Chapter 14

Design and Simulation of Anfis Controller for Virtual-Reality-Built Manipulator

Yousif I. Al Mashhadany

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/48383

1. Introduction

Fuzzy logic (FL) and artificial neural networks (ANNs), despite their successful use in many challenging control situations, still have drawbacks that limit them to only some applications. Their combined advantages have thus become the subject of much research into ways of overcoming their disadvantages. Neuro-fuzziness is one resulting rapidly emerging field. ANFIS network, proposed by Jang, is one popular neuro-fuzzy system [1-4].

For specific-problem training of an ANFIS network, [1] proposes use of hybrid learning rule, which combines gradient descent technique and least-square estimator (LSE). Being a method of supervised learning, it needs a teaching signal, which can be difficult to provide when the ANFIS network is to be a feedback controller, as the desired control actions that the teaching signal represents are unknown. Literatures have proposed several ANFIS learning methods in which ANFIS is applied as a MIMO controller. Djukanović *et al.*, for example, uses a special ANFIS learning technique called temporal back propagation (TBP); control of a nonlinear MIMO system is by considering both the controller and the plant as a single unit each time step. The method, however, is complex and distinctly computation-heavy [5-9].

Another training approach for ANFIS-controller of nonlinear MIMO systems is inverse learning; the ANFIS network is trained to learn the inverse dynamics of the plant it controls. Its success, however, is crucial on three elements: accurate modeling of the original system (a problem when the system is complex), availability of the system's inverse dynamics (they do not always exist), and appropriate distribution of the training data (could be impossible, given the constraints of the system's dynamics). [10-13] has another training approach besides the ones already mentioned.

Present robot navigation systems demand controllers that can solve complex problems under uncertain and dynamic environments. ANFIS garners interest because it offers the benefits of both neural network (NN) and FL, and removes their individual disadvantages by combining them on their common features. ANN is a new motivation for studies into FL. It can be used as a universal learning paradigm in any smooth parameterized models, including fuzzy inference systems [14-15].

Traditional robot control methods rely on strong mathematical modeling, analysis, and synthesis. Existing approaches suit control of mobile robots operating in unknown environments and performing tasks that require movement in dynamic environments. Operational tasks in unstructured environments such as remote planets and hazardous waste sites, however, are more complex, yet the analytical modeling is inadequate. Many researchers and engineers have tried to solve the navigational problems of mobile robot systems [16-18]. Though fuzzy systems can use knowledge expressed in linguistic rules (and thus could implement expert human knowledge and experience), fuzzy controller lacks a systematic design method. Tuning of membership-function parameters takes time. NN learning techniques can automate the process so development can be hastened and performance improved. The combination of NN and FL has produced neuro-fuzzy controllers and created their present popularity. In real-time autonomous navigation, a robot must be able to sense its environment, interpret the sensed information to obtain knowledge of its position and environment, and plan a route that gets it to the target position from an initial position and with obstacle avoidance and control of its direction and velocity. Ng et al. [19-23] propose a neural-integrated fuzzy controller that integrates FL representation of human knowledge with NN learning to solve nonlinear dynamic control problems. Pham et al. focus on developing intelligent multi-agent robot teams capable of both autonomous action and dynamic-environment collaboration in achieving team objectives. They also propose a neuro-fuzzy adaptive action selection architecture that enables a team of robot agents to achieve adaptive cooperative control of cooperative tasks, track dynamic targets, and push boxes. Crestani et al. defines autonomous navigation in mobile robots as a search process within a navigation environment that contains obstacles and targets, and propose a fuzzy-NN controller that considers navigation direction and navigation velocity as controllable. Rutkowski et al. derived a flexible neuro-fuzzy inference system; their approach increases structural and design flexibility in neuro-fuzzy systems. Hui et al. and Rusu et al. discuss neuro-fuzzy controllers for sensor-based mobile robot navigation. Garbi et al. implemented an adaptive neuro-fuzzy inference system in robotic vehicle navigation [24-26].

Robots are one way to improve industrial automation productivity. Robotic manipulators have been used in routine and dangerous-environment manufacturing jobs. They are highly nonlinear dynamic systems subject to uncertainties. Obtaining accurate dynamic equations for their control laws is thus difficult. Uncertainties in their dynamic models include unknown grasped payloads and unknown frictional coefficients. Adaptive control or model-free intelligent control has been much proposed as able to compensate for those uncertainties [26-28].

Virtual reality (VR) has become important to applications in engineering, medicine, statistics, and other areas where 3D images can aid understanding of system complexity. The interactability of a virtual system can in many applications be enhanced by touch sensing. Haptic feedback can convey to a human user, virtual environment forces. It has become useful in tele-surgery, where a master manipulator guides robotic surgical tools while providing realistic-force feedback to the surgeon. It is already available in many systems under development but these often still are specifically developed research prototypes (i.e., providing specific-force feedbacks and for specific problems) [29, 30].

Kinematics analysis is key to motion control of humanoid manipulators. Its main problems are forward kinematics and inverse kinematics. The inverse kinematics of a 7-DOF manipulator has multiple solutions; obtaining anthropomorphic solutions is thus a problem. Detailed research in it has yet to be found, though works on design, control, and obstacle avoidance in humanoid manipulator exist. In serial manipulators, between forward kinematics and inverse kinematics, solution of the latter is a lot more difficult. Solution methods of inverse kinematics generally are numerical, analytical, or geometric. Numerical method is most widely used, but it cannot obtain all possible solutions. Analytical method can derive all possible solutions but is much more difficult. Geometric method is simple and easy to understand but suits only a few types of manipulators [31, 32].

A human-manipulator skeleton has 7-DOF mechanism; other degrees of freedom of the arm are performed by the tendon. A 7-DOF anthropomorphic arm has been developed; it has the manipulability criterion and uses 3-DOF planar manipulator theory (i.e., the mechanism of the human limb can be modeled by three moving links). 3-DOF planar mechanism is fundamental to an anthropomorphic arm; its evaluation criteria can be used to analyze the arm's operational performance. 3-DOF planar manipulator theory is a supposition that the three link lengths are of the upper arm, the forearm, and the hand, and the three joints are 3-DOF shoulder joint, 1-DOF elbow joint, and 3-DOF wrist joint [33].

This chapter presents the design of an ANFIS controller of a VR manipulator model and simulation of the ANFIS-controlled system's command execution. Simulation results of the 7-DOF human-manipulator show an improved control system. Section II presents the manipulator's kinematic model whereas Section III the structure of the ANFIS controller and its learning methods. Section IV presents the manipulator's VR model, whereas Section V, the simulation of the case study, the system design, and the conclusions.

2. Kinematics of the 7-DOF manipulator

Forward kinematics was used to calculate the racket's posture according to the joint angles. It is quite useful for analysis of the manipulator's workspace and verification of the inverse kinematics. Denoting the shoulder width as D, the upper arm length as L_1, and the lower arm length as L_2, the position of the shoulder is thus $P_1(0, -D, 0)$ and the position of the elbow is P_2 (x_{p2}, y_{p2}, z_{p2}). Denoting the joint angles as $q_1 \ldots q_7$, the posture of joint i relative to joint i-1 can thus be described by a 4×4 homogeneous matrix ${}^{i-1}\boldsymbol{T}_i$ [34]:

$$\left.\begin{array}{l}
{}^0T_1 = \begin{bmatrix} cq_1 & 0 & sq_1 & 0 \\ 0 & 1 & 0 & -D \\ -sq_1 & 0 & cq_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^1T_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cq_2 & -sq_2 & 0 \\ 0 & sq_2 & cq_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^2T_3 = \begin{bmatrix} cq_3 & -sq_3 & 0 & 0 \\ sq_3 & cq_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^3T_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cq_4 & -sq_4 & 0 \\ 0 & sq_4 & cq_4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^4T_5 = \begin{bmatrix} cq_5 & -sq_5 & 0 & 0 \\ sq_5 & cq_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^5T_6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cq_6 & -sq_6 & 0 \\ 0 & sq_6 & cq_6 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^6T_7 = \begin{bmatrix} cq_7 & 0 & sq_7 & 0 \\ 0 & 1 & 0 & -D \\ -sq_7 & 0 & cq_7 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{array}\right\} \quad (1)$$

Where: $sq_i \equiv \sin(q_i)$; $cq_i \equiv \cos(q_i)$

Assuming the position of Joint-7 in the fixed coordinate is P_3 (x_{p3} , y_{p3} , z_{p3}) and its pose as described by RPY (Roll Pitch Yaw) angles is (ϕ, θ, ψ), the posture of Joint-7 can thus be described by a homogenous matrix ${}^0\boldsymbol{T}_7$:

$${}^0T_7 = \begin{bmatrix} c\varphi c\theta & c\varphi s\theta s\psi - s\varphi c\psi & c\varphi s\theta c\psi + s\varphi s\psi & x_{p3} \\ s\varphi c\theta & s\varphi s\theta s\psi + c\varphi c\psi & s\varphi s\theta c\psi - c\varphi s\psi & y_{p3} \\ -s\theta & c\theta s\psi & c\theta c\psi & z_{p3} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Matrix ${}^0\boldsymbol{T}_7$ also stands for the racket's posture and can be derived also as:

$${}^0T_7 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 {}^6T_7 \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

From (1) and (2), the racket's posture can be calculated as:

$$\left\{\begin{array}{c} \varphi = a\tan 2(n_y, n_x) \\ \theta = a\tan 2(-n_z, c\varphi n_x + s\varphi n_y) \\ \psi = a\tan 2(s\varphi a_x - c\varphi a_y, -s\varphi o_x + c\varphi o_y) \end{array}\right. \quad \& \quad \left.\begin{array}{c} x_{p3} = p_x \\ y_{p3} = p_y \\ z_{p3} = p_z \end{array}\right\} \quad (4)$$

with atan2( ) being the four-quadrant inverse tangent function and (3) the manipulator's forward kinematics. All the commands (e.g., the racket's hitting position and hitting speed) are given by the visual system, in the operation space. They must be transformed into joint space values. Jacobian matrix is used to calculate the joint space speed according to the operation space speed. The mapping relationship between them is [35]:

$$V = J(q)\dot{q} \quad (5)$$

with $\boldsymbol{V}$ being the racket's speed in the operation space and $\dot{q}$ the joint space speed. $\boldsymbol{J}$ is the 6×7 Jacobian matrix and can be derived by differential transformation method. The i^{th} item of $\boldsymbol{J}$ is:

$$J_i = \begin{bmatrix} (p_i \times n_i)_k \\ (p_i \times o_i)_k \\ (p_i \times o_i)_k \\ n_{ik} \\ o_{ik} \\ a_{ik} \end{bmatrix} \tag{6}$$

Where ***n**i*(*nix*, *niy*, *niz*), ***o**i*(*oix*, *oiy*, *oiz*), ***a**i*(*aix*, *aiy*, *aiz*) and ***p**i*(*pix*, *piy*,*piz*) are the items of matrix ${}^{i-1}T_i$

$${}^{i-1}T_7 = {}^{i-1}T_i \ldots\ldots {}^{6}T_7 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

The parameter *k* stands for the rotational axis of joint *i*. For example, if the joint rotates around the *x*-axis, then *k* is *x*. Racket speed ***V*** can be calculated by (4) whereas joint space speed $\dot{q}$ cannot be uniquely determined because ***J*** is not a square matrix. This problem can be solved by Moore-Penrose method:

$$\dot{q} = J^{+}V \tag{8}$$

with $J^{+} = (J^{T}J)^{-1}J^{T}$ being the Moore-Penrose pseudo inverse matrix of ***J***.

The inverse kinematics is used to calculate the joint angles according to the racket's posture. The manipulator is redundant, so the elbow's position is not uniquely determined when the racket's posture is given. The motion characteristics of a human arm show the mapping relationships between elbow position and racket posture. In a general configuration of the manipulator (see Fig. 2), any three points on the manipulator's neck, shoulder, elbow, and wrist are not collinear. The position of elbow P_2, whose axis is P_1P_3, is in circle O'. The position of the center point $O' = (x_{O'}, y_{O'}, z_{O'})$ and the radius *r* of the circle are determined by positions P_1 and P_3, and arm length parameters L_1 and L_2. Ω_1 denotes the plane constructed by points P_1, P_2, and P_3, and Ω_2 the plane constructed by points O, P_1, and P_2. α denotes the separation angle between Ω_1 and Ω_2. Once the position of P3 is known, α can uniquely determine P2 elbow position. If the points O, P_1, P_3 are collinear whereas P_1, P_2, P_3 are not, the plane Ω_2 does not exist. Angle α can be defined as the separation angle between Ω_1 and the horizontal plane. Also, if the points P1, P2, P3 are collinear, the plane Ω_1 does not exist and P2 position can be calculated according to positions P_1 and P_3 and arm length parameters L_1 and L_2. Considering the general configuration of the manipulator (see Fig. 2), the inverse kinematics can be solved if angle α and the racket's posture are given. According to the motion characteristics of human arms, the mapping relationship between α and the racket posture can be built offline. Denoting with ***i*** (i_1, i_2, i_3) the unit vector pointing from P_3

to P1, and assuming the angle α to have been calculated by a well-trained ANN model, the circle $O^{'}$ can thus be expressed as [34-36]:

$$\begin{Bmatrix} i_1(x-x_{o^{'}})+i_2(y-y_{o^{'}})+i_3(z-z_{o^{'}})=0 \\ (x-x_{o^{'}})^2+(y-y_{o^{'}})^2+(z-z_{o^{'}})^2=r^2 \end{Bmatrix} \tag{9}$$

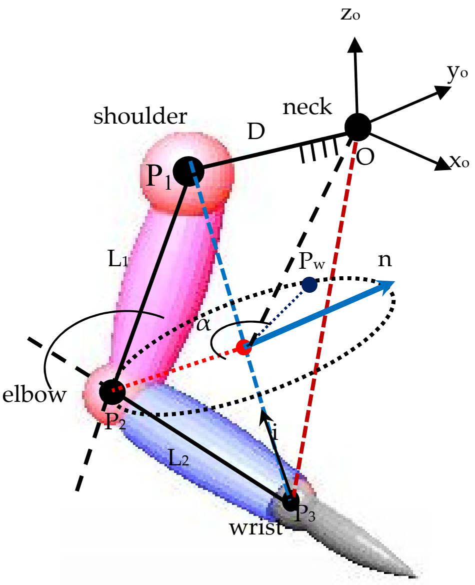


Figure 1. General configuration of the manipulator

The plane Ω2 and the circle $O^{'}$ intersect at two points. One point near the neck is Pw (*xw*, *yw*, *zw*) and angle ∠P2O$^{'}$ Pw is the separation angle α between Ω1 and Ω2. Denoting Ω2 norm vector as ***n*** and position as Pw should satisfy this equation:

$$\overrightarrow{OP_w}.n=0 \tag{10}$$

Pw position can be calculated from (9) and (10). P2 position (*xp2*, *yp2*, *zp2*) satisfies this equation:

$$\sqrt{(x_{p2}-x_m)^2+(y_{p2}-y_m)^2+(z_{p2}-z_m)^2}=2r\sin\frac{\alpha}{2} \tag{11}$$

P2 position can be calculated from (9) and (11). With P1, P2, P3 positions and the cosine theorem, Joint-4 angle can be calculated as:

$$q_4=\pi-a\cos(\frac{L_1^2+L_2^2-\left\|\overline{P_1P_3}\right\|^2}{2L_1L_2}) \tag{12}$$

With P2 position, this can be obtained:

$$\begin{bmatrix} x_{p2} & y_{p2} & z_{p2} & 1 \end{bmatrix}^T ={}^0 T_1{}^1T_2{}^2T_3\begin{bmatrix} 0 & 0 & -L_1 & 1 \end{bmatrix}^T \tag{13}$$

The angles of Joints 1 and 2 can thus be derived as:

$$\left.\begin{array}{ll} q_1 = a\tan 2(\dfrac{-x_{p2}}{cq_2L_1},\dfrac{-z_{p2}}{cq_2L_1}) + k\pi & (k \in N) \\ q_2 = a\sin((y_{p2} + D) / L_1) & \end{array}\right\} \tag{14}$$

With P3 position, this can be obtained:

$$\begin{bmatrix} x_{p3} & y_{p3} & z_{p3} & 1 \end{bmatrix}^T ={}^0 T_1{}^1T_2{}^2T_3{}^3T_4\begin{bmatrix} 0 & 0 & -L_2 & 1 \end{bmatrix}^T \tag{15}$$

Joint-3 angle can thus be derived as:

$$\left\{\begin{array}{l} q_3 = a\tan 2(\dfrac{y_3 + D - sq_2L_1 - sq_2cq_4L_2}{cq_1cq_2sq_4L_2}sq_1sq_2 - \dfrac{x_3 + sq_1cq_2(1 + cq_4)L_1}{sq_2cq_1L_1} \\ \qquad\qquad\qquad ,\dfrac{y_3 + D - sq_2L_1}{cq_2sq_4L_2} - \dfrac{sq_2cq_4}{cq_2sq_4}) \end{array}\right\} \tag{16}$$

From (2),

$$T_5^{\,4}T_6^{\,5}T_7^6 = \left[T_1^{\,0}T_2^{\,1}T_3^{\,2}T_4^3\right]^{-1}T_7^6 = T^* \tag{17}$$

The angles of Joints 5-7 can thus be derived as:

$$\left.\begin{array}{ll} q_5 = a\tan 2(-T_{13}^*, T_{23}^*) + k\pi & (k \in N) \\ q_6 = a\tan 2((sq_5T_{13}^* - sq_5T_{23}^*, T_{33}^*) & \\ q_7 = a\tan 2((-cq_5T_{12}^* - sq_5T_{22}^*, cq_5T_{11}^* + sq_5T_{21}^*) & \end{array}\right\} \tag{18}$$

with T_{ij}^* being the i^{th} row and j^{ij} the column item of T

3. General structures of ANFIS

ANFIS integrates ANN with FIS. The ANFIS analyzed here was a first-order Takagi Sugeno Fuzzy Model. The analysis has four inputs: front obstacle distance (x_1), right obstacle distance (x_2), left obstacle distance (x_3), and target angle (x_4). The output is steering angle. The 'if-then' rules are [26, 37]:

$$\begin{array}{l} Rule: \; IF \;\; x_1 \;\; is \;\; A_j; \;\; x_2 \;\; is \;\; B_k; \;\; x_3 \;\; is \;\; C_m \;\; and \;\; x_4 \;\; is \;\; D_n \\ THEN \quad F_i = p_ix_1 + r_ix_2 + s_ix_3 + t_ix_4 \end{array} \tag{19}$$

Where

$$\begin{array}{l} F_i = p_i x_1 + r_i x_2 + s_i x_3 + t_i x_4 + u_i \quad \text{for} \quad \text{steering} \quad \text{angle} \\ J = 1 \quad \text{to} \quad q_1; \quad k = 1 \quad \text{to} \quad q_2; \quad m = 1 \quad \text{to} \quad q_3; \quad n = 1 \quad \text{to} \quad q_4; \quad \text{and} \quad i = 1 \quad \text{to} \quad q_1.q_2 \quad q_3.q_4 \end{array} \tag{20}$$

A, B, C, and D are the fuzzy membership sets defined for input variables x_1, x_2, x_3, and x_4. q_1, q_2, q_3, and q_4 are the number of membership functions, respectively for the fuzzy systems of inputs x_1, x_2, x_3 and x_4. fi is the linear consequent functions defined in terms of inputs x_1, x_2, x_3, and x_4. q_i, r_i, s_i , ti, and ui are consequent parameters of an ANFIS fuzzy model. Same-layer nodes of an ANFIS model have similar functions. Output signals from the nodes of a preceding layer are input signals to a present layer. The output obtained through the node function will be input signals to the next layer (see Figure 2) [38, 39].

Layer 1: This is the input layer, which defines obstacles as either static or moving and also the tracker robot's target position. It receives signals from x_1, x_2, x_3, and x_4.

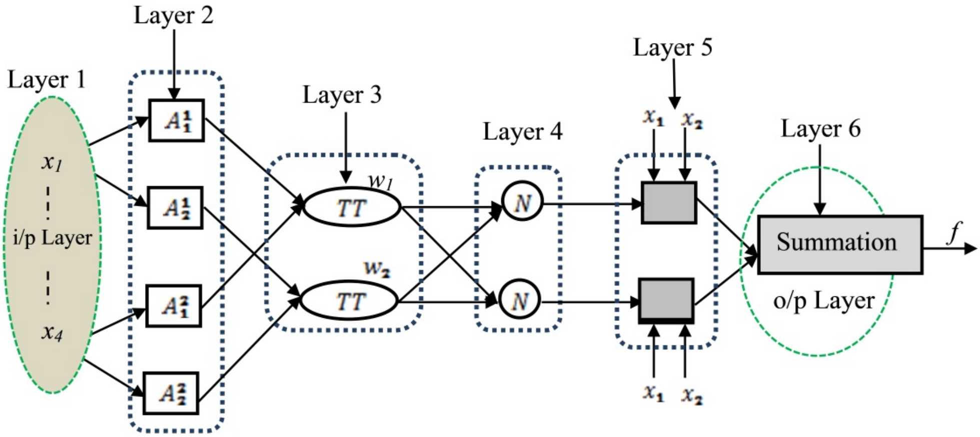


Figure 2. Structure of a six-layer ANFIS

Layer 2: Every node in this layer is an adaptive node (square node) with a particular fuzzy membership function (node function) specifying the degrees to which the inputs satisfy the quantifier. For four inputs, the node outputs are:

$$\begin{array}{llll} L_{2g} = U_{ag}(x) & \text{for} & g = 1,.....,q_1 & (\text{for} \quad \text{input} \quad x_1) \\ L_{2g} = U_{Bg}(x) & \text{for} & g = q_1 + 1,.....,q_1 + q_2 & (\text{for} \quad \text{input} \quad x_1) \\ L_{2g} = U_{Cg}(x) & \text{for} & g = q_1 + q_2 + 1,.....,q_1 + q_2 + q_3 & (\text{for} \quad \text{input} \quad x_1) \\ L_{2g} = U_{Dg}(x) & \text{for} & g = q_1 + q_2 + q_3 + 1,.....,q_1 + q_2 + q_3 + q_4 & (\text{for} \quad \text{input} \quad x_1) \end{array} \tag{21}$$

The membership functions considered here for A, B, C, and D are bell-shaped functions and defined as:

$$\left.\begin{aligned}
\mu_{Ag}(x) &= \frac{1}{1+\left\{\left(\frac{x-c_g}{a_g}\right)^2\right\}^{b_g}}; \quad g=1 \quad to \quad q_1 \\
\mu_{Bg}(x) &= \frac{1}{1+\left\{\left(\frac{x-c_g}{a_g}\right)^2\right\}^{b_g}}; \quad g=q_1+1 \quad to \quad q_1+q_2 \\
\mu_{Bg}(x) &= \frac{1}{1+\left\{\left(\frac{x-c_g}{a_g}\right)^2\right\}^{b_g}}; \quad g=q_1+q_2+1 \quad to \quad q_1+q_2+q_3 \\
\mu_{Bg}(x) &= \frac{1}{1+\left\{\left(\frac{x-c_g}{a_g}\right)^2\right\}^{b_g}}; \quad g=q_1+q_2+q_3+1 \quad to \quad q_1+q_2+q_3+q_4
\end{aligned}\right\} \tag{22}$$

with a_g, b_g, and c_g being the parameters for fuzzy membership function. The bell-shaped function changes its pattern with changes to the parameters. This change will give various contours of the bell-shaped function, as needed and in accordance with the data set for the problem considered.

Layer 3: Every node in this layer is a fixed node (circular) labeled 'π'. L_{2i} output is the product of all incoming signals.

$$\begin{aligned}
&L_{3i} = W_i = U_{ag}(x), U_{Bg}(x), U_{Cg}(x), U_{Cg}(x); \\
&For\, i = 1,....,q_1+q_2+q_3+q_4 \quad and \quad g = 1,....,q_1+q_2+q_3+q_4
\end{aligned} \tag{23}$$

Each of the second layer's node output represents the firing strength (degree of fulfillment) of the associated rule. The T-norm operator algebraic product $\{T_{ap}(a,b) = ab\}$ was used to obtain the firing strength (W_i).

Layer 4: Every node in this layer is a fixed node (circular) labeled "N". The output of the ith node is the ratio of the firing strength of the ith rule (W_i) to the sum of the firing strength of all the rules [40].

$$L_{4i} = \overline{W}_i f_i = \frac{W_i}{\sum_{r=1}^{r=q_1 \cdot q_2 \cdot q_3 \cdot q_4} W_r} \tag{24}$$

This output gives a normalized firing strength.

Layer 5: Every node in this layer is an adaptive node (square node) with a node function.

$$L_{5i} = \overline{W}_i f_i = \overline{W}_i (p_i x_1 + r_i x_1 + s_i x_3 + t_i x_4 + u_i) \tag{25}$$

with $\overline{W_i}$ being the normalized firing strength form (output) from Layer-3 and {p$_i$, r$_i$, s$_i$, t$_i$, u$_i$} the steering-angle parameter set. Parameters in this layer are consequent.

Layer 6: The single node in this layer is a fixed node (circular) labeled "Σ". It computes the overall output as the summation of all incoming signals.

$$L_{6i} = \sum_{r=1}^{r=q_1 \cdot q_2 \cdot q_3 \cdot q_4} \overline{W}_i f_i = \frac{\sum_{i=1}^{i=q_1 \cdot q_2 \cdot q_3 \cdot q_4} \overline{W}_i f_i}{\sum_{i=1}^{i=q_1 \cdot q_2 \cdot q_3 \cdot q_4} W_i} \tag{26}$$

This work's ANFIS development has six-dimensional space partitions and q$_1$, q$_2$, q$_3$, and q$_4$ regions. Each region is governed by a fuzzy if-then rule. The first layer is the input layer. The second contains premise or antecedent parameters of the ANFIS and is dedicated to fuzzy sub-space. Consequent parameters of the fifth layer were used to optimize the network. During the forward pass of the hybrid learning algorithm, node outputs go forward until Layer-5 and the consequent parameters are identified by least-square method. In the backward pass, error signals propagate backwards and the premise parameters are updated by gradient descent method [40, 41].

4. VR Modeling of the 7-DOF manipulator

Design requirements for Virtual Reality Modeling Language (VRML) are described in finite processing allocations, autonomy, consistent self-registration, and calculability. VRML design procedure will be presented. Design in VRML depends on the information available to the designer and his imaging of the object. There are two choices for VR design: one is standard configuration such as sphere, cone, cylinder, etc., another is free design by selecting indexed face set button to get many configurations with free rearrangement of points; every real-form design is thus considered to be the latter [42-45], which starts with building parts one by one and comparing the shape's similarity against that of the real manipulator part. Manipulator parts cannot be simulated in VR when the VR library's standard shapes (they are not uniform) are used. Designing thus uses indexed face set in VR. The second choice to be made in design work is very important as it is about connecting all parts to get the final object, and limiting the object's original point. This is the starting point of the design work. The first shape (e.g., the shoulder) is first set, and then the next shape (forearm) is connected to the "children" button. The same procedure is repeated for other parts. Figure 3 presents the full design of the 7-DOF human arm manipulator [46-50].

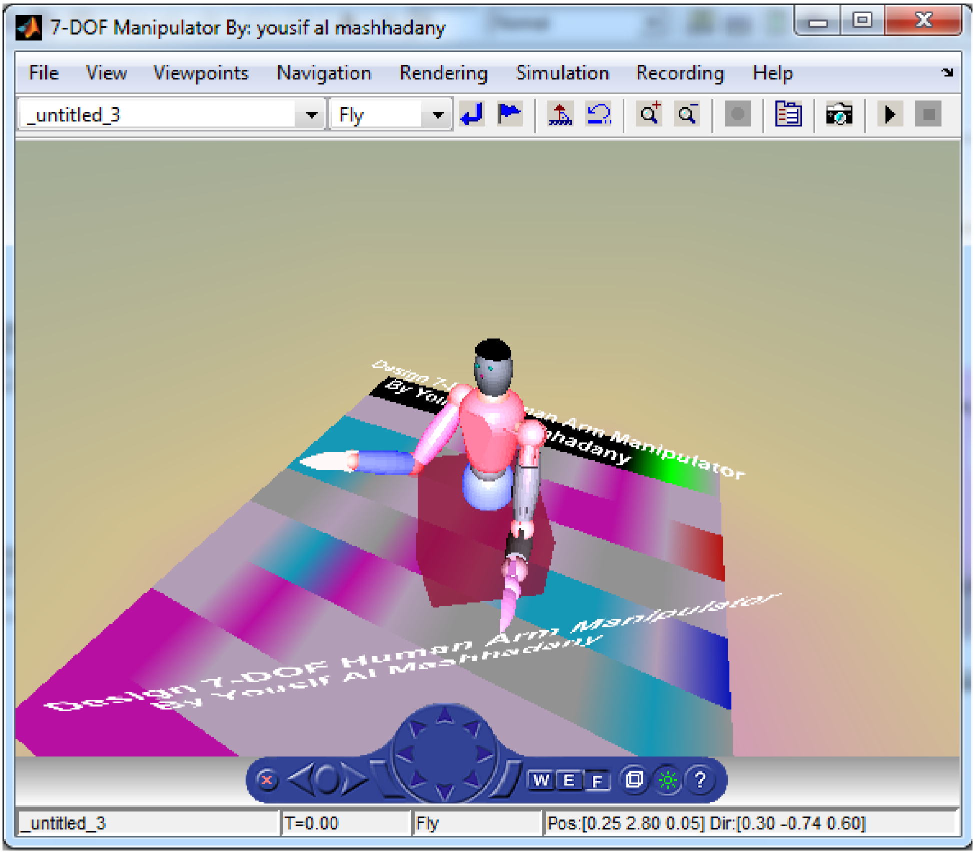


Figure 3. The VR-modeled 7-DOF human-arm manipulator

5. Design of the ANFIS controller for the 7-DOF manipulator

The design work considered many parameters related to the system's real values (see Table 1 for real limits of the joints of a human arm). The joints motor is considered a real DC-motor with a transfer function similar with that in Equation (1).

$$T.F_m(s) = \frac{1}{s^2 + 6s + 8} \tag{27}$$

Figure 4 is a block diagram of the control system. The inputs to the system design have two sets of targets: orientation (θ_{T1}, θ_{T2}, θ_{T3}) and position (T_x, T_y, T_z). In the proposed technique, each joint has its own controller, so in all, seven identical (same structure, same training algorithm) ANFIS controllers were used. Inputs of the training algorithm were the desired joint angle and the actual angle. The desired values were calculated by using analytical solution of the IKP algorithm shown previously. Actual values of the joint angle were obtained by feedback from the virtual model.

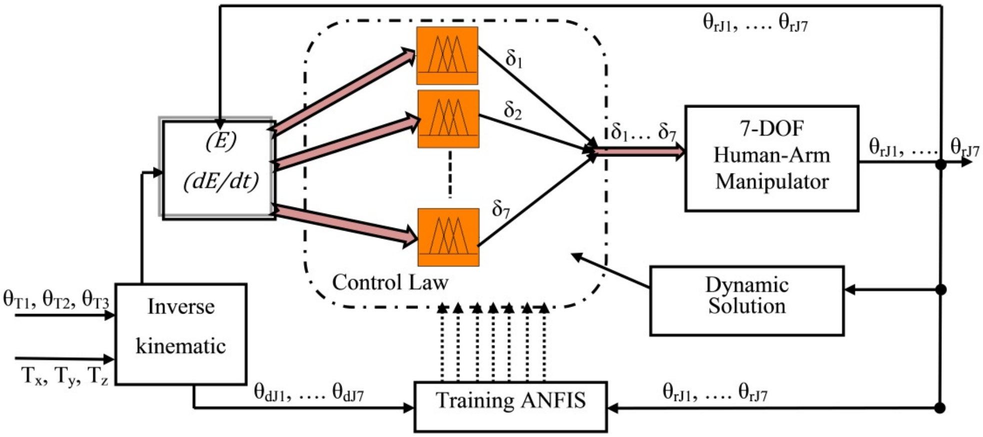
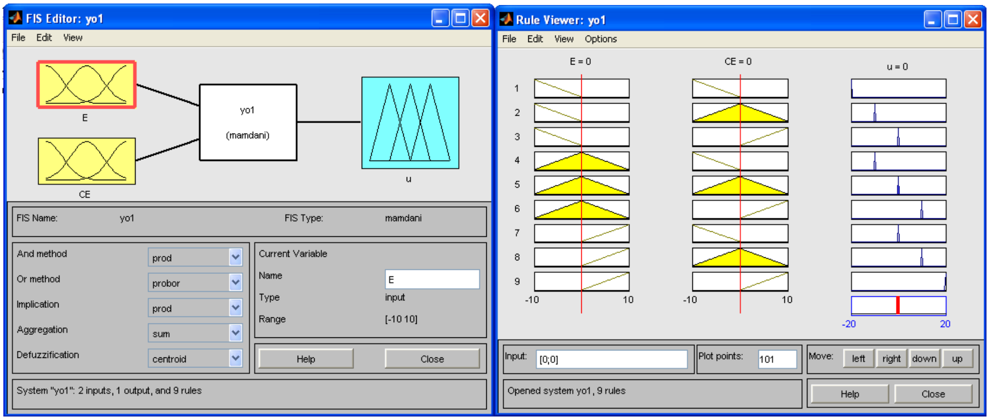

θ_{rJ1}, …. θ_{rJ7}
(E)
(dE/dt)
δ_1
δ_2
δ_7
δ_1… δ_7
7-DOF Human-Arm Manipulator
θ_{rJ1}, …. θ_{rJ7}
Control Law
Dynamic Solution
θ_{T1}, θ_{T2}, θ_{T3}
Inverse kinematic
T_x, T_y, T_z
θ_{dJ1}, …. θ_{dJ7}
Training ANFIS
θ_{rJ1}, …. θ_{rJ7}

Figure 4. ANFIS control of the 7-DOF human-arm manipulator

The structure of the ANFIS controller was built in Matlab software Ver.2011b, with two inputs, an error signal, and change in the error. The fuzzy inference method used was Mamdani's, because it is intuitive, widely accepted, and well-suited to human input, and, for the proposed control structure, it gives better results than does Sugeno inference method. In designing the controller, types of membership functions were tried before selecting the best: triangular built-in membership function (trimf). The trial-and-error design approach for the ANFIS controller, i.e., selecting the interference type, the membership function type, and the number of this membership function in the hidden layer, gave optimal results: minimum number of rules and simple simulation. The design uses seven parallel-connected ANFIS to compute the optimal deflection of the joints and get the desired angle. Figures (5-7) show the procedure of the Matlab-fuzzy-toolbox-based design.



FIS Name: yo1

FIS Type: mamdani

And method: prod

Or method: probor

Implication: prod

Aggregation: sum

Defuzzification: centroid

Current Variable

Name: E

Type: input

Range: [-10 10]

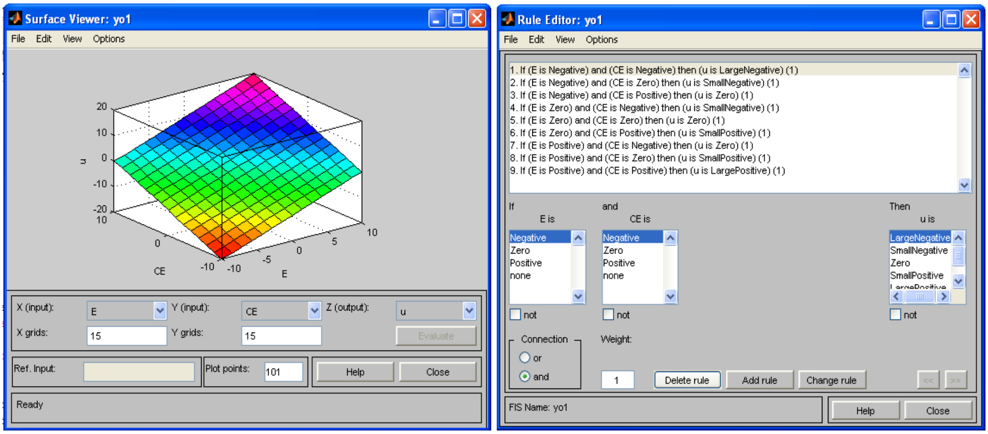
System "yo1": 2 inputs, 1 output, and 9 rules

Input: [0;0]

Plot points: 101

Opened system yo1, 9 rules

Figure 5. Mamdani's fuzzy inference and the set of rules for training of the ANFIS controller



X (input): E Y (input): CE Z (output): u

X grids: 15 Y grids: 15

Ref. Input: Plot points: 101

Ready

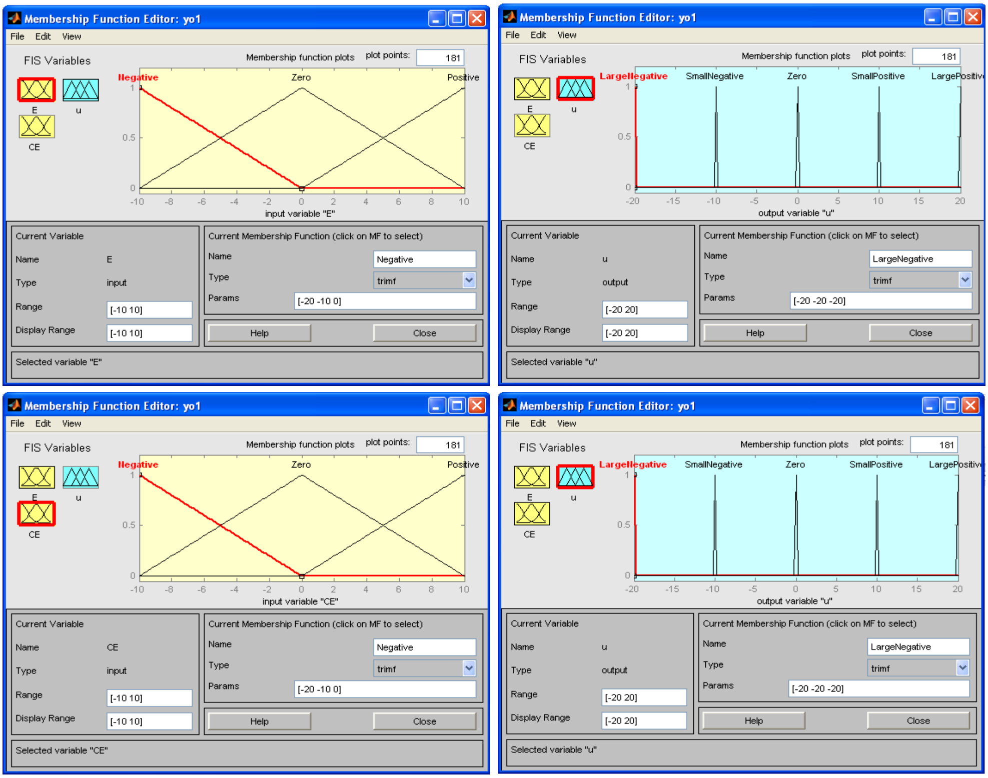
1. If (E is Negative) and (CE is Negative) then (u is LargeNegative) (1)
2. If (E is Negative) and (CE is Zero) then (u is SmallNegative) (1)
3. If (E is Negative) and (CE is Positive) then (u is Zero) (1)
4. If (E is Zero) and (CE is Negative) then (u is SmallNegative) (1)
5. If (E is Zero) and (CE is Zero) then (u is Zero) (1)
6. If (E is Zero) and (CE is Positive) then (u is SmallPositive) (1)
7. If (E is Positive) and (CE is Negative) then (u is Zero) (1)
8. If (E is Positive) and (CE is Zero) then (u is SmallPositive) (1)
9. If (E is Positive) and (CE is Positive) then (u is LargePositive) (1)

□ not □ not □ not

Connection ○ or ● and Weight: 1

FIS Name: yo1

Figure 6. The surface error and the set of rules for training of the ANFIS controller



plot points: 181

Current Variable — Name: E; Type: input; Range: [-10 10]; Display Range: [-10 10]

Current Membership Function (click on MF to select) — Name: Negative; Type: trimf; Params: [-20 -10 0]

Selected variable "E"

Current Variable — Name: u; Type: output; Range: [-20 20]; Display Range: [-20 20]

Current Membership Function (click on MF to select) — Name: LargeNegative; Type: trimf; Params: [-20 -20 -20]

Selected variable "u"

Current Variable — Name: CE; Type: input; Range: [-10 10]; Display Range: [-10 10]

Current Membership Function (click on MF to select) — Name: Negative; Type: trimf; Params: [-20 -10 0]

Selected variable "CE"

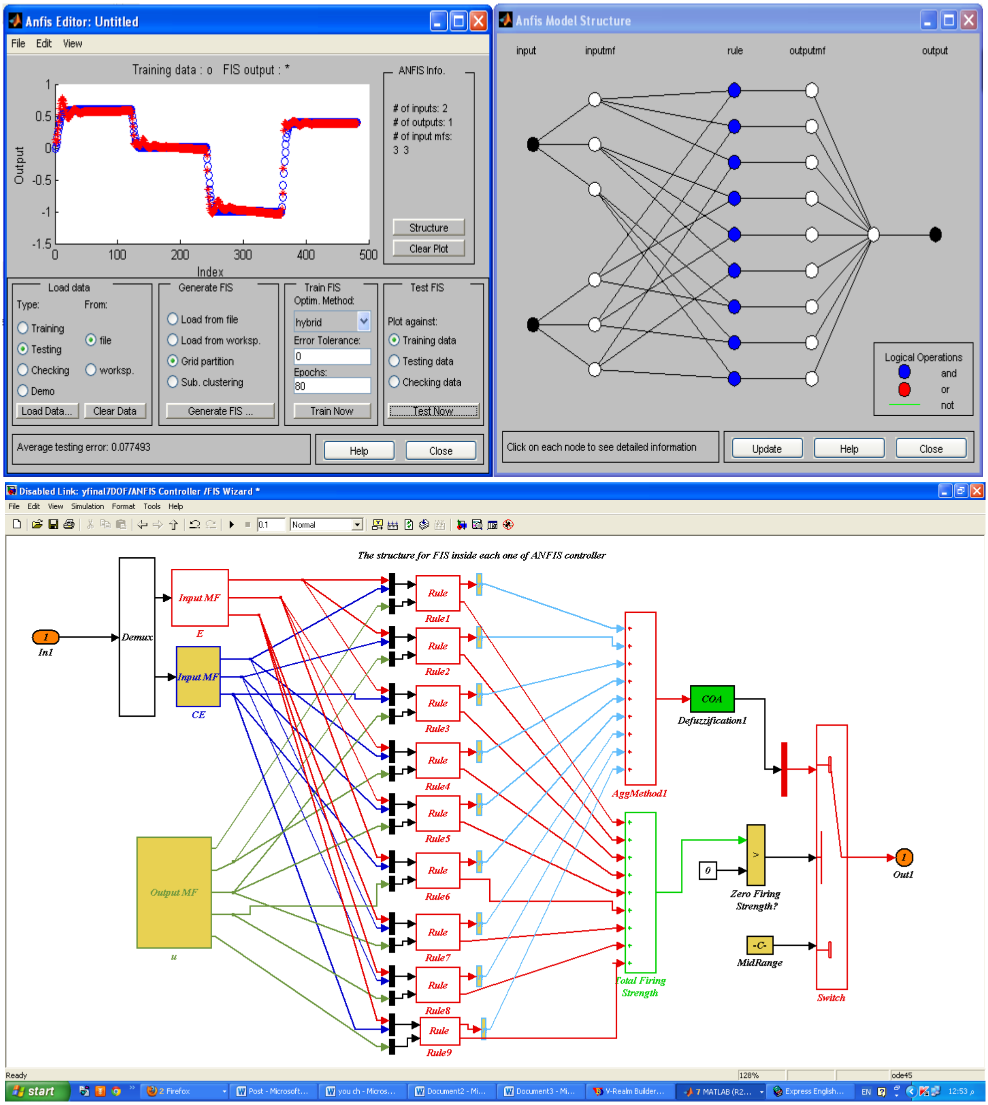
Current Variable — Name: u; Type: output; Range: [-20 20]; Display Range: [-20 20]

Current Membership Function (click on MF to select) — Name: LargeNegative; Type: trimf; Params: [-20 -20 -20]

Selected variable "u"

Figure 7. The triangular built-in memberships function with error signal and change in error signal

Figure 8 shows in Matlab Simulink GUI window the internal structure of the ANFIS controller and the training output for the input signal, with the various steps used in instructing the manipulator's movement. The ANFIS training used hybrid training algorithm, with the input nodes (3, 3) to the membership functions each having nine rules (see Figure 5). Epoch length was used in training eighty iterations for each sample, with 0.01s Simulink sampling time.



Anfis Editor: Untitled

Training data : o FIS output : *

ANFIS Info.
# of inputs: 2
# of outputs: 1
# of input mfs:
3 3

Load data — Type: ○ Training ● Testing ○ Checking ○ Demo — From: ● file ○ worksp.

Generate FIS — ○ Load from file ○ Load from worksp. ● Grid partition ○ Sub. clustering

Train FIS — Optim. Method: hybrid — Error Tolerance: 0 — Epochs: 80

Test FIS — Plot against: ● Training data ○ Testing data ○ Checking data

Average testing error: 0.077493

Anfis Model Structure

Click on each node to see detailed information

Logical Operations: 🔵 and, 🔴 or, not

Disabled Link: yfinal7DOF/ANFIS Controller /FIS Wizard *

The structure for FIS inside each one of ANFIS controller

Figure 8. The ANFIS structure in Matlab-Simulink GUI and its training response

6. Simulation results

Figure 9(a) is a block diagram of the simulation done for the ANFIS-controlled manipulator. Seven ANFIS controllers were used, one each for each degree of freedom of the three joints (3 for the shoulder, 1 for the elbow, and 3 for the wrist). The seven were effective in tracking the trajectory desired for the 7-DOF manipulator. The controller's rules base has 9 rules, each determined by fuzzy neural network (FNN). The desired position and orientation were, in simulation entered as input signal, whereas actual positions for the joints were given as feedback from the output signal. Figure 9(b) shows the results of using both ANFIS and PID controllers on the joints and for the ANFIS controller's implementation into the virtual model. Performance of the ANFIS-controlled joint was better than that of the PID-controlled one. The ANFIS controller effected fast response in the manipulator and reduced errors, for various complex trajectories of the manipulator.

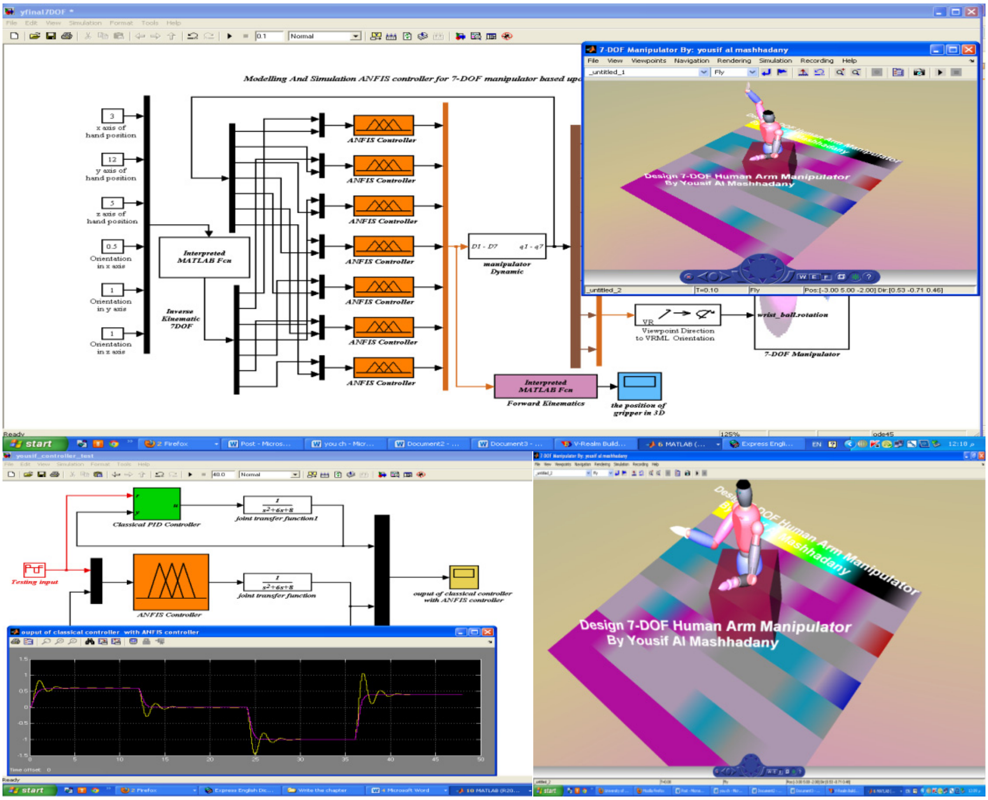


Figure 9. (a, b, c). Simulation of the ANFIS controller for the VR-implemented 7-DOF manipulator

The manipulator's movements used the link between Matlab-Simulink and VR environment. Computation for the order of movement was done in Matlab-Simulink. The order was then sent to the VR model to implement, with considerations for the axes between

Matlab's and VR's to get the real movement (i.e., for general motion: elbow flexion/extension, elbow rotation (supination/pronation), shoulder adduction/abduction, shoulder flexion/extension, shoulder interior/exterior rotation, wrist flexion/extension, wrist ulnar/radial deviation, shoulder horizontal flexion/extension; for special motion: arm reaching towards the head, arm reaching to the right and the head, arm reaching to the left and the head, and arm reaching to the right) (see Figure 10 for examples).

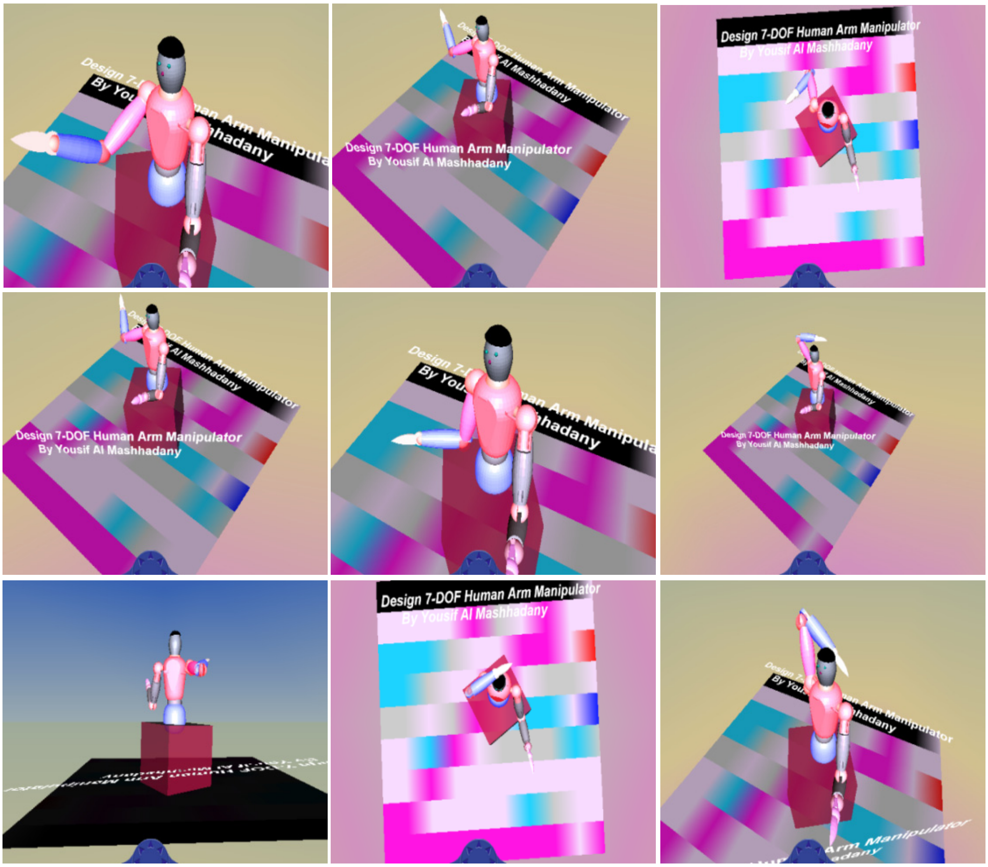


Figure 10. Postures of the ANFIS-controlled 7-DOF manipulator

7. Conclusion

This work aimed to design an ANFIS-based controller that overcomes the general problems of FL and NN in a dynamic system. To obtain excellent manipulator postures, training of the ANFIS controller for all the elements (fuzzy interference, membership function, number of neurons, number of rules) was by trial and error. The controller was compared with classical PID controller, in tracking the speed and the joint angle's accuracy. The simulation results allowed these conclusions to be drawn:

- For any speed, the ANFIS controller has the better transient response and the steadier state response than does a PID controller; even the best-tuned PID controller is unable to perform well in both slow-speed and high-speed ranges.
- With only 9 neurons, 4 layers, and 9 rules, the proposed ANFIS controller is simpler than other adaptive neuro-fuzzy controllers reported by many other researches.
- The ANFIS controller does not require an accurate model of the plant. Its relative simplicity makes it fairly easy to construct and implement. High-level knowledge of system is not needed to build a set of rules for a fuzzy controller or for the identification needed in an NN controller.
- The ANFIS controller was used in both simulation and experiment in trajectory tracking of multiple manipulators. Kinematic analysis of the 7-DOF manipulator produced the calculation for the envelope of similar manipulators.
- VR implementation of the manipulator was able to show the latter's accuracy. The manipulator can be depended on for many high-accuracy applications such as collecting of minute/fragile items.

Author details

Yousif I. Al Mashhadany
Electrical Dept., Al Anbar University, Engineering College, Baghdad, Iraq

Acknowledgement

Special thanks to Ms. Wirani M. Munawir of Verdana Inc. (verdana.inc@live.com) for her editing of the text.

8. References

[1] AWARE M. V, KQTHARI A.G, CHOUBE S.O, (2000), "Application of adaptive neuro-fuzzy controller (ANFIS) for voltage source inverter fed induction motor drive", IEEE Power Electronics and Motion Control Conference, 2000. Proceedings. IPEMC 2000. The Third International 945-939

[2] Choon Y. L, Lee J, (2005)," Multiple Neuro-Adaptive Control of Robot Manipulators Using Visual Cues", IEEE transactions on industrial electronics, Vol. 52, No.1, 320-326.

[3] Hui C, Gangquan S, Yanbin Z, Xikui M, (2007),"A Hybrid Controller of Self-Optimizing Algorithm and ANFIS for Ball Mill Pulverizing System", Proceedings of the IEEE International Conference on Mechatronics and Automation, Harbin, China

[4] Swasti R. K, Sidhartha P, (2010), "ANFIS Approach for TCSC-based Controller Design System Stability Improvement Design for Power" , IEEE ICCCCT-10, 149-154

[5] Prabu D, Surendra K, Rajendra P, (2011), "Advanced Dynamic Path Control of the Three Links SCARA using Adaptive Neuro Fuzzy Inference System", IN BOOK, Robot Manipulators, Trends and Development, InTech, ch 18, ISBN: 978-953-307-073-5. pp. 399-412

[6] Zhiqiang G, Thomas A. T, James G. D, (2000),"A Stable Self-Tuning Fuzzy Logic Control System for Industrial Temperature Regulation", IEEE Industry Applications Conference, Vol 2, 1232-1240

[7] Jeich M, Feng-J. L, (2001), "An ANFIS Controller for the Car-Following Collision Prevention System", IEEE transactions on VEHICULAR TECHNOLOGY, Vol. 50, No. 4, 1106-1113.

[8] Hassanzadeh, S. K, liang G. A, (2002), "Implementation of .I Functional Link Net-ANFIS Controller for a Robot Manipulator", Third International Workshop on Robot Motion and Control, 399-404

[9] Saifizul A. A., Zainon M. Z, Abu Osman N. A, (2006), "An ANFIS Controller for Vision-based Lateral Vehicle Control System", Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference on IEEE ICARCV, 1-4.

[10] Ravi S, Balakrishnan P .A, (2010), "Modeling and control of an ANFIS temperature controller for plastic extrusion process", IEEE Communication Control and Computing Technologies, ICCCCT-10, 314-320.

[11] Omar F. L, Samsul B. M. N, Mohammad H. M, (2011), "A Genetically Trained Simplified ANFIS Controller to Control Nonlinear MIMO Systems", International Conference on Electrical, Control and Computer Engineering Pahang, Malaysia, 349-354

[12] Sgg Pedro P. C, Maya R. E, (2004), "Vector Control using ANFIS Controller with Space Vector Modulation", Universities Power Engineering Conference, 2004. UPEC 2004. 39th International, 545-549

[13] Thair S. M, Mohammed H. M, Tang S. H, Sokchoo N, (2008), "ANFIS Controller with Fuzzy Subtractive Clustering Method to Reduce Coupling Effects in Twin Rotor MIMO System (TRMS) with Less Memory and Time Usage", IEEE International Conference on Advanced Computer Control, 19-24

[14] Mohammad A, (2006), "ANFIS Based Soft-Starting and Speed Control of AC Voltage Controller Fed Induction Motor", 0-7803-9525-5/06/6 IEEE Power India Conference.

[15] Srinivasan A, Nigam M. J, (2008), "Adaptive Neuro-Fuzzy Inference System based control of six DOF robot manipulator", Journal of Engineering Science and Technology Review ,106- 111

[16] Sanju S, Sarita R, (2012), "Temperature Control Using Intelligent Techniques", IEEE Second International Conference on Advanced Computing & Communication Technologies, 138-145

[17] Abolfazl H. N, Abolfazl V, Hassan M, (2006), "ANFIS-Based Controller with Fuzzy Supervisory Learning for Speed Control of 4-Switch Inverter Brushless DC Motor Drive", Power Electronics Specialists Conference, PESC '06. 37th IEEE , 1-5

[18] Chitra V, (2010), "ANFIS Based Field Oriented Control for Matrix Converter fed Induction Motor", IEEE international conference on power and energy (PECon2010), Malaysia, 74-78

[19] Muthukumar G. G, Albert V. T. A, (2011), "A DE – ANFIS Hybrid Technique For Adaptive Deadbeat Controller", IEEE 978-1-61284-379-7/11, 1st International Conference on Electrical Energy Systems, 134-139

[20] Yajun Z, Tianyou C, Yue F, Hong N, (2009), "Nonlinear Adaptive Control Method Based on ANFIS and Multiple Models", Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference Shanghai, P.R. China, 1387-1392I

[21] Al-Mashhadany Y. I, (2011), "Modeling and Simulation of Adaptive Neuro-Fuzzy Controller for Chopper-Fed DC Motor Drive", IEEE applied power electronic colloquium (IAPEC), 110-115

[22] Ginarsa M, Soeprijanto A, Purnomo M.H, (2009), "Controlling Chaos Using ANFIS-Based Composite Controller (ANFIS-CC) in Power Systems", Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME).

[23] Yajun Z, Tianyou C, Hong W, (2011), "A Nonlinear Control Method Based on ANFIS and Multiple Models for a Class of SISO Nonlinear Systems and Its Application", IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 22, NO. 11, 1783-1795

[24] JafarT, AfsharShamsi J, Muhammad A. D, (2012), "A new method for position control of a 2-DOF robot arm using neuro– fuzzy controller", Indian Journal of Science and Technology Vol. 5 No. 3, ISSN: 0974- 6846, 2253-2258

[25] Giuliani P. G, Gamarra R, Francisco J. G, (2007), "Multivalued Adaptive Neuro-Fuzzy Controller for Robot Vehicle", Proceedings on Intelligent Systems and Knowledge Engineering (ISKE).

[26] Singh M. K, Parhi D. R, Pothal J. K, (2009), "ANFIS Approach for Navigation of Mobile Robots", 2009 International Conference on Advances in Recent Technologies in Communication and Computing, 727-731

[27] Ming-Y. S, Ke H. C, Chen Y. C, Juing S. C, Jeng H. L, (2007), "ANFIS based Controller Design for Biped Robots", IEEE Proceedings of International Conference on Mechatronics TuA2-B-1 Kumamoto Japan, 1-6

[28] Widodo B, Achmad J, Djoko P, (2010), Indoor Navigation using Adaptive Neuro Fuzzy Controller for Servant Robot", IEEE Second International Conference on Computer Engineering and Applications, 582-586

[29] Greg R. L, (2011), "Haptic Interactions Using Virtual Manipulator Coupling with Applications to Underactuated Systems", IEEE transactions on robotics, Vol. 27, No. 4, 730-740.

[30] Al-Mashhadany Y. I,(2012), "A Posture of 6-DOF Manipulator By Locally Recurrent Neural Networks (LRNNs) Implement in Virtual Reality", IEEE Symposium on Industrial Electronics and Applications (ISIEA), Penang, Malaysia, 573-578.

[31] Jacob R, Joel C. P, Nathan M, Stephen B, Blake H, (2005), "The Human Arm Kinematics and Dynamics During Daily Activities – Toward a 7 DOF Upper Limb Powered Exoskeleton", This work is supported by NSF Grant #0208468 entitled "Neural Control of an Upper Limb Exoskeleton System" - Jacob Rosen (PI) 0-7803-9177-2/05/ IEEE, 532-539

[32] Erico G, Travis D, (2012), "Robotics Trends for 2012", IEEE Robotic & Automation Msgazine, 1070-9932/12/, 119-123

[33] Shahri M. R. A, Khoshravan H, Naebi A, (2011) " Design ping-pong player robot controller with ANFIS", IEEE Third International Conference on Computational Intelligence, Modelling & Simulation, 165-189

[34] Younkoo J, Yongseon L, Kyunghwan K, Yeh-Sun H, Jong-Oh P, (2001), "A 7 DOF Wearable Robotic Arm Using Pneumatic Actuators", Proceedings of the 32nd ISR(International Symposium on Robotics), 19-21

[35] Tie-jun Z, Jing Y, Ming-yang Z, Da-long T, (2006), "Research on the Kinematics and Dynamics of a 7-DOF Arm of Humanoid Robot", IEEE International Conference on Robotics and Biomimetics , Kunming, China, 1553-1558

[36] Zhang B, Xiong R, WU J, (2011), "Kinematics Analysis of a Novel 7-DOF Humanoid Manipulator for Table Tennis", NSFC-2008AA042601/978-1-4577-0321-8/11/IEEE, 1524-1528

[37] Patricia M, Oscar C, (2005), "Intelligent control of a stepping motor drive using an adaptive neuro–fuzzy inference system", Elsevier Information Sciences 170,133–151 www.elsevier.com/locate/ins

[38] Al-maliki K. H, Wali W.A, Hameed L. J, Turky Y. A, (2011), "Force / Motion Control for Constrained Robot Manipulator Using Adaptive Neural Fuzzy Inference System (ANFIS)", it is available with (the last visit at 12/4/2012) http://www.nauss.edu.sa /En/DigitalLibrary/Researches/Documents/2011/articles_2011_3157.pdf

[39] Oscar C, Patricia M, (2003), "Intelligent adaptive model-based control of robotic dynamic systems with a hybrid fuzzy-neural approach", Elsevier Applied Soft Computing, 363–378

[40] Mukhtiar S, Chandra A, (2012), "Real Time Implementation of ANFIS Control for Renewable Interfacing Inverter in 3P4W Distribution Network", This article has been accepted for publication in a future issue of IEEE journal

[41] Ouamri B, Ahmed Z, (2012), "Adaptive Neuro-fuzzy Inference System Based Control of Puma 600 Robot Manipulator", International Journal of Electrical and Computer Engineering (IJECE) Vol.2, No.1, ISSN: 2088-8708, 90~97

[42] Greg R. L, (2011), "Haptic Interactions Using Virtual Manipulator Coupling with Applications to Underactuated Systems", IEEE transactions on robotics, Vol. 27, No. 4, 730-740.

[43] Zappi M., Maltina R., Cerveri,P, (2010), "Modular micro robotic instruments for transluminal endoscopic robotic surgery: new perspectives", 978-1-4244-7101-0110/IEEE journal, 440-446

[44] Eckhard F, Jürgen R, (2003), "Integrating Robotics and Virtual Reality with Geo-Information Technology: Chances and Perspectives", In L. Bernhard, A. Sliwinski, K. Senkler (Hrsg.): Geodaten- und Geodienste-Infrastrukturen – von der Forschung zur praktischen Anwendung. Beiträge zu den Münsteraner GI-Tagen 2003, Reihe IfGI Prints, Uni Münster, 285-296

[45] Michal J, Michal P, Michal C , Peter Nand O, (2012), "Towards Incremental Development of Human-Agent-Robot Applications using Mixed-Reality Testbeds", This article has been accepted for publication in IEEE Intelligent Systems but has not yet been fully edited. Some content may change prior to final publication.

[46] Ben C, (2009), "Humanoid Robotic Language and Virtual Reality Simulation" in book "Humanoid Robots", InTech ,chapter 1, 1-20

[47] Syed I. A, Nageli V. S, Sangeeta S, Rakshit S, (2012), "Digital Sand Model using Virtual Reality Workbench", IEEE International Conference on Computer Communication and Informatics (ICCCI -2012), Coimbatore, INDIA

[48] Manuela C, Fabio S, Silvio P, (2011), "Virtual Reality to Simulate Visual Tasks for Robotic Systems", IN BOOK In tech, ISBN: 978-953-307-518-1 " Virtual Reality"ch 4,, 71-92

[49] Patton J. L, Dawe G, Scharver C, Mussa-Ivaldi1 F. A, Kenyon R, (2012), "Robotics and Virtual Reality: The Development of a Life-Sized 3-D System for the Rehabilitation of Motor Function" Supported by NIDRR RERC. 0330411Z, NIH 1 R24 HD39627-0, NIH 1 R01- NS35673-01, and the Falk Trust

[50] Rong-wen H, Chia-hui L, (2007),"Development of Fuzzy-based Automatic Vehicle Control System in a Virtual Reality Environment", IEEE International Conference on Emerging Technologies, 10.1109/ICET.2007.4516340, pp 184 – 189.