

8th International Conference Interdisciplinarity in Engineering, INTER-ENG 2014, 9-10 October
2014, Tirgu-Mures, Romania

ANFIS based Solution to the Inverse Kinematics of a 3DOF Planar Manipulator

Adrian-Vasile Duka*

"Petru Maior" University of Tg. Mureș, No. 1 N.Iorga St., Tg.Mureș, 540088, Romania

Abstract

This paper investigates a method for finding a neuro-fuzzy based solution to the inverse kinematics problem of robotic manipulators. For this purpose, the case of a hypothetical 3 degree of freedom (DOF) planar robot is considered, for which computer simulation is performed in order to outline the effectiveness of the approach. By transforming the inverse kinematics problem to a fitting problem, an Adaptive Neuro-Fuzzy Inference System (ANFIS) is trained using the inverse mapping of the data provided by the forward kinematics and learns, with acceptable accuracy, the end-effector's localization to joint angles mapping.

© 2015 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of "Petru Maior" University of Tirgu Mures, Faculty of Engineering

Keywords: robotic arm; planar manipulator; inverse kinematics; trajectory; ANFIS

1. Introduction

Forward and inverse kinematics of robotic manipulators are two of the most common problems in robotics addressed in almost all major textbooks on this subject. In the forward kinematics (FK) problem the end-effector's location in the Cartesian space, that is, its position and orientation, is determined based on the joint variables. The joint variables are the angles between the links, for rotational joints, or the link extension, for prismatic joints. Conversely, given a desired end-effector position and orientation, the inverse kinematics (IK) problem refers to finding the values of the joint variables that allow the manipulator to reach the given location.

* Corresponding author. Tel.: +40-0265-233212

E-mail address: adrian.duka@ing.upm.ro

While FK is rather straightforward, solving the IK problem is a difficult and challenging task, since the mapping between the joint space and Cartesian space is nonlinear in nature and involves transcendental equations having multiple solutions. Therefore, the required computations are time consuming and do not guarantee convergence to an accurate solution.[15,16]

The scientific literature reports many methods of solving the IK problem which can be classified as closed form, numeric and iterative approaches.[16] The later one includes, among others, soft-computing techniques which are better suited for real-time control applications. These techniques, modeled upon the human mind, are characterized by approximate solutions to computationally expensive tasks, and are able to perform input-output mapping faster than the serial analytical or numerical representations.[11,20]

For instance, the use of Neural Networks (NN) for computing the solution to the IK has been reported in [14], where the NN were trained with large sets of data having the Cartesian position and orientation (given as Euler angles) as inputs and producing the angular position of every joint. In [7] a similar approach used the components of the homogeneous transformation matrix as inputs to the NN. Other reports for the use of NN in this matter include [3,6,9,12,17].

A solution to the IK, using Fuzzy Logic is presented in [11], where, this soft-computing technique is applied to the Micro-robot with 3 DOF. In [19] a Fuzzy Inverse Kinematic Mapping is produced that can be applied to both redundant and non-redundant robots.

Another successful approach uses Adaptive Neuro-Fuzzy Inference Systems (ANFIS) [13], which is a combination of both NN and Fuzzy Logic. ANFIS is applied for the computation of the IK for 2 and 3 DOF planar manipulators in [1,2,4,5,8,11]. These types of manipulators are often used in Robotics as testbeds for various algorithms or theories. There are also numerous reports of ANFIS being applied to solve the IK of robots with more than 3 DOF. For example in [18], ANFIS is applied to solve the IK of the SCARA robot, in [10] it is applied to a redundant planar 4 DOF manipulator with rotational joints and in [16] for a 6 DOF human upper limb.

However, in all these papers which report the ANFIS based solution to the IK only the position of the end-effector is taken into account. Its orientation is not used to compute the angular position of the joints.

This paper investigates the use of ANFIS to produce the solution to the IK problem for a three-link robotic manipulator. The ANFIS is trained using the data provided by the FK to learn the inverse forward mapping of the configuration space. This means, the end effector's position, as well as its orientation are given as inputs and the ANFIS identifies which joint configuration corresponds to the given localization of the end effector.

The remainder of this paper is organized as follows: Section 2 describes the model and motion constraints of the considered robot. Section 3 describes a generic structure of ANFIS networks, and creates the premises for the ANFIS system, which produces the solution to the IK problem, described in Section 4. Section 5 shows the simulation results for several trajectory tracking experiments carried out to validate the solution, and finally, Section 6 contains the conclusions extracted from this work.

2. Structure, workspace and proposed trajectory of the robotic arm

The structure of a hypothetical robotic arm, discussed in [9], is considered as testbed for the proposed algorithm. It is a three-link planar manipulator with rotational joints, having l_i , $i=\{1,2,3\}$ as link lengths and q_i , $i=\{1,2,3\}$ as joint angles.

Considering the length of the arms, $l_i=2$, $i=\{1,2,3\}$, and the joint angle constraints, $q_1 \in [0, \pi]$, $q_2 \in [-\pi, 0]$, $q_3 \in [-\pi/2, \pi/2]$, the Cartesian coordinates of the end-effector (x_E, y_E) are given by the forward kinematic equations:

$$\begin{aligned} x_E &= l_1 \cos q_1 + l_2 \cos(q_1 + q_2) + l_3 \cos(q_1 + q_2 + q_3) \\ y_E &= l_1 \sin q_1 + l_2 \sin(q_1 + q_2) + l_3 \sin(q_1 + q_2 + q_3) \end{aligned} \quad (1)$$

To completely locate the end-effector within the plane its orientation is also needed. While position (x_E, y_E) is defined by equations (1), the orientation of the end-effector Θ_E is related to the actual joint displacements as:

$$\Theta_E = q_1 + q_2 + q_3 \quad (2)$$

Equations (1) and (2) describe the position and orientation of the robot end-effector viewed from the fixed coordinate system (Ox_0y_0) attached to the base of the robot in relation to the joint variables $q_i, i=\{1,2,3\}$.

Within its workspace, the robotic arm is required to perform a circular trajectory defined by the parametric equation of the circle as:

$$\begin{aligned}x_d &= x_C + r \cos \varphi \\y_d &= y_C + r \sin \varphi \\\varphi &= [0: 2\pi]\end{aligned}\quad (3)$$

where r is the radius of the circle, (x_C, y_C) are the coordinates of the circle's centre and φ is the parametric variable in the range 0 to 2π .

Equations (3) correspond to the desired position of the end-effector. The desired orientation of the end-effector is expressed by the angle between the positive x coordinate axis and the line that connects the origin of the frame attached to the base of the robot and a point on the circle trajectory. This is equivalent to:

$$\Theta_d = \text{atan}\left(\frac{y_d}{x_d}\right) \quad (4)$$

To produce the above specified trajectory the computation of the inverse kinematics is necessary for each point on the circle. To validate the design, other trajectories (e.g. square, triangle) in the Cartesian space are proposed in Section 5 and the system's tracking capabilities are also shown there.

3. Adaptive Neuro Fuzzy Inference System (ANFIS)

This section contains an introduction of the basics of ANFIS networks. To simplify explanations, the structure of a two input type-3 ANFIS with 4 rules is considered. The structure of the considered ANFIS network is shown in Fig. 2. and the complete set of rules is shown in (5).

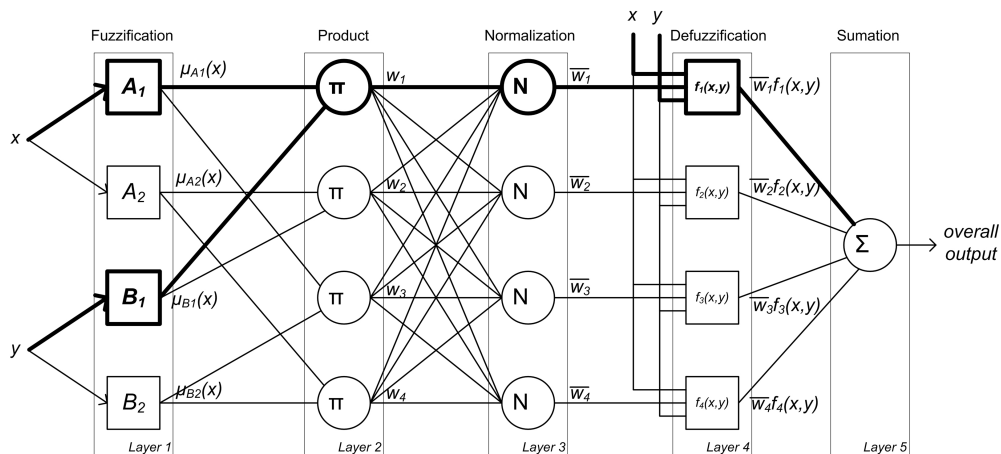


Fig 2. ANFIS architecture emphasizing rule 1 in (5)

According to [13], type-3 ANFIS uses Takagi-Sugeno if-then rules of the following form:

Rule 1: IF x is A_1 and y is B_1 THEN $f_1 = p_1 x + q_1 y + r_1$

Rule 2: IF x is A_2 and y is B_1 THEN $f_2 = p_2 x + q_2 y + r_2$

(5)

Rule 3: IF x is A_1 and y is B_2 THEN $f_3 = p_3x + q_3y + r_3$

Rule 4: IF x is A_2 and y is B_2 THEN $f_4 = p_4x + q_4y + r_4$

where: x and y are two input variables, $f_i(x,y)$ ($i = \overline{1,4}$) is one output variable, A_j and B_j ($j = \overline{1,2}$) are linguistic variables that cover the input variables universe of discourse and p_i , q_i and r_i ($i = \overline{1,4}$) are linear consequent parameters.

As seen, the output of each rule is a linear combination of input variables and a constant.

The typical ANFIS consists of a 5 layer structure. Each layer in the network is associated with a particular step in the fuzzy inference process, as described in the following section.

3.1. Layer 1: fuzzification layer

The two crisp inputs x and y are fuzzified by the nodes of this layer. For input x , each node in layer 1 has a fuzzy set A_j ($j = \overline{1,2}$) attached to it and a corresponding membership function $\mu_{A_j}(x)$ that produces a degree of membership of variable x with respect to fuzzy set A_j . This represents the output of the node. The same structure is considered for the second antecedent variable y , which is defined over a universe of discourse made of two fuzzy sets B_j ($j = \overline{1,2}$) characterized by their particular membership functions $\mu_{B_j}(x)$.

There are various choices for the membership functions of this layer (triangular, trapezoid, Gaussian, bell-shaped) which can easily be parameterized. These parameters, called premise or antecedent parameters, modify the shape of the function and are adapted during the learning stage.[8,13]

3.2. Layer 2: product (T-norm operation) layer / rule layer

The nodes of this layer implement the *and* connective in the antecedent part of the rules by using the algebraic product, or T-norm operations. This means, that the nodes of this layer produce at their output a weight function w_i ($i = \overline{1,4}$) representing the firing strength of the particular rule they represent. For example, the firing strength of Rule 1 in (5) is given by:

$$w_1 = \mu_{A_1}(x) \cdot \mu_{B_1}(y) \quad (6)$$

3.3. Layer 3: normalization layer

Each node in this layer receives inputs from all the nodes in the previous layer and calculates the normalized firing strength of a given rule. This is done by calculating the ratio of each rule's weight to the sum of all rules weights. For example the output of node 1, represents the normalized firing strength of Rule 1, that is, the contribution of that rule to the final result. Its output is given by:

$$\overline{w}_1 = \frac{w_1}{\sum_{i=1}^4 w_i} = \frac{w_1}{w_1 + w_2 + w_3 + w_4} \quad (7)$$

3.4. Layer 4: defuzzification layer

Each defuzzification node in this layer calculates the weighted consequent value of the rule it represents as:

$$\overline{w}_i \cdot f_i(x, y) = \overline{w}_i \cdot (p_i x + q_i y + r_i) \quad (8)$$

where: \overline{w}_i represents the normalized weight of rule i ; $f_i(x,y)$ is the output function in the rule's consequence p_i , q_i and r_i ($i = \overline{1,4}$) are linear consequent parameters and $i = \overline{1,4}$ represents the total number of rules in the rule base (5).

3.5. Layer 5: aggregation (summation) layer

The overall output of the ANFIS network is produced by the neuron in layer 5 which sums all the incoming signals and is expressed as:

$$\text{overall output} = \sum_{i=1}^4 \overline{w_i} \cdot f_i(x, y) = \frac{\sum_{i=1}^4 w_i \cdot f_i(x, y)}{\sum_{i=1}^4 w_i} \quad (9)$$

As seen from Fig. 2, the ANFIS network consists of square nodes and circle nodes. The nodes in layers 1 and 4 are square (adaptive), meaning that the premise and consequent parameters change during the learning process. The other layers have circle nodes (non-adaptive), which implement basic functions (product, normalization, summation) and have fixed parameters.

For training ANFIS networks a hybrid learning procedure is used most of the time, which combines the backpropagation gradient descent method to tune the premise parameters of the membership functions and the least squares estimate (LSE) to identify the consequent parameters of the output functions. [8,16]. A detailed reference to ANFIS and its learning procedure is available in [13].

4. ANFIS structure for computing the IK

To compute the solution to the IK problem specified by equations (3) and (4) (as well as for the other experiments detailed in Section 5) the ANFIS architecture of Fig. 3 is used. It consists of three ANFIS networks, one for each joint angle.

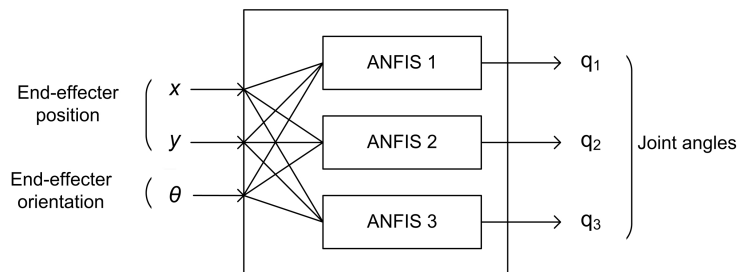


Fig. 3. Proposed ANFIS architecture for computing the inverse kinematics of the 3DOF planar manipulator

In order to produce the training set for the ANFIS networks, the same approach as in [9] was used. This means that random joint angle values $(q_1, q_2, q_3)^T$, which cover the specified ranges, were generated and, using the FK equations (1) and (2), the corresponding localization of the end-effector was computed $(x_E, y_E, \theta_E)^T$.

ANFIS networks provide a method to learn information about a data set. In this case the inverse information contained in the data set is learned, meaning finding the inputs that yield a desired output. For this reason, as seen in Fig. 3, the input data for training ANFIS network i ($i=\{1,2,3\}$) is the localization vector $(x_E, y_E, \theta_E)^T$ resulted from the FK and the target data is the corresponding joint parameter q_i of the random generated vector $(q_1, q_2, q_3)^T$.

After generating the primary training set, using the above mentioned procedure, the duplicates in the input data are identified and the corresponding training set (input data and target data) is removed. This way, each configuration the ANFIS network is learning refers to a unique mapping from Cartesian space to joint space.

Each of the three ANFIS networks uses 6 membership functions for each of the three input variables x , y and θ , meaning a rule base made of 216 rules is generated for each network. A standard choice for the input membership functions was considered, which uses the generalized bell function of equation (10).

$$\mu(x; a, b, c) = \frac{1}{1 + \left| \frac{x - c}{a} \right|^{2b}} \quad (10)$$

The shape of the membership function in (10) depends on three parameters a , b (usually positive), and c (locates the center of the curve). These parameters represent the premise parameters, adapted during the learning process.

The output membership functions are linear and have the following general form:

$$f(x, y, \Theta; p, q, r, t) = p \cdot x + q \cdot y + r \cdot \Theta + t \quad (11)$$

where p, q, r, t are the consequent parameters, which are also adapted during the learning process.

Matlab's Fuzzy Logic Toolbox software, through its *anfis* function constructs the necessary fuzzy inference system and computes (adjusts) the membership function parameters in (10) and (11) that best allow the designed ANFIS to match the input/output data specified by the training set. It does this through a learning process which is a combination of least squares estimation and backpropagation for membership function parameter estimation.

This structure of the ANFIS networks resulted after running several training scenarios with different configurations of the fuzzy inference system. These scenarios considered (for each network) combinations of 3, 5, 6 membership functions per input / output, training sets ranging from 100 to 1000 input-output data samples and up to 200 training epochs. Using a validation set made of input/output data, the difference between the desired and the resulted output value of the ANFIS network has been evaluated in order to identify the resulting ANFIS structure.

5. Results

After each ANFIS network is trained using a training set made of 1000 input-output data samples for 200 epochs, the IK solution, corresponding to a desired trajectory, can be computed by applying the Cartesian coordinates and the orientation of the points on that trajectory as inputs to the ANFIS networks in Fig. 3. Each of the three ANFIS networks will produce it's corresponding joint parameter value, which combined will result in a joint vector that will place the end-effector on the required trajectory with the specified orientation.

Fig. 4 shows the resulting joint space trajectories corresponding to the circular trajectory defined in the Cartesian space, which was specified by equations (3) and (4).

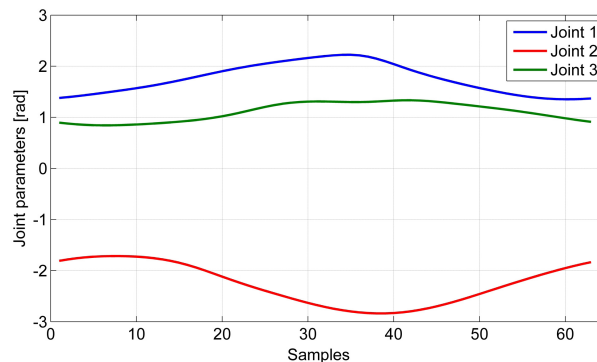


Fig. 4. Joint space trajectories.

The points on the joint space trajectories can be easily interpolated to 5th order polynomials and the trajectory repeated without having to appeal each time to the computational requirements of the ANFIS network. However,

this is possible only if the required Cartesian trajectory is continuous. In case of discontinued Cartesian trajectories this interpolation can be applied only for the continuous sections.

Fig. 5 shows the tracking capabilities of the robot in case the localization of the end-effector, corresponding to the proposed circular trajectory, is applied to the ANFIS networks. Fig. 5.a. shows how the end-effector tracks the desired position and Fig. 5.b. shows the end-effector's tracking capabilities for the desired orientation.

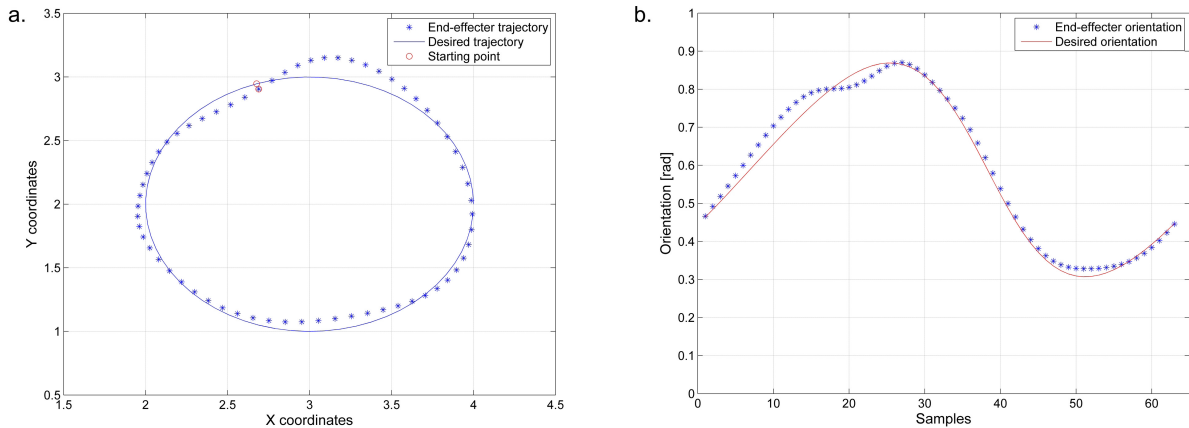


Fig.5. Circle trajectory: (a) End-effector position; (b) End-effector orientation.

To further test the ANFIS capabilities to produce the solution to the IK other trajectories have been considered. For example, Fig. 6.a. shows the robot's tracking capabilities for a square trajectory defined within its workspace. Again, the solution to the IK is computed by the ANFIS networks. For this trajectory a constant orientation of the end-effector is imposed: $\Theta=0$ rad (parallel to the horizontal axis).

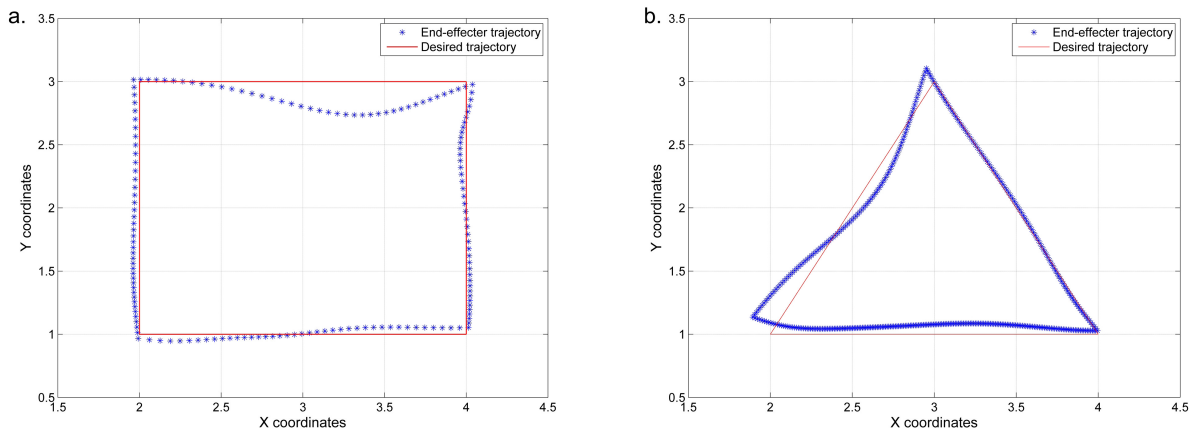


Fig.6. End-effector position. (a) Square trajectory; (b) Triangle trajectory.

In Fig. 6.b., the trajectory of a triangle in the Cartesian space is considered, and an orientation of the end-effector as specified in equation (4), corresponding to the first example. In both cases the orientation error was under 10%.

As seen in Fig. 5 and 6 there are sections where the generated trajectory could be improved. Probably a longer training is required, or the training samples should have provided a better coverage of that particular area from the workspace. This issue should be further addressed. Probably another DOF should be added to the robotic arm, resulting in a redundant robot with increased mobility. However, this solution would complicate the structure of the

ANFIS network and would increase the duration of the training and the necessary amount of memory needed for the training process.

6. Conclusion

All results shown in the previous section were obtained using Matlab modeling of the proposed system.

This paper addressed the problem of finding a solution to the IK for a 3 DOF robotic manipulator using a neuro-fuzzy approach. While this problem has been also studied by other researchers, none considered introducing the orientation of the end-effector as a condition used to compute the angular position of the joints by means of an ANFIS network. While the tracking capabilities of the system could be improved this paper shows that the inverse kinematic problem can be transformed to a fitting problem and that ANFIS networks have the ability to fit multi-dimensional mapping problems arbitrarily well, given consistent data, enough fuzzy sets to cover the input variable ranges and a proper training algorithm.

References

- [1] Alavandar S., Nigam M.J., Neuro-Fuzzy based Approach for Inverse Kinematics Solution of Industrial Robot Manipulators, *International Journal of Computers Communications & Control*, 2008; 3:224-234.
- [2] Alavandar S, Nigam MJ. Inverse Kinematics Solution of 3DOF Planar Robot using ANFIS, *International Journal of Computers Communications & Control*, Suppl. issue: Proceedings of ICCCC 2008; 3:150-155.
- [3] Ali T Hasan, Hayder MAA Al-Assadi, Ahmad Azlan Mat Isa. Neural Networks' Based Inverse Kinematics Solution for Serial Robot Manipulators Passing Through Singularities, *Artificial Neural Networks. Industrial and Control Engineering Applications*, Prof. Kenji Suzuki (Ed.), InTech, 2011.
- [4] Ankarali A. ANFIS inverse kinematics and precise trajectory tracking of a dual arm robot, *Proceedings of the 2012 International Conference on Modelling, Simulation and Visualization Methods*, 2012, pp. 270-274.
- [5] Ankarali A, Cilli M. ANFIS Inverse Kinematics and Hybrid Control of a Human Leg Gait Model, *Academic Platform Journal of Engineering and Science* 2013; 1(2):34-49.
- [6] Bassam Daya, Shadi Khawandi, Mohamed Akoum. Applying Neural Network Architecture for Inverse Kinematics Problem in Robotics. *J. Software Engineering & Applications* 2010; 3:230-239.
- [7] Bingul Z, Ertunc HM, Oysu C. Applying Neural Network to Inverse Kinematic Problem for 6R Robot Manipulator with Offset Wrist. *Adaptive and Natural Computing Algorithms*, 2005, p. 112-115.
- [8] Chen CH, Subbaram Naidu D. Hybrid control strategies for a five-finger robotic hand, *Biomedical Signal Processing and Control* 2013; 8(4):382-390.
- [9] Adrian-Vasile Duka, Neural Network based Inverse Kinematics Solution for Trajectory Tracking of a Robotic Arm, *Procedia Technology* 2014; 12:20-27.
- [10] M.P.Flower Queen, M.SasiKumar, P.BabuAurtherson, Inverse kinematics solution of 4 d.o.f Industrial manipulator using neuro-fuzzy Network, *International Journal of Emerging Technology and Advanced Engineering* 2012; 2(8):276-280.
- [11] Howard David W, Zilouchian A. Application of Fuzzy Logic for the Solution of Inverse Kinematics and Hierarchical Controls of Robotic Manipulators, *Journal of Intelligent and Robotic Systems*, Kluwer Academic Publishers 1998; 23(2-4):217-247.
- [12] Jack H, Lee DMA, Buchal RO, Elmaraghy WH. Neural networks and the inverse kinematics problem. *Journal of Intelligent Manufacturing* 1993; 4:43-66.
- [13] Jang JSR. ANFIS: adaptive-network-based fuzzy inference system, *IEEE Transactions on Systems, Man and Cybernetics* 1993; 23(3):665-685.
- [14] Karlik B, Aydin S. An improved approach to the solution of inverse kinematics problems for robot manipulators. *Engineering Applications of Artificial Intelligence* 2000, 13(2):159–164.
- [15] Kucuk S, Bingul Z. Robot Kinematics: Forward and Inverse Kinematics. *Industrial Robotics: Theory, Modelling and Control*, Sam Cubero (Ed.), InTech., 2006
- [16] Pérez-Rodríguez R. et.al., Inverse kinematics of a 6 DoF human upper limb using ANFIS and ANN for anticipatory actuation in ADL-based physical Neurorehabilitation, *Expert Systems with Applications* 2012; 39(10):9612-9622.
- [17] Sreenivas Tejomurtula, Subhash Kak, Inverse kinematics in robotics using Neural Networks. *Information Sciences* 116, 1999, pp. 147-164.
- [18] Wesam Mohammed Jasim, Solution of Inverse Kinematics for SCARA Manipulator Using Adaptive Neuro-Fuzzy Network, *International Journal on Soft Computing (IJSC)* 2011; 2(4):59-66.
- [19] Yangsheng Xu, Nechyba MC. Fuzzy inverse kinematic mapping: rule generation, efficiency, and implementation, *Intelligent Robots and Systems '93*, Proceedings of the 1993 IEEE/RSJ International Conference on 1993; 2:911-918.
- [20] Zadeh L. Fuzzy Logic, Neural Networks, and Soft Computing, *Communications of the ACM* 1994, 37(3):77-84