Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 41 (2012) 700 – 709

Engineering Procedia

www.elsevier.com/locate/procedia

International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012)

ANFIS-Inverse-Controlled PUMA 560 Workspace Robot with Spherical Wrist

Yousif I. Al Mashhadany (MIEEE, IIE)

Electrical Engineering Department, Engineering College, University of Anbar, Baghdad, Iraq.
yousif_phd@hotmail.com.

Abstract

Cognitive architecture is used here to create a portfolio of movement in a spherical-wristed PUMA 560 robot. Adaptive Neuro-Fuzzy Inference System (ANFIS) was used, and the robot's postures and trajectories were executed in Virtual Reality (VR). The design aims to enable complex movements through use of the dual phases of ANFIS: one, solution by Inverse Kinematic Problem (IKP) with ANFIS identifier, for the end-effector's position and orientation as allowed by the 3-DOF wrist (the identifier offers high computation and accuracy of the IKP solution), two, implementing an inverse-ANFIS controller for all the robot's joint angles. VR implemented the robot's movements through the controller's use of forward dynamics. The IKP of determining a set of joint angles to achieve a given command for the manipulator's postures is addressed. The design's simulation was enabled by connecting the VR environment with Simulink/MATLAB Ver. 2012a. Forward and inverse kinematics implemented the manipulator's movements. Results validated the robot's range of possible postures and trajectories.

Keywords: Adaptive Neuro-Fuzzy Inference System (ANFIS), Virtual Reality (VR), PUMA 560.

Nomenclature

E_{IKP}	Error signal
ΔE_{IKP}	the rate of change of the error
$O_{J,I}$	the output of the *ith* node in layer-jth
x , y	the input to nodes
*ai ,bi ,*ci	the parameter set
w_i	the normalized firing
p_i ,q_i , r_i	the node's parameter set
p_{th}	target output vector
η_{ai}, η_{bi}, η_{ci}	the NN's learning rates
a_i, d_i, α_i, θ_i	the DH parameters
A_i	the transformation matrix
T	the sampling time.
ϕ	the learning step.
$u_{(j,k)}$	the input at k-th trial
$yd_{(j,k)}$	the desired output
$y_{(j,k)}$	the plant's actual output
$e_{(j,k)}$	the system's overall error

* Corresponding author. Tel.: +6-010-421-7541;
E-mail address: yousif_phd@hotmail.com

1877-7058 © 2012 Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
doi:10.1016/j.proeng.2012.07.232

1. Introduction

A robot has a series of rigid links joined by revolute or prismatic joints. A revolute joint rotates about, whereas a prismatic joint slides along, a motion axis. The location of any of the joints is usually relative to a neighbouring joint. Linking of successive joints is described by 4x4 homogeneous transformation matrices containing the manipulator's orientation and position. The manipulator's degree of freedom (DOF) is set by the number of transformation matrices; multiplying the matrices gives the manipulator's final position and orientation. Joint coordinates contain the control actions, whereas Cartesian coordinates specify the manipulator motions. Converting the manipulator's end-effector position and orientation from Cartesian space to joint space is an inverse kinematics problem and fundamental to calculating the desired joint angles in manipulator design and control [1-3].

For a manipulator with n-DOF, at any instant, the joint variables are denoted by ($\theta_i = \theta_i(t)$, i = 1;2;3;....N) and position variables (xj = x(t), j = 1;2;3;….M). End-effector position x(t) and joint angle θ(t) relate through forward kinematic equation: x(t) = *f* (θ(t)), with *f* being nonlinear, continuous, and differentiable. Given a desired end-effector position, however, finding values of the joint variables is an inverse-kinematics problem solvable by θ(t) = f(x(t)). The governing equations are nonlinear, uncertain, and time-varying, so the solution to (θ(t)) is not unique. Fig. 1 gives the forward, and the inverse, kinematics. For other solution techniques to inverse kinematics, read Kottege [2], who simulated a Six-axis Virtual Robot arm (SVR) program for a real robot manipulator [4,5].

Some works have discussed the problem of visually recognizing VR-animated hand postures. Exemplifying such animation is a 5-DOF robot arm. Intelligent control systems increase the effectiveness of control strategies and overcome limits of classical and adaptive controllers. They are expected to have in-built adaptation/learning and decision-making capabilities that handle uncertainties towards achieving the desired performance. To overcome traditional computing paradigm limits, researchers look for new computational approaches to partial modeling of neural system functionality and effective solving of real-world problems [6,7].

Novel computational models have emerged, collectively called soft computing. Incorporating intelligence into soft computing are ANN (artificial neural network), FL (fuzzy logic), ANFIS, GA (genetic algorithm), and knowledge-based expert systems. Intelligent self-learning or self-organizing control of expert systems, AI (artificial intelligence), FL, NN (neural networks), hybrid networks, etc. are the new performance-improving tools of industrial power-electronics-based drive systems. Control that combines intelligence and adaptation is the most promising in research into implementation and control of electrical drives [8].

FL, introduced by Zadeh in 1965 [9], processes data via partial, not crisp, set membership. As with NN, it can generate definite conclusions from vague, ambiguous, imprecise, and missing input information. A proper FIS is essential to correcting fuzzy membership functions (MFs) and rules but constructing it takes time and needs profound expertise. Converting domain knowledge into if-then fuzzy rules is another difficulty, even to domain experts [10].

ANFIS combines FL and NN in a hybrid intelligent system that enhances automatic learning and adaptation. Researchers have used it in various engineering systems to predict and model. Basic to neuro-adaptive learning techniques is a fuzzy modeling procedure that learns the data set and automatically computes the MF parameters that best allow the associated FIS to track the given input/output data [10].

The MF parameters are tuned by a combination of least-squares estimation and back-propagation algorithm for MF parameter estimation. MF parameters change through a learning process akin to NN's. They are adjusted by a gradient vector that measures how well a FIS models the input/output data of a given set of parameters. Once the gradient vector is obtained, any of several optimization routines can adjust the parameters to reduce the error between the actual and the desired outputs. The fuzzy system then learns from the data it is modeling. Adjustment of the MF boundaries does not need a human operator; this is an advantage over pure-fuzzy paradigm [11].

Continuous and repetitive tasks of production lines are tiring, and fatigue can cause accident/injury or long-term health problems. A robot manipulator arm can repeatedly carry huge and heavy parts from one production line to another, so is nowadays extensively used in factories to increase efficiency and handle repetitive/hazardous/delicate tasks (e.g., producing semiconductor wafers). Redundancy of a robot manipulator can be advantageous when the DOF of a given task is less than that of the manipulator. When a joint failure occurs, the robot manipulator is still able to perform its task reliably. Singular configuration of the robot manipulator can be avoided by controlling redundancy through robust inverse Jacobian singularity [12,13].

2. Adaptive neuro fuzzy identifier

Find This incorporates FL algorithm into a five-layer ANN structure (see Fig. 1) [14,15]. A tuning block adjusts the fourth-layer parameters to correct any angle-joint deviation. The inputs to a neuro-fuzzy identifier are the error between a robot IKP's analytical solution (E_{IKP}) and the rate of change of that error (ΔE_{IKP}):

$$\left.\begin{aligned} Input_1 &= E_{IKP} = y_{AIKP} - y_{ANFIS} \\ input_2 &= \Delta E_{IKP} = \frac{E_{IKP}(n) - E_{IKP}(n-1)}{T} \times 100\% \end{aligned}\right\} \tag{1}$$

The proposed identifier uses Sugeno fuzzy model with five-layer ANN structure. The first layer is the input layer, the second the fuzzification layer, the third and the fourth the fuzzy-rule-evaluation layers, and the fifth the defuzzification layer. Fig. 2 is a 2-input first-order Sugeno fuzzy model with two rules. Every node i in layer-1 is an adaptive node with the function:

$$\left.\begin{aligned} O_{I,i} &= \mu_{Ai}(x) \qquad for \quad i = 1,2 \quad or \\ O_{I,i} &= \mu_{Bi-2}(y) \qquad for \quad i = 3,4 \end{aligned}\right\} \tag{2}$$

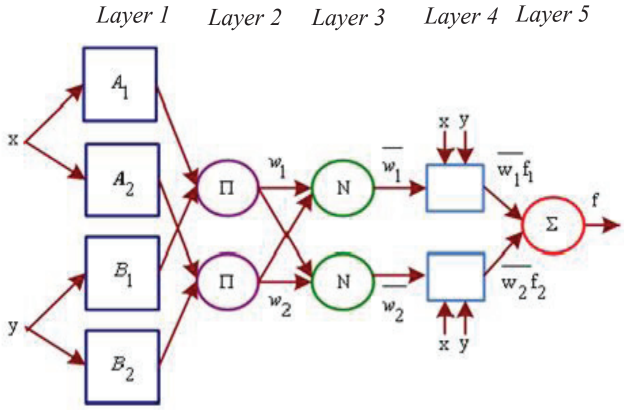


Fig. 1: Architecture of 2-input Sugeno-fuzzy ANFIS identifier [4]

$O_{1,1}$ denotes the output of the *ith* node in layer-1, with x (or y) the input to nodes i and Ai *(or Bi-2)* and is a linguistic label such as 'small' or 'large' associated with the node. The MF for A can be any appropriate parameterized MF. The proposed scheme uses as MF the generalized bell function as given by Eqn. (3) [16-18]:

$$\mu_A(x) = \left(1/(1 + \left| \frac{x - c_i}{a_i} \right|^{2bi}) \right) \tag{3}$$

with $\{ai\ ,bi\ ,ci\}$ being the parameter set. As the values of the parameters change, various forms of bell-shaped MFs can be obtained for fuzzy set A. Parameters in this layer are referred to as premise parameters. Every node in layer-2 is a fixed node labeled Π, with the output the product of all the incoming signals.

$$O_{2,i} = w_i = \mu_{Ai}(x)\mu_{Bi}(y), \qquad i = 1,2 \tag{4}$$

Each node output represents the firing strength of a rule. Every node in layer-3 is a fixed node labeled N. The outputs of this layer are normalized firing strengths as given by Eqn. (5):

$$O_{3,i} = \overline{w_i} = \frac{w_i}{w_1 + w_2}, \qquad i = 1,2 \tag{5}$$

Every *i*-node in layer-4 is an adaptive node with a function as that given by Eqn. (6).

$$O_{4,i} = \overline{w}_i f_i = \overline{w_i}(p_i x + q_i y + r_i) \tag{6}$$

w_i is the normalized firing strength from layer-3 and $\{p_i, q_i, r_i\}$ the node's parameter set. Parameters in this layer are referred to as consequent parameters. Layer-5 is the single node layer with a fixed node and labeled Σ; it sums up all the incoming signals to produce the overall output.

$$O_{5,i} = \sum \overline{w}_i f_i = \frac{\sum i w_i f_i}{\sum i w_i} \tag{7}$$

The hybrid approach converges much faster as it reduces the search-space dimension of the original pure back-propagation. In hybrid learning, for back propagation, the objective function to be minimized is defined by Eqn. (8) [19, 20].

$$E_p = \sum_{m=1}^{l} (T_{m,p} - O_{m,p})^2 \tag{8}$$

with $T_{m,p}$ being the m_{th} component of the p_{th} target output vector and O_{mp} the m_{th} component of the actual output vector produced by presenting the p_{th} input vector. Eqn. (9) gives the overall error measure [21]..

$$E = \sum_{p}^{P} = 1^{E_P} \tag{9}$$

The learning rules are:

$$\left.\begin{aligned} a_i(n+1) &= a_i(n) - \eta_{ai}\left(\partial E / \partial a_i\right) \\ b_i(n+1) &= b_i(n) - \eta_{bi}\left(\partial E / \partial b_i\right) \\ c_i(n+1) &= c_i(n) - \eta_{ci}\left(\partial E / \partial c_i\right) \end{aligned}\right\} \tag{10}$$

3. Case study

The spherical-wristed PUMA 560 robot exemplifies application of both the analytical solution and the ANFIS technique (see Fig. 2) [22].

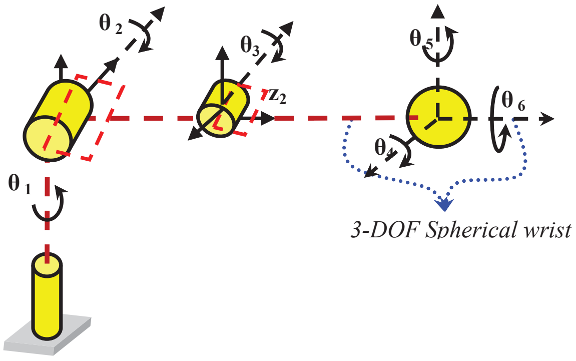


Fig. 2: The spherical-wristed PUMA 560 robot

An offset in the manipulator's shoulder joint slightly complicates both the forward and the inverse kinematics [7]. Coordinate frames for the joints are first established through DH convention (Table 1 lists the DH parameters). Computation of matrices Ai is straightforward as below [5].

Table 1. DH parameters for the PUMA 560 robot (*≡Joint varaiable)

Link	d_i	a_i	α_i	θ_i
1	d_1	0	π/2	θ^*
2	0	a_2	0	θ^*
3	0	a_3	0	θ^*
4	0	0	-π/2	θ^*
5	0	0	π/2	θ^*
6	d_6	0	0	θ^*

$$A_1 = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}; A_2 = \begin{bmatrix} c_2 & 0 & s_2 & a_2 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

$$A_3 = \begin{bmatrix} c_3 & 0 & -s_3 & a_3 \\ s_3 & 0 & c_3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; A_4 = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

$$A_5 = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; A_6 = \begin{bmatrix} c_6 & 0 & -s_6 & 0 \\ s_6 & 0 & c_6 & 0 \\ 0 & 1 & 0 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

The forward kinematics as represented by $A_6^0\ is.\quad A_6^0 = A_1 \ldots . A_6$

$$A_6^0 = \begin{bmatrix} & & & \\ & R_{3\times3} & & t_{1\times3} \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

o is the manipulator's end-effector at the point with coordinates, whereas R=(r_{ij}) is the orientation of the end-effector. Coordinates of the wrist center o_c are:

$$o_c^o = o - d_6 R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{15}$$

R is the orientation of the frame $o_6x_6y_6z_6$ relative to the base. If o_x,o_y,o_z denotes the components of the end-effector position (o), x_c,y_c,z_c denotes the components of the wrist center o_c^o , giving Eqn. (16):

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} o_x - d_6 r_{13} \\ o_y - d_6 r_{23} \\ o_z - d_6 r_{33} \end{bmatrix} \tag{16}$$

The inverse kinematics is achieved by a closed solution of the above equation. General solutions for the rotation angles are:

$$\left.\begin{aligned}
&\theta_1 = a\tan 2(x_c, y_c)\\
&\theta_3 = a\tan(D, \pm\sqrt{1-D^2})\\
&where: D = \frac{x_c^2 + y_c^2 + (z_c - d_1)^2 - a_2^2 - a_3^2}{2a_2a_3}\\
&\theta_2 = a\tan 2(\sqrt{x_c^2 + y_c^2}, z_c - a\tan(a_2 + a_3c_3 + a_3s_3)\\
&\theta_4 = a\tan 2(c_1c_{23}r_{13} + s_1c_{23}r_{23} + s_{23}r_{33},\\
&\qquad\qquad -c_1s_{23}r_{13} - s_1s_{23}r_{23} + c_{23}r_{33})\\
&\theta_5 = a\tan 2(s_1r_{13} - c_1r_{23}, \pm\sqrt{1-(s_1r_{13} - c_1r_{23})^2})\\
&\theta_6 = a\tan 2(-s_1r_{11} + c_1r_{21}, s_1r_{12} - c_1r_{22})\\
&where:\\
&c_i = \cos(\theta_i), and \quad s_i = \sin(\theta_i)
\end{aligned}\right\} \tag{17}$$

4. Modelling the design in VR

Virtual Reality Modeling Language (VRML) design requirements are contained in *finite processing allocations, autonomy*, *consistent self-registration,* and *calculability* [2]; they are necessary considerations to the designer, whose imagination of the object and available information determines his VRML design. VR design can be standard configuration (sphere, cone, cylinder, etc.) or free design (through selection of the indexed set face button to get many configurations whose points can be freely rearranged). Every real-form design is thus free design [23].

Free design starts with building the parts one by one and checking their shape against the real manipulator part, which cannot be simulated in VR when the standard shape in the VR library had been used. Next, all parts of the design must be connected to produce the final object (whose point of origin must, incidentally, be limited). This is done by first setting the shape (i.e., the base) and then connecting the next shape (joint two) in the "children" button. The procedure is repeated with the other parts. Fig. 3 is one such design in VR, i.e., the spherical-wristed elbow manipulator.

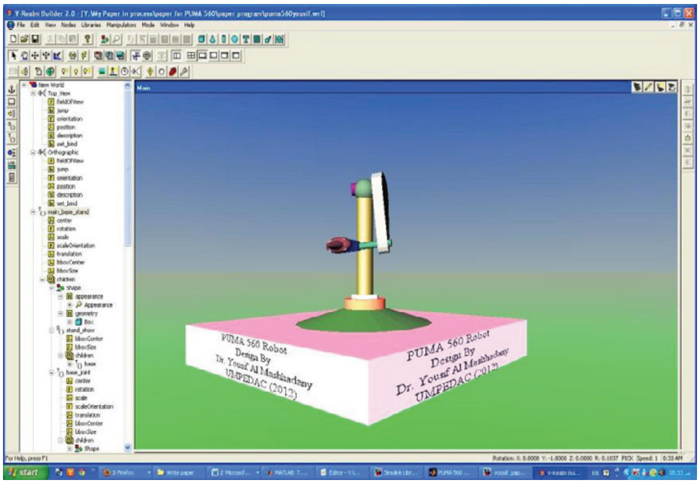


Fig. 3: The PUMA 560 robot in VR

5. The Inverse ANFIS controller design

The inverse-ANFIS control system has mainly the ANFIS inverse network acting as system controller and placed in series with the plant. Fig. 4 shows the training obtaining the plant's inverse ANFIS model. The input-output data set used reflects the plant's input-output characteristics.

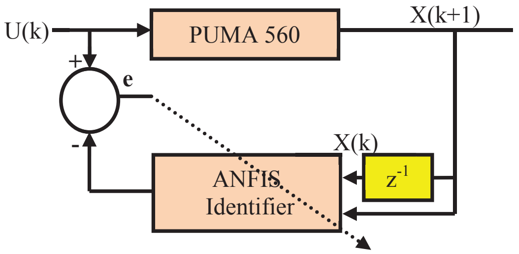


Fig. 4: Training of the inverse ANFIS model

The training data set is

$$(X^N(k), X(k+1)^N; U^N) \tag{18}$$

With N being the number of training data. To minimize the error cost function E, the ANFIS network can be trained by least-square estimation:

$$E = \sum_{i=1}^{N} (U_m - O_m^i)^2 \tag{19}$$

U_m is the control input signal and $O_m{}^i$ the corresponding actual output of the ANFIS identifier of the m-th component. The iterative learning tuner is designed to improve the tracking performance of inverse-ANFIS control which repeats the desired task over a finite interval. The system's inputs, outputs, and states are propagated along two sets, one describing the variables in terms of time history, the other reflecting the learning iteration. Fig. 5 is a basic block diagram of the proposed structure.

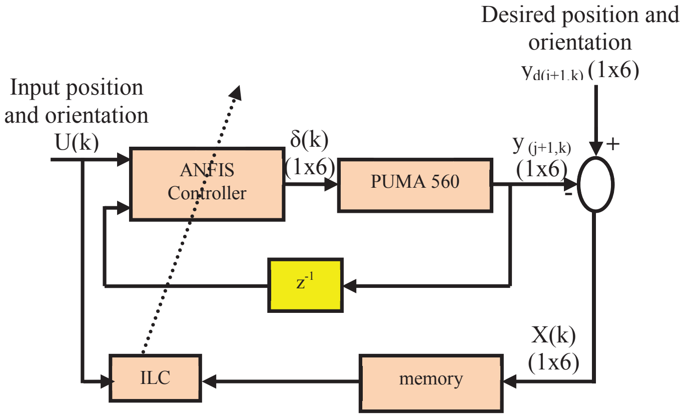


Fig. 5: The proposed controller

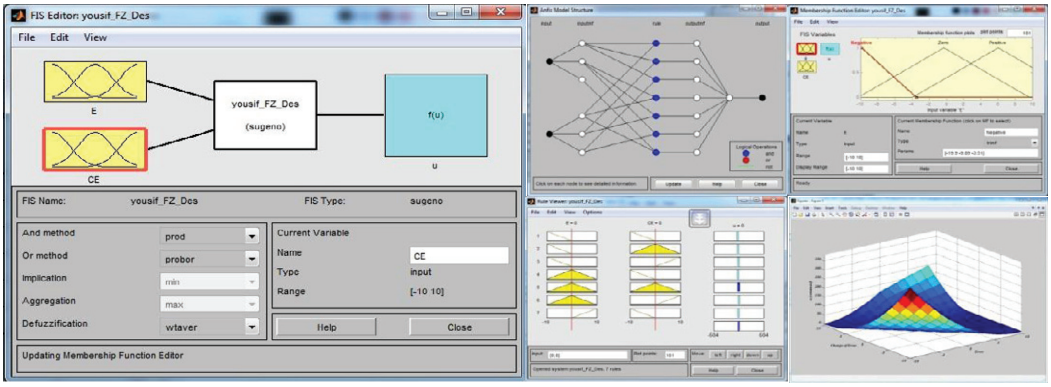
The learning law adjusts the control input continuously:

$$u_{(j,k+1)} = u_{(j,k)} + \phi e_{(j,k)} \quad \& \quad e_{(j,k)} = yd_{(j,k)} + y_{(j,k)} \tag{20}$$

$$y_{(j,k)} \rightarrow yd_{(j,k)} \;\; as \;\; k \rightarrow \infty \tag{21}$$

At each trial, the inverse-ANFIS controller is updated according to the controlled plant's actual output. This manages the model's error and uncertainties through minimized system error. ILC improves the control system iteratively; higher performance and precision. Assumptions made to ensure the proposed method's stability and convergence include repeatability of the desired output signal (*j*, *k*) *yd* over a finite time interval, system stability through the inverse-ANFIS control before introduction of the iterative learning, and bounded control input signal (*j*, *k*) value *u*.

The simulation used Sugeno fuzzy model for the inverse-ANFIS network so the consequent part of the fuzzy rules is a linear equation. The inputs were Triangular membership function (TRIMF); seven MFs to each input. Initialization of the prime parameters of the ANFIS network was such that the centers of the MFs were equally spaced throughout the entire range of the input variable, with the initial value of the consequent parameter set assumed zero. P-type learning law with 0.15 learning step was used. The collected input-output data of the coupled tank trained the ANFIS to do the inverse of the tank system. The training data included the target point's position and orientation, and the level corresponded with that of the delayed signal. Fig. 5 is the trained model as the inverse controller. Fig. 6 gives the inverse-ANFIS controller design with surface error and FL rules.



FIS Name: yousif_FZ_Des
FIS Type: sugeno

And method: prod
Or method: probor
Implication: min
Aggregation: max
Defuzzification: wtaver

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

Updating Membership Function Editor

Fig. 6: The inverse-ANFIS controller for the PUMA 560 robot

Fig. 7 compares the analytical solution by IKP and the solution by ANFIS identifier. Figs. 7 and 8 are the NN identification for inverse kinematics calculation with maximum normalized values for the angles of the joints. This work finds the IKP solution by using the LRNNs as forward direction, post-collection of the neuro-identifier database for use by the NN (of dimensions 6-19-6 and with LMBP learning algorithm).

Fig. 9 is the implementation of various movements of the PUMA 560 robot as controlled by the inverse-ANFIS controller. The results show use of the inverse-ANFIS to be effective for 6-DOF. The IKP is first calculated, then the forward kinematics, to produce the commands for the VR movement.

The method's accuracy depends on the envelope range and the accuracy of the identification process. Highly accurate data is needed for calculation of the IKP when the data is used in forward NN. Fig. 7 compares the results; the angles do not differ (the same values were used in the identification process). An accurate identifier in the ANFIS controller will reduce the end-effector's straying from target point. The identification values must range widely. Values of the end-effector must be in the manipulator's envelope of movements.

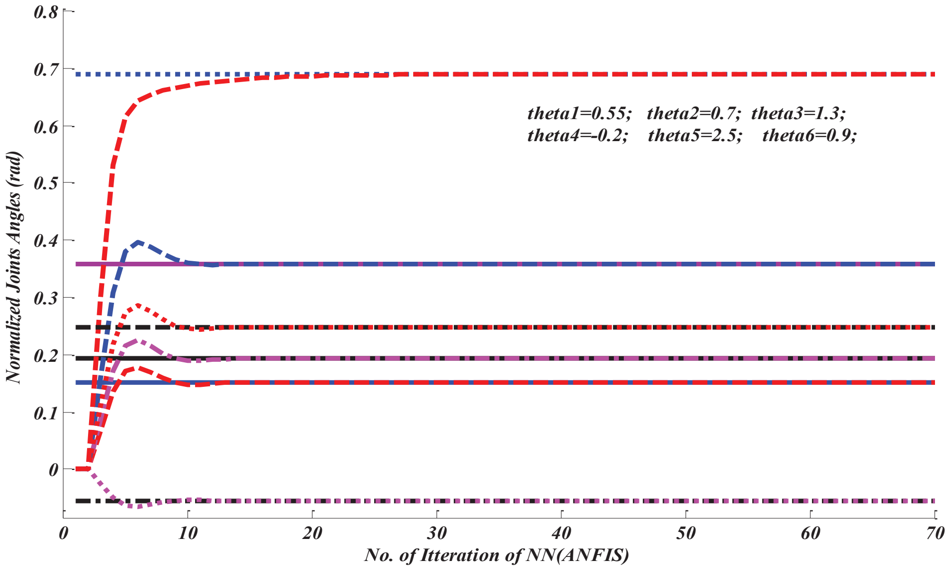


Fig. 7: The IKP analytical solution vs. the ANFIS-identifier solution

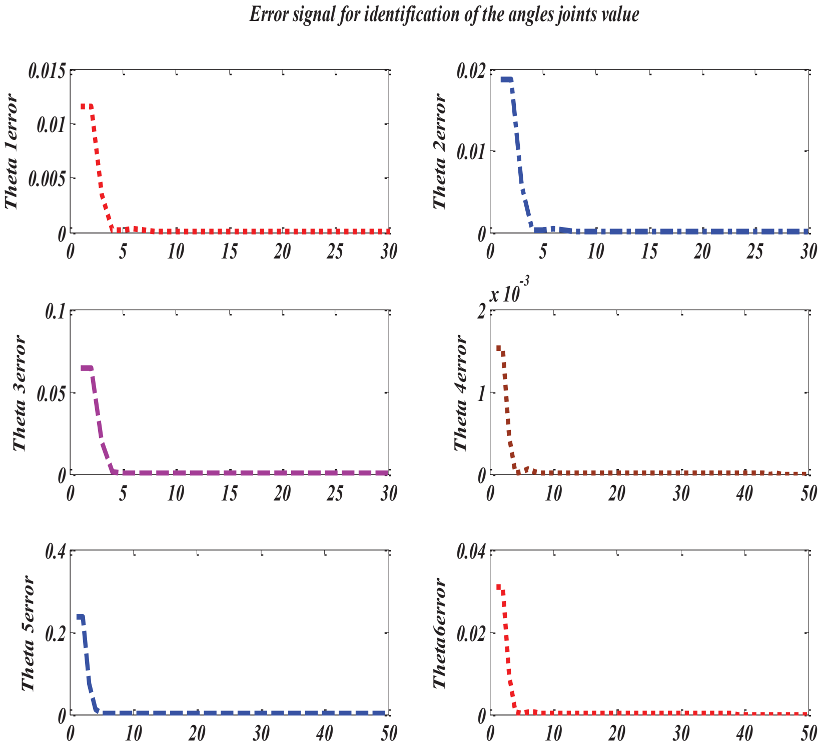


Fig. 8: The values of error for the joint angles identified by ANFIS

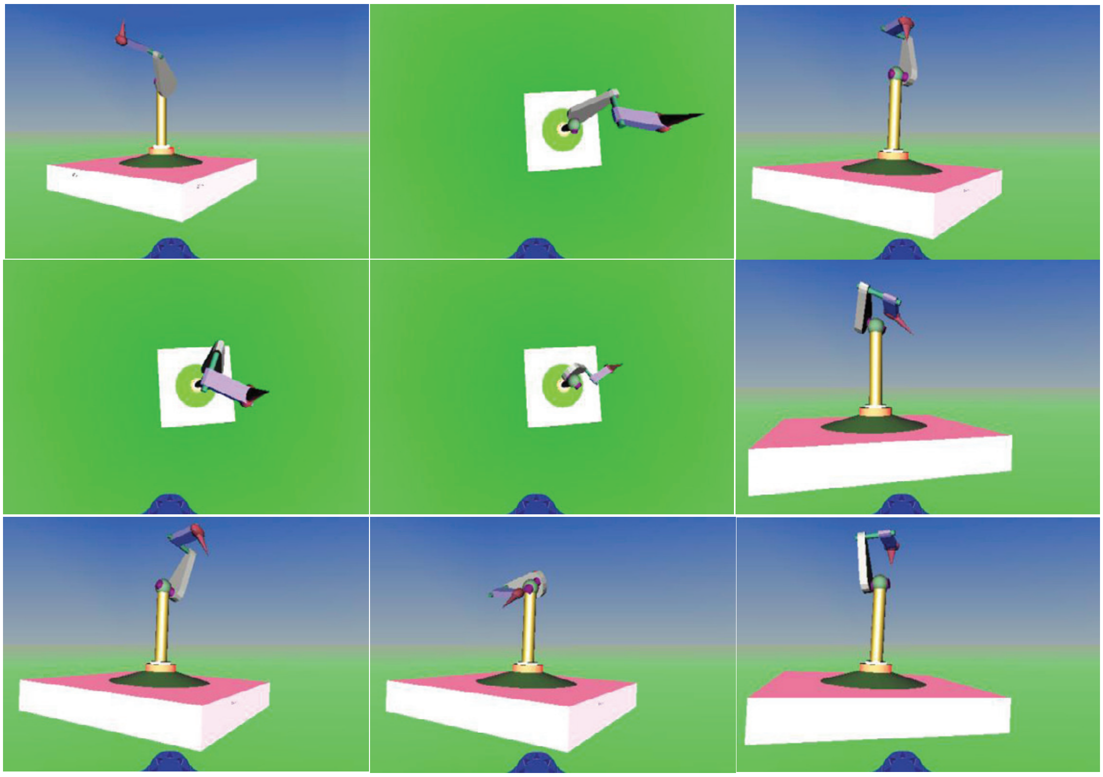


Fig. 9: Various VR postures of the PUMA 560 robot with the inverse-ANFIS controller

6. Conclusions

ANFIS has been shown as able to identify and solve IKP highly accurately and simply. NN can identify any-size envelope for the manipulator's movement, and with shorter computation than is afforded by closed-form analytical solution. NN calculation is sensitive to sampling band identification. This can be overcome by training the network for every point and by many tries, for optimal results. Any additional movement constraints to the IKP may be easily resolved by the proposed NN approach rather than by analytical solution.

Increased identification accuracy increases controller facility. For control of the joint angles, the inverse-ANFIS controller uses the same amount of value sets, each iteration. VR implementation of the manipulator's movements effectively realized various postures and trajectories of the PUMA 560 robot through commands produced from connecting the VR environment with MATLAB/Simulink. The spherical-wristed PUMA 560 robot can benefit many industrial applications. A future work may fully implement the VR-simulated robot in a physical model.

References

[1] Alavandar S., Nigam M.J., 2008. Inverse Kinematics Solution of 3DOF Planar Robot using ANFIS, Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844. Vol. III, Suppl. issue: Proceedings of ICCCC 2008, p. 150-155

[2] Kottege D., 2003.“ Virtual Robot Arm Control Model”, Ph.D thesis, University of Colombo, Srilanka.

[3] Izadbakhsh A., 209. “Closed-Form Dynamic Model of PUMA 560 Robot Arm”, IEEE Proceedings of the 4th International Conference on Autonomous Robots and Agents, Feb 10-12, p. 675-680

[4] Chen X., Xiong C., Xiong Y., 2009. “Kinematics Modeling and Simulation of a Passive Four Bar Linkage”, IEEE International Conference on Mechatronics and Automation, Changchun, China, p. 4139-4144.

[5] Spong M. W., Hutchinson S., Vidyasagar M., 2004. Robot Dynamics and Control, Second Edition.

[6] Carey R., Bell G., Marrin, C., 2008. “ISO/IEC 14772-1:1997 Virtual Reality Modeling Language (VRML97),” At ttp://www.vrml.org.

[7] Leottau L., Melgarejo M., 2011. “An Embedded Type-2 Fuzzy Controller for a Mobile Robot Application”, chapter 18, Recent Advances in Mobile Robotics, p.365-384

[8] Mahmoud M. S., Saif A. A., 2012. Robust Quantized Approach to Fuzzy Networked Control Systems, Intelligent Control and Automation journal, Vol.3, No.1, p. 59-70.

[9] Ibrahim A.M., 2004. “Fuzzy Logic for embedded system application “, ISBN 0-7506-7699-X, copyright by Elsevier science.

[10] Piltan F., Sulaiman N., Ferdosali P. , Rashidi M. & Tajpeikar Z., 2011. Adaptive MIMO Fuzzy Compensate Fuzzy Sliding Mode Algorithm: Applied to Second Order Nonlinear System, International Journal of Engineering (IJE), Volume (5) : Issue (5), p: 380-398

[11] Mazhari S. A., Kumar S., 2008. Heuristic Search Algorithms for Tuning PUMA 560 Fuzzy PID Controller, International Journal of Electrical and Computer Engineering, Vol, 3:4, p: 286-277

[12] Benitez A., Huitzil I., Ramos C., Medina A, 2011. “ A 3D simulation environment for kinematic task of the PUMA 560 robot”, 21st IEEE International Conference on Electrical, Communications, and Computers, Cholula, Puebla, Mexico. pages 260-265.

[13] Corke P. I., 2012. “Robotics Toolbox for ATLAB”, http://petercorke.com/Robotics_Toolbox.html, the last visit at 13/03/2012.

[14] Ibrahim A. M., 2004. Fuzzy Logic for embedded system application, ISBN 0-7506-7699-X, copyright by Elsevier Science.

[15] Subramaniam M., Manju A., Nigamc J, 2011. A Study of parameter tuning of weighted fuzzy rulebas using genetic algorthim for trajectory control of PUMA 560, International Journal of Engineering Science and Technology (IJEST), ISSN : 0975-5462 Vol. 3 No. 9, p, 7016-7028

[16] 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, p. 573-578.

[17] Aghajarian M., Kiani K. 2011. “Inverse Kinematics Solution of PUMA 560 Robot Arm Using ANFIS, IEEE The 8th International Conference on Ubiquitous Robots and Ambient Intelligence, p 574-578

[18] Omar F. L.,Samsul 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, p. 349-354

[19] 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,p. 1783-1795

[20] 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, p. 582-586

[21] 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, p. 165-189

[22] 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, p. 90-97

[23] Patton J. L., Dawe G ., Scharver C., 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