

## **ROBOT MANIPULATOR INVERSE KINEMATICS USING ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM**

MUSTAFA T. HUSSEIN<sup>1,\*</sup>, ALI S. GAFER<sup>2</sup>, ESSAM Z. FADHEL<sup>1</sup>

<sup>1</sup>Department of Mechanical Engineering, University of Babylon, Iraq

<sup>2</sup>College of Engineering, University of Babylon, Iraq

\*Corresponding Author: hussein\_m0214@daad-alumni.de

### **Abstract**

The solution of Inverse kinematics problem using an intelligent approach that combines fuzzy systems with the field of neural networks “adaptive Neuro-Fuzzy inference system ANFIS” is demonstrated in this contribution. A four DoF robot manipulator (IRIS) is used as a model to solve its inverse kinematics problem. The forward kinematics equations of the robot system are derived. The kinematics equation of IRIS robot system is highly nonlinear and coupled. The ANFIS solve the inverse kinematics problem through learn from training data. Using the kinematics equations, the data created to cover the all 3D workspace of the robot system. The Gaussian membership function with hybrid learning algorithm used; which are useful to solve similar problems. The ANFIS networks validity was tested by two different simulations first by identification of trajectory generation and the second by drawing a circle in the 3D workspace. The simulation shows that the Adaptive Neuro-Fuzzy Inference system can be used effectively to produce acceptable solutions of the inverse kinematics problem.

Keywords: ANFIS, Inverse kinematics, Robot 3D workspace, Robot motion, Robot simulation.

## **1. Introduction**

Robots perform at several of the tasks in a variety of applications. Robots must perform these tasks as quickly as possible while maintaining the highest degree of accuracy. The inverse kinematics problem is the key feature for robot control. More high precision method for the inverse kinematics problem is called for; hence the traditional mathematical tools are even so not perfectly suited to resolve complex and uncertain systems [1]. The artificial intelligent approaches used in robotics design due to their advantageous in handling complex mechanical systems control, their ability to learn, ability to generalize what it has learned, and the ability to decide. Artificial Neural Networks (ANN) is well known for their learning and adapting ability of the system behaviour, on the other hand. A fuzzy-logic approach (FL) is based on linguistic patterns with an IF-THEN general structure which gets to the FL appear to exhibit some human intelligence or decision capability [2].

The most known and widely applied intelligent control architecture is the “adaptive-network-based fuzzy inference system”. This architecture which introduced for the first time in the '90s, is well recognized for its ability dealing with the nonlinearities and uncertainties in robotics' systems [3]. Recent developments show that superior results can be obtained using adaptive neural networks and fuzzy logic resulting in “Adaptive Neuro-Fuzzy Inference System”. The ANFIS approach can be used effectively for modelling, estimation and control of complex engineering system. The ANFIS paradigms equal to a system contains fuzzy and neural network in one system. Moreover, the fuzzy system's ability of decision making can be immersed inside neural network learning and adaptation ability. Furthermore, both nodes and parameters of the two intelligent systems will hold specific significance and corresponding to the membership function or the inference process of fuzzy systems [4].

Recent improvements in the area of robot kinematics are due to a large extent to the advances made on the subject of rigid body kinematics. In fact, as a robot manipulator normally modelled as a series of links which undergo 'rigid-body' motions. Revolute or prismatic joints are normally used to connect the robot links to each other. In the joint space, tasks are first mapped into joint space through homogeneous transformation 4x4 matrices which relate successive joints mathematically. These matrices provide the state of the orientation and location of the robot end-effector. Furthermore, they represent the degrees of freedom of a robot. Robot control depends directly on the forward and the inverse kinematics processes. The control action (inputs) in the robotic system is applied to the joints; while robot motions of the end-effector are stated in the Cartesian coordinates. The inverse kinematics process converts of the required end-effector position from Cartesian coordinates ( $x, y, z$ ) to joint space (motor angles). The inverse kinematic process produces the desired joint angles for robot desired motion, and it is the key point of the robot controller success [5].

Several articles reported the technique of implementing of Neural networks and Fuzzy logic in combination as a solution for the robot inverse kinematics problem. The authors in [6] employed the “Neural network” approach to determine the desired joint angles for the planar robotic manipulator for a given desired end-effector. The inverse kinematics problem of PUMA robot is addressed in [7]; the system is considered as 4DoF the ANFIS successes to calculate the inverse kinematics with average error 27.974 mm due to low number of membership

function. A planar robot system with 3DoF is investigated in [8] to solve kinematics problem using ANFIS, the end-effector is used as a condition to compute the angular position of each joint, also the authors transferred the ANFIS into fitting tool to improve tracking capability of the system in plane. The ANFIS networks is implemented to solve kinematics problem of Delta Parallel robot in [9], the method is used to reduce the complexity of solution of Delta robot kinematics problem. A training data is prepared using the geometric equation of the Delta robot and the trained network determined the required joint angles to follow the suggested trajectories. The author in [10] presented the solution of inverse kinematics problem multi degrees of freedom robot using the ANFIS network, the out off-plane motion is excluded from the study due to complexity and the time-consuming training process and the system is considered as planar robot.

From the above literature, it found that the researchers studied the inverse kinematics with ANFIS in planar not 3D workspace due to the complexity of the training and calculations. Also, a high average error is recorded in the results presented. In this paper, Neuro-fuzzy systems “ANFIS” which implement the neural network systems for automatic parameter tuning of fuzzy systems is applied to figure out the inverse kinematics problem. A 4DoF IRIS robot is used as a case study; the forward kinematics equations for IRIS robot are calculated first. Based on the derived kinematics equations the training data is prepared for the 3D workspace. An offline training process is done, and the error tremendously reduced to ensure the convergence of the solution. The simulations for using ANFIS show the inverse kinematics successfully estimated for the whole 3D workspace of the robot system. As an outline for the paper; robot kinematics addressed in the next section, then the ANFIS network detailed, later the simulation results for the robot system is presented, and the conclusion of the paper is presented in the last section.

## 2. Robot Inverse Kinematics

The motivation for the growth of robotic systems branches from the wish to improve the efficiency of robot operations. Robotics control is an important step in in the robot development process. Furthermore, the control in the robotics field represents an important branch in control engineering. Besides the properties of intelligent, nonlinearity, real-time computing; the inverse kinematics problem can arise as a non-solvable problem [11]. In order to achieve the desired motion for a robot end-effector following a path, robot control is necessary. The central point of the robot control problem is the Kinematics of a robot arm, which dispenses with the motions of the arm with respect to a specified coordinate system as a role of time. The forward kinematics relate the joint angles of the robot (DoF) to the end effector position in a straightforward manner, while the inverse kinematics is the problem of finding the joint angles required to reach specific end-effector 3D position shown in Fig. 1.

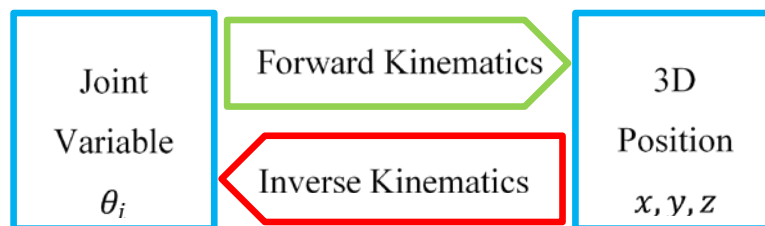


Fig. 1. Forward and inverse kinematics of robot.

In traditional robot analysis, the ground base of a robot is used up as a reference, the rest of the points in the kinematic chain are measured from the base reference point. This analysis approach utilizes the inverse kinematics solution to generate the required joint angles set to reach the user definite Cartesian coordinates of the end-effector. Each robot manipulator system has unique kinematics equations describing the motion of the system and they cannot be applied to another robot system. On the other hand, due to the complexity of the transformation matrix to the robot base reference, the deriving process of the kinematics equations yields a set of highly coupled non-linear equations.

The geometric coordinates frames set for the IRIS robot manipulator studied in this paper is illustrated in Fig. 2, in this figure  $q_i = \theta_i \quad i = 1 \dots 4$ . The kinematics parameters based on the frames orientation from Fig. 2 are listed in Table 1.

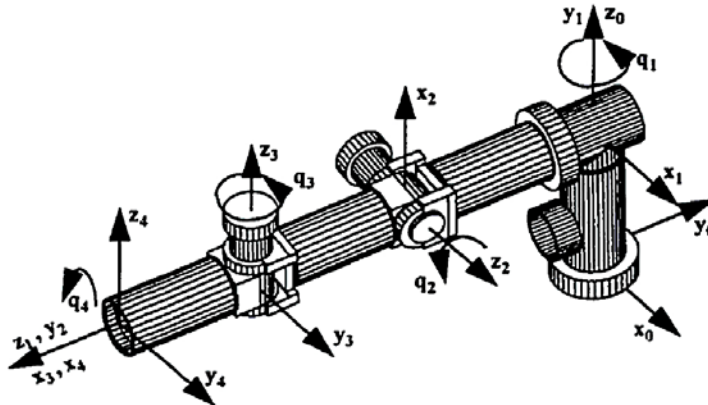


Fig. 2. The desired configuration of the IRIS robot with the link coordinate frames [12].

Table 1. kinematics parameters of IRIS robot under study [13].

Link <i>i</i>	$\theta_i$	$\alpha_i$ [deg]	$a_i$ [m]	$d_i$ [m]
1	$\theta_1$	0,180°	0.28	0
2	$\theta_2$	-45°, 60°	0.32	0
3	$\theta_3$	0,90°	0	0.25
4	$\theta_4$	0,180°	0	0.19

Several ways can be used to represent the robot end-effector position in 3D workspace, one way is to employ the four joint angles  $\theta_1, \theta_2, \theta_3$ , and  $\theta_4$ . This approach is known as the representation in “joint space”, it can be defined as:

$$P_i = f(\theta_1, \theta_2, \theta_3, \theta_4) \tag{1}$$

By using the parameters shown in Table 1 and the DH Matrix equations described in [12], the forward kinematics problem is solved for IRIS robot. The DH matrix convention consists of a homogeneous transformation matrix  $T_{i-1}^i$  from frame  $i - 1$  to frame  $i$  which can be represented with the basic matrix:

$$T_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

The homogeneous transformation from the robot end-effector to the robot base is given by multiplying the transformation matrices in order:

$$T_0^4 = T_0^1 T_1^2 T_2^3 T_3^4 = \begin{bmatrix} O_{xyz} & P_{xyz} \\ 0 & 1 \end{bmatrix} \quad (3)$$

Here:

$$T_0^1 = \begin{bmatrix} c_1 & 0 & s_1 & a_1 c_1 \\ s_1 & 0 & -c_1 & a_1 s_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_1^2 = \begin{bmatrix} c_2 & 0 & s_2 & a_2 c_2 \\ s_2 & 0 & -c_2 & a_2 s_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_2^3 = \begin{bmatrix} c_3 & 0 & s_3 & 0 \\ s_3 & 0 & -c_3 & 0 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_3^4 = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}, P_{xyz} = \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix},$$

$$s_i = \sin(\theta_i), c_i = \cos(\theta_i).$$

After few simplifications the position coordinates  $x_e, y_e$  and  $z_e$  of the end-effector of the IRIS robot system in the 3D workspace produced as:

$$x_e = d_4(c_1 c_2 s_3 - s_1 c_3) + d_3 c_1 s_2 + a_2 c_1 c_2 + a_1 c_1 \quad (4)$$

$$y_e = d_4(s_1 c_2 s_3 + c_1 c_3) + d_3 s_1 s_2 + a_2 s_1 c_2 + a_1 s_1 \quad (5)$$

$$z_e = d_4 s_2 s_3 - d_3 c_2 + a_2 s_2 \quad (6)$$

When the Cartesian coordinates of the end-effector are given, the inverse kinematics problem must be resolved in order to find the values of the joint angles' variables. The solution can be written as:

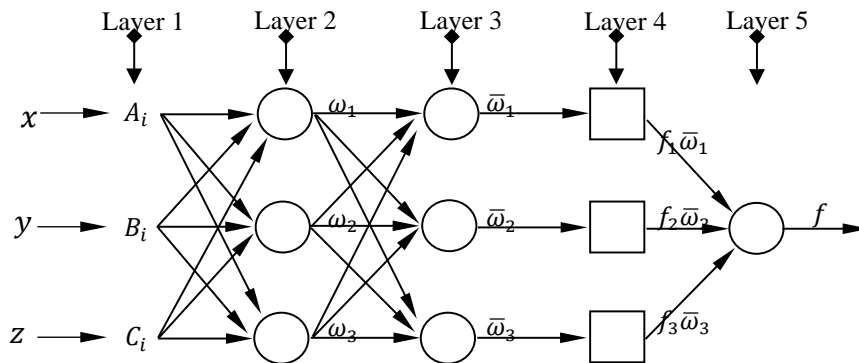
$$\theta_i = f(x, y, z) \quad (7)$$

Due to nonlinearity, uncertainty and the time-varying nature of the inverse kinematics equations, the closed form solutions of Eq. (7) cannot be guaranteed using an algebraic method. The result can be moved over to a single solution using the iterative methods, yet these methods depend on the initial starting point and face difficulty near singularities.

### 3. Structure of ANFIS neural network

An adaptive network is a network grid containing of nodes linked to each other through directed links to transfer signals. Furthermore, part or all of the nodes inside the network are adaptive. In other words, the outputs of the nodes depend on the parameter(s) which related to these nodes. The parameters are changed systematically based on the learning rule which stipulates the parameters to minimize the output error. A multilayer feedforward network shown in Fig. 3 represents an adaptive network. Each node in this network works particularly on the input signals through a set of parameters affecting the input signals. There is a specific mathematical representation for each node function in the network. Moreover, choosing of this

mathematical representation depends on the functionality of the network or the task which the network built to carry-out. It is also important to know that the arrows in the network only the direction of signals moving between two nodes; these arrows have no effect on the values of the flow signal.



**Fig. 3. Adaptive neuro-fuzzy inference network structure.**

The ANFIS, as the name indicates, is a combination of fuzzy systems and neural network. In ANFIS, a neural network is used to manipulate parameters of the fuzzy system. The optimization process of parameters inside the fuzzy system is not necessary for the ANFIS system. The Neural Network part of ANFIS adjusting the fuzzy parameters automatically, in other words, the membership functions limits which produce an improved performance is altered automatically without operator involvement. The zero-order and first-order Sugeno methods are traditionally used for ANFIS architecture, consequently, the output membership functions must be either linear or constant.

In Inverse Kinematics problem, this means that the desired 3D coordinates and the desired joint angles data set are fed to a fuzzy system. The parameters of the fuzzy system can be optimized using the neural network system [2]. For the inverse kinematics complex system behaviour, ANFIS improves fuzzy if-then work with a larger choice of membership functions to use. The architecture of an ANFIS for inverse kinematics solution is shown in Fig. 3. A more detailed explanation of ANFIS can be found in [3]. The role of each layer is briefly presented as follows:

**Layer 1:** In this layer, each node  $i$  is an adaptive node generates output according to a node function defined by [6]:

$$\begin{aligned}
 O_i^1 &= \mu_{A_i}(x) \\
 O_i^1 &= \mu_{B_i}(y) \\
 O_i^1 &= \mu_{C_i}(z)
 \end{aligned} \tag{8}$$

Here,  $x, y, z$  are the end-effector 3D coordinates which are the inputs to node  $i$ ,  $A_i, B_i$  and  $C_i$  are the fuzzy sets,  $\mu_i$  is the membership. The output of these layers is membership values (MF).

**Layer 2:** The multiplication of all received signals is applied in the nodes of this layer to compute the fuzzy operator, as [6]:

$$\omega_i = \mu_{A_i}(x) \cdot \mu_{B_i}(y) \cdot \mu_{C_i}(z) \quad (9)$$

Each node output in this layer represents the strength of a rule.

**Layer 3:** The nodes in this layer are defined; they are utilized to complete the scale conversion and calculate the ratio of each rule firing strength to the totality of all rules firing strengths as [6]:

$$\bar{\omega}_i = \frac{\omega_i}{(\omega_1 + \omega_2 + \omega_3)} \quad (10)$$

where,  $\omega_i$  referred to as the ratio of the firing strengths.

**Layer 4:** Each square node  $i$  in the layer is an adaptive one with a node function [5]:

$$O_i^4 = f_i \bar{\omega}_i = \bar{\omega}_i (p_i x + q_i y + r_i z + s_i) \quad (11)$$

The  $p_i, q_i, r_i$  and  $s_i$  are the parameter set of this node.

**Layer 5:** The individual node in this layer completes the non-fuzzification process of fuzzy systems, here the layer uses the summation in order to estimate the total output of all incoming signals [6], which is shown equally:

$$O_i^5 = \frac{\sum_i \bar{\omega}_i f_i}{\sum_i \bar{\omega}_i} \quad (12)$$

The network structure with first-order Sugeno model is considered in this work. The model contains 216 rules; at the fuzzification level the Gaussian membership function with multiplication inference rule is used [14]. An algorithm which combines the steepest-descent and least-squares method is used to regulate the parameters of membership function, which called hybrid learning method.

## 4. Simulation Results

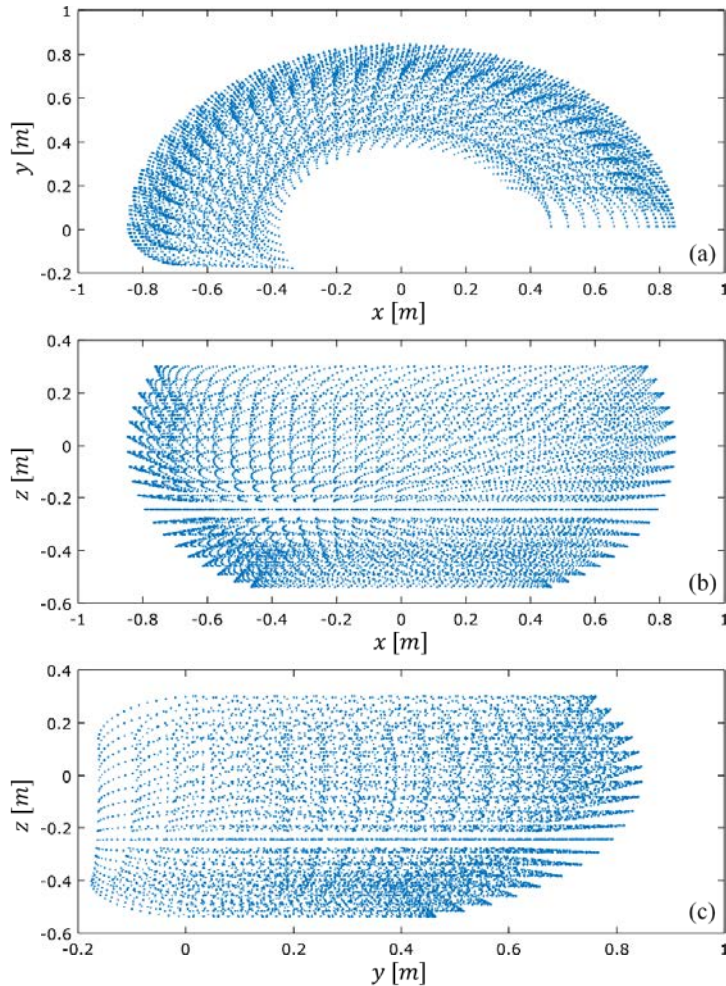
### 4.1. Training Data ANFIS Neural Network

The kinematics parameters for the 4DoF robot system under study are listed in Table 1 earlier. The robot 3D workspace in  $x, y$ , and  $z$  Cartesian coordinates, can be generated from the forward kinematic Eqs. (4-6) and the joint angles' ranges (i.e. third column in Table 1). The mesh grid for every two sets of the coordinates is presented as shown in the Figs. 4(a)-(c). The figures show all possible data points of the  $x_e, y_e, z_e$ , the range for each joint angle is divided into steps each step equal to 0.05 rad then the  $x, y$ , and  $z$  coordinates deduced for each.

Due to the fact that the  $\theta_4$  has no effect on the forward kinematics Eqs. (4-6) (i.e.  $\theta_4$  is not included in Eqs. (4-6) which means it has no effect on 3D Cartesian coordinates of the end effector), thus the problem here is simplified to three unknown joint angles required to predict. Consequently, the ANFIS solution for this problem is to build three ANFIS networks, first to predict  $\theta_1$ , second to predict  $\theta_2$  and the third one to predict  $\theta_3$ . In order to make the ANFIS networks able to solve the inverse kinematics problem, the following procedure described in the next paragraphs had been followed.

The input-output dataset for each of the angles should be prepared, these datasets include the set of  $x, y$ , and  $z$  3D coordinates dataset as inputs to the network, and  $\theta_i$  as an output. The data is generated using the joint angles limits in Table 1. The data have to cover all the 3D workspace of the robot system. Each of the ANFIS networks has to be trained with the generated input-output dataset, the

first ANFIS network will be trained with  $x, y,$  and  $z$  coordinates as input and corresponding  $\theta_1$  values as output. The data sets required for the training of the network are  $x - y - z - \theta_1$ ,  $x - y - z - \theta_2$ , and  $x - y - z - \theta_3$ . The same training procedure is also repeated for the second and the third network. Check the training error until the network reaches an acceptable value to ensure the effectiveness of the trained network.



**Fig. 4. 3D coordinates workspace using forward kinematics equations.**

The training procedure and simulation have been carried out using MatLab software. The Gaussian membership function in conjunction with hybrid-learning algorithm was used in this work as mentioned earlier. The training data of the three ANFIS networks resulted for each of the joint angles in the robot system are shown in Figs. 5-7. The 3D end-effector position coordinates act as inputs ( $x - input1, y - input2, z - input3$ ) to the network and the  $i$ th joint angle of the robot system induced from the network as the output. The learning procedure explains to the ANFIS network how to estimate the appropriate joint angles required to fulfil



the Cartesian coordinates through training. The premise parameters that define membership functions results are determined by using the steepest-decent method and the consequent parameters are determined by employing the least-squares method. In the training stage, the membership functions type and the weights value are selected based on two conditions: the minimum error is acceptable or if the number of epochs reached.

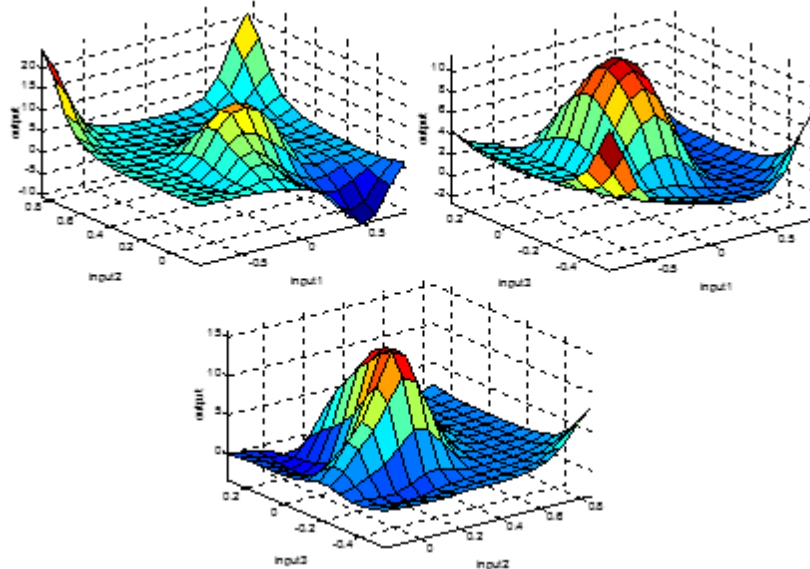


Fig. 5. Training data of  $\theta_1$  for IRIS robot manipulator.

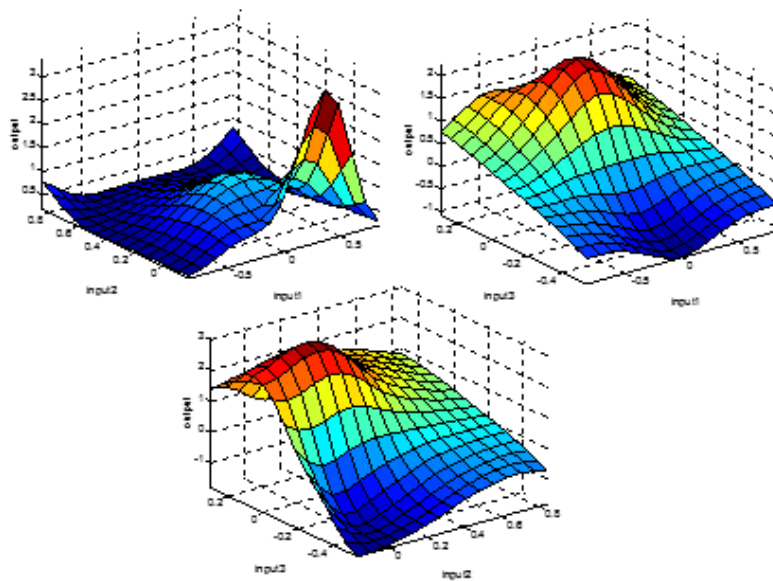


Fig. 6. Training data of  $\theta_2$  for IRIS robot manipulator.

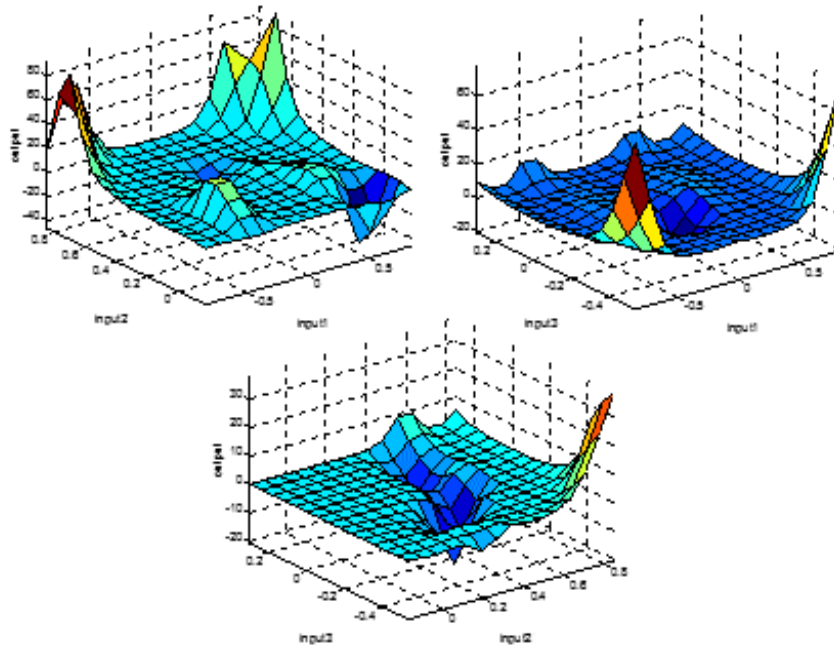


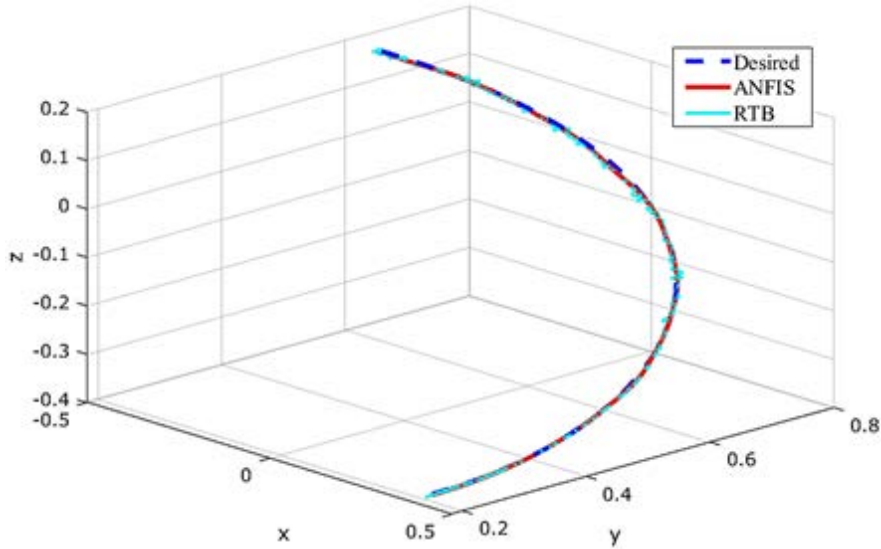
Fig. 7. Training data of  $\theta_3$  for IRIS robot manipulator.

#### 4.2. Validation of ANFIS Network for Inverse Kinematics Solution.

For the sake of ensuring the effectiveness of the trained networks presented earlier, a simulation on an IRIS robot manipulator inverse kinematics is illustrated. Furthermore, a comparison with the Robotics Toolbox [15] Inverse kinematics approach is also presented. Here, the joint angles vector is  $q = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4]$ , the 3D position vector is  $P = [x_e \ y_e \ z_e]^T$  as stated earlier.

The first simulation test is aimed to accomplish the task of end-effector tracking a specific path in the 3D workspace of the robot. The desired path is created in the 3D-workspace; by using the robotics toolbox (RTB) for MatLab [15]. The toolbox is implemented to generate the joint angles vector  $q$ , by defining the initial and final joint angles for fifth order polynomial in trajectory as  $q_i = [0 \ 30^\circ \ 0 \ 0]$ ,  $q_f = [120^\circ \ 45^\circ \ 60^\circ \ 0]$ , and each of these angles are divided into 500 steps. After that, the generated joint angles substituted in the Eqs. (4-6) to generate the desired path. The end-effector path point position  $(x_e \ y_e \ z_e)$  now applied as the input of the designed ANFIS network to find the required joint angles (inverse kinematics solution). The inverse kinematics problem for IRIS robot under study is also solved using the RTB by implementing the IRIS parameters in the toolbox to generate automatic solution. Figure 8 illustrates the generated path based on Eqs. (4-6). It's also demonstrates the resulted path using inverse kinematics solution from the ANFIS network and RTB.

Additionally, the error  $e_{\theta_i}$  between the joint angles from trajectory generation and predicted joint angles from the ANFIS network and the RTB are shown in Fig. 9. It is clear from the figure presented for the first simulation that the ANFIS network gives better solution results.

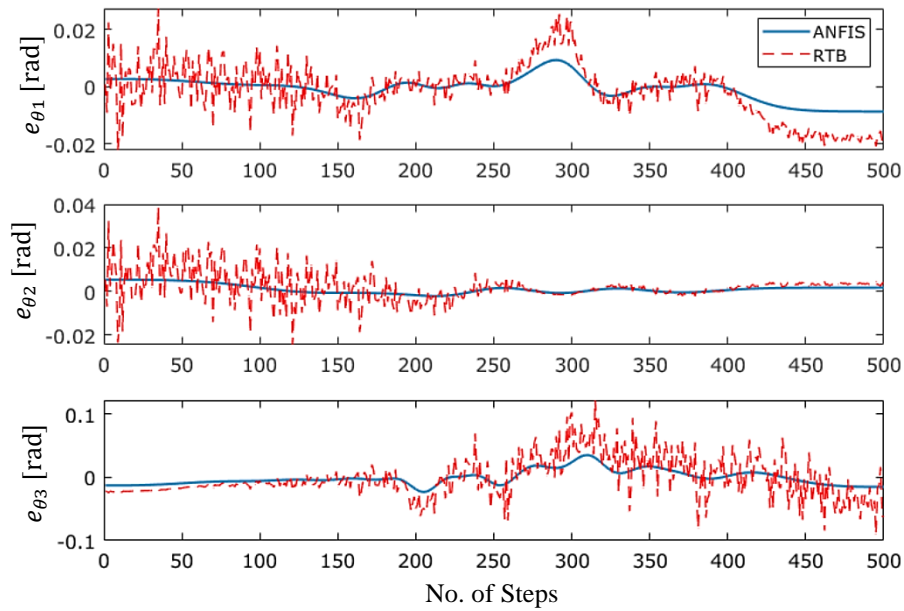


**Fig. 8. End effector 3D path in the first simulation test.**

The second simulation test, which includes the task of following a circular path for end-effector in the 3D workspace is also performed using ANFIS network and RTB. The desired path selected for the end-effector position vector is described by:

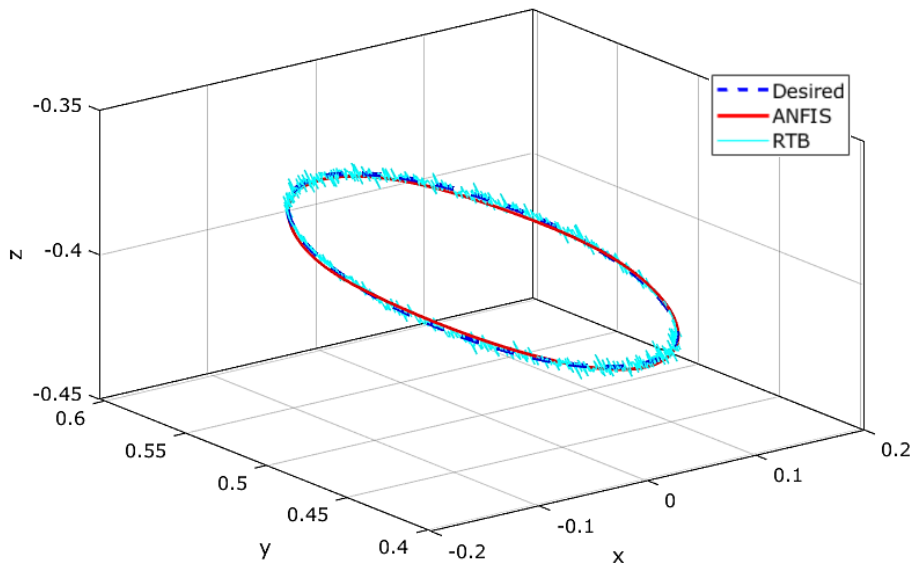
$$x = ra * \cos(t) + x_0, y = ra * \sin(t) + y_0, z = -0.4 + 0.01 * \sin(t) \quad (13)$$

The centre of a circle is  $[x_0, y_0] = [0, 0.5]$  and the radius is  $ra = 0.1$  m.



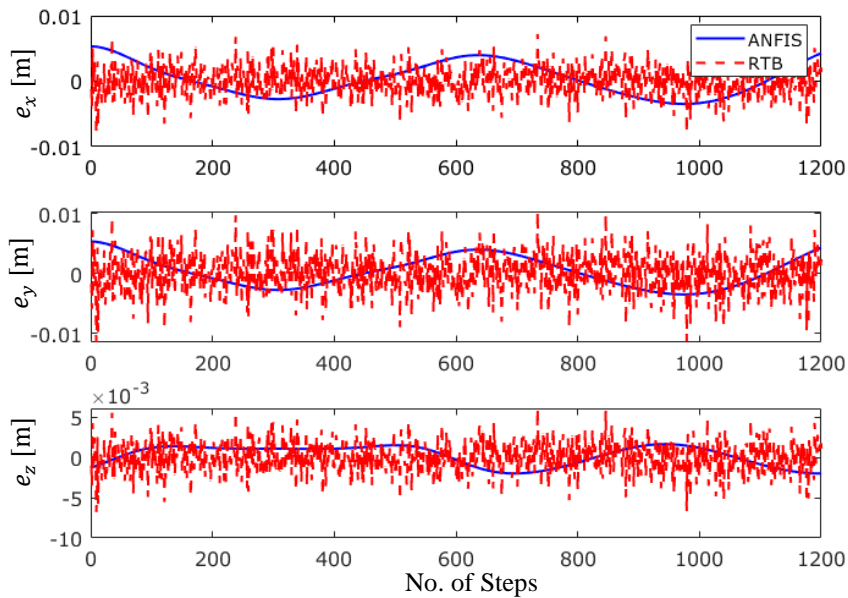
**Fig. 9. Joint angles error between the desired and the estimated (predicted) for 3D path in first simulation test.**

Figure 10 shows the desired and the predicted circular path, here the joint angles are divided into 1200 steps. Both the ANFIS network and RTB are followed the circular path in the figure; still it can be noticed that the ANFIS results unities with the desired path better than the RTB results.



**Fig. 10. End effector 3D path in the second simulation.**

Figure 11 presents the errors between the desired Cartesian coordinates  $x$ ,  $y$ , and  $z$  from the circular path (Eq. (13)) and estimated ones through using ANFIS networks and the RTB. It is clear that the network determined the correct value of the 3D coordinates with small errors.



**Fig. 11. Difference between the desired and the estimated (predicted) end-effector 3D path Cartesian coordinates in second experiment.**

Moreover, the time required for the inverse kinematics solution in case of ANFIS network is extremely small due to fact the ANFIS search nearest point related to the 3D coordinated and find the required joint angles. While the method in RTB depends algebraic solution of matrices with trigonometric function which could lead to good results but slow and with small deviation. In Table 2, a comparison between the time required for each method is addressed. The data provided in the table show that the ANFIS has the better performance in time manner over the RTB inverse kinematics function.

**Table 2. Inverse Kinematics estimation time required in seconds.**

Method	First Simulation Path		Second Simulation Path	
	One point	Whole path	One point	Whole path
<b>ANFIS</b>	0.0019	0.95505	0.0019	2.3883
<b>RTB</b>	0.0116	5.8	0.0113	14.2041

## 5. Conclusions

In this work, the problem of inverse kinematics of the robot system in the 3D workspace has been solved using an ANFIS neural network algorithm. Here, a procedure can be considered as a fast and simple method to solve the inverse kinematics problem.

The approach utilizing the motion analysis (forward kinematics) equations and trained the ANFIS network on the data produced from these equations. The solution technique was effectively verified using the simulation tests of a 4DoF IRIS robot, and the features of effectiveness, robustness, and applicability capability have been confirmed through the simulations results in the previous section.

Adaptive Neuro-Fuzzy Inference System is a convenient method which can be used to produce satisfactory solutions of the inverse kinematics problem. The results presented in this work also shown that the solution has very fast convergence times; which make the method very appropriate for practical application.

Numerous possibilities exist for the neural networks to work in conjunction with fuzzy logic, if the two methods implemented together can serve successfully the control problems in the robotics arena.

### Nomenclatures

$A_i$	The Fuzzy Set Related to the First Input.
$B_i$	The Fuzzy Set Related to the Second Input.
$C_i$	The Fuzzy Set Related to the Third Input.
$a_i$	Shifting of the $i^{th}$ Joint in z-axis direction, m.
$d_i$	Shifting of the $i^{th}$ Joint in x-axis direction, m.
$O_i^1$	Membership Function Value.
$O_i^4$	Node Function of Layer 4.
$O_i^5$	Non-fuzzification Node Function.
$P_i$	Position vector of $i^{th}$ Joint, m.
$T_{i-1}^i$	Homogeneous Transformation Matrix
$x_e$	Motion of the End-Effector in the x-axis Direction, m.

$x_p$	Predicted Motion of the End-Effector in the x-axis Direction, m.
$y_e$	Motion of the End-Effector in the y-axis Direction, m.
$y_p$	Predicted Motion of the End-Effector in the y-axis Direction, m.
$z_e$	Motion of the End-Effector in the z-axis Direction, m.
$z_p$	Predicted Motion of the End-Effector in the z-axis Direction, m.
<b>Greek Symbols</b>	
$\alpha_i$	Twist Angle of $i^{th}$ Joint about x-axis, deg.
$\theta_i$	Angle of rotation of $i^{th}$ Joint about z-axis, deg.
$\mu_i$	The Membership Function of the $i^{th}$ Node.
$\omega_i$	Rule strength of the Membership Function $i^{th}$ Node.
$\bar{\omega}_i$	Rule Firing Strength $i^{th}$ Node.
<b>Abbreviations</b>	
3D	Three Dimensional.
ANFIS	Adaptive Neuro-Fuzzy Inference System.
ANN	Artificial Neural Networks.
DH	Dinavet Hartenberg Matrix
DoF	Degrees of Freedom.
FL	Fuzzy Logic.
IRIS	Institute of Robotics and Intelligent Systems
RTB	Robotics Toolbox for use with MatLab

## References

1. Carlos, L.F.; Jesus, H.B.; Alma Y.A; and Nancy A.D. (2018). A soft computing approach for inverse kinematics of robot manipulators. *Journal of Engineering Applications of Artificial Intelligence*, 74, 104-120.
2. Al-Mashhadany, Y.I. (2012). ANFIS-inverse-controlled puma 560 workspace robot with spherical wrist. *Journal of Procedia Engineering*, 41, 700-709.
3. Jang, J.S.R. (1993). ANFIS: adaptive-network based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3), 665 - 685.
4. Masoudi, S.; Sima, M.; and Tolouelrad, M. (2018). Comparative study of ANN and ANFIS models for predicting temperature in machining. *Journal of Engineering Science and Technology*, 13(1), 211-225.
5. Wei, L.X.; Wang, H.; and Li, Y. (2003). A new solution for inverse kinematics of manipulator based on neural network. Proceedings of the Second *International Conference on Machine Learning and Cybernetics*. Xi'an, China, 1201-1203.
6. Alavandar, S.; and Nigam, M.J. (2008). Neuro-fuzzy based approach for inverse kinematics solution of industrial robot manipulators. *International Journal of Computers, Communications & Control*, 3(3), 224-234.
7. Hendarto, H.A.; Munadi; and Setiawan J.D. (2014). ANFIS Application for Calculating Inverse Kinematics of Programmable Universal Machine for Assembly (PUMA) Robot. *International Conference on Information Technology, Computer and Electrical Engineering*. Semarang, Indonesia, 35-40.

8. Duka, A.V. (2015). ANFIS based solution to the inverse kinematics of 3DoF planar manipulator. *Journal of Procedia Technology*, 19, 526-533.
9. Tho, T.P.; Thinh, N.T.; Tuan, N.T.; and Nhan, M.T. (2015). Solving inverse kinematics of delta-robot using ANFIS. *International Conference on Control, Automation and Systems*. Bexo, Busan, Korea, 790-795.
10. Patel, K. (2016). Adaptive Neuro Fuzzy Inference System for Kinematics of 6DoF Robotic Arm. *International Journal of Scientific & Engineering Research*, 7(1), 936-943.
11. Angeles, J. (2003). *Fundamentals of robotic mechanical systems: theory, methods, and algorithms*. Second Edition, Springer-Verlag.
12. Selig, J.M. (1992). *Introductory robotics*. Prentice Hall Inc.
13. Emami, M.R. (1997). *Systematic methodology of fuzzy-logic modelling and control and application to robotics*. Master thesis, Dept. of Mechanical and Industrial Engineering, University of Toronto.
14. Math Works Inc. (2014). Fuzzy logic toolbox user's guide. Retrieved November, 2018, from [https://www.mathworks.com/help/pdf\\_doc/fuzzy/fuzzy.pdf](https://www.mathworks.com/help/pdf_doc/fuzzy/fuzzy.pdf)
15. Corke, P.I. (2017). *Robotics, vision, and control: fundamental algorithms in MatLab* (2<sup>nd</sup> ed.). Springer.