Design, Analysis, Simulation, and Virtual Reality Verified Intelligent Controller for Industrial Application SCARA Robot

Yousif Al Mashhadany (SMIEEE, MIIE)
Electrical Engineering Department, Engineering College,
University of Anbar, Baghdad, Iraq
Email:yousif.almashhadany@uoanbar.edu.iq

ABSTRACT

Robot manipulators are designed to execute required movements. Their controller design is equally important. Selective Compliant Assembly Robot Arm (SCARA) has four degrees of freedom (DOFs), with three (shoulder, elbow, wrist) controlled by servo motors and one by pneumatics. Presented here is the development of a complete mathematical model of an industrial-application SCARA robot including the servomotor dynamics and simulation of the dynamics, also the analytical inverse kinematic problem (IKP) and the forward kinematic solution by D-H parameters. The DC servomotor driving each of the robot-arm joint is studied and modeled. The robot arm is built for trajectories of drilling, manufacture, assembly, etc. It is realized by a 3D virtual reality (VR) model, which builds and receives commands through a MATLAB/Simulink link for the design's simulation on MATLAB Version R2012a. The control scheme is Adaptive Neuro Fuzzy Inference Strategy (ANFIS). A neural network with fuzzy logic controller (FLC) selects the proper rule base through back propagation algorithm. The integrated approach improves system performance, cost-effectiveness, efficiency, dynamism, and controller reliability. The method is effective, and the response (settling) is fast. The SCARA here is VR-verified. Real-time application is possible through interface cards.

Keywords-SCARA robot, mathematical modeling, kinematic solutions

1. INTRODUCTION

SCARA manipulators frequently perform tasks such as defect removal, pick-and-place, brushing, hole pegging, circuit board assembly, and mechanical assembly, all which require accurate tracking and high-speed maneuvering of the end-effector. Some tasks involve large payloads (10kg-20kg) and high speeds. Conventional high-speed robots cannot handle large payloads, whereas robots handling large payloads cannot reach high speeds.

Vibration, too, can be an issue. Servo hydraulics make high speeds with high payloads possible. High speeds or torques usually can be obtained only from large electric motors, which have high inertia, limiting the performance of the robot. Parallel kinematic mechanisms with stationary actuators often produce the high speeds. They are very accurate and very stiff [1]. Some can reach accelerations of up to 785 ms−2 (almost 80 times the gravitational acceleration), but unusable with large payloads. Their operating envelope is limited, and they do not suit planar operations. To overcome the problem of inertia, belt drives isolate the drive system components including the actuator and the power transmission components, from the moving parts. This, however, introduces backlash and friction to the system. Some studies propose options such as the wire-driven FALCON (FAst Load CONveyance) robot. 3-DOF M.I.T. direct-drive manipulator was proposed by Toumi and Kuo. Its tracking speed is 3m/s, and it can accelerate to up to 3.8 times the gravitational acceleration (accurate to between 0.05mm and 0.1mm). There is yet to be any study on a hydraulic solution to the high-speed high-payload problem [2 – 5].

Planar operations extensively use SCARA manipulators. Conventional commercial SCARA manipulators are driven by brushless DC motors, which can produce very high torques but also add to system inertia. Overcoming the problem are revolute joints coupled by gears or belt drives, however introducing the problems of delay, slip, and friction. At very high speeds, large-payload handling in electric-motor-driven SCARA becomes impossible. Studies on servo-hydraulics systems have been on high-torque/force-related applications only. A few have used servo-hydraulic actuators to drive serial manipulators. Some used linear actuators to drive revolute joints, varying the torque throughout the operating envelope, limiting performance. There is insufficient research on rotary actuators. Bilodeau, studied rotary hydraulic actuators for a dexterous humanoid robot. Heintze, and Hera, considered rotary actuators for high-torque applications. None so far have used servo hydraulics to produce high speeds in serial manipulators [6-10].

This work developed with D-H formulation, the kinematic equations of the SCARA with robot dynamics and the actuators-dc servomotors for each joint. Actuator characteristics; dc servo motors were studied in detail. It develops through Virtual Reality Modeling Language (VRML) a 4-axis SCARA system for handling. The structure to be built depends on the principles of solid-body modeling with VR technology. Simulation on MATLAB/Simulink software will reinforce the results obtained by SD program. The results of both will be presented and discussed. The paper is organized as follows: Section 2 introduces robotics and robot kinematics, Section 3 presents the robot's inverse kinematics, Section 4, the dynamics, Sections 5 and 6, the robot's actuators and transmission elements, Section 7 a review of ANFIS, Section 8 the VRML design of the model, Section 9 the ANFIS controller, and Section 10 the simulation, results, and conclusions.

2. ROBOT KINEMATICS

Table 1 defines the Denavit-Hartenberg (D-H) parameters specifying the SCARA robot.

Table.1 D-H parameters of the robot

i	*Θi*	*d_i*	*a_i*	*α_i*
1	*θ_1*	*0*	*L_1*	*0*
2	*θ_2*	*0*	*L_2*	*0*
3	*0*	*d_3*	*0*	*0*
4	*θ_4*	*d_4*	*0*	*0*

By using (D-H) convention [11], the transformation matrices result in:

$$T_1^0 = A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & L_1c_1 \\ s_1 & c_1 & 0 & L \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (1)$$

$$T_2^1 = A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & L_2c_2 \\ s_2 & c_2 & 0 & L_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2)$$

$$T_3^2 = A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3)$$

$$T_4^3 = A_4 = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & -d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (4)$$

After the multiplication and use of addition matrices, one gets the total transformation matrix:

$$T_4^0 = \begin{bmatrix} c_{124} & -s_{124} & 0 & L_2c_1 + L_1c_1 \\ s_{124} & c_{124} & 0 & L_2s_{12} + L_1s_1 \\ 0 & 0 & 1 & -d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (5)$$

3. INVERSE KINEMATICS OF THE ROBOT

3.1. Inverse Solution for Position:

Desired location of the SCARA robot

$$T_H^R = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (6)$$

The final equation representing the robot is [12-16]:

$$T_H^R = A_1A_2A_3A_4 = T_4^0 \qquad (7)$$

To solve for the angle θ4, both sides of equation (7) are successively pre-multiplied with $A_3^{-1}A_2^{-1}A_1^{-1}$ matrices, such that:

$$A_3^{-1}A_2^{-1}A_1^{-1}T_H^R = A_4 \qquad (8)$$

The left side of the equation (8) ($A_3^{-1}A_2^{-1}A_1^{-1}T_H^R$) is:

(9)

(9)

$$\left.\begin{array}{l}\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&-d_3\\0&0&0&1\end{bmatrix}\times\begin{bmatrix}c_2&s_2&0&-L_2\\-s_2&c_2&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\times\begin{bmatrix}c_1&-s_1&0&-L_1\\-s_1&c_1&0&0\\0&0&1&-d_1\\0&0&0&1\end{bmatrix}\times\begin{bmatrix}n_x&o_x&a_x&p_x\\n_y&o_y&a_y&p_y\\n_z&o_z&a_z&p_z\\0&0&0&1\end{bmatrix}=\\ \begin{bmatrix}n_xc_{12}+n_ys_{12}&o_xc_{12}+o_ys_{12}&a_xc_{12}+a_ys_{12}&p_xc_{12}+p_ys_{12}-L_1c_2-L_2\\-n_xs_{12}+n_yc_{12}&o_xs_{12}+o_yc_{12}&-a_xs_{12}+a_yc_{12}&-p_xs_{12}+p_yc_{12}-L_1s_2\\n_z&o_z&a_z&p_z+d_3\\0&0&0&1\end{bmatrix}\end{array}\right\} \tag{9}$$

From 1,4 and 2,4 elements of the equations (5) and (6):

$$p_x = L_1c_1 + L_2c_{12} \tag{10}$$

$$p_y = L_1s_1 + L_2s_{12} \tag{11}$$

From equation 10 and equation 11,

$$c_2 = \frac{1}{2L_1L_2}\left(p_x^2 + p_y^2 - L_1^2 - L_2^2\right) \tag{12}$$

$$s_2 = \pm\sqrt{1-c_2^2} \tag{13}$$

$$\theta_2 = \tan^{-1}\frac{s_2}{c_2} \tag{14}$$

Rearranging equation (10) and equation (11) yields:

$$p_x = (L_1 + L_2c_2)c_1 - L_2s_2s_1 \tag{15}$$

$$p_y = L_2s_2c_1 + (L_1 + L_2c_2)s_1 \tag{16}$$

Solving equations (15) and (16) by kramer's rule:

$$\Delta = \begin{bmatrix}L_1+L_2c_2 & -L_2s_2\\ L_2s_2 & L_1+L_2c_2\end{bmatrix} = (L_1+L_2c_2)^2 + (L_2c_2)^2 \tag{17}$$

$$\Delta s_1 = \begin{bmatrix}L_1+L_2c_2 & -p_x\\ L_2s_2 & -p_y\end{bmatrix} = (L_1+L_2c_2)p_y - (L_2s_2)p_x \tag{18}$$

$$\Delta c_1 = \begin{bmatrix}p_x & -L_2s_2\\ p_y & L_1+L_2c_2\end{bmatrix} = (L_1+L_2c_2)p_x + (L_2s_2)p_y \tag{19}$$

$$s_1 = \frac{\Delta s_1}{\Delta} = \frac{(\mathrm{L}_1+L_2c_2)p_y - L_2s_2p_x}{(\mathrm{L}_1+L_2c_2)^2 + (L_2s_2)^2} = \frac{(\mathrm{L}_1+L_2c_2)p_y - L_2s_2p_x}{p_x^2+p_y^2} \tag{20}$$

$$c_1 = \frac{\Delta c_1}{\Delta} = \frac{(\mathrm{L}_1+L_2c_2)p_x - L_2s_2p_y}{(\mathrm{L}_1+L_2c_2)^2 + (L_2s_2)^2} = \frac{(\mathrm{L}_1+L_2c_2)p_x - L_2s_2p_y}{p_x^2+p_y^2} \tag{21}$$

$$\theta_1 = \tan^{-1}\frac{s_1}{c_1} = \tan^{-1}\frac{(\mathrm{L}_1+L_2c_2)p_y - L_2s_2p_x}{(\mathrm{L}_1+L_2c_2)p_x - L_2s_2p_y} \tag{22}$$

From 4,4 elements of the equation (5) and (6):

$$d_3 = -p_z - d_4 \tag{23}$$

We have

$$\theta_3 = 0 \tag{24}$$

From 1,1 and 2,1 elements of the equation (4) and (9):

$$c_4 = n_xc_{12} + n_ys_{12} \tag{25}$$

$$s_4 = -n_xs_{12} + n_yc_{12} \tag{26}$$

$$\theta_4 = \tan^{-1}\frac{-n_x\sin(\theta_1+\theta_2) + n_y\cos(\theta_1+\theta_2)}{n_x\cos(\theta_1+\theta_2) + n_y\sin(\theta_1+\theta_2)} \tag{27}$$

3.2. Inverse Solution for Velocity

From equation (11) and equation (12)

$$\dot{p}_x = -L_1s_1\dot{\theta}_1 - L_2s_{12}\left(\dot{\theta}_1 + \dot{\theta}_2\right) \tag{28}$$

$$\dot{p}_y = +L_1c_1\dot{\theta}_1 - L_2c_{12}\left(\dot{\theta}_1 + \dot{\theta}_2\right) \tag{29}$$

So,

$$\dot{p}_x = -(L_1s_1 + L_2)\dot{\theta}_1 - L_2s_{12}\dot{\theta}_2 \tag{30}$$

$$\dot{p}_y = -(L_1s_1 + L_2c_{12})\dot{\theta}_1 + L_2c_{12}\dot{\theta}_2 \tag{31}$$

Using Kramer's rule to solve equation (31) and equation (32)

$$\dot{\theta}_1 = \frac{\dot{p}_xc_{12} + \dot{p}_ys_{12}}{L_1s_2} \tag{32}$$

$$\dot{\theta}_2 = \frac{-\dot{p}_y(L_1s_1 + L_2s_{12}) - -\dot{p}_x(L_1c_1 + L_2c_{12})}{L_1L_2s_2} \tag{33}$$

Translational velocity:

$$\dot{d}_3 = -\dot{p}_z \qquad (34)$$

By differentiating the equation (27):

$$c_4 d\theta_4 = -\left[dn_x s_{12}\left(d\theta_1 + d\theta_2\right)\right] + d_{ny} c_{12} - n_y s_{12}\left(d\theta_1 + d\theta_2\right) \qquad (35)$$

So,

$$d\theta_4 = -\frac{d\theta_1 + d\theta_2}{c_4}\left(n_x c_{12} + n_y s_{12}\right) - \frac{s_{12}}{c_4} dn_x + \frac{c_{12}}{c_4} dn_y \qquad (36)$$

And finally:

$$\dot{\theta}_4 = \frac{d\theta}{dt} = \frac{c_{12}\dot{n} - s_{12}\dot{n}_x - \left(n_x c_{12} + n_y s_{12}\right)\dot{\theta}_{12}}{c_4} \qquad (37)$$

3.3 Inverse Solution for Acceleration

$$\ddot{\theta}_1 = \frac{-\left(\dot{p}_x s_{12} + \dot{p}_y c_{12}\right)\dot{\theta}_{12} + \left(\ddot{p}_x c_{12} + \ddot{p}_y s_{12}\right) - L_1 c_2 \dot{\theta}_1 \dot{\theta}_2}{L_1 s_1} \qquad (38)$$

$$\ddot{\theta}_2 = \frac{\left[\left(\ddot{p}_x s_1 - \ddot{p}_x c_1\right)L_1 + \left(\ddot{p}_x s_{12} - \ddot{p}_x c_{12}\right)L_2 + \left(\dot{p}_y c_1 - \dot{p}_x s_1\right)L_1\dot{\theta}_1 + \left(\dot{p}_y c_{12} + \dot{p}_x s_{12}\right)L_2\dot{\theta}_{12} + L_1 L_2 \dot{\theta}_2^2\right]}{L_1 L_2 s_2} \qquad (39)$$

$$\ddot{d}_3 = -\ddot{p}_x \qquad (40)$$

$$\ddot{\theta}_4 = \frac{\ddot{n}_y c_{12} - \ddot{n}_x s_{12} - \left(2\dot{n}_y s_{12} + 2\dot{n}_x c_{12}\right)\dot{\theta}_{12} - \left(n_y c_{12} - n_x s_{12}\right)\dot{\theta}_{12}^2 - \left(n_x c_{12} + n_y s_{12}\right)\ddot{\theta}_{12} + s_4 \dot{\theta}_4^2}{c_4} \qquad (41)$$

4. Dynamics of the Robot

For SCARA robot figure 3, torques exerted on the robot joints are [17]:

$$T_1 = b_{11}\ddot{\theta}_1 - b_{12}\ddot{\theta}_1 - b_{13}d_3 - b_{14}\dot{\theta}_1\dot{\theta}_2 + b_{15}\dot{\theta}_2^2 \qquad (42)$$

$$T_2 = b_{21}\ddot{\theta}_1 + b_{22}\ddot{\theta}_2 + b_{23}\ddot{d}_3 + b_{24}\dot{\theta}_2^2 \qquad (43)$$

$$T_3 = -b_{31}\ddot{\theta}_1 + b_{32}\ddot{\theta}_2 + b_{33}\ddot{d}_3 - b_{34} \qquad (44)$$

Where:

$$b_{11} = r_1^2 m_1 + j_1 + g_{r1}^2 j_{m1} + \left(L_1^2 + r_2^2 + 2L_1 r_2 c_2\right)m_2 + L_1^2 m_{m2} + j_2 + j_{m2} + \left(L_1^2 + L_2^2 + 2L_1 L_2 c_2\right)\left(m_3 + m_{m3}\right) + j_3 + j_{m3}$$

$$b_{12} = \left(r_2^2 + L_1 r_2 c_2\right)m_2 + j_2 + g_{r2} j_{m2} + \left(L_1^2 + L_1 L_2 c_2\right)\left(m_3 + m_{m3}\right) + j_3 + j_{m2}$$

$$b_{13} = g_{r3} j_{m3} \;;\quad b_{14} = 2L_1 s_2\left[m_2 r_2 + \left(m_3 + m_{m3}\right)L_2\right]$$

$$b_{15} = L_1 s_2\left[m_2 r_2 + \left(m_3 + m_{m3}\right)L_2\right] \;;$$

$$b_{15} = L_1 s_2\left[m_2 r_2 + \left(m_3 + m_{m3}\right)L_2\right]$$

$$b_{21} = m_2\left(r_2^2 + L_1 r_2 c_2\right) + j_2 + g_{r2} j_{m2} + \left(L_1^2 + L_1 L_2 c_2\right)\left(m_3 + m_{m3}\right) + j_3 + j_{m3}$$

$$b_{22} = r_2^2 m_2 + j_2 + g_{r2}^2 j_{m2} + L_2^2\left(m_3 + m_{m3}\right) + j_3 + j_{m3}$$

$$b_{23} = g_{r3} j_{m3} \;;$$

$$b_{24} = L_1 s_2\left[m_2 r_2\left(m_3 + m_{m3}\right)L_2\right] b_{31} = b_{32} = g_{r3} j_{m3} \;;$$

$$b_{33} = m_3 + g_{r3} j_{m3} \;;\quad b_{34} = m_3 g$$

5. ACTUATOR EQUATIONS (ACTUATOR MODELING)

Actuator is the device moving a robot. The device types include pistons (pneumatic, hydraulic) and motors (DC, stepper). Most robots use DC motors, so a detailed model for this actuator type will be derived for use throughout this research. To control the voltage supply, the motor drive will use PWM control, which use microcontrollers for these advantages [18-20]:

1) their size and lightness

2) the fewer inputs and outputs, and

3) remote operation using

4) changeable with minimum loss of the armature voltage

Equations governing permanent-magnet operation with separate-wound or shunt-wound DC motors (see Figure 4) when the flux is constant are [12]:

$$V_a = Ri_a + e_a + L\frac{di_a}{dt} \qquad (45)$$

$$e_a = k_e \phi \omega_m = k_\omega \omega_m \qquad (46)$$

$$T = k_e \phi i_a = k_T i_\omega \qquad (47)$$

$$T = T_L + j_m \frac{d\omega_m}{dt} + b\omega_m \qquad (48)$$

6. TRANSMISSION EQUATIONS

The many types of transmission elements in a robot convey mechanical power from the actuator to the load. Gear is the most common transmission element. A common robotic revolute-joint transmission element is harmonic drive [13] - compact in-line parallel shafts with very high transmission ratio. The torque transmitted to the motor shaft (T) can be calculated from Figure 5 as:

$$T = T_L / g_r \eta \Rightarrow T \frac{\omega_m}{\omega_L} \eta = T \frac{\theta_m}{\theta_L} \eta = T_L \qquad (49)$$

Also, the transmitted inertia is

$$J = j_m + J_L / g_r^2 \qquad (50)$$

For the third joint, the translational variable (linear velocity) can be derived as:

$$\omega_m = \frac{D}{2} = \dot{d}_3 \qquad (51)$$

7. ADAPTIVE NEURO FUZZY INFERENCE SCHEME (ANFIS)

This section reviews the ANFIS concepts used in controlling a plant's system parameters. Neural network (NN) concepts began as an attempt at transliterating human thought. NNs have had successful application in speech recognition, image analysis, and adaptive control, constructing software agents or autonomous robots and controlling machines. ANNs are a family of intelligent algorithms that can be used for time-series prediction, classification, and control and identification. NNs have the ability to train with induction motor's various parameters. As a non-linear function, they can identify extremely nonlinear system parameters with high accuracy. Use of NNs to identify and control nonlinear dynamic systems has been proposed because they can approximate a wide range of non-linear functions to any desired accuracy. They also have extremely fast parallel computation and high fault tolerance characteristics. NNs in power electronics and AC drives have been investigated, including in speed estimation. NN technique estimates speed fairly well and is robust to parameter variation. For better performance, an NN speed estimator should be trained sufficiently with various patterns. Fuzzy logic (FL) can control various parameters of real-time systems. Its combination with NNs gives significant results. NNs can learn from data but understanding that knowledge is difficult, especially in giving meaning to each neuron and each weight [21-23].

Fuzzy-rule models are easier to understand because they use linguistic terms and the IF-THEN rule structure, but FL cannot learn on its own. Its learning and identification adopt techniques from fields such as statistics and system identification. NNs can learn; merging it with FL is only natural. Merging NN's learning with FL's knowledge representation is a hybrid technique called 'neuro fuzzy networks' [23]. ANFIS design starts with a pre-structured system; DOF for learning is thus limited, i.e., the MF of the input variables and the output variables contain more information that NN has to drive from sampled data sets. Knowledge on the systems being designed can be used straightaway. Part of the system can be excluded from training. ANFIS is thus more effective. The rules are linguistic, easing analysis and interpretation of intermediate results. Rules can be modified during training and optimization can be manual. ANFIS also supports TS-based systems. To start ANFIS learning, a training data set containing the desired input/output data pairs of the target systems to be modelled it. The design parameters to any ANFIS controller are viz., number of data pairs, training data set, checking data sets, fuzzy inference systems for training, number of epochs to be chosen to start the training, learning results to be verified after mentioning the step size [24-28].

In this work, the following is the general ANFIS control structure for the control of any plant. The structure contains the same components as does FIS, except for the NN block. The network structure is a set of units (and connections) arranged into five connected network layers, viz., l1 to l5.

Layer 1: This layer consists of input variables (MFs), viz., inputs 1 and 2. Triangular or bell-shaped MF can be used here. This layer just supplies the input values i x to the next layer, where I = 1 to n.

Layer 2: This layer (the membership layer) checks the weight of each MF. It receives the input values i x from the 1st layer and acts as MF to represent the fuzzy sets of the input variables. It also computes the membership values specifying the degree to which the input value i x belongs to the fuzzy set, inputting the next layer.

Layer 3: Called the rule layer, each node (neuron) in it pre-condition matches the fuzzy rules, i.e., computing the activation level of each rule, the number of layers equalling the number of fuzzy rules. Each node calculates the normalized weights.

Layer 4: Called the defuzzification layer, it provides the output values y resulting from rule inference. Connections between layers l3 and l4 are weighted by the fuzzy singletons that represent another set of parameters for the neuro fuzzy network.

Layer 5: Called the output layer, it sums up all the inputs from layer 4 and transforms fuzzy-classified results into crisps (binary).

The ANFIS structure is tuned automatically by least-square estimation and back propagation algorithm. The algorithm shown above is used in the next section to develop the ANFIS controller to control the various parameters of the induction motor. ANFIS's flexibility enables its use in many control applications.

8. THE MODEL in VIRTUAL REALITY

The requirements for design in VRML are explained in *finite processing allocations*, *autonomy*, *consistent self-registration* and *calculability*. Design in VRML depends on the designer's information and his imagination of the object. VR design choices are standard configurations (sphere, cone, cylinder, etc.) and free-form (the indexed face set button is selected, to get many configurations with points that can be rearranged). Every real-form design is considered free-form designing, which starts with building parts one by one and checking the shape against a related, real manipulator part. That robot part cannot be simulated in VR when the standard shape from the VR library is used, where they are not uniform shape. The design uses the indexed face set in VR. The next design step is connecting all the parts to produce the object and to limit the object's point of origin. This job was made by setting the first shape (e.g. the base) and then connecting the next shape (joint two) in the "children" button; the same procedure is repeated with other parts. Fig. 1 is the design, in full VR, of a SCARA robot with vacuum handling wrist [29-31].

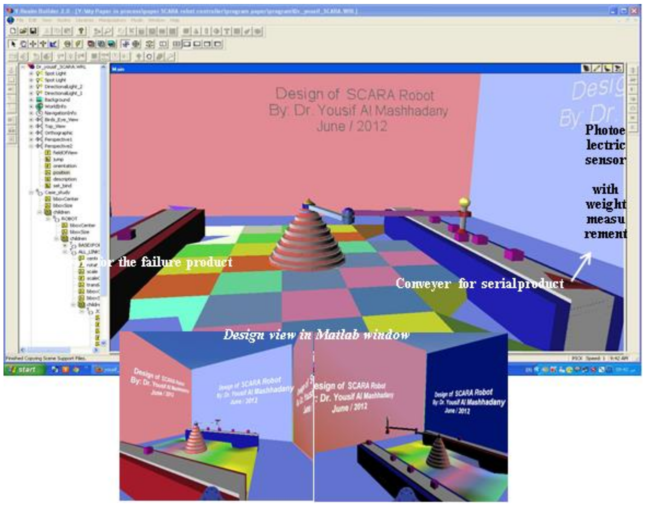

Design of SCARA Robot
By: Dr. Yousif Al Mashhadany
June / 2012
Photoelectric sensor with weight measurement
the failure product
Conveyer for serial product
Design view in Matlab window

Fig. 1. A SCARA robot left arm, in VR view through Matlab

9. ANFIS CONTROLLER DESIGN

A controller is a device controlling each and every system operation through the decisions it makes. A control system brings stability during disturbances, safeguarding from equipment damages. It may be hardware-based, software-based, or both. This section develops an ANFIS control scheme (see Fig. 2) for SCARA robot parameters such as speed, accuracy, envelop and high performance. Fuzzy logic is one of the successful applications of fuzzy set in which the variables are linguistic rather than the numeric variables. Linguistic variables, defined as variables whose values are sentences in a natural language (such as large or small), may be represented by fuzzy sets. Fuzzy set is an extension of a 'crisp' set where an element belongs to either a set (full membership) or to none (no membership). Fuzzy sets allow partial membership, i.e., membership to more than one set.

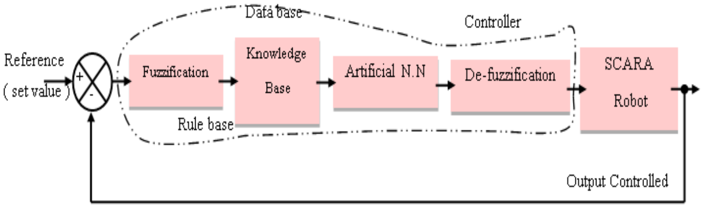


Fig. 2: The ANFIS control scheme for speed control of the IM

A fuzzy set A of a universe of discourse X is represented by a collection of ordered pairs of generic element and its membership function μ : X → [0 1], which associates a number μA(x) : X → [0 1] to each element x of X. An FLC uses a set of control rules called fuzzy rules, expressed among the linguistic variables as conditional statements. The basic structure of this work's ANFIS controller comprises 4 blocks: fuzzification, knowledge base, neural network, and de-fuzzification; each will be explained briefly in the next paragraphs. The inputs to the ANFIS controller, i.e., the error and the change in error, are modelled through Eqn. (51) as

Table 2
The rule base for speed control

ΔE\E	**NB**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NM	NM	NS	ZE	PS
NS	NB	NM	NS	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PS	PM	PB
PM	NS	ZE	PS	PM	PM	PB	PB
PB	ZE	PS	PM	PB	PB	PB	PB

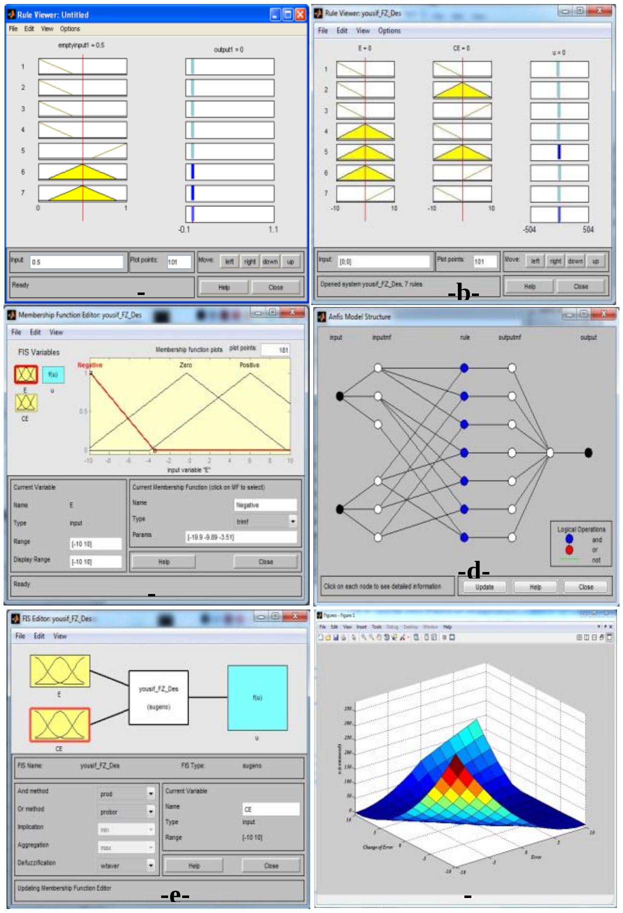
$$\left.\begin{aligned} e(k) &= \omega_{ref} - \omega_r \\ \Delta e(k) &= e(k) - e(k-1) \end{aligned}\right\} \qquad (51)$$

Where w_{ref} is the reference speed, w_r the actual rotor speed, e(k) the error, and Δe(k) the change in error. The fuzzification unit converts the crisp data into linguistic variables, given as inputs to the rule-based block. The set of 49 rules are based on past knowledge/experiences in the rule-based block, which connects to the NN block. Back propagation algorithm trains the NN to select the proper set of rule base. In developing the control signal, the training is very important for selection of the proper rule base. Selection, and then firing of the proper rules, generates the control signal needed for optimal outputs. The output of the NN unit is given as input to the de-fuzzification unit and the linguistic variables are re-converted into numeric data, as crisps. In fuzzification, the crisp variables, the speed error, and the change in error are converted into fuzzy variables or linguistics variables. The fuzzification maps the 2 input variables to linguistic labels of the fuzzy sets. The fuzzy coordinated controller uses the linguistic labels. Each fuzzy label has an associated MF. Triangular MF was used here (see Fig. 3-a,b). The inputs are fuzzified through the fuzzy sets and given as input to the ANFIS controller. Table II lists the rule base for selection of the proper rules through back propagation algorithm.

$$y = \left(\sum_{i=1}^{R} \mu^i a_1^i x_1 \right) + \Lambda + \sum_{i=1}^{R} \mu^i a_q^i x_q \Bigg/ \sum_{i=1}^{R} \mu^i \qquad (52)$$

The control decisions are based on the fuzzified variables in Table 2. The inference uses a set of rules in determining the output decisions. As there are 2 input variables and 7 fuzzified variables, the controller has a set of 49 rules for the ANFIS controller. From the 49 rules [Fig. 4-c,d], the proper rules are selected by NN training helped by back-propagation algorithm, before the selected rules are fired. Further, it has to be converted into numerical output, i.e., they have to be de-fuzzified. This process is defuzzification, which produces a quantifiable result in FL.

Defuzzification transforms fuzzy set information into numeric data information. Methods of defuzzification include centre of gravity, centre of singleton, maximum, marginal properties of centroid, etc. This work used the centre-of-gravity method. The output of the defuzzification unit generates the control commands that are given as input (crisp input), through the inverter, to the plant. Any deviation in the controlled output is feedback and compared with the set value and the error signal generated, and given as input to the ANFIS controller, which restores the output to the normal value, maintaining system stability. Eqn. (52) gives the controlled output signal y, which is the controller's final output and is the weighted average of the proper rule-based outputs selected by the back-propagation algorithm.



Input: 0.5 Plot points: 101

Ready

Input: [0;0] Plot points: 101

Opened system yousif_FZ_Des, 7 rules

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 [-19.9 -9.89 -3.51]

Ready

Click on each node to see detailed information

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. 3. Learning and analysis of the ANFIS controller

10.SIMULATION RESULTS & DISCUSSION

Fig. 4 is the Matlab Simulink R2012a VR model for the neuro-fuzzy controller tracking the SCARA trajectory. Starting off the simulations is the invoking of the 49-fuzzy-rule set from the Matlab command window; the fuzzy file where the rules are written with T-S control strategy incorporated is opened, then the fuzzy editor (FIS) dialogue box opens (see Fig. 3). The .fis file (yousif_FZ_Des) is imported through the command window from the source, and then opened (through file-open command) in the fuzzy editor dialog box. Figs. 4 a-f shows opening of the file activating the TS fuzzy-rule file. The data is exported to the workspace, and the simulations are run for e.g., 60s.

The fuzzy MF editor is next obtained from the menu bar, through view membership command (see Figs. 3 a-b). The rule-view command enables viewing of the TS-fuzzy rules written. It is the pictorial rule viewer for the 2 inputs and 1 output. Post preliminary operations, the VR model is called up through the interface block between it and Matlab. Fig. 4e shows the ANFIS editor opening in the command window. The workspace-data variables are loaded onto the ANFIS editor (see Figs. 3 a-f). The .fis file is next generated in the ANFIS editor through loading of the workspace data. Once the .fis file is generated, the ANFIS has to be trained properly through a proper algorithm with enough epochs. In rule-training, this work used back-propagation algorithm with an appropriate number of epochs. The 2 items are selected in the train window of the ANFIS editor, and the NN is trained for proper-rule-base selection. Next the trained data is exported to the workspace, through file-export command. Fig. 4f is the surface plot for error signal and the change in output error. The ANFIS controller designed trained the NN through the fuzzy-rule base, to select the proper and optimal rule. The hidden layers use 7x7 rules. Neuron-1 connects to 7 fuzzy rules, as does Neuron-2. The hidden layers have 49-49 neurons selecting the proper rule base. The 49 fuzzy rules are fired, de-fuzzified output obtained as output neuron. Through the value of the calculated command, the de-fuzzified output generates the firing pulse to be applied to the actuator of each motor joint. Post-simulation, the performance of the SCARA is evaluated through its movements, which are recordable in video or photograph and observable according to scope.

The control system with intelligent controller for the industrial-application SCARA, with the joints connected, in VR modeling (see Fig. 2), was simulated by using Matlab R2012a. The results prove the effectiveness of the proposed neuro-fuzzy controller.

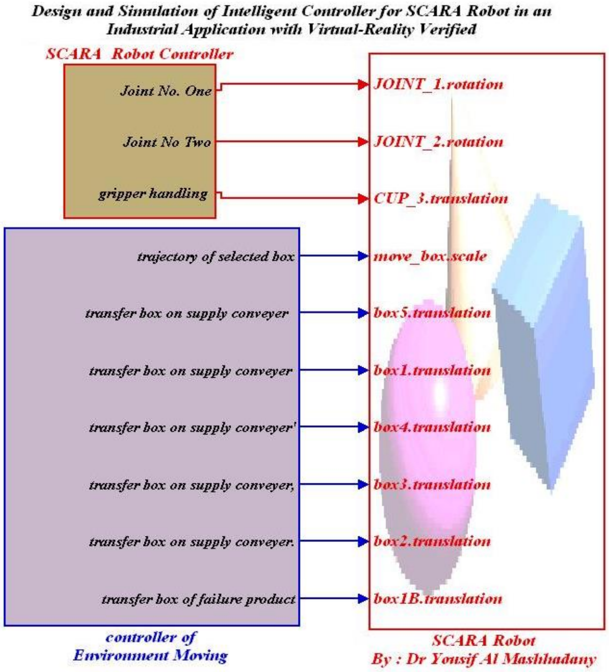
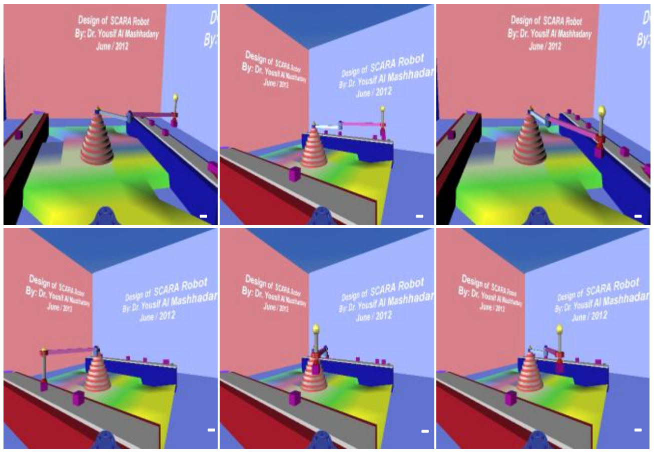

Design and Simulation of Intelligent Controller for SCARA Robot in an Industrial Application with Virtual-Reality Verified
SCARA Robot Controller
Joint No. One
Joint No Two
gripper handling
JOINT_1.rotation
JOINT_2.rotation
CUP_3.translation
trajectory of selected box
move_box.scale
transfer box on supply conveyer
box5.translation
transfer box on supply conveyer
box1.translation
transfer box on supply conveyer'
box4.translation
transfer box on supply conveyer,
box3.translation
transfer box on supply conveyer.
box2.translation
transfer box of failure product
box1B.translation
controller of Environment Moving
SCARA Robot
By : Dr Yousif Al Mashhadany

Fig. 4. Simulation of the Intelligent Controller for the Virtual-Reality-Verified Industrial-Application SCARA Robot

The highly accurate model of the system shows the drive speeding up with faster dynamism. The response characteristics curve of the proposed neuro-fuzzy controller shows, as compared with [3], [4], [5], faster settling and reaching of steady state. Figs. 5 a-f show the robot trajectory in a factory application of quality-control defect screening. Defective items are selected by an electro photo sensor with a weight measurement sensor that limits failure in weighing the pieces in the production line. Compared with other methods, the ANFIS control achieves system stability faster through the ANN training and use of the proper rule-base. The desired trajectory is accurately achieved through accurate positioning and orientation of the end-effector (see Figs. 5 a-f).



Figs. 5 (a-f): Movements of the SCARA in production-line screening

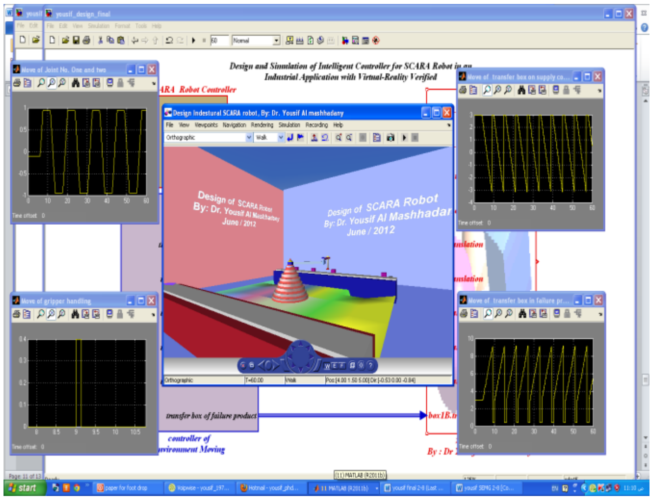

Design and Simulation of Intelligent Controller for SCARA Robot in an Industrial Application with Virtual-Reality Verified
Design indesturial SCARA robot, By: Dr. Yousif Al mashhadany
Design of SCARA Robot
By: Dr. Yousif Al Mashhadany
June / 2012
Orthographic
Walk
controller of Environment Moving
transfer box of failure product
box1B.t
start

Fig 6. Controlled signal for movement robot joints and environment

Fig. 6 shows the rotation signal of the SCARA joints and translation of the gripper movements through a sixty-second delivery-period simulation test. The SCARA very accurately detects defective items as signaled by the photoelectric sensor before transferring them to another conveyor, which returns them. The SCARA moves five defective pieces per minute.

11. CONCLUSIONS

The results verify successful mathematical modeling of the SCARA and its servo-actuator dynamics. Accurate equations of the forward and inverse kinematics will benefit investigation of the SCARA parameters. The SCARA Simulink model has been developed, as its ANFIS controller, which can be used in tasks as complex as, e.g., production-line screening. ANFIS has effective computation and works well with linear, optimization, and adaptive techniques. Compared with other control strategies, it operates much faster. The ANFIS controller developed here settles and stabilizes quickly and has very good dynamic response. The use of photoelectric sensor as sensing element in the control system increased effectiveness. This sensor has sufficient real specification for its practical implementation in the control system proposed.

ACKNOWLEDGEMENT

Special thanks are due to University of Anbar – Iraq / Renewable Energy Research Centre for supporting me with the work with Grant No. RERC-TP17.

REFERENCES

[1] M. Isaksson, T. Brogårdh, M. Watson, S. Nahavandi, P. Crothers, The Octahedral Hexarot A novel 6-DOF parallel manipulator, Journal of Mechanism and Machine Theory, Vol. 55, pp. 91–102, 2012

[2] S. Yamacli, H. Canbolat, Simulation of a SCARA robot with PD and learning controllers , Journal of Simulation Modelling Practice and Theory, Vol. 16, pp. 1477–1487, 2008

[3] F. Bravo, G. Carbone, J. Fortes, Collision free trajectory planning for hybrid manipulators, journal of mechatronics, http://dx.doi.org/10.1016/j.mechatronics.2012.05.001 , 2012.

[4] A. Djuric, R. AlSaidi,W. ElMaraghy , "Dynamics solution of n-DOF global machinery model, Journal of Robotics and Computer-Inegrated Manufacturing, Vol. 28, pp. 621– 630, 2012

[5] S. Toroghia , M. Gharibb, A. Ramezanic, K. Rahmdelb, Modeling and Robust Controller Design for an Industrial Boiler, 2011 2nd International Conference on Advances in Energy Engineering, Journal Energy Procedia, Vol. 14, pp. 1471-1477, 2011

[6] T.Benjanarasuth, N. Sowannee, N. Naksuk, "Two-Degree-of-Freedom Simple Servo Adaptive Control for SCARA Robot", International Conference on Control, Automation and Systems, Oct. 27-30, in KINTEX, Gyeonggi-do, Korea, pp. 489 – 484, 2010

[7] M.Liyanage, N. Krouglicof , R. Gosine, "Development and Testing of a Novel High Speed SCARA Type Manipulator For Robotic Applications", IEEE International Conference on Robotics and Automation Shanghai International Conference Center , May 9-13,Shanghai, China, pp. 3226 – 3242, 2011

[8] A. Nagchaudhuri, "Experience with Introducing Robotics Toolbox for MATLAB in a Senior Level Undergraduate Course" ASME International Mechanical Engineering Congress and Exposition November 13-19, Lake Buena Vista, Florida, USA, pp. 1-8, 2009

[9] C. Lee, K. Son, J. Lee, Integrated SCARA Robot Control System with OLP and Its Performance Evaluation, PR0001-3/97/0000-l125 SICE, pp.1125-1129, 1997.

[10] M. Short, K. Burn, A generic controller architecture for intelligent robotic systems, Journal of Robotics and Computer-Integrated Manufacturing, Vol. 27, No.2, pp. 292–305, 2011.

[11] M.Taylan Das, L. Canan Dulger, Mathematical modeling, simulation and experimental verification of a scara robot, Journal of Simulation Modelling Practice and Theory Vol. 13, No. 3,pp. 257–271,2005.

[12] H. Souley Ali, L. Boutat-Baddas, Y. Becis-Aubry and M. Darouach," H∞ control of a SCARA robot using polytopic LPV approach" 14th Mediterranean Conference on Control and Automation, MED '06,pp. 1 - 5,2006.

[13] J. Antonio, J. de Lope, M. Santos, A method to learn the inverse kinematics of multi-link robots by evolving neuro-controllers Journal of Neurocomputing, Vol. 72, No.13–15, pp. 2806–2814, 2009.

[14] M. Taylan Das, L. Canajournal on Dulger, Mathematical modeling, simulation and experimental verification of a scara robot, Journal of Simulation Modelling Practice and Theory, Vol.13, No.3, pp. 257–271, 2005.

[15] F.Bravoa, G. Carboneb, J.C. Fortes, Collision free trajectory planning for hybrid manipulators, Journal of Mechatronics, In Press, 2012.

[16] P. Cheng, K. Cheng, Evaluation of the dynamic performance variation of a serial manipulator after

eliminating the self-weight influence, journal of Mechatronics, Vol.21, No.6, pp. 993–1002, 2011.

[17] Z.Liang, S.Meng, D.Changkun," Accuracy Analysis of SCARA Industrial Robot Based on Screw Theory" IEEE International Conference on Computer Science and Automation Engineering (CSAE), Vol.3, pp.40-46, 2011.

[18] A.Visioli,G.Legnani, " On the Trajectory Tracking Control of Industrial SCARA Robot Manipulators" IEEE Transaction on Industrial Electronics, Vol. 49, No. 1, pp.224-232, 2002.

[19] M. Nkomo, M. Collier, "A Color-Sorting SCARA Robotic Arm", 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), pp. 763- 768, 20012.

[20] S.Takagi, N. Uchiyama, Robust Control System Design for SCARA Robots Using Adaptive Pole Placement, IEEE Transaction on Industrial Electronics, Vol. 52, No. 3, pp.915-921,2005.

[21] Sahin, Y., Ankarali, A.; Tinkir, M.," Neuro-Fuzzy trajectory control of a scara robot" IEEE 2nd International Conference on Computer and Automation Engineering (ICCAE), pp. 298- 302, 2010.

[22] R.Rodriguez, A. Cedeno , U.Costa , J. Solana, C. Caceres, E.Opisso , J.M. Tormos , J. Medina , E. J. Gomez, Inverse kinematics of a 6 DoF human upper limb using ANFIS and ANN for anticipatory actuation in ADL-based physical Neurore habilitation, Journal of Expert Systems with Applications ,Vol.39, No.10.

[23] S.Volker, Optmized SCARA kinematic description and examples, International Symposium on Robotics (ISR), 2010 41st and 2010 6th German Conference on Robotics (ROBOTIK), pp.1-5, 2010.

[24] S.Yamacli , H.Canbolat, Simulation of a SCARA robot with PD and learning controllers, Journal of Simulation Modeling Practice and Theory, Volume 16, Issue 9,pp. 1477–1487,2008.

[25] M. Isaksson, T. Brogardh, I. Lundberg, S.Nahavandi, "Improving the Kinematic Performance of the SCARA-Tau PKM" IEEE International Conference on Robotics and Automation, pp. 4683-4690,2010.

[26] S. Majidabad, A. Kalat, H. Shandiz" Neuro-Fuzzy-Discrete Sliding Mode Control of a Tree-Link SCARA Robot" 19th Iranian Conference on Electrical Engineering (ICEE),pp.1, 2011 .

[27] M. Joo, M. T. Lim, H. S. Lim, Real-time hybrid adaptive fuzzy control of a SCARA robot, Journal of Microprocessors and Microsystems, Vol.25, No.8, pp. 369–378, 2001.

[28] M. Plius, M. Yilmaz, U. Seven and K. Erbatur" Fuzzy Controller Scheduling for Robotic Manipulator Force Control, 12th IEEE International Workshop on Advanced Motion Control, pp.1-8, 2012.