Implementation Of 5 Degree Of Freedom Robotic Arm Manipulator With ANFIS Algorithm

Sandip A. Mehta
Assistant Professor
Nirma University, Ahmedabad

Jatin Patel
Mtech Control and Automation
Nirma University, Ahmedabad

Abstract

Inverse kinematics is difficult in computation and can result in significant delays in real time. For a redundant robot, additional computations are required for the inverse kinematic solution through optimization schemes. Getting inverse kinematics result using equation would be considered under expensive efforts. So this optimization can be done efficiently by fuzzy logic. Based on the fact that exact inverse kinematics cannot be calculated by human, but can do defined positioning from trial and error. The implementation of the projected scheme has established that it is practical for both redundant and non redundant cases, and that it is very computationally efficient. The result provides sufficient precision. This paper discusses (1) the automatic generation of the Fuzzy Inverse Kinematic Mapping (FIKM) from specification of the Denavit-Hartenberg parameters (2)Programming that in microcontroller and implementation on hardware.

1. Introduction

People who work efficiently in complex, unstructured environments acquire their skills through various kinds of learning. It may be necessary to implement similar abilities in robots to enable them to work in the same kinds of environments. The task of calculating all of the joint angles that would result in a specific position/orientation of an end effecter of a robot arm is called the inverse kinematics problem[4]. An inverse kinematics solver using fuzzy mapping that learns the inverse kinematics system of a robot arm has been used in much research. In the case of redundant manipulators and non redundant manipulators in singular configurations, the problem is compounded by the fact that throughout the workspace of the manipulator, multiple solutions exist. The inverse kinematics of redundant manipulators therefore requires that a choice be made among the set of all possible solutions. Arriving at such a decision through some optimization scheme is difficult and the lengthy computations can result in However, Humans do not have to calculate exact inverse kinematics every time we move an arm or a leg. Experience and knowledge, rather than complex computations, allow humans to effectively move with ease. In this paper, we propose to characterize this human knowledge by proposing a general method of computing the inverse kinematics for an arbitrary n-DOF manipulator through a fuzzy logic approach[5]. Firstly, this paper, presents an algorithm which automatically generates the fuzzy model for an arbitrary manipulator based only on the Denavit-Hartenberg (DH) parameter. Second we use this parameter to implement real robotic arm which is designed using servo motors and controlled by PIC18f4550 chip. To obtain a precise inverse kinematics model of a robot with fewer degrees of freedom, some modifications are necessary. We modified the Gauss-Newton method for finding the joint angle vector trajectory from the initial posture of the arm to the given desired end-effecter position/orientation. By the modifications proposed in this paper, the fuzzy logic mapping system can obtain a precise inverse kinematics model of a general robotic arm. Numerical experiments of the inverse kinematics learning were performed in order to evaluate the performance of the improved system.

2.FUZZY MAPPING

2.1 Overview

As shown in Fig. 1, our fuzzy inverse kinematic mapping (FIKM) takes as input the actual and desired locations of the end-effector, and the current joint variable values. From these inputs, the fuzzy controller generates as output the necessary trajectories for the joint variables, so that the actual and desired end-effector locations onverge to zero steady-state error.

The Jacobian matrix relates the differential Cartesian rates dr to the differential joint rates , such as

$$dr = J(\theta)d\theta \quad (1)$$

but we want to solve inverse problem

$$d\theta = J(\theta)^{-1}dr \qquad (2)$$

Consider each Jij term in the Jacobian separately along with dri the ith component of the dr vector. We define a new variable dij which relates dri and Jij.

$$J_{ij}d\theta_{ij}=dr \qquad (3)$$

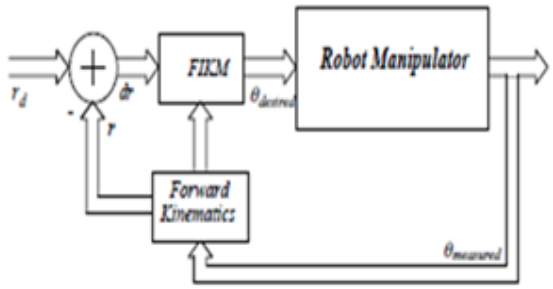


Fig. 1 signal flow for fuzzy controller.

Therefore, dij relates how much dθj contributes todri . This relationship gives a good understanding of which joints will contribute more to reducing dri and which ones will contribute less. Thus with proper scaling of each of the d$\theta$is the fuzzy mapping can arrive at an intelligent set of joint angles that will drive the end-effector to the desired position. The function that we will actually apply the fuzzy mapping to is given by,

$$d\theta_{ij} = \frac{dri}{Jij} \qquad (4)$$

2.2 JACOBIAN CALCULATION AND RANGE DETERMINATION

There are many computationally-efficient methods for calculating the forward Jacobian [6]. Orin and Schrader present several methods, one of which requires (30n - 55) multiplications, (15n - 38) additions, and (2n -2)
sine/cosine evaluations for both position and orientation tracking. Here and the twist angle_ in the DH parameters is restricted 0 or 90. In order to minimize the inference error of the fuzzy model, we want to fuzzify relationship over the full range of values thatJij may assume. Therefore it is useful to determine, before the fuzzy mapping, the range for each element Jij i of 1,2,3..... , , in J(θ) . Each Jij will be of the form,

$$J_{ij} = l_1f_1(\Theta_1,\ldots\ldots\ldots \Theta_n) + = l_2f_2(\Theta_1,\ldots\ldots\ldots \Theta_n)+\ldots.$$
$$+ l_kf_k(\Theta_1,\ldots\ldots\ldots \Theta_n)+ d_1f_{k+1}(\Theta_1,\ldots\ldots\ldots \Theta_n)+\ldots\ldots$$
$$+d_mf_{k+m}(\Theta_1,\ldots\ldots\ldots \Theta_n)$$

2.3 ANFIS ARCHITECTURE

This section introduces the basics of ANFIS network architecture and its hybrid learning rule. Adaptive Neuro-
Fuzzy Inference System is a feedforward adaptive neural network which implies a fuzzy inference system through its structure and neurons. Jang was one of the first to introduce ANFIS[11]. He reported that the ANFIS architecture can be employed to model nonlinear functions, identify nonlinear components on-line in a control system, and predict a chaotic time series. It is a hybrid neuro-fuzzy technique that brings learning capabilities of neural networks to fuzzy inference systems. The learning algorithm tunes the membership functions of a Sugeno-type Fuzzy Inference System using the training input-output data. A detailed coverage of ANFIS can be found in[1-2-3].

For a first order Sugeno type of rule base with two inputs x, y and one output, the structure of ANFIS is shown
in Figure 1. The typical rule set can be expressed as,

Rule 1: If x1 is A1 AND x2 is B1, THEN f1 = p1x+q1y+r1
Rule 2: If x1 is A2 AND x2 is B2, THEN f2 = p2x+q2y+r2

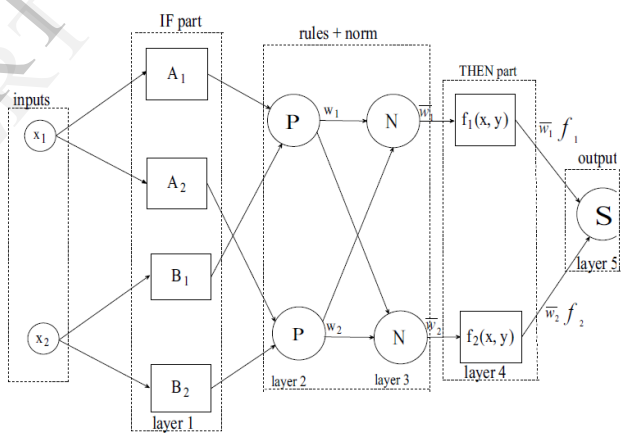


Fig. 1 ANFIS Architecture.

In the first layer, each node denotes the membership functions of fuzzy sets Ai,Bi, i = 1,2 be μAi(x1),μBi(x2). In the second layer the T-norm operation will be done related to AND operator of fuzzy rules. Considering T-norm multiplication:

$$w_i = \mu_{Ai}(x_1)* \mu_{Bi}(x_2) \qquad (4)$$

In the third layer, the average is calculated based on weights taken from fuzzy rules,

$$\overline{w}_i = \frac{wi}{w1+w2}$$

IJERT

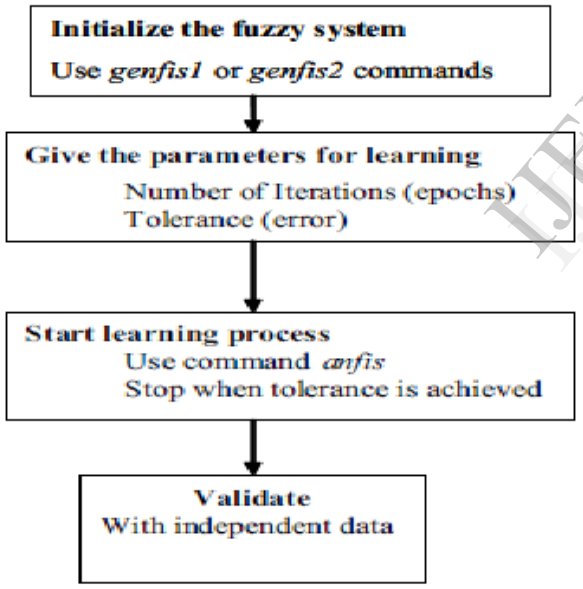
In the fourth layer, the linear compound is obtained from the input of the system as THEN part of Sugeno-type
fuzzy rules as,

$\overline{w}_i * f_i = \overline{w}_i \ (p_i x_1 + q_i x_2 + r_i)$

In the fifth layer, defuzzification process of fuzzy system (using weighted average method) is obtained by,

$$f = \sum fi . \overline{wi} = \frac{\sum wi.fi}{\sum wi}$$

This paper considers the ANFIS structure with first order Sugeno model containing 49 rules. Gaussian membership
functions with product inference rule are used at the fuzzification level. Hybrid learning algorithm that
combines least square method with gradient descent method is used to adjust the parameter of membership function.
The flowchart of ANFIS procedure is shown in Figure 3



.

Fig. 3 ANFIS procedure.

3 CALCULATION OF INVERSE KINEMATICS FOR THREE DEGREE OF FREEDOM.

Out of 5 Degree of freedom 3 degree of freedom is used for elbow, shoulder and joint. 1 Degree of freedom is used for 3 dimensions and last degree of freedom is used for gripping purpose.

So that's why we will calculate inverse kinematics for 3 dof and assume the 2 dimension plane is already given.

Forward kinematics equations for calculation of 3 dof.

$x = l1\cos(\theta_1) + l2\cos(\theta_1 + \theta_2) + l3\cos(\theta_1 + \theta_2 + \theta_3)$

$y = l1\sin(\theta 1) + l2\sin(\theta 1 + \theta 2) + l3\sin(\theta 1 + \theta 2 + \theta 2)$
$\phi = (\theta 1 + \theta 2 + \theta 2)$

From above equations we can find the coordinates for given constraint angles. After calculation, these all co-ordinates can relate it with individual angles. This input/output table we can be used to train the data for ANFIS network.

In verse Kinematics equations.

$\theta_2 = \text{atan2}(\sin \theta_2, \cos \theta_2)$
$\theta_1 = \text{atan2}(k_1 y_n - k_2 x_n), (k_1 x_n - k_2 y_n)$
$\theta_3 = \phi - (\theta_1 + \theta_2)$

where $K_1 = l_2$, $\sin \theta_2 \cos \theta_2 = \frac{x^2 + y^2 - l1^2 - l2^2}{2l1l2}$, $\sin \theta_2 \sqrt{1 - \cos^2 \theta 2}$, $X_n = x - l_3 \cos \phi$ and $y_n = y - l_3 \sin \phi$. For simulation, the length for three links are $l_1 = 10$, $l_2 = 7$ and $l_3 = 5$ with joint angle constraints $0 < \theta_1 < \frac{\pi}{3}$, $0 < \theta_2 < \frac{\pi}{2}$ and
$0 < \theta_3 < \pi$ the same procedure is repeated, Figure shows the training data of three ANFIS networks for three joint angles. Figure shows the difference in theta deduced analytically and the data predicted with ANFIS.

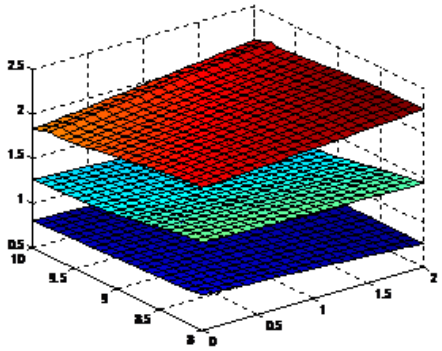


Fig. 4 Angles Correspond to co-ordinates.

4 IMPLEMENTATION OF ANFIS IN EMBEDDED SYSTEM.

In the hazardous area it is not always possible to carry a computer and calculate the angles through matlab. So I have decided to make the system itself powerful such to calculate the angles on board without involvement of any computer.

IJERT

To implement ANFIS in embedded, easiest while powerful membership function is triangle. Using line equation for given co-ordinates we can create a triangle membership function and from the training data gathered from the forward kinematics equations can be used for weighting the rules. Using centroid as defuzzification method I have finally achieved angles with least error than it is achieved from the matlab.

The coding for implementation of ANFIS in embedded is developed in C for PIC 18f4550. C18 compiler is used to compile this code. The code is so efficient that it don't use much memory of micro controller rather by means of only equations it solves the angles. Figure below shows the comparison between matlab angles and embedded angles.

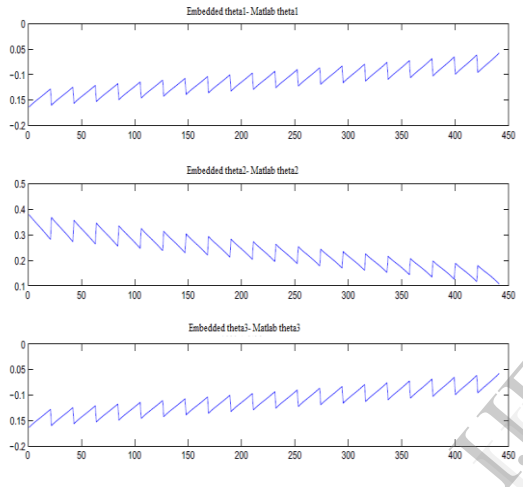


Fig.5 VALIDATING EMBEDDED ANGLES WITH MATLAB ANGLES

5 CONCLUSIONS

Scheming exact inverse kinematics in real-time is computationally too heavy for all but the most simple kinematic configurations. Here, we have presented a method of calculating inverse kinematics which has been shown to be robust to singular configurations, and is applicable to almost every manipulator. After calculating angles PIC micro controller can take end effecter to desire position. At last we achieved an embedded system with inbuilt code to calculate the angles using ANFIS algorithm.

6. REFERENCES

[1] J.-S.R. Jang, "ANFIS: Adaptive Network based fuzzy inference systems," IEEE transactions on Systems, Man, and Cybernetic,23(03),665-685, May1993.

[2] Jand, J., Sun, C., and Mizutani, E., Neuro Fuzzy and Soft computing, Pritice-hall, upper Saddle River, NJ, 1997.

[3] H.Sadjadian, H.D. Taghirad, and A. Fatehi "Neural Networks approaches for computing the forward kinematics of a Redundant parallel manipulator,' International journal of Computing Intelligence vol. 2, No.1, 40-47, 2005.

[4] J.J. Craig, Introduction to Robotics: mechanics and Control, 2nd ed. New York, Addison-wesley Publishing company, 1989.

[5] A. Schacherbauer and Y. Xu, "Fuzzy Control and Fuzzy Kinematic Mapping for a Redundant Space Robo," Technical Paper, CMU-RI-TR-92-12, Carnegie Mellon University, 1992.

[6] S. Araki, H. Nomura, I. Hayashi and N. Wakami, "A Self-Generating Method of Fuzzy Inference Rules," in *Proceedings: 1991 IFES Conference*, pp. 1047-1058, 1991.

[7] Y. Xu and M. Nechyba, "Fuzzy Inverse Kinematic Mapping: Rule Generation, Efficiency, and Implementation," Technical Paper, CMU-RITR- 93-02, Carnegie Mellon University, 1993.

[8] Robot Manipulator Control Theory and Practice - Frank L.Lewis

IJERT