

Robust Hybrid Intelligent Based Motion Control of Industrial Robot Manipulators for Unstructured Dynamic Environment

M. Dev Anand*, N. Ramasamy

Department of Mechanical Engineering, Noorul Islam Centre for Higher Education,
Kumaracoil - 629180, Tamil Nadu, India

*Corresponding author: E-Mail: anandpmt@hotmail.com

ABSTRACT

The intelligent robot system is credited with a commendable task of moving in an unstructured dynamic environment with a little period information without a chance of being collided. This interesting capability is apparently the subject of research for more scholars than researchers in robotics. This study investigates moving of a robotic arm in the presence of any obstacle. The robots that are capable of working in unpredictable and changing environment will be suitable and effectively used in wide range of applications. Moving a robot arm around obstacle needs a special skill which is akin to intuitive skill for human beings. Intuitive skill is translated in to instructions for a robotic arm is a Himalayan task. Neuro fuzzy systems are excellent in determining the input and output relationships using a simple data. The traditional fuzzy logic system is replaced with the back propagation neural network. A robotic arm could be trained to plan its movements averting collision with obstacle in its area. This is possible only with neuro fuzzy techniques since several of its controllers are trained using simple data received from human control of robotic arm and this study quantifies and compares their performance. The suggestions for possible avenues for future work are also explained.

KEY WORDS: Neuro-fuzzy, Robot Manipulator, Trajectories, Motion Control, Obstacle, Environment.

1. INTRODUCTION

Robots used in space exploration and construction should possess the quality of sensing their surrounding environment, regardless of objects present in work area. The process of translating the human skills of moving an obstacle to a goal, into instructions for a robot is tough job to perform. But the goal of translation can be achieved through artificial intelligence. In the area of artificial intelligence, Neural Network and fuzzy logic are two important results of research that can be effectively applied to everything from voice and image recognition to toaster and automobile transmissions. Neural network associated with learning capabilities, while neural network uses human skills and thinking abilities in machines. Both the things added a new dimension to many engineering fields of study regardless of its weakness. The training of neural network requires millions of iterative calculations may result in inadequate learning. On the other hand fuzzy logic acquires the knowledge from an expert who encodes his knowledge in the system through IF / THEN rules. It is easy to understand as it imitates the thinking of human beings. The problem arises when system have arises many inputs and outputs and it also makes the acquiring of rule base for large systems as difficult. Researches are made an idea to combine both the systems, as individual strengths and compensate their weakness. These combinations replace the rule base of a traditional fuzzy logic system with a back propagation network. The combination is called as neuro-fuzzy technique that provides an effective training to robot, to make its movements in a way without any obstacle collision in its locality.

Literature Survey: The method of artificial intelligence progressed in the order of neural networks, fuzzy logic and neuro-fuzzy technology. Among that neural networks and fuzzy logic proved their value in control systems. In recent days neuro-fuzzy technique also shows improvement in quality of intelligent control.

Robot Control: The robot control system should have the capability to execute the planned sequence of motions and forces in a correct way, in presence of unforeseen errors (Mckerrow, 1991). The in-accuracies in the model of robot, mechanical compliance in linkages, and limitation of precision in computation are the reason for errors. At robot level, the action is decomposed into a sequence of robot motions and forces in Cartesian space. It is decomposed as parallel joint motion and joints torques in case of joint level. The precise control is necessary for both Cartesian and joint space for position velocity, force and torque. The feedback and control laws helps to achieve this control, the feedback was deal by orchestrator and control laws was deal by robot controller. For the robot control, different approaches like open loop control, adaptive control, feed forward and feedback control are present. The feedback control attains an accurate control of process during execution of an action. It continuously measures the parameter to be controlled and compares it to a reference for error and action is modified according to control law to overcome error. Much research has been gone in the past in this area to achieve it through different methods. Among those object oriented programming is one of the approach. A robotic manipulator simulation package that is both platform and vendor independent (Gourdeau, 1997).

For the design of intelligence robot major approaches are taken. One focus attention on local sensor information and the other stored internal state (Gat, 1993). The first classical approach assumes that an intelligent robot needs a significant number of world models, including models of its perceptual environment, models of other

agents and models of itself. The second approach (reactive approach) deals robots as a collection of relatively independent behaviors that are handled mostly by sensor which requires minimal modeling. In our approach use reunite both classical and reactive approaches by integrating fuzzy and neural tools to facilitate the purpose of in a flexible and learn-as-needed manner.

Motion planning task is simple for humans but in case of robots it turns out to be quite difficult to achieve. It arises difficult computational questions especially with the problems associated with real time motion control, sensing and task planning. Different methodologies are available for motion control planning can be seen in literature. Among that three approaches: road maps, cell decomposition and potential field methods are used to categorize the variants. In the road map approach, the connectivity of robots free space in a network of one dimensional curve was captured, which we say R . The motion planning is reduced to connecting the initial and goal configurations to points in R and searching R for a path between these points. There are several methods available to compute different types of road maps. For example: visibility graphs, voronoi diagrams, freeway diagrams and silhouettes (Latombe, 1991). In cell decomposition approach the robot's free space was decomposed into simple regions called cells, whose path between two configuration can be generated. A non-directed graph called the connectivity graph is used to represent the adjacency relation between the cells and is constructed and searched to produce a sequence of cells called channel out of which a continuous free path can be computed.

In general, both the roadmaps and cell decomposition method capture global connectivity of the robots free space into a condensed graph, which is later on searched for a path. Potential field methods handle the robot as a particle under the impact of an artificial potential field U that represents structure in the free space. In free space potential function as described as sum of an attractive potential pulling the robot towards the goal configuration and often repulsive potential to move the robot away from the barrier. Originally the above method was developed as on-line collision avoidance approach, that applies in absence of robot's prior model of obstacles but has capability to sense it during motion (Hwang and Ahuja, 1992; Khatib, 1986). The dynamic motion planning problem comes into deal when obstacles are moving in workspace, which is difficult to solve by roadmaps, cell decomposition, or potential field methods.

The problem of robot to avoid obstacles, while moving in unknown terrain was examined by many researchers (Ahrikencheith and Seireg, 1994; Rao, 1989). Fuzzy methodologies are also used for investigation that uses fuzzy rules to plan robot's motion in unknown terrain, for quantifying a robot's reactive behavior. Different reactive behavior was also developed (Bourbakis, 1995; Tuscillo and Bourbakis, 1996; Terano, 1992). Array of sensors are used to give inputs regarding path widths, distance between robot and obstacle to left, front and right locations and also size of the obstacle. Based on inputs, the speed control unit, direction control of motor, a heading angle is commanded. The performance of robot is reasonably well in unstructured motion-planning – situation. The usage of expert systems and fuzzy logic to design and implement multi agent system for robotic-soccer control (Daniel, 2007), have capability to recognize games situation and deduce speed vector for every possible case. Low conceptual distance confirms easy transfer of expert knowledge to inference system. The experimental implementation of controlling the real robotic soccer match, as performance is not good enough. Necessary areas of improvement are in synchronization of agents and knowledge base.

Robotic soccer is good tool for development and testing of individual and social behavior of robots. It facilitates easy generation of various situations and compares the effects of their control mechanism by playing against each other. No special input element available, it uses only camera with image recognition. Part of the system is also main PC machine with higher levels of the control and with image recognition (Evin, 2005).

Hall (2007), describes the design development and simulation of real time controller for intelligent vision-guided robot. The use of a creative controller that can select its own tasks is demonstrated. The creative learning system is proposed and described as an adaptive critic learning system. It is easy for the system to obtain the global optima and escape local optima by learning the domain knowledge. It is a step toward generalizing the highest level of human learning imagination. The requirements of complex and unstructured environment can be met by expanding the dynamic database.

The ideas is to establish a large number of autonomous mobile robots that able to self-organize in order to collected the information in knocked urban sites and to dynamically maintain the communication links between rescuers. Robots engaged in rescue operations must be reactive and smart enough to deal with complex situations. Hybrid agents seem to be valuable architectures for controlling such robots (Noury and Serge, 2007). The evolution of robot resources and unplanned variation of surrounding are not taken into account in resulting control architecture.

The artificial neural network (ANN) technique is used to control the mobile robot movement in dynamically changing environments (environment with mobile obstacles). The proposed structure (Lids and Marcelo, 2007) is a backward neural one. The path generator is based on neural memorization of a path previously planned and facilitates robot movement, avoiding obstacles and position errors. It encourages the usage of ANN topology in hybrid path planning module and its use as a predictive controller. The sensors data case local planning that monitors the

environment at real time (Becker, 2001). The optimal global planning methods solve navigation problems in static environments, as it is difficult from time to time and presence of unknown obstacles that must be detected to avoid collisions.

The work is devoted to planning and control of a pattern formation of mobile robots when moving between goal points in a known and static environment. A feasible trajectory can be generated by a search coupled with a proposed smoothing technique, along with non-holonomic constraints of mobile robots taken into account. The work (Ngo, 2007) includes (i) fast and feasible path finding achieved by the modified A* search and vector approach coupled with the two proposed smoothing algorithms, that consider kinematic and dynamic constraints of mobile robots and (ii) the maintenance and changing of formations, in curvilinear co-ordinates to accomplish the required task while formation safety is concerned illustrative simulation for a three robot wedge was performed for two scenarios.

In most existing works of autonomous navigation, solution is achieved by separating the planning and control into two sequential stages. This may have some consequences (e.g.) algorithms convergence. The problem of motion planning and control can be taken into account with Artificial Potential Field (APF) along with robot motion planning. The APF was first proposed by Khatib (1986). He put forward the force involving artificial repulsion from the surface, should be non-negative continuous and differentiable. However, the potential field introduced, reveals local minima other than the goal position of robot. Volve (1990), developed new elliptical potential functions called a super quadratic artificial potential function that does not generate local minima in physical space. They have shown that super quadratic potential fields can be constructed only for simple shapes like square or triangular Figures. The endowments of Koditsche are worth mentioning because an analytical potential field in the configuration space was introduced without local minima. But topology is limited to obstacles that must be ball or star shaped, otherwise no solution obtained. Conaolly, 1994 contributes most successful methods which concerns robot motion planning with potential fields. Harmonics functions are used simultaneously to build a potential field in the configuration space without local minima.

Several autonomous systems have been developed using rule-based methods to control the motion of robot manipulators. Tsoukalas (1997), presented neuro fuzzy methodology for a robot to navigate it in dynamic environments. Ding and Li (1999), used fuzzy logic system to solve obstacle problem for a redundant manipulator. The structured and unstructured uncertainties are not considered by these systems, which is limited to stationary by these obstacles. It can also be dealt with a robust fuzzy obstacle avoidance scheme.

Figure.1, was proposed to accomplish this task, which illustrates system architecture for a described combination of neural networks and for the described combination of neural networks and fuzzy logic. The rule base of a fuzzy system is replaced with a trainable neural network. Controller to approximate imperfectly known and / or un-modeled to robot dynamic parameters, complex input-output relationships can be achieved which cannot be easily specified by IF / THEN rules. The ability of generating and find minimizing weights can be realized with fuzzification and defuzzification stages increasing improvement in training time of a neural network. The modified Elman neural network was preferred to a multilayer procedure for the selection of optimum multilayer neural network architecture for a given problem. The term is used in the control law to control the weights of modified Elman neural network from growing without bounds. The closed loop stability of the system is rigorously tested.

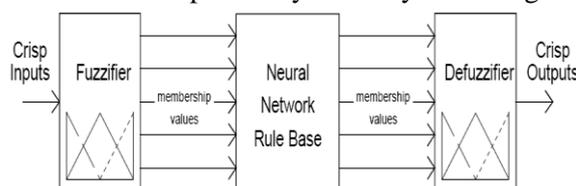


Figure.1. A Fuzzy System with Neural Network Rule Base

2. METHODS AND METERIALS

Robot Model: The equation 1 shown below is the dynamic non-linear equation of ‘n’ degree of freedom robot manipulator in continuous time in closed form.

$$M(q(\ddot{q})) + C(q, \dot{q}) + F(\dot{q}) + G(q) = \tau + \tau_d \tag{1}$$

Where q and τ are the (n × 1) vectors of joint variables and driving joint torques. M is (n × n) symmetric positive definite inertia matrix, C is the vector of coriolis and centrifugal forces, G is the vector of gravitational forces, F is the vector of frictional torques and τ_d is a quantity including un-modeled disturbances un-modeled dynamics.

Robust Intelligent Motion Control: From the Figure.2, it is noticed that Intelligent Dynamic Motion Planning (IDMP) algorithm is the base for the collision avoidance. Here h (q) is the transformation from joint co-ordinates to Cartesian co-ordinates. The neuro fuzzy decision for any robot motion is observed as its brain, where d_a>d_{min} is

the end-effector. It may cause the complicated problems as there is no unique mapping available to convert a coordinate in Cartesian space to a specific pint in joint space. In fact for a given end-effectors desired position, there exist an infinite solutions for possible joint angles which facilitates the selection of arm movements based on power efficiency or a straight line trajectory, or even on the observed behavior of an expert’s control of robot arm.

Assumptions are made regarding the goal to lie within the possible reach of end-effector and obstacle to be lie outside the path of link, L1. But the limitation is that the vertical position must be greater than are equal to zero for all possible locations of the end-effectors. Thus the problem is summarized as follows:

Given a robotic arm with:

- starting position (x_s, y_s) ,
- an obstacle position, (x_o, y_o) ,
- an obstacle size (x_{os}) ,
- path size (x_p) ,
- and a desired position, or goal, (x_g, y_g) ;

The following points are considered for robot arm motion control:

- the end-effector reaches the goal,
- the robot arm does not touch the obstacle,
- and the calculations can be performed in real-time with current hardware,
- proper direction and correct speed,
- minimum distance to the goal;

Neuro Fuzzy Solution: Many research works are made in determining the trajectory of a robotic arm that adds new dimension to the problem, more degrees of freedom and more non-linear equations. Consider an obstacle in the robot working area and the problem becomes a challenge of optimizing differential equations, iterative numerical solutions and “gues and shoot” methods. The drawback of previous solutions is computer a resource is needed to solve equations or optimize a performance index. Researchers accredited that “minutes” are necessary to plan one trajectory on some of the world’s fastest computers. It is burdensome thing to determine the equations to be solved. The way of approaching needed thing is easy to apply and work in real time. It is easy to predict input-output relationship from the sample data, using the neuro fuzzy system. The neuro fuzzy technique replaces fuzzy logic rule because of fuzzy logic system with a multilayer back propagation neural network. The Figure 4 demonstrates the proposed neuro fuzzy system used in robot motion control problems.

The benefit of the system is that inborn ability of a child to move an arm around an obstacle to reach a desired goal can be performed by putting instructions into a neat, fuzzy logic. Here IF/THEN rule base is not easy. So need arises for the neural network to learn the rules. The fuzzifiery and defuzzifiery provides an interface between expert’s control of a robot arm and the neural network.

The objective of robot arm control is to initiate a robust and intelligent controller for robot dynamics as shown in equation (1) to achieve a certain motion to reach the prescribed goal. The robot’s trajectory is the unknown parameter and not calculated in advance. A suitable approximation based controller is used by the robust computed torque type control.

The MATLAB program having robot SIMULINK tool box was utilized to program our generic fuzzy system. The training set for our neuro-fuzzy system is the recorded data. The model of robot arm and neuro fuzzy system was combined to simulate the robot arm under neuro fuzzy contol.

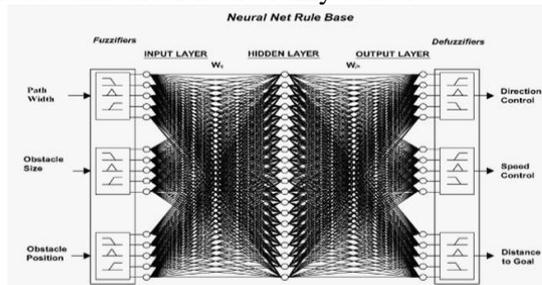


Figure.4. A Sample Neuro Fuzzy System

State Representation: The state representation of the system is the primary consideration in neuro fuzzy controller that determines the ability of final neuro fuzzy controller to generalize a solution from training data. The question about whether to input path width, obstacle size, obstacle position or a combination of these into the neuro fuzzy network will also determine the convergence of the system during training.

Robotic Arm Simulation: The Robotic simulink tool of a MATLAB program was used to implement neuro-fuzzy control, to get the training data and test the robotic arm controller. The datas are normalized to be din the range of (0, 1). The training samples were obtained by cycling through a series of goal and obstacle positions. The training

samples included two starting positions of the arm ($\theta_1 = 0; \theta_2 = 0; \theta_3 = 0; \theta_4 = 0; \theta_5 = 0$) and ($\theta_1 = 90; \theta_2 = 0; \theta_3 = 0; \theta_4 = 0; \theta_5 = 0$). Goal and obstacle positions were taken from the entire reachable region. The recorded training sample creates a new input to the neuro fuzzy system, at each movement. Selection of starting positions of arm at the two extremes of possible movements generated training samples that extended over the entire state space, which is important for neuro fuzzy control.

Neuro Fuzzy Programming: After the completion of training samples normalization, neuro fuzzy system was simulated using robotic MATLAB Toolbox. The neuro fuzzy engine is completely void of any constants of programming related to a robotic arm. As a result it can be used for any input/output data. The number of inputs, the number of nodes and even the fuzzy membership functions are defined for individual systems.

The Neuro Fuzzy Controller: A neuro-fuzzy system was trained using obtained training sample and controller was tested on the robotic arm, by placing obstacles randomly and assigning random starting positions. The controller simulation must quantify the performance of the proposed neuro fuzzy controller for that purpose, the problem statement must be originally addressed. The model specification of above section listed the following as the goals of the system:

- The end-effector reaches the goal (minimum approach distance),
- The arm does not touch the obstacle (collision avoidance)
- The calculations can be performed in real-time with current hardware.

The simulated neuro fuzzy controller decides the percentage error in reaching the goal, based on the above guidelines that determines whether collision occur or not, and performance judgment based on real time application. The collision of arm with obstacle can be detected from the path and obstacle Position.

Motion Control in Environments: SCORBOT ER 5U Plus robot model was used to stimulate intelligent neuro fuzzy motion control. Robot navigates through an unknown building, depending on the rough instructions about obstacle avoidance. It uses internal sensor, motion detectors, distance measurement capability and is self-propelled. Sensor may also be used to determine errors in joint positions, velocities, or forces that can be compensated by computer, by using feedback control.

The robot model is programmed to move from some start point say X_s to some goal point say X_y along a path containing moving obstacles as shown in Figure 5. Assumption was made regarding the path's size as space between left and right boundary. Outline of obstacle ahead was identified by the sensing equipment that defines it by their size and position relative to the path. The Robot's motion control facilitates arm movement around the obstacle and reach goal at minimum distance. The Robot's motion control facilitates arm movement around the obstacle and reach goal at minimum distance. The Figure 5 gives the view of the above statement. The behavior of this can be altered after suitable instruction that enhances the robot to recognize the presence of corners and obstacles at a different path. An Instructor teaches the robot arm to swear right on the path to go around the obstacle, which is on the left hand side of the path and vice versa. Thus it can be calculated that the robot arm autonomously avoids obstacles, on moving towards goal, by receiving obstacle position from sensor, fed to fuzzy controller.

The robot arm continuously processes information about its environment and changes the speed and direction based on the behaviors of the motion planner. The information for processing includes path's size, obstacle size and its relative position on path. On a straight path (no obstacle) the robot arm moves straight, have freedom to move in any direction left or right. The path's width determines the robot arm's speed human path planning involves fuzzy rules such as:

If path width is 'medium' with 'medium size object' at 'right side' of the path then go around the object by making a 'left-medium' turn at 'medium' speed.

The Robot arm encodes from human automobile driver intelligence, while driving a car. The processes are getting speeded up by using fuzzy values such as SLOW, MEDIUM AND FAST etc. to define the system. As a result final training error gets increased. Use of fuzzy values like VERY SLOW, SLOW MEDIUM, FAST and VERY FAST, etc. increases rules that complicates it but the final error get reduced by increasing number of fuzzy values in a fuzzy set. The motion control planner acquires several rules, through instruction which is somewhat limited but typical repertoire for collision avoidance sensors provide environmental data as input to robot. The left hand side variables are defined as follows: the current width of the path (denoted as X_1) the size of the obstructing object (denoted as X_2) and the position of the obstructing object (denoted as X_3).

The system output is usually a control signal to robot's motor that facilitates the movement of robot arm away from a feasibility obstructing obstacle to ensure its safety. This is defined by three action variables called direction (denoted as Y_1), speed (denoted as Y_2) and minimum distance of the goal to the robot (Y_3). The value of direction defines the direction of robot arm movement. The robot arm can turn positive or negative or in any direction in between. The speed value defines the speed of the robot arm while it is making a turn. Fuzzy variables describe any object of interest to the system X as a pair $\mu_A(X) = X$, where 'A' is a fuzzy category and $\mu_A(X)$ is the grade of

membership of the object X in category A. The following data objects are defined as fuzzy variables with a term indicated in the following section figures.

Path Width (x₁): The Figure.6, shows the membership functions of the values of X₁, which are triangular fuzzy numbers {NARROW, MEDIUM, WIDE}. The asymmetry observed with respect to the partition of universe of discourse has to do with the fact that the robot arm ordinary will stick to the right side of the positive. The fuzzy category width is nothing but the difference between the values whose membership equals ‘0’. The crisp path width determined from robot’s sensor, at any time has a fuzzy membership value in each of these categories. Fuzzy values for each category are calculated using equation (2) below:

$$\mu_A(x) = 1 - \{[\text{peak value} - x]/\text{path width}\} \tag{2}$$

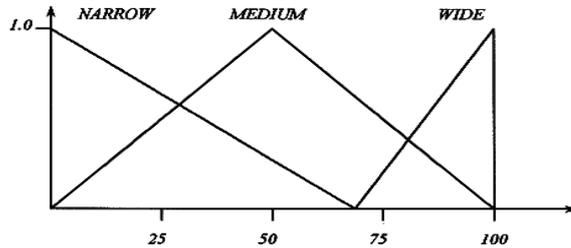


Figure.6. The Values of the Fuzzy Variable in Path Width x₁

Obstacle Size (x₂): The control variable for robot’s movement is the size of the obstacle in the robot’s path, which is described as a fuzzy variable (X₁) whose values are shown in Figure.7.

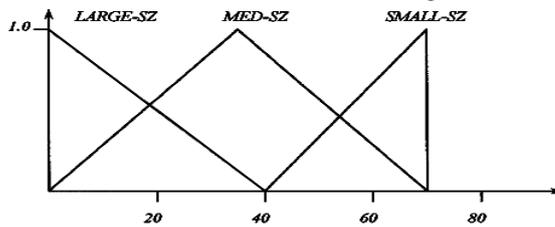


Figure.7. The Fuzzy Values of the Variable in Obstacle Size x₂

Obstacle Position (x₃): SCORBOT robot’s arm navigation was determined by the position of the obstructing object X₃. The robot arm can go around the object by either going left or right of the obstacle. The direction control of robot depends on the relative position of the obstacle in the path. Fuzzy variable X₃ has the values as shown in Figure.8.

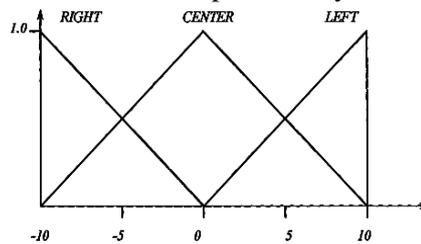


Figure.8. The Fuzzy Values of the Variable in Obstacle Position x₃

Robot Movement Direction (y₁): Direction control is important parameter to be informed to Scorbot arm for its movement, whose values are in degrees, can change from ‘-90’ i.e. a wide left-turn to ‘+90’ a wide right turn. In this system the fuzzy variable Y₁ may take the three values shown in Figure.9.

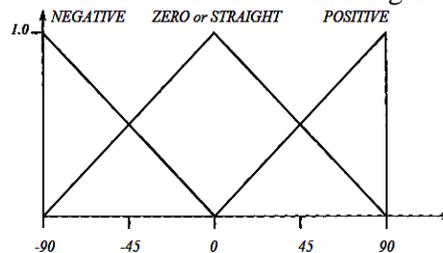


Figure.9. The Fuzzy Values of the Variable in Robot Movement Direction y₁

Robot Speed Control (y₂): The robot’s motor control system was driven by the fuzzy variable Y₂ whose speed value ranges from 0 to 60 in /sec. At the speed value ‘0’, no arm movement takes place. This and the other values of Y₂ are shown in Figure 10. A fuzzy rule base was constructed by the robot arm as soon as it receives the instructions. An example of a rule is:

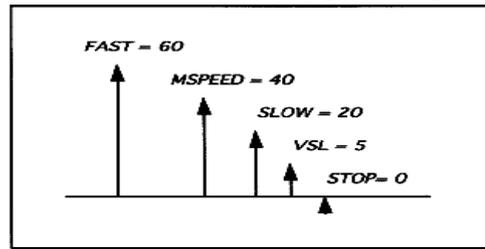


Figure.10. The Fuzzy Values of the Variable in Robot Speed Control y_2

If x_1 is WIDE AND x_2 is SMALL SZ AND x_3 is LEFT then y_1 is POSITIVE AND y_2 is MEDIUM-SPEED; (3)

Robot Goal Distance (y_3): The information regarding obstacle position, received from sensors, by the fuzzy controller, is required by robot to intelligently avoid the obstacles and reach the target at shortest distance. The range universe of discourse is discretized into levels with three fuzzy sets, where levels and membership functions are chosen heuristically Y_3 are shown in Figure.11. All rules have three fuzzy values in their left hand side and 3 values, direction control, speed control goal distance, in the right hand side.

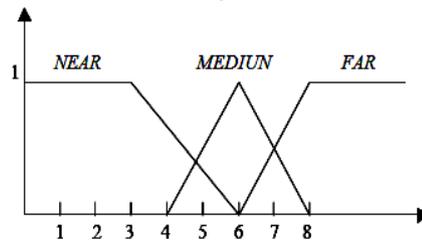


Figure.11. The Fuzzy Values of the Variable in Distance y_3

The max-min composition technique was used by Scorbot ER 5U robot to evaluate its fuzzy rules. The membership value of a rule is the minimum membership value of all the fuzzy data objects in the rule. Here the input to the system includes current path width, obstacle size and position, calculated from robot sensors. These values are changed to fuzzy categories and the membership value in each category is calculated e.g. the input values path = 100; object size=65; object position=18 are calculated to fuzzy values.

path = (WIDE=1.0;MEDIUM=0.0;NARROW=0.0);
 object size = (SMALL-SZ=0.0;MED-SZ=0.2; LARGE-SZ=0.8);
 object position = (RIGHT=0.0;CENTER=0.2;LEFT=0.8);

The resultant membership value is given by:

$$\mu(r_i) = \min\{\mu(x_1); \mu(x_2); \mu(x_3)\} \tag{4}$$

The action-part is derived by taking an average of all matched rules (the average of all degrees of fulfillment (DOF's))

$$Direction\ Angle = \frac{\sum_i \mu(r_i) * joint\ angle\ (r_i)}{\sum_i \mu(r_i)} \tag{5}$$

and

$$Speed = \frac{\sum_i \mu(r_i) * speed\ (r_i)}{\sum_i \mu(r_i)} \tag{6}$$

Simulations are performed on a Windows Xp machine using the MATLAB Simulink. A sample run of the system is shown below. Assuming, for example, that the current path size is 80 in, current obstacle size 40 in, and current obstacle position is 5 in (from the center to the right). This input is converted to fuzzy sets:

current path size (x_1) = (MED=0.7;WIDE=0.3);
 current obstacle size (x_2) = (MED-SZ=0.8;LARGE-SZ=0.2);
 current obstacle position(x_3) = (CENTER=0.4;RIGHT=0.6);

All combinations of the fuzzy values are matched against the 27 rules in the rule-base of this instructed behavior. A membership value is calculated for each of the matched rules:

r4: if x_1 is WIDE AND x_2 is CENTER AND x_3 is MEDIUM SIZE,
 then y_1 is NEGATIVE AND y_2 is 40 AND y_3 is FAR DOF = 0.3

r5: if x_1 is WIDE AND x_2 is CENTER AND x_3 is LARGE SIZE,
 then y_1 is ZERO AND y_2 is 30 AND y_3 is NEAR, DOF = 0.2

r7: if x_1 is WIDE AND x_2 is RIGHT AND x_3 is MED SIZE,
 then y_1 is POSITIVE AND y_2 is 40 AND y_3 is MEDIUM, DOF = 0.3

r8: if x_1 is WIDE AND x_2 is RIGHT, x_3 is LARGE SIZE,

then y_1 is NEGATIVE AND y_2 is 30 AND y_3 is FAR, $DOF = 0.3$

r22: if x_1 is MED AND x_2 is CENTER AND x_3 is MED SIZE, then y_1 is NEGATIVE AND y_2 is 30 AND y_3 is FAR, $DOF = 0.4$

r23: if x_1 is MED AND x_2 is CENTER AND x_3 is LARGE SIZE, then y_1 is ZERO AND y_2 is 30 AND y_3 is NEAR, $DOF = 0.2$

r25: if x_1 is MED AND x_2 is RIGHT AND x_3 is MED SIZE, then y_1 is POSITIVE AND y_2 is 30 AND y_3 is MEDIUM, $DOF = 0.6$

r26: if x_1 is MED AND x_2 is RIGHT AND x_3 is LARGE SIZE, then y_1 is NEGATIVE AND y_2 is 30 AND y_3 is FAR, $DOF = 0.2$.

JINT14The final outputs are defuzzified using the “weighted averages” technique commonly used for computing by taking the average of the results (DOF’s) of all the matched rules for singleton output sets. The final output is: speed = 13.0 inches/secs, direction angle = 37.9° and distance to goal = 2 inches. After the robot arm has been instructed to make a turn so that it may avoid obstacle in its rule base has modified membership functions. Scorbot proceeds by taking into account all things, i.e., avoid obstacle and make left or right turn with proper speed based on the minimum approach distance of goal.

$$Z = \frac{\sum_{i=1}^n \mu_i z_i}{\sum_{i=1}^n \mu_i} \tag{7}$$

Where z_i is the output of the μ_i rule and Z is its degree of truth.

3. RESULTS AND ANALYSIS

To implement neuro fuzzy system on robotic arm control, a model is required first, that generates training samples from a human’s intelligence of controlling an arm, which is used to train seven different neuro-fuzzy controllers namely NF1,NF2.....,NF7 for reference. The controller’s are varied in the definition of their fuzzy membership functions definition on network training to a sinusoidal function. The section constitutes the results of training sessions and testing of each controller on the simulated arm. Three different types of graphs are used to display the results. The fuzzy membership function graph gives the details regarding crisp values, the way that are converted into fuzzy values. The seven controllers constitutes its own fuzzy membership function graph. The RMS training error graph indicates the capability of neuro-fuzzy controller in learning the training data. The controller’s performance in moving robot’s arm from starting position to goal without any interruption was to quantify by the performance histograms. The fuzzy membership function of NF1 that assigns a crisp value’s degree of membership to three fuzzy values – NEGATIVE, ZERO AND POSITIVE. On the other hand, the fuzzy membership function NF2 and NF3 have seven fuzzy values – NEGATIVE LARGE, NEGATIVE MEDIUM, NEGATIVE SMALL, ZERO, POSITIVE SMALL, POSITIVE MEDIUM, POSITIVE LARGE. The learning capabilities of these three controllers’ are shown in Figure 12 and its performance are compared in Figure.13.

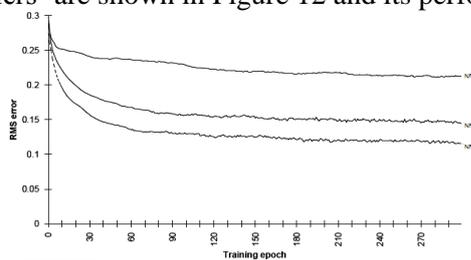


Figure.12. The RMS Training Error of NF1, NF2, and NF3

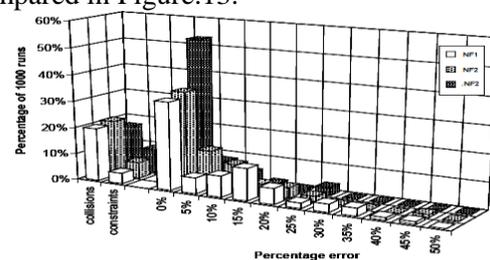


Figure.13. Performance Histogram of the Neuro Fuzzy Controller forNF1, NF2 and NF3

Different controllers are compared with the performance histogram. The collision with obstacle are recorded in collision category of histogram that displays the values in percentage of runs that ends in collision when the neuro fuzzy controller attempted to move the arm past the joints capabilities, the constraint” category was incremented. The percentage error categories record how close the end effectors were to the goal at the arm’s final position. The final RMS training error of seven controllers was compared in Figure.14. Attempts are made to train a back propagation neural network to control the simulated robot arm. The absence of fuzzification layer makes the learning of neural network from sample data as complicated one.

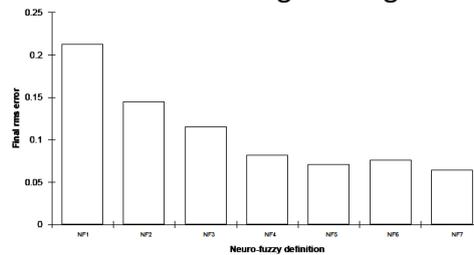


Figure.14. A Comparison of the Final RMS Training Errors

The process of programming and debugging of neuro fuzzy model was extensive one. The first function used to test the neuro fuzzy system was $y=x$. The outputs are produced at after 800 training epochs. The function $y=\sin(x)$ is used to test the ability to neuro fuzzy system to learn more complex data. The output of network is checked after training epochs. A programming bug exist is not a problem for the single input, single output function of neuro fuzzy model. The error ws identified by testing the model with the function $z=x|\sin(y)|$. The network learns the function only after correcting the programming bug and train it for epochs. The learning rate constant β , is another important part of back propagation neural network that determines the adjustment of weights corresponding to the error, which is back propagated. The effect of the learning rate constant on three training sessions for the sine functions, which began the networks with same initial weights. The network with higher learning rate constitutes less training error. It becomes apparent during the early stages of this work that the neuro fuzzy system would benefit from the addition of a momentum term in the learning rule. The quick convergence of system was indicated by momentum term in the result. The neuro fuzzy model was tested for several functions with varying membership function definitions, to test the fuzzy membership function features. The higher granularity ensures lowering of error in fuzzifier, defuzzifier.

4. CONCLUSION

This paper deals with formation of a new, stable, robust and intelligent controller that helps integrate of neural networks and fuzzy logic and this robust based computerized torque controller is proposed to control the movement of robot manipulators in an unstructured dynamic environment. The replacement of fuzzy system with neural network helps achieve the complex input output relationship which is normally difficult to be specified by 'IF/THEN' rules. This new system demonstrates that this strategy effectively controls the robotic manipulators. Another important result is effective autonomous capability of the proposed scheme. Now the robot can automatically avoid the obstacles and chooses its way to its target point. Another merit of proposed controller is that it is capable of functioning without a basic requirement of knowledge of the robot manipulator parameters. For instance the conventional techniques find its very difficult to model the friction but the neural networks learnt them on-line which facilitate the system with the high level of autonomy which in turn helps the controller to use various dynamic applications. The manipulator dynamics now can experience parameter variations, load changes and any other possible external disturbances. Finally from the results, we can arrive at two conclusions,

- A. The membership function plays an important role in neuro fuzzy system.
- B. Figure shows that the increasing number of fuzzy values in a fuzzy sets results in the reduction of final training error by 50%.

REFERENCES

- Ahrikencheikh C and Seireg A, Optimized-Motion Planning, John Wiley and Sons, New York, 1994.
- Becker M, Obstacle Avoidance Procedures for Mobile Robots, XVI Brazilian Congress of Mechanical Engineering, 15, 2001, 126-135.
- Bourbakis N.G, Knowledge Extraction and Acquisition during Real-Time Navigation in Unknown Environments, International Journal of Pattern Recognition Artificial Intelligence, 9 (1), 1995, 83-99.
- Connolly C.I, Harmonic Functions as a Basis for Motor Control and Planning', Ph.D. Dissertation, Department of Computers Science, University Massachusetts, Amherst, 1994.
- Daniel H, Multi-Agent Fuzzy Control of the Robotic Soccer, 5th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence and Informatics, Poprad, Slovakia Technical University of Kosice, Slovakia, 2007.
- Ding and Li H.X, Fuzzy Avoidance Control Strategy for Redundant Manipulators, Engineering Application Artificial Intelligence, 12, 1999, 513-521.
- Evin M, Intelligent Control Design in Robotic Soccer, In Proceedings of Selected Extended Abstracts Based on Final M. Sc. Projects from the Branch of Artificial Intelligence, 2005, 43-48.

Gat E, on the role of stored internal state in the control of autonomous mobile robots, *Artificial Intelligence Magazine*, 1993, 64-73.

Gourdeau R, Object Oriented Programming for Robotic Manipulator Simulation, *IEEE Robotics and Automation Magazine*, 1997, 21-28.

Hall E.L, Ghaffari M, Liao X and Alhaj A.S.M, Intelligent Robot Control Using an Adaptive Critic with a Task Control Center and Dynamic Database, Center for Robotics Research, University of Cincinnati, Cincinnati, OH 45221-0072 USA, 2007, 1-12.

Hwang Y.K and Ahuja N, A potential field approach to path planning, *IEEE Transactions of Robotics and Automation*, 8 (1), 1992, 23-32.

Hwang Y.K and Ahuja N, Gross Motion Planning-a Survey, *ACM Computing Surveys*, 24 (3), 1992.

Khatib O, Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, *International Journal of Robotics Research*, 5, 1986, 90-98.

Kim J.O and Khosla P.K, Real-Time Obstacle Avoidance Using Harmonic Potential Functions, *IEEE Transactions of Robotics and Automation*, 8, 1992, 338-349.

Koditschek D.E, Some Applications of Natural Motion, *Journal of Dynamic System, Measurement and Control*, 113, 1991, 552-557.

Koditschek D.E, The Control of Natural Motion in Mechanical Systems, *Journal of Dynamic System, Measurement and Control*, 113, 1991, 547-551.

Latombe J.C, *Robot Motion Planning*, Kluwer Academic Publishers, Boston, 1991.

Lids E.Z and Marcelo B, An Artificial Neural Network Structure Able to Obstacle Avoidance Behavior Used in Mobile Robots, Pontifical Catholic University of Minas Gerais, Av. Dom José Gaspar, Belo Horizonte - MG - CEP 30535-610, Brazil, 2007.

Mbede J.B, Huang X and Wang M, Fuzzy Motion Control among Dynamic Obstacles Using Artificial Potential Functions for Robot Manipulators, *Robotics Automation and System*, 32 (1), 2000, 61-72.

McKerrow P.J, *Introduction to Robotics*, Pearson Education International, 1991, 640.

Ngo V.T, Nguyen A.D and Ha Q.P, Integration of planning and control in robotic formations, ARC Centre of Excellence for Autonomous Systems, University of Technology, Sydney, Australia, 2007.

Noury B and Serge S, Towards adaptive robot control architecture', 2nd National Workshop on Control Architectures of Robots: from Models to Execution on Distributed Control Architectures, Paris - France, 2007, 1-15.

Pham D.T and Liu X, Identification of Linear and Nonlinear Dynamic Systems Using Recurrent Neural Networks, *Artificial Intelligence Engineering*, 8, 1993, 67-75.

Rao N.S.V, Algorithmic framework For Learned Robot Navigation in Unknown Terrain's, *IEEE Computer*, 1989, 37-43.

Rimon E and Koditschek D.E, Exact Robot Navigation Using Artificial Potential Functions, *IEEE Transactions of Robotics and Automation*, 8, 1992, 501-518.

Rimon E and Koditschek D.E, The Construction of Analytic Differ Orphisms for Exact Robot Navigation on Star Worlds, In *Proceedings IEEE Conference on Robotics Automation*, 1989, 21-26.

Terano T, Asai K and Sugeno M, *Fuzzy Systems Theory and its Applications*, Academic Press, Boston, 1992.

Tsoukalas L.H, Houstis E.N and Jones G.V, Neuro-Fuzzy Motion Planner for Intelligent Robots, *Journal of Intelligent Robotics and System*, 19, 1997, 339-356.

Tuscillo A and Bourbakis N.G, A Neural and Fuzzy Control of a Robotic Hand, *IEEE Transactions on SMC*, 1996.

Volpe R and Khosla P.K, Manipulator Control with Super-Quadric Artificial Potential Functions, *Theory and Experiments, IEEE Transactions of System, Man and Cybernetics*, 20, 1990, 1423-1436.