ANFIS Application for Calculating Inverse Kinematics of Programmable Universal Machine for Assembly (PUMA) Robot

by Joga Setiawan

ANFIS Application for Calculating Inverse Kinematics of Programmable Universal Machine for Assembly (PUMA) Robot

Hugo Adeodatus Hendarto
Master Degree of Mechanical Engineering,
Diponegoro University, Indonesia
hugo.hendarto@yahoo.com

Munadi
Department of Mechanical Engineering,
Diponegoro University, Indonesia
munadi@undip.ac.id

Joga Dharma Setiawan
Department of Mechanical Engineering,
Diponegoro University, Indonesia
joga.setiawan@gmail.com

This paper focused on the robot arm's kinematics problem or the connection between angle in each joint and the end-effector's position.f Forward kinematics problem will be deduced using D-H (Denavit– Hartenberg) parameter method. The inverse kinematics problem will be solved using ANFIS (Adaptive Neuro-Fuzzy Inference System) instead of calculating the solution. ANFIS is a feature in MATLAB using ANFIS toolbox. PUMA 560 robot arm virtual model is used in this paper. Three ANFIS training conditions are made to test the influence of training conditions with the result's performance. The difference between end effector's position that using ANFIS and from calculation in forward kinematics will be calculated to test the end effector's position error. By making ANFIS solutions with three different MFs (Membership Functions), influence of MF number are known. With more MF will decrease the position's error. The most MF in this paper is 10 MFs resulting position error by 27.974mm.

Keywords—Arm robot; PUMA 560; forward kinematics; inverse kinematics; ANFIS

Fig. 1. PUMA 560 dimension [2].

I. INTRODUCTION

Consumer demand of products are always increasing both in terms of quantity and quality with a good price. Automated production system will be very useful to overcome this situation. In Indonesia, there are many industries that still using non automated system. This is caused by automating the production system's cost in Indonesia are still high, or man power are still not so expensive compared with automating the system.

However, non automated system still have many drawbacks in production time, stability of the products quality and quantity. Because such condition, automation developing is needed, so that the automation system will be more affordable in Indonesia.

There are a lot of automated machines used in automation. One of the common automated machine is an arm robot or a manipulator robot. This robot can be used for many tasks e.g. moving objects, welding, painting, assembling, etc. In this paper, PUMA 500 series configuration is used. This industrial robot is developed by Victor Scheinmann under Unimation robot industry [1].

PUMA 500 series' configuration can be seen on Fig. 1[2]. This arm robot has six DOF (degree of freedom), with servo motors as the actuator [2]. This robot will be modeled and simulated, so forward and inverse kinematics can be solved and proved.
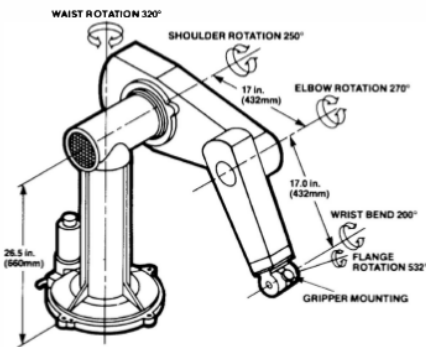
To develop arm robot's controller, forward and inverse kinematics are needed. D-H parameter will be used for solving the forward kinematics problem [3]. However, inverse kinematics solution has always been a problem in every robot's controller designing due to its complexity and more than one possible solution for each position, whereas inverse kinematics is used in defining motor's rotation from desired end effector's position. Inverse kinematics also used to smoothen robot's movement and moving the end effector in line, circle, or another trajectory. To overcome this difficulty, ANFIS is used to solve the inverse kinematics so that there is no need to calculate inverse kinematics solution. And then, the result from those solution will be tested in the simulation.

ANFIS has been used for DC-DC step down converter [4], controlling autonomous flight system [5], ANFIS tuning of PID controller for hydraulically driven parallel robot machine [6], water flow rate [7], etc. Another uses of ANFIS is to solve inverse kinematics problem. There are some papers that use ANFIS in solving the inverse kinematics problem.

Srinivasan Alavandar, M. J. Nigam made a paper about Neuro-Fuzzy based approach for inverse kinematics solution of industrial robot manipulators [8]. The paper solved 3 Degree of Freedom (DOF) arm robot's kinematics problem. They made inverse kinematics solution with both ANFIS and calculation to test the result's performance by comparing the deduced angle with ANFIS's angle.

Yousif I. Al Mashhadany from Iraq also made ANFIS-Inverse-Controlled PUMA 560 [9]. He also compared between ANFIS's angle solution and inverse kinematics analytical solution.

Ouamri Bachir, Ahmed-foitih Zoubir from Oran made a paper about controlling PUMA 600 series using ANFIS. They compares between ANFIS and fuzzy controller, and shows that ANFIS controller is better to fuzzy controller in robustness [10]. ANFIS's performance in time to target position is discussed in this paper.

From those papers about ANFIS based inverse kinematics solution, there is no paper that analyze position error between model's end-effector position that use angle from ANFIS solution and calculated end-effector's position used forward kinematics. By comparing with forward kinematics result, there is no need to calculate inverse kinematics solution to test the result. ANFIS's performance with different amount of membership functions (MFs) also has not discussed in the existing papers.

The PUMA virtual model will be described in section 2. After the model simulation is finished, the forward kinematics solution is made in section 3. Inverse kinematics solution using ANFIS toolbox described in section 4. ANFIS's performance in solving inverse kinematics also discussed in this section.

II. PUMA 560 SIMULATION

To simulate this PUMA, a virtual model is needed to get a better form. This virtual model was made using SolidWorks software. After the model is finished, it will imported into MATLAB SimMechanics using SimMechanics Link feature.

A. PUMA 560 model in SolidWorks

SolidWorks is used for making the PUMA model in order to get a realistic arm robot's virtual model. Fig. 2 shows the model made in SolidWorks. Furthermore, this model is exported into .xml file, and then SimMechanics will import this file.

Fig. 2. PUMA 560 model in SolidWorks.

B. PUMA Model in SimMechanics
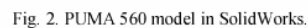
Due to paper's focus is on the forward and inverse kinematics, the actuation for the model is in angle. SimMechanics first generation is used, in which there is no angle actuating in SimMechanics second generation.
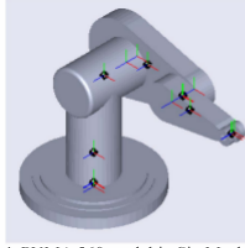
After .xml file is generated from 2.A section, then the file can be imported in SimMechanics, and creates blocks as shown in Fig. 3. The revolute blocks represent joints of the robot, and then actuating block with angle value will be connected to that block. When the simulation is running, the imported model's body can be seen as shown in Fig. 4.

Fig. 3. SimMechanics blocks of PUMA model.

Fig. 4. PUMA 560 model in SimMechanics.

III. FORWARD KINEMATICS

Forward kinematics solution is used for calculating the end-effector's position with known motor's angle values. There are four frames in this model. The frame configuration shown in Fig. 5. This configuration focused on the arm's movement, so the model used is an arm robot without gripper.
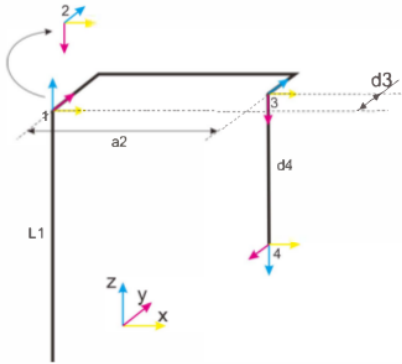

Fig. 5. Coordinate system of the model.

Based on coordinate system of the model which is shown in Fig. 5, the D-H parameter can be defined. This parameter shown in Table I. The method to get the D-H parameter discussed in "Introduction to Robotics Mechanics and Control Second Edition" [3].

TABLE I. D-H PARAMETER OF PUMA MODEL

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	-90°	0	0	θ_2
3	0	a_2	d_3	θ_3
4	-90	0	d_4	θ_4

After D-H parameter defined, then forward kinematics solution can be solved in matrix form. These matrices are made by using transformation matrix based on the D-H parameter. Matrices 1, 2, 3, 4 represent joints from base to end-effector. And the forward kinematics represented by matrix T_4^0 in equation 5.

$$T_1^0 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$T_2^1 = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$T_3^2 = \begin{bmatrix} c_3 & -s_3 & 0 & a2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & d3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$T_4^3 = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ 0 & 0 & 1 & d3 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$T_4^0 = T_1^0 \, T_2^1 \, T_3^2 \, T_4^3 \quad (5)$$

$$T_4^0 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

where :

$$r_{11} = c_4 c_1 c_{23} + s_1 s_4 \quad (7)$$
$$r_{21} = c_4 s_1 c_{23} - c_1 s_4 \quad (8)$$
$$r_{31} = -c_4 s_{23} \quad (9)$$

$$r_{12} = c_4 s_1 - s_4 c_1 c_{23} \quad (10)$$
$$r_{22} = -c_1 c_4 - s_4 s_1 c_{23} \quad (11)$$
$$r_{32} = s_4 s_{23} \quad (12)$$

$$r_{13} = -c_1 s_{23} \quad (13)$$
$$r_{23} = -s_1 s_{23} \quad (14)$$
$$r_{33} = -c_{23} \quad (15)$$

$$p_x = -d_3 s_1 - d_4 c_1 s_{23} + a_2 c_1 c_2 \quad (16)$$
$$p_y = c_1 d_3 + d_4 s_1 s_{23} + a_2 c_2 s_1 \quad (17)$$
$$p_z = -a_2 s_2 - d_4 c_{23} + L_1 \quad (18)$$

Based on the above equation, the position of the end-effector is represented with Px, Py, Pz from equation 16, 17, and 18, in which L_1, a_2, d_3, d_4 is length of arms as shown on Fig. 5. $C_i = \cos(i)$, $S_i = \sin(i)$, $C_{ij} = \cos(i+j)$, $S_{ij} = \sin(i+j)$.

IV. INVERSE KINEMATICS

Inverse kinematics problem will be solved using ANFIS in MATLAB. Defined motor's minimum rotation angle is 5°, maximum rotation angle in joint 1, 2 and 3 are 270°, 90°, and 180°, respectively.

A. ANFIS Training

To train the ANFIS, training data is needed. This data set use the equation based on equation 16, 17 and 18. The data set also can be used to show the model's workspace by plotting the data as shown in Fig. 6. There are four figures in Fig. 6 which are show the workspace in different plane. The PUMA robot stands on (0, 0, 0) coordinate.

PUMA Model Workspace

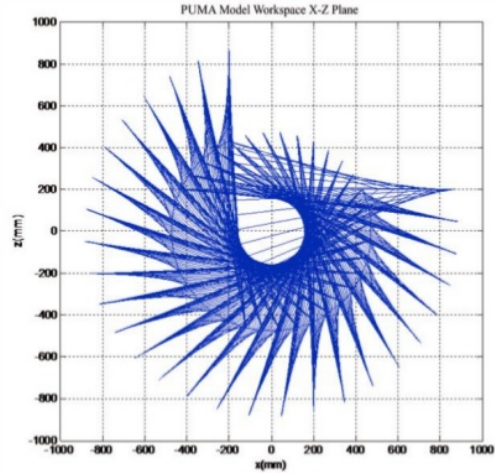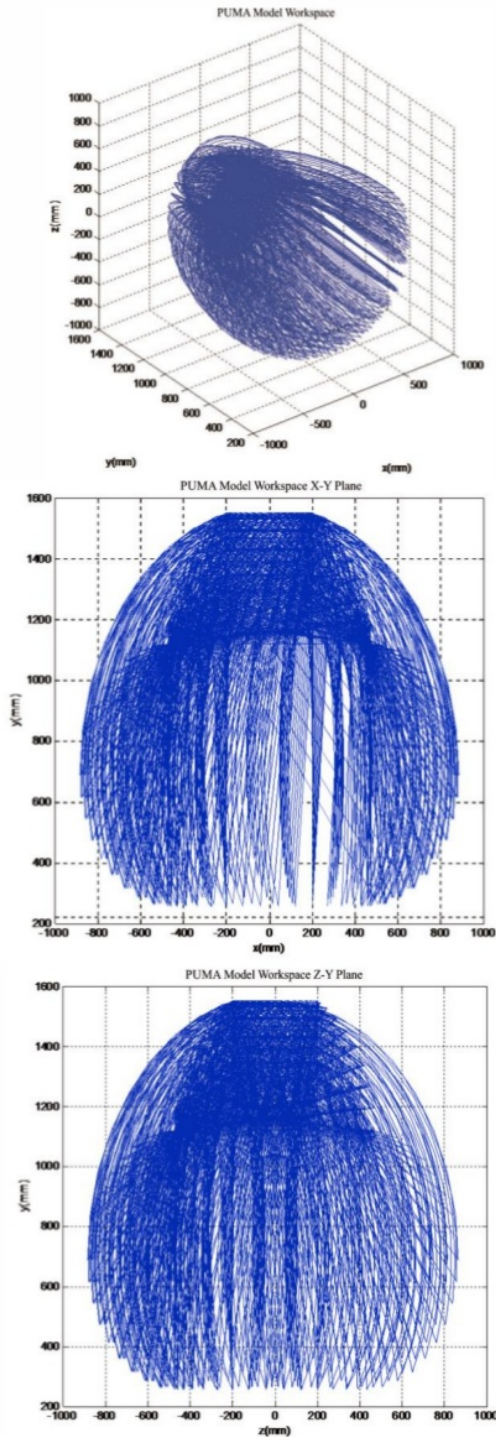PUMA Model Workspace X-Z Plane

PUMA Model Workspace X-Y Plane

Fig. 6 PUMA model's workspace

After the training data set is ready, then train ANFIS using ANFIS toolbox. One training, is only for single output which is the motor's position. There are three trainings for each motor which use different amount of MFs which are 3, 7, and 10 MFs. These trainings are using Gaussian Membership Function [9], backpropagation optimization method.
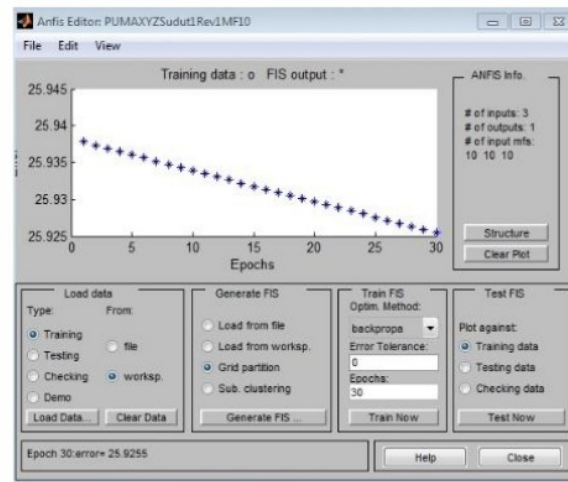

PUMA Model Workspace Z-Y Plane

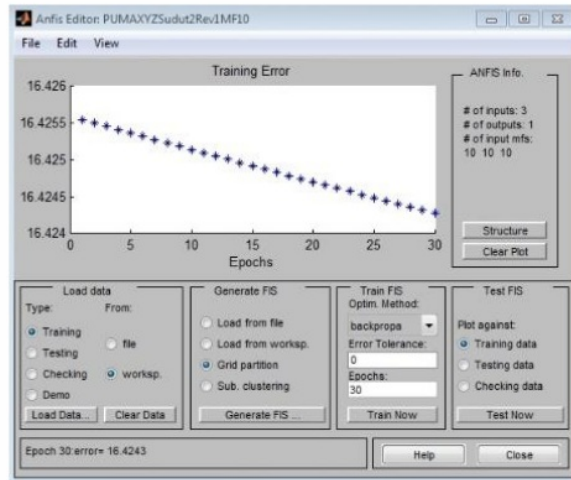

Fig. 7. Training result for joint 1

38

Fig. 8. Training result for joint 2
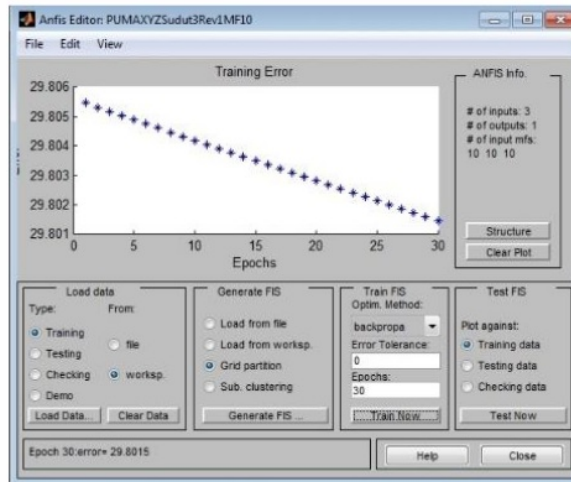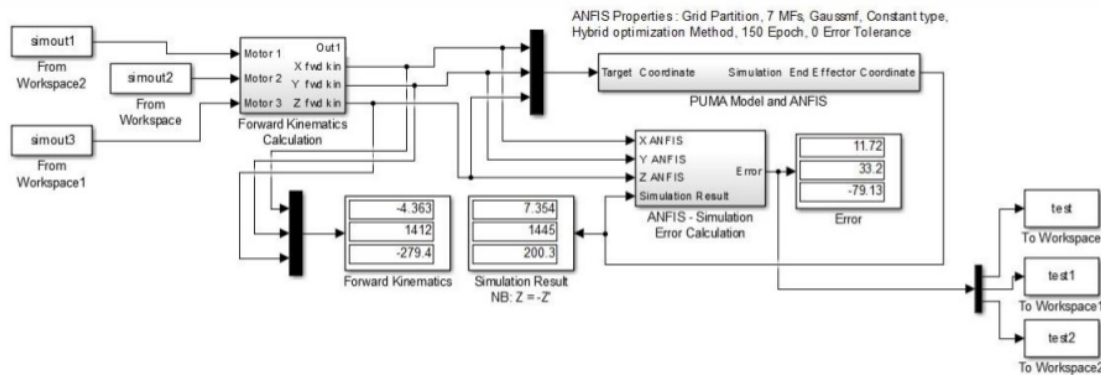

Fig. 9. Training result for joint 3

Fig. 7, 8 and 9 shows the training result for each joint that using 10 MFs. The number of epoch used is 30. This training result can be exported into a .fis file. This file is then used in the simulation to give angle input to the joint block with desired end-effector coordinate.

B. ANFIS Performance Analysis

To analyze the ANFIS performance, a simulation in Simulink was made. This simulation will calculate the error between known angle and ANFIS's angle. Fig. 10 shows the block diagram used for analyze the ANFIS performance. The data in simout1, simout2, and simout3 blocks are 500 different angles with random values in range of each motor's capability. Then, these values are calculated into end-effector's position with forward kinematics from equation 16, 17 and 18. The results from this calculation are used as inputs for Fuzzy blocks with generated .fis files from the ANFIS training.

After end-effector's positions with ANFIS's angle are known by simulating them in the PUMA model (shown in Simulation Result block), these values are then compared with the position's values from the forward kinematics calculation (shown in the Forward Kinematics block). By using this method, end effector's position error will known, and then these position's difference presented in the Error block. All this position error data's average shown in Table II.

TABLE II. ANFIS PERFORMANCE ERROR DATA

Membership functions	Axis	Average error (mm)	Total average error (mm)
3	X	657.812	
	Y	56.355	388.226
	Z	450.510	
7	X	21.028	
	Y	36.846	32.575
	Z	39.851	
10	X	12.227	
	Y	34.262	27.974
	Z	37.433	

Fig. 10. Block diagram of ANFIS performance simulation

Table II shows the average error of the ANFIS. There are three ANFIS different training conditions i.e. the amount of MFs. ANFIS with 10 MFs has the smallest error. So, the more amount of MFs, the smaller error of ANFIS's result.

This error could be caused by the smallest rotation angle that can be done with the motor i.e. 5°. These problems can be solved by increase the motor capability. Moreover, bigger amount of MF is also needed to decrease the error.

V. CONCLUSIONS

ANFIS has capability to solve inverse kinematics problem. However, there are still some position error in this paper. This error decreased with increase of the MFs. The error difference between ANFIS that use 3 MFs and 7MFs is 355.651. And then, the error decreases again to 27.914mm by increasing the MFs into 10 MFs. So, performance of the ANFIS can be improved by increasing the amount of MFs. However, it will take more time to train ANFIS with more MFs. This ANFIS solution also can be applied into another inverse kinematics problems with different configuration e.g. arm robot with SCARA configuration, robot leg, and another complex robot configuration.

REFERENCES

[1] "Programmable Universal Machine for Assembly," http://en.wikipedia.org/wiki/ Programmable_Universal_Machine_for_Assembly.

[2] "Unimation," Unimate PUMA Series 500 Industrial Robot, 1984.

[3] John J. Craig, "Introduction to Robotics Mechanics and Control Second Edition,", 1955, pp. 88-93.

[4] Ömer Deperlioğlua, Uçman Ergünb ve Gür Emre Güraksınc, "Design of ANFIS Controller for DC-DC Step-Down Converter," Afyon Kocatepe University, Afyonkarahisar, Turkey, 2010, pp. 17-29.

[5] Sefer Kurnaz, Omer Cetin, Okyay Kaynak, "Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles," Turkish Air Force Academy, Bogazici University, Istanbul, Turkey, 2010, pp. 1229-1234.

[6] Mazin I. AL-Saedi, Huapeng Wu, and Heikki Handroos, "ANFIS and Fuzzy Tuning of PID Controller for Trajectory Tracking of a Flexible Hydraulically Driven Parallel Robot Machine," Journal of Automation and Control Engineering, Vol. 1, No. 2, June 2013pp. 70-77.

[7] Vandna Kansal, Amrit Kaur, "Application of ANFIS for Water Flow Rate Control in a Rawmill of Cement Manufacturing Process," University College of Engineering, Punjabi University, Patiala, Punjab- 147002, India, Vol. 3, 2013, pp. 459-463.

[8] Srinivasan Alavandar, M. J. Nigam, "Neuro-Fuzzy based Approach for Inverse Kinematics Solution of Industrial Robot Manipulators," Department of Electronics and Computer Engineering, Roorkee - 2477667, Uttarkhand, INDIA, Vol. III, No. 3, 2008, pp. 224-234.

[9] Mashhadany, Yousif I. Al, "ANFIS-Inverse-Controlled PUMA 560 Workspace Robot with Spherical Wrist," International Symposium on Robotics and Intelligent Sensors 2012, IRIS 2012, pp. 700-709.

[10] Ouamri Bachir, Ahmed-foitih Zoubir, "Adaptive Neuro-fuzzy Inference System Based Control of Puma600 Robot Manipulator," International Journal of Electrical and Computer Engineering (IJECE), Vol. 2, No.1, February 2012, pp. 90-97.

ANFIS Application for Calculating Inverse Kinematics of Programmable Universal Machine for Assembly (PUMA) Robot