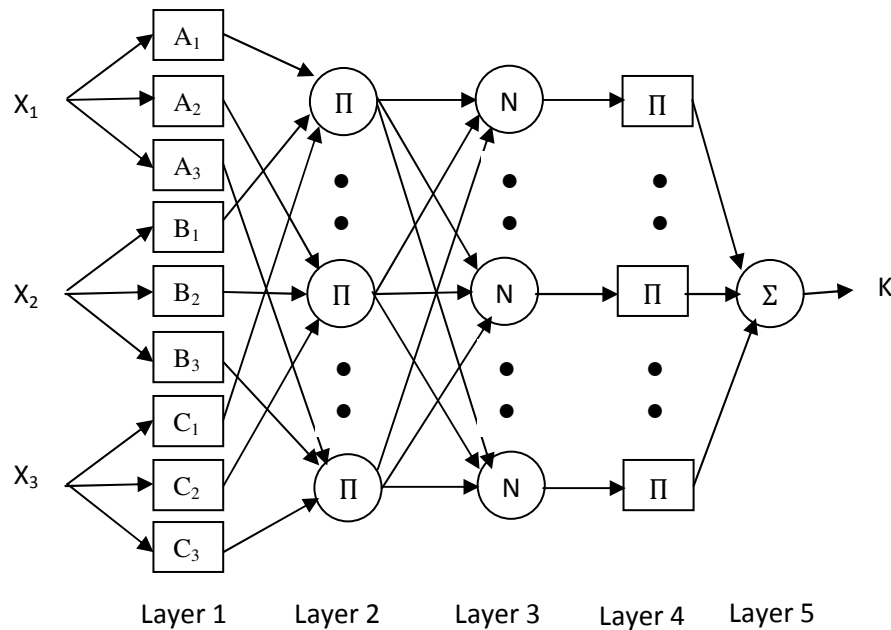


Figure.2 .Neuro-fuzzy (ANFIS) architecture.

4. COMPUTER SIMULATION AND RESULTS

Four different ANFIS are designed for the computer simulation, for solving the inverse kinematics parameters of 4-DOF SCARA manipulator θ_1, θ_2, d_3 and θ_4 respectively, each one have three Gaussian membership functions in each input and 27 rules in second layer.

The adaptation process of the parameters of the ANFIS has two steps. For the first step of the consequent parameters training, the Least Squares method is used, because the outcome of the ANFIS is a linear integration of the consequent parameters.

The premise parameters are static at this step. After the consequent parameters have been regulated, the approximation error is back-propagated through every layer to update the premise parameters as the second step. This part of the adaptation procedure is based on the gradient descent principle, which is similar to the training of the Back-Propagation BP neural network.

A MATLAB package 7.8.0 R2009a is used to implement the simulation code.

Figure 3 shows the error of θ_1 , θ_2 and θ_4 , figure 4 shows the error of d_3 , figure 5, figure 6, figure 7 and figure 8, shows the testing data of four ANFIS networks for θ_1 , θ_2 , d_3 and θ_4 respectively.

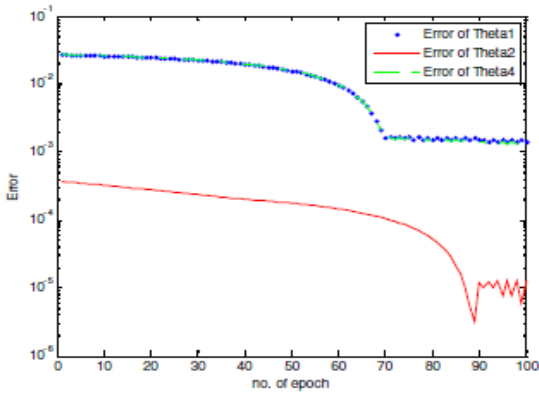


Figure 3. Error of Theta 1 , 2 and 4.

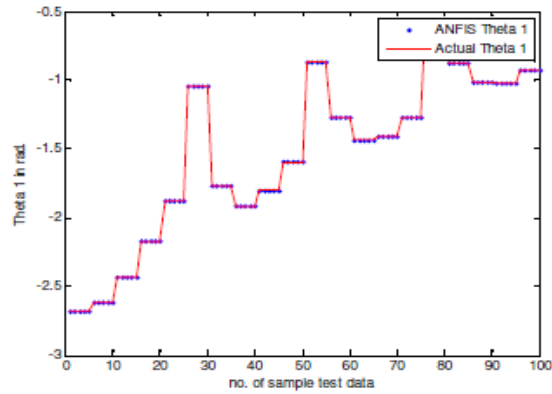


Figure 5. Test of Theta 1

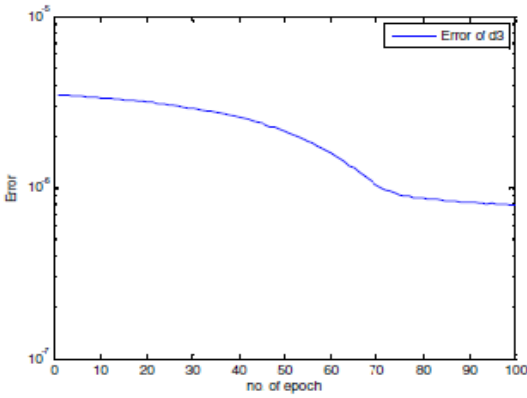


Figure 4. Error of d_3 .

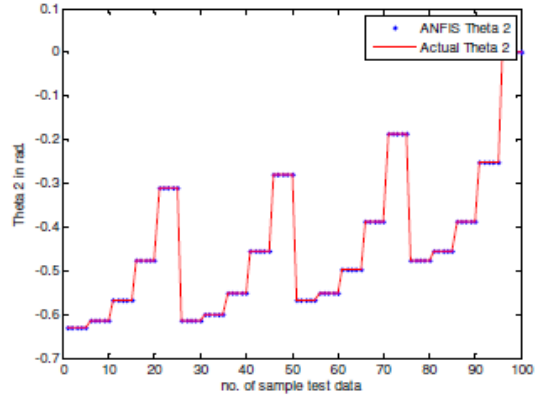


Figure 6. Test of Theta 2

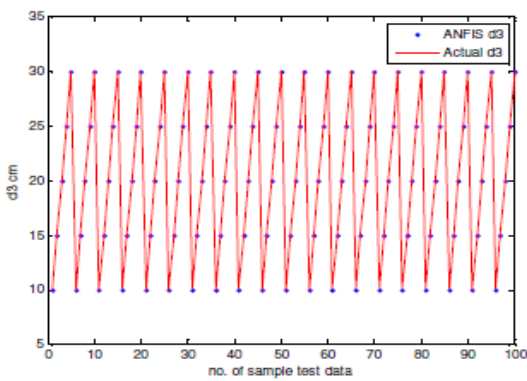


Figure 7. Test of d_3 .

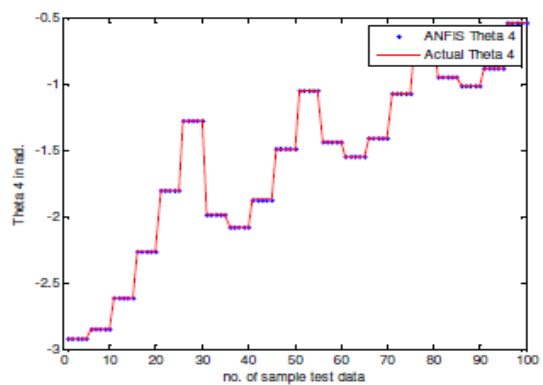


Figure 8. Test of Theta 4.

5. CONCLUSIONS

This work demonstrate that is possible to use Simulated ANFIS in order to present a very good and reliable solution to the Inverse kinematics problem. The results of used ANFIS shows that it have good and acceptable training error and small number of epoch in training all inverse kinematic parameters of 4-DOF SCARA manipulator. At the same time, is important to say that this technique is easy because its very simple to implement and calculating. Then use of ANFIS give fast and acceptable solution of inverse kinematic parameters.

6. REFERENCES

- [1] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar, 2006, “ Robot Modeling and Control “ John Wiley & Sons, New York.
- [2] E. Sariyildiz, and H. Temeltas, 14-17 July, 2009, “ Solution of Inverse Kinematic Problem for Serial Robot Using Dual Quaternion and Plucker Coordinates “ IEEE/ASME, Singapore, pp.338-343.
- [3] P. Kalra, P.B. Mahapatra, and D.K. Aggarwal, 2003, “ On the Solution of Multimodel Robot Inverse Kinematic Functions using Real-coded Genetic Algorithms “ IEEE, pp. 1840-1845.
- [4] YangshengXu and Michael C. Nechgba, 1993,“ Fuzzy Inverse Kinematic Mapping “ IROS Conference.
- [5] RamakrishnanMukundan, , 2008, “ A Fast Inverse Kinematics Solution for an n-link Joint Chain” ICITA, pp. 349-354.
- [6] Victor H. and etc., October 2010,“ Kinematics for the SCARA and the Cylindrical Manipulators “ ICIC, Vol. 4, No. 5, pp. 1-6.
- [7] Samuel R. Buss, October, 2009, “ Introduction to inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods “ University of California press, pp. 1-19.
- [8] Takehiko O. and H. Canada, 2010, “Solution for Ill-Posed Inverse Kinematics of Robot ArmbyNetwork Inversion “ Hindawi Publishing Corporation, Journal of Robotics, Volume 2010, doi:10.1155/2010/870923.
- [9] Panchanand J., 2009, “Novel Artificial Neural Network Application for Prediction of Inverse Kinematics of Manipulator” M.Sc. Thesis, National Institute of Technology, India.
- [10] Dr. Bob John “Adaptive Network Based Fuzzy Inference Systems (ANFIS) “www.cse.dmu.ac.uk/~hseker/ANFISnotes.doc

Author

Wesam Mohammed Jasim is presently working as deputy dean in Computer College, University of Anbar, Ramadi, Iraq, and completed B.Sc. and M.Sc. in Electrical and Electronic Engineering/Control at 1996 and 2002 respectively, Al-Rasheed College for Engineering and Science, University of Technology, Baghdad, Iraq.

