Design and Simulation of Proportional-Derivative Controller of A Servo Systems Based on Fuzzy Logic and Hybrid Fuzzy Neural

Laith Ali Abdul-Rahaim
Electrical Engineering Department, Babylon University

Abstract

In the past few years, fuzzy-rule-based modeling has become an active research field because of its good merits in solving complex nonlinear system identification and control problems. A servo system (SS) is a class of a nonlinear position system that needs to be positioned accurately and fastly on a commanded position. The strategy followed in this paper in designing digital controller for such system; first building a neuro-model that represents the open loop servo system. This is accomplished by sufficiently collecting input-output data and used it off-line to build the neural network that will represent the plant for the second design stage. Second design fuzzy controller through NN simulation to reach the required closed –loop behavior. The design technique is based on the adjustment of the scale factors, rule base and membership functions of the controller; it and was accomplished by fine tuning and heuristic corrections linked to the knowledge of the process to be controlled. For the specified plant, there are certain parameters, which produce a well-controlled response.

Keywords— *Fuzzy Logic Control, Neural Networks, Internal Model Control, Servo system.*

الخلاصة

أصبحت النمذجة المعتمدة على استخدام قواعد البيانات الضبابية (fuzzy rule base) من المجالات الفعالة وذلك بسبب مميزاتها الجيدة في حل معضلات السيطرة والتعرف على معلمات المنظومة غير خطية. أن منظومة المؤازرة هي نوع من أنواع المنظومات الغير خطية والتي تتطلب السيطرة على خرج المنظومة، الموقع المطلوب بدقة وسرعة عالية. إن الإستراتيجية المعتمدة في هذا البحث لتصميم المسيطر الرقمي الضبابي (digital fuzzy logic controller) للسيطرة على منظومة المؤازر تعتمد على اولا": بناء نموذج عصيبي (neuro-model) والذي يمثل المنظومة المؤازرة المفتوحة (open-loop s.s) ويتطلب ذلك معرفة بيانات الإدخال والإخراج لتلك المنظومة، حيث تستخدم تلك البيانات لبناء شبكة عصيبية منفصلة (off-line) لتمثيل المنظومة (plant) لمرحلة التصميم اللاحقة. ثانيا": تصميم المسيطر الضبابي من خلال التمثيل باستخدام الحاسوب للحصول على الاستجابة المطلوبة للدارة المغلقة (closed-loop). تعتمد تقنية التصميم على تنظيم عوامل القياس (scale factors) والدوال العضوية (membership functions) للمسيطر الذي تم بناءه باستخدام التنغيم الدقيق والتصحيحات المؤرخة (heuristic corrections) المرتبطة بمعرفة النظام المراد السيطرة عليه. ومن الجدير بالذكر أن لكل منظومة هناك معلمات خاصة تحقق الاستجابة المطلوبة.

1. Introduction

Increasing control systems are required to have dynamic performance and robust behaviors, yet are expected to cope with more complex, uncertain and highly nonlinear dynamic processes. Along with this increased process, complexity is

increased abstraction and uncertainty in the models and their mathematical representation .One significant approach in dealing with major changes and uncertainty in nonlinear dynamic processes is through intelligent modeling and control. Intelligent controllers are generally self-organizing or adaptive and are naturally able to cope with the significant changes in the plant and its environment, while satisfying the control design requirement [Radu et al2009]. As with any advanced control theory, a central issue is the representation and development of appropriate process models with known approximation errors. As processes increase in complexity, they become less amenable to direct mathematical modeling based on physical law, since they may be [Vengerov et al., 2005]: Distributed, stochastic, nonlinear and time varying, Subject to large unpredictable environmental disturbances, and Have variables that are difficult to measure, have unknown casual relationships or are expensive to be evaluated in real time. The emergence of AI technology has undoubtedly expanded the frontier of the power electronics and motor drives area, which is already a complex and multidisciplinary technology. Among all AI disciplines, the present trend indicates that neural networks will possibly have the maximum impact on power electronics. Most of our current applications are based on backpropagation (often called MLP or multilayer perceptron) networks, although a large number of feedforward and recurrent topologies are already available. Many novel applications can be explored with these networks. Unfortunately, large fully functional ASIC neural chips are not yet available [Vengerov et al 2005], and most current applications use DSPs. For this reason, currently, industrial applications of ANNs are very few. Again, by far, the majority of ANN applications use off-line training. However, in a parameter varying or unfamiliar signal environment, adaptive ANN with on-line training is essential. Fortunately, NNW training time is decreasing progressively because of improvements in the training algorithms and rapid increases in computer speeds. Some of the off-line techniques can be used for on-line if the training time demand is not too stringent. Currently, much emphasis is being placed on R&D for hybrid-AI techniques, which will have a significant impact on power electronics. The conventional controllers encounter difficulties when facing nonlinear, uncertain, temporal behavior. In recent years, a great deal of attention has been paid to the application of Artificial Neural Networks (ANN) in modeling, identification, and control of dynamic processes [Cheng 2007]. ANNs provide an excellent

mathematical tool for dealing with nonlinear problems. They have an important property, according to which nearly any continuous nonlinear relationship can be approximated with acceptable accuracy using a neural network with suitable architecture and weight parameter [Dongmei et al., 2008]. There is another attractive property is the self-learning ability. The performance of fuzzy controllers depends on two significant issues, namely the soundness of knowledge acquisition techniques and the availability of human experts. These two issues restrict the application domains of FLCs. Adaptive-Network-based Fuzzy Inference System bypasses these issues by tuning the FLC directly from a desired input-output data set [Shahnazi 2008]. The objective of this study was to design an FLC for a servomechanism, based on acquired data from a human expert. We used an ANFIS to build and tune a fuzzy controller that would be able to accomplish the control task. Unlike conventional control techniques such as LQR (where the availability of a mathematical model of the system is paramount), we demonstrate that by simply acquiring input-output data from a human expert in a human-in-the-loop fashion and tuning the FLC parameters by incorporating these data directly into an ANFIS structure, one can avoid the burden of obtaining the mathematical model for the system and design an FLC that outperforms the existing conventional controllers.

A neural network (NN) can extract the system feature from historical training data using the learning algorithm, requiring a little or no a prior knowledge about the process. This provides modeling of nonlinear system a great flexibility. These features allow one to design adaptive control system for complex, unknown and nonlinear dynamic process [Zurada 1996].

As opposed to many effective applications, e.g. in pattern recognition problems, approximation of the nonlinear function, the application of NN in control systems requires taking into consideration the dynamic of the processes being investigated. Another important application area, where the dynamic NN can be effectively used, is diagnostics of industrial process

Recently, fuzzy logic controllers (FLCs) are finding increasing use in industry. The application of fuzzy reasoning to process control has opened up a new approach in this field. A controller is built from a set of fuzzy rules naturally incorporate commonsense expert knowledge; it may be easier to build and to maintain this than a conventional controller [Shahnazi 2008]. The advantage of fuzzy control lies in its ability to implements the action of expert operator without the need of accurate mathematical model. The main benefits of this approach can be summarized in [Reznik 1997, Feng 2007]. The main objective of this work is to design and implementation of fuzzy logic and fuzzy neural controllers to the servo system. The work is directed towards to identify the servo system by the neural method MRNN, using the neuro-identified model to design the fuzzy logic and fuzzy neural controllers, and testing the performance of the proposed models on the servo system [Ahmed et al., 2007].

2. System Identification:

In general, exciting the system and observing its input and output over a time interval performs an identification experiment. These signals are normally recorded using computer with mass storage. The first step is to determine an appropriate form of the model and in the second step some statically based method is used to estimate the unknown parameter of the model. Finally the model obtained is tested to see whether it is representation of the system [Crockett 2006]. The position system under experiment for collecting the input-output data depends on the feedback principle that comparison the controlled variable, whatever it may be with a desired value of that variable, so that errors signal or a measure of the error can be formed. The servo was arranged so that operates in a sense to reduce the error to zero then output equal the demanded input. The objective of the servomechanism is to position a massive object by means of a motor and gearbox as shown in fig.(1). The performance of RLS and MRNN for system identification will be examined, by considering the input output data collection from the plant. The parallel identification scheme, which is used for identification, based on modified recurrent neural network (MRNN) is illustrated in fig. (2). The network is trained using the backpropagation-training algorithm [Pham 1997]. The collected data from the SS if 1volt-step input is applied to the preamplifier is shown in fig. (3). The input signal $U_p(k)$ is applied to the plant as well as the network and the error signal is then feedback to the network. The aim of the learning is to minimize the $R.M.S$ error. The learning rate was chosen by trial and error for the MRNN, typical value to be chosen is (0.01). For the recursive least square (RLS), the initial value of estimation parameter chooses to be zero $(\theta(0)=0)$, and initial covariance matrix $P(0)=\sigma_0 I$ with $(\sigma_0 = 300)$. Fig.(4) shows the responses of the RLS method and the SS, which is represented by the input output data collected. Figure (5a) shows the modeling error of the RLS method, and fig. (5b) show the covariance matrix trace of the RLS, and fig (6) show the parameter estimates of the RLS $(a_1, a_2, b_1 \ \ and \ \ b_2)$. To demonstrate the capability of this neuro-identifier, the MRNN is selected with one input, six hidden and context units, and one output unit. The learning rate is chosen by trail and error and it is notice that, large learning rates cause oscillations or even instabilities to the training process as shown in fig. (7). When suitably small learning rates are adopted so that no oscillations or instabilities occur, training $R.M.S$ errors are extremely slow to reach an acceptable error level for good results. Increasing the number of hidden units makes the achievable $R.M.S$ error levels smaller. However the number of hidden units cannot be too large because the permissible learning rates become even smaller and the training is even slower [Paulo 2000]. Also, the initial values of the weights are effective in the training process; this is due to the fact that the starting point of the learning process is determined by the initial values of the weights.

The choice of activation function is not a vital problem. Practically, in the field of using NN identification, if the system to be identified is linear, a linear activation function is used. And in the case of nonlinear system, a nonlinear activation function will be used [Mohan et al., 2008]. However, many tests have been carried in order to show the effect of the type of the activation function on the identification results. For this work the sigmoid activation function was chosen for the hidden and output layer. The initial values of all trainable weights are initialized at small random values between (0.5, -0.5). Figs. (8) shows the response of MRNN model and the SS [Bimal 2006, Nicholas 2008].

It is clear from Figs. (4) and (8), that the MRNN represent the system under test more accurately than the RLS. The transfer function obtained by the RLS can be written as follows:

$$G(z)=\frac{b_1 Z^{-1}+b_2 Z^{-2}}{1-(a_1 Z^{-1}+a_2 Z^{-2})} \qquad (1)$$

where $a_1=-1.7417, a_2=0.7419, b_1=0.051, b_2=0.051$ are the convergence parameters. All the software has been written using MATLAB commands and Simulink application tool, version 7.4

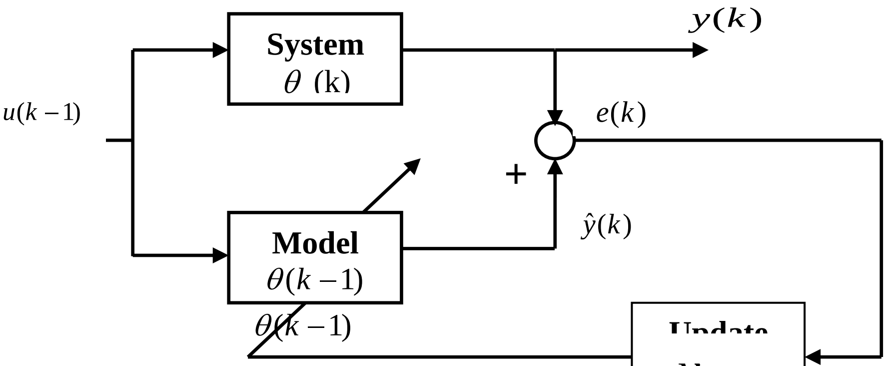


Figure (1) V... Recursive Estimation

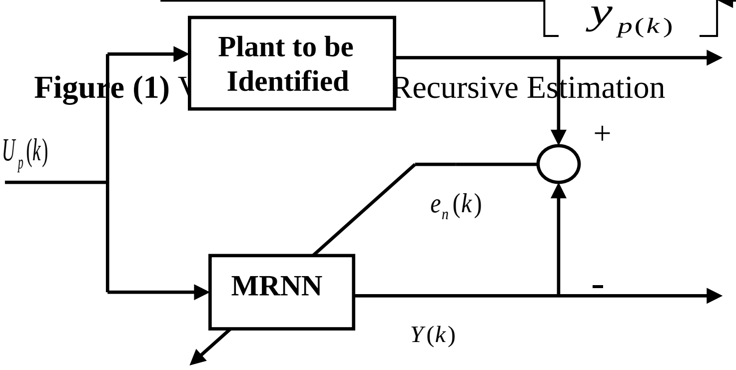
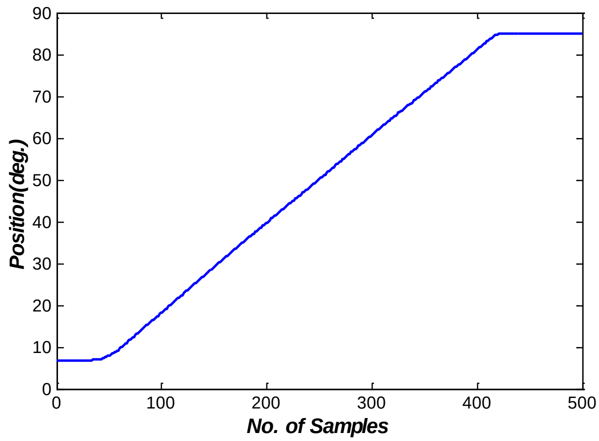
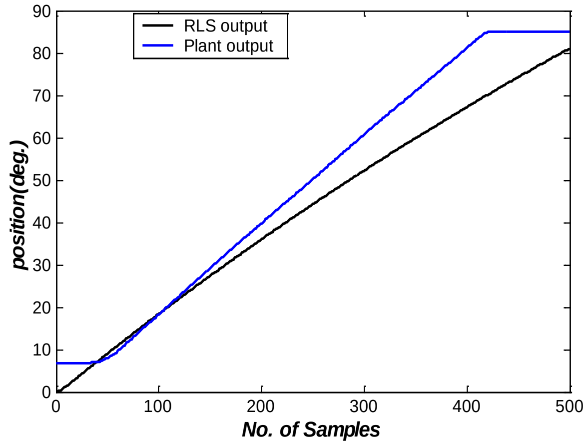


Figure (2) The Parallel Identification Scheme Based on MRNN





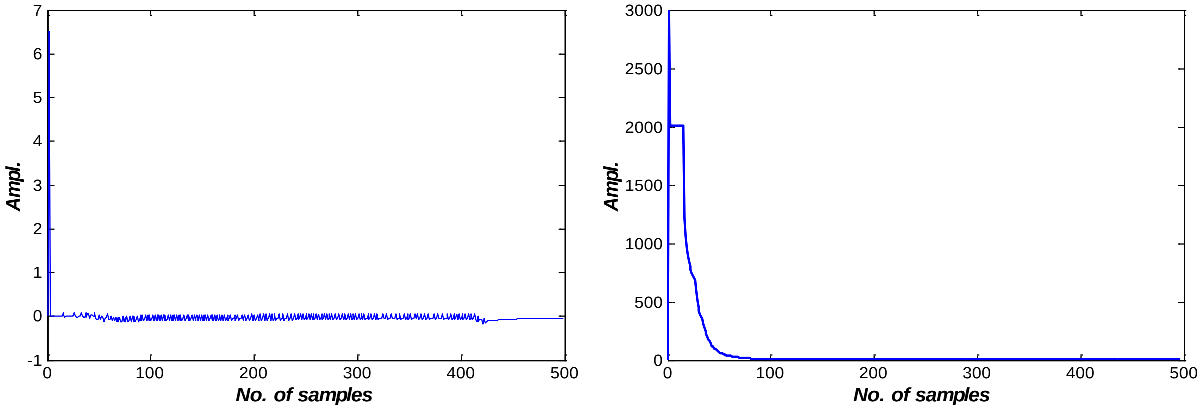


Figure (4) modeling error and trace of the RLS

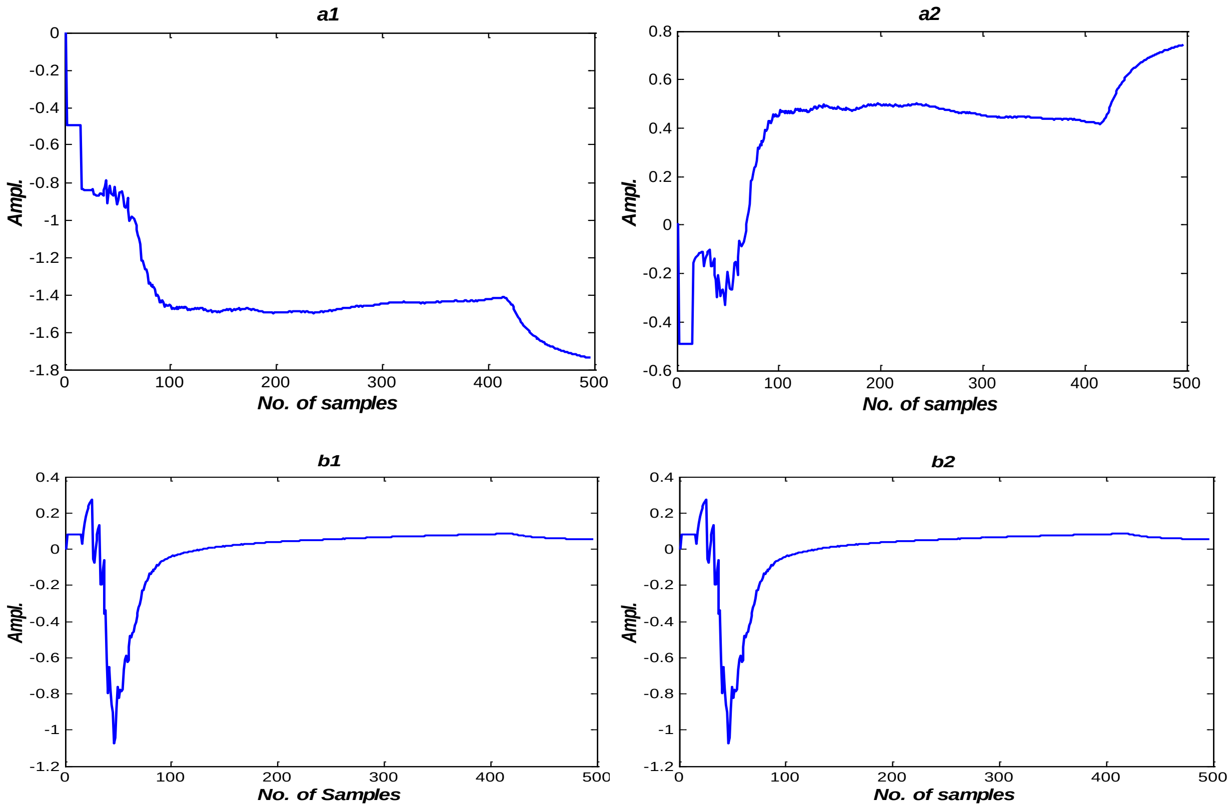
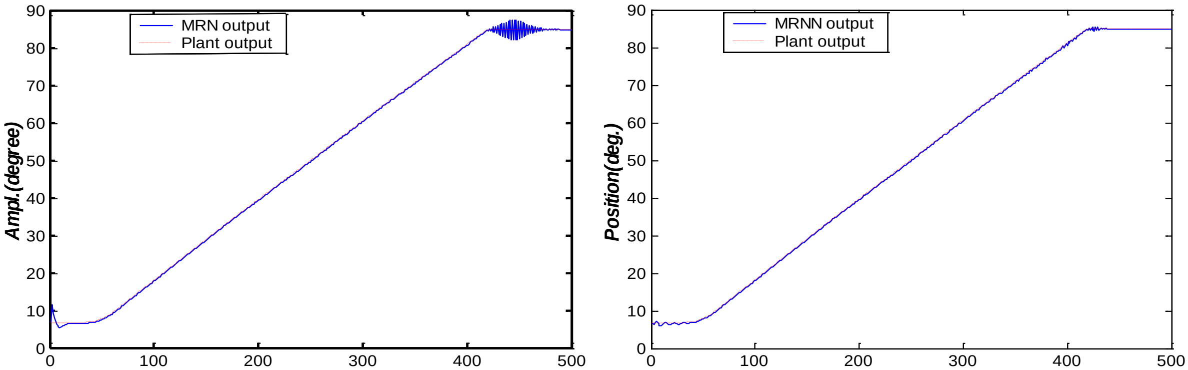


Figure (5) parameters estimate of the RLS



.

3. Design of PD Fuzzy Logic and fuzzy neural Controllers:

The block diagram of the plant with the Proportional-Derivative Fuzzy Logic Controller is shown in fig. (7).

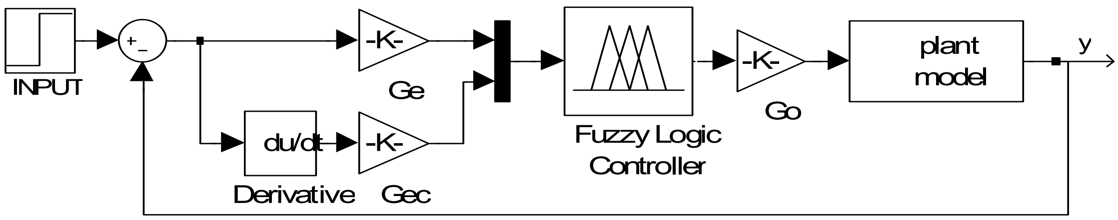


Figure (7) PD fuzzy logic controller

The input to the FLC is the position error $e(k)$ and position error change, $\dot{e}(k)$ that is:

$$e(k) = r(k) - y(k) \text{ and } \dot{e}(k) = \Delta e = \frac{e(k) - e(k-1)}{T}$$

Where $r(k)$ is the reference input which represents desired angular position.

$y(k)$ is the process output which represent the actual angular position.

T=(t2-t1) sampling period.

The output of the fuzzy controller is denoted by $U(k)$, which is the input to the plant.

The following steps describe the general PD fuzzy logic controller algorithm:

1. Initialize the fuzzy inference parameters, say centers (c_i) and widths (σ_j) of membership functions and consequent parts parameters, say z_m.

2. While stopping condition is false, do steps 3-5:

3. Compute the normalized system error signal: $e(k) = \dfrac{\omega_r(k) - \omega(k)}{\omega_{base}}$

Compute the normalized change in system error signal:

$\Delta e(k) = e(k) - e(k-1)$

$\omega_r(k)$ is the speed reference.

$\omega(k)$ is the estimated speed.

ω_{base} is the base speed.

4. Compute the change in controller output, say $\Delta u(k)$, using Sugeno inference mechanism for the inputs e(k) and $\Delta e(k)$.

5. Compute the overall controller output: $u(k) = u(k-1) + \Delta u(k)$

6. Test for the stopping condition (If it is required to terminate the program, then stop, else Continue).

The structure of control system is shown in Fig. 8. NNM is the neural network prediction model. FNNC is fuzzy-neural networks controller [Crockett 2006]. It consists of four layers NN shown in Fig. (9) ω, Δω is the input and u* is the output. Individual input has seven linguistic variables. They are described by *NB*, *NM*, *NS*, *ZE*, *PS*, *PM*, *PB*, respectively. The corresponding membership function is the type of Gauss, i.e., $\mu(x) = \exp(-(x-a)^2 / b)$. The fuzzy sets are all designed for standard, uniform, and perfect.

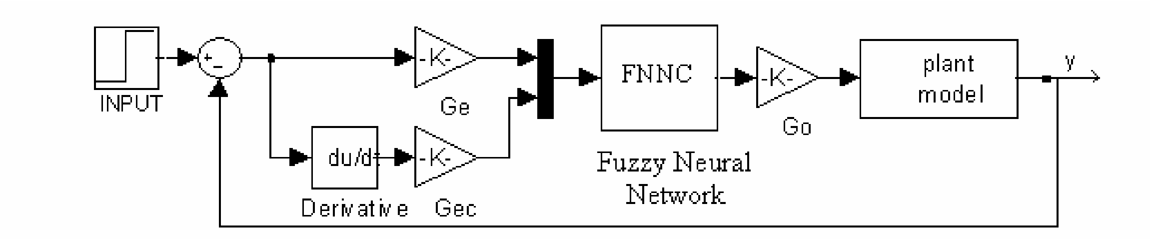


Figure (8) PD Fuzzy Neural Network Controller

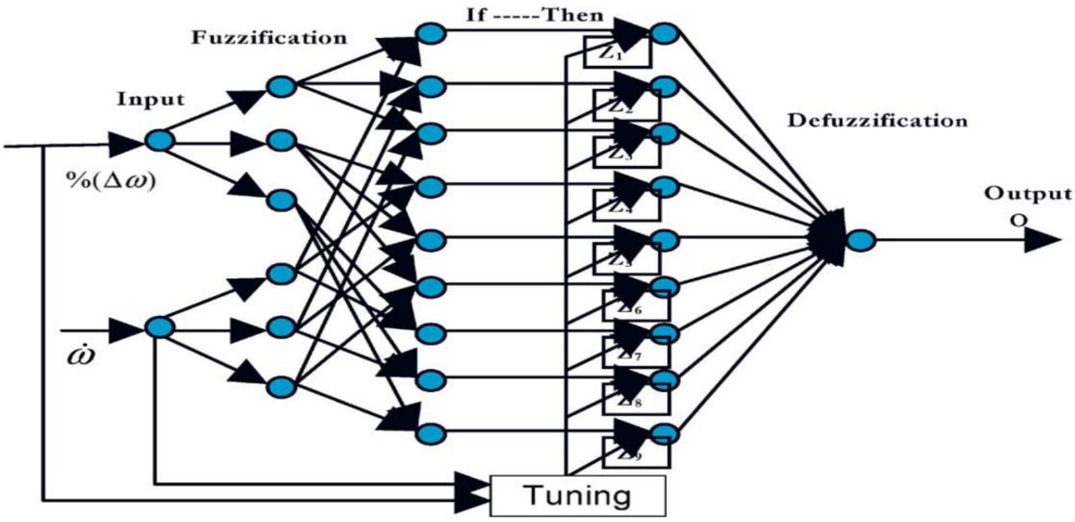


Figure (9) FNNC controller structure

The following steps describe the PD fuzzy neural controller algorithm that present in this work:

1. Initialize the input membership functions parameters, say c_i and σ_j.
Initialize the rules outputs parameters, say z_m.
Initialize the learning factors ($0<\eta_{1,2,3}<1$).
Initialize the tolerance.

2. While the stopping condition is false do steps 3-11:

3. Calculate the normalized speed error at the k^{th} time instant:

$$e(k) = \frac{\omega_r(k) - \omega(k)}{\omega_{base}} \qquad (2)$$

$\omega_r(k)$ is the speed reference and $\omega(k)$ is the estimated speed and ω_{base} is the base speed.

4. Calculate the normalized change in speed error:

$$\Delta e(k) = (e(k) - e(k-1)) \qquad (3)$$

5. For the two inputs $e(k)$ and $\Delta e(k)$, do steps 6-10:

6. Compute the output of ANFIS; say Δu(k), which corresponds to the inputs e(k) and Δe(k).

7. Adjust the membership functions parameters and rules outputs such that the cost function E is minimized by employing equations $c_i(t+1)=c_i(t)-\eta_1\frac{\partial E^k(c_i(t),\sigma_j(t),z_m(t))}{\partial c_i(t)}$, $\sigma_j(t+1)=\sigma_j(t)-\eta_2\frac{\partial E^k(c_i(t),\sigma_j(t),z_m(t))}{\partial \sigma_j(t)}$ and $z_m(t+1)=z_m(t)-\eta_3\frac{\partial E^k(c_i(t),\sigma_j(t),z_m(t))}{\partial z_m(t)}$

8. Test for stopping condition:

If the change in parameters in step 7 is greater than the tolerance then do step 7.

Else, continue.

9. Compute the output of the ANFIS, i.e. Δu(k), after training.

10. Compute the overall output:

$$u(k)=u(k-1)+\Delta u(k) \tag{4}$$

11. Test for the stopping condition:

If it is required to terminate the program, then stop.

Else go to step 2.

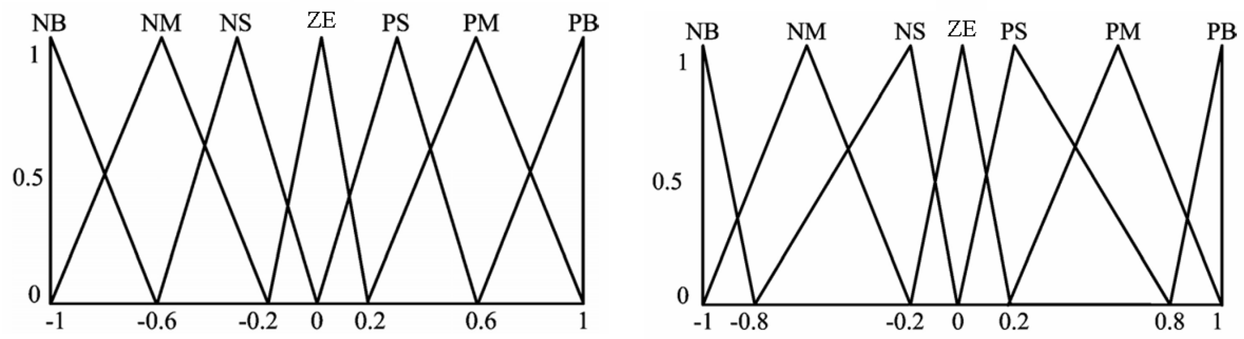
4. Simulation Results

In the simulation results, five triangular membership functions for each inputs and output variables are used, which are uniformly distributed across their universes of discourse for inputs and output membership functions. The fuzzy system is normalized which means the effective universes of discourse are all given by (-1,1). The linguistic values of these membership functions (for inputs and output) are NB,NM,NS,ZE,PS,PM, and PB which stand for (negative big, negative small, zero, and positive small and positive big respectively). The associated distributions of membership functions are shown in Fig. 10. The complete set of rules is shown in tabulated form in table (1), the premises for the input $e(k)$ are represented by the linguistic values in the left-most column, the premises for the input $\dot{e}(k)$ are represented by the linguistic values found in the top row, and the linguistic values representing the consequent for each of the rules can be found at the intersections of the row and column of the appropriate premises. Table (1) is constructed based initially on the characteristics of the system, then they are fine-tuned by repeated trials, and this table represents abstract knowledge about how to control the process given the error and its derivative as input [Golob 2001].

The fuzzy operation is executed using center of gravity (COG) defuzzification method. The controller was found to have best performance when the values of the scaling factors are $(g_e = 0.29, g_{ec} = 0.09, g_o = 3)$. This PD Fuzzy Logic Controller is used to control the position of the Servo System. The neuro-model obtained previously is used to represent the dynamic behavior of the actual SS. The SS response under PD FLC is shown in fig.(11) and the controlled voltage of the FLC applied to the amplifier stage of the SS is traced in fig. (12).

At sampling time 250, a step disturbance on the plant output of magnitude (6) was added, so the error is suddenly increases to make a high peak overshoot, as shown in figures (13) and (14). It is clear that the FLC is capable to handle this disturbance.

In the case of a conventional controller (such as PD controller) a design problem includes a proper choice of the PD controller coefficients. In the FLC design, one needs to choose many more parameters, number of rules, membership functions a scalar factors fuzzification and defuzzification procedures. These extra parameters make a FLC more robust and much difficult for analysis [Golob 2001].



(a) **(b)**

Fig. 10 Membership functions of: (a) input variables. (b) Output variable.

Table (1) Linguistic output control rules matrix

Δe \ e	PB	PM	PS	ZE	NS	NM	NB
PB	NB	NB	NM	NM	NS	NS	PM
PM	NB	NM	NS	NS	NS	ZE	PM
PS	NB	NM	NS	NS	ZE	PS	PB
ZE	NB	NS	NS	ZE	PS	PS	PB
NS	NB	NS	ZE	PS	PS	PM	PB
NM	NM	ZE	PS	PS	PS	PM	PB
NB	NM	PS	PS	PM	PM	PB	PB

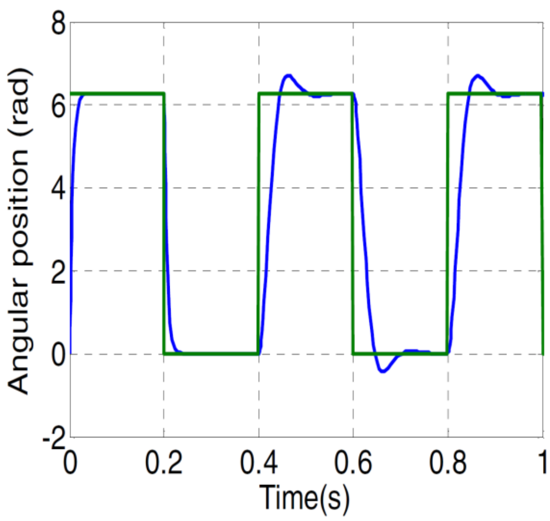


Figure (11) the output of the SS under FLC

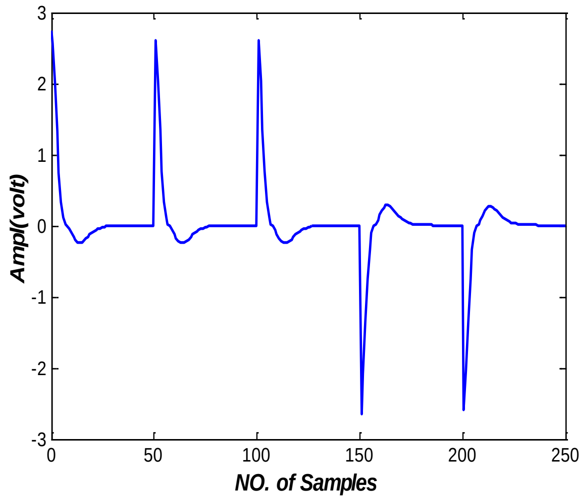


Figure (12) Control action of the FLC

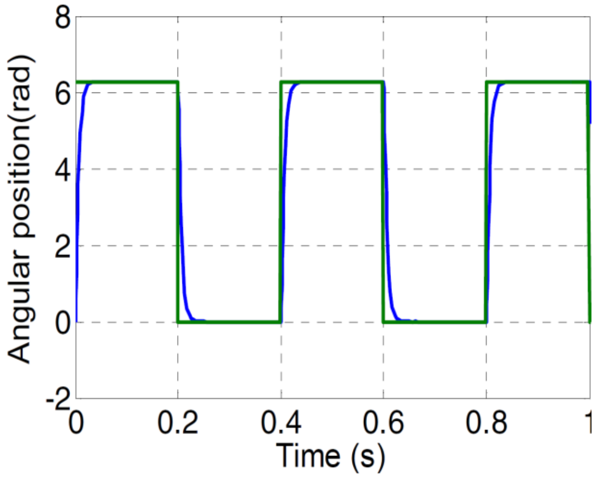


Figure (13) the output of the SS under FLC with disturbance

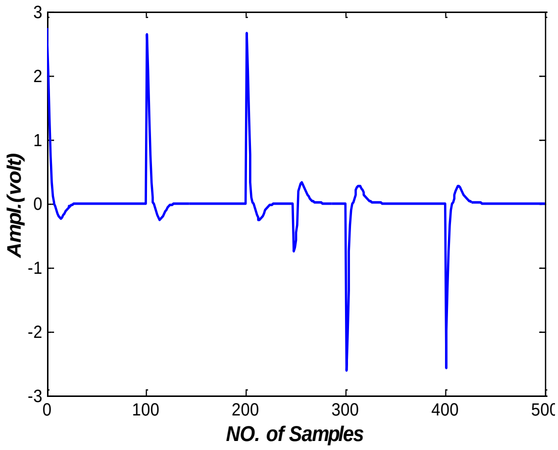


Figure (14) Control action of the FLC with disturbance

Following results are concerned with the system under study with fuzzy neural controller algorithm described above as the speed control strategy. So, the fuzzy logic parameters are adjusted to obtain an optimal output. Results under No-Load Condition

No-load results had been obtained by setting the load torque TL to zero to study the behavior of the system under study when the motor is unloaded. The motor torque can be described by $\frac{d\omega(t)}{dt} = \frac{K_m}{J} \cdot i_a(t) - \frac{B}{J}\omega(t) - \frac{T_L}{\omega}$. The results that will be discussed are the response, error signal, armature current, and armature torque. Note that the response figures contain the reference signal too. Results for Step Reference Input A step reference input of step instant 1-second is applied to the system under study. The system response, armature current, system error signal, motor torque is shown in fig.15. It is clear that the steady state error is zero, the rising time (tr), and the settling time for 5% criterion (ts) are: tr=34.2 msec ,ts=58.8msec, The peak overshoot is 1.8%. The error signal is shown in fig.15.c. It is obvious that the system response is satisfactory and has specifications better than that of fuzzy logic controller based system due to adaptation in controller parameters that make the overall output as close as possible to the desired output. The steady state error is zero because a PI-like fuzzy neural controller is utilized. The armature current and torque are shown in figures 15.b and 16.d. respectively. It is clear that figures 15.b and 15.d are of the same shape due to the fact that the torque is proportional to the armature current.

It is also plain that the starting current is too high due to high performance of the fuzzy neural controller, which may cause a fault in the motor operation. So, soft starting is normally utilized instead of sudden starting.

A square wave reference input is applied to the system to investigate the tracking of the system response to the reference input. The square wave has a 2-second period and 50% duty cycle. The system response, error signal, armature current, and motor torque are shown in figure 16

The square wave can be considered as duplication of step input with successive starts, step, and final times. It is apparent that the performance of the fuzzy neural controller forces the overall system response to be as close as possible to the reference input (see figure 16.a).

No-load step response had been obtained that has specifications of (see figure 16.a) with following specifications, rising time (tr) is 34.2 msec, settling time (ts) for 5%

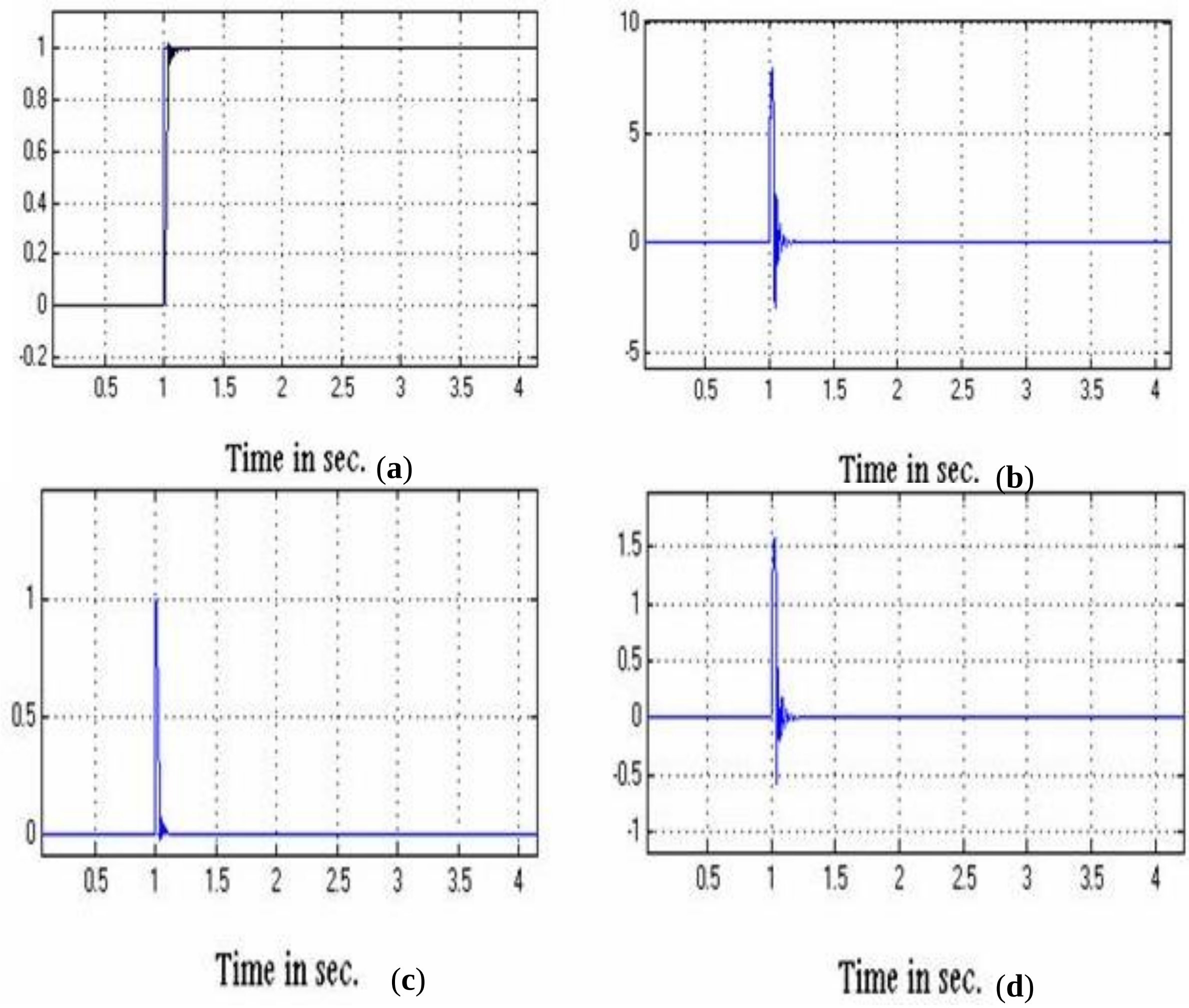


Figure (15) system results for step input with neural fuzzy controller: a. step response. b. armature current (A). c. normalized system error. d. torque (N.m).

It is clear that the specifications above are excellent for many practical applications such robot manipulators. At step instant, the response has a small oscillation around the required final value, because the integral term of the PI-like fuzzy neural controller, slightly, increases the response oscillation. On the other hand, the integral part suppress the steady state error (ess), this is why the PI is preferred on the PD-like fuzzy neural controller especially in cases of load insertion (as will be discussed later). Due to high value of armature current at step times, motor operation with high steps should be restricted to motors that withstand high armature current. Consequently, the maximum allowable armature current is the major factor that limits the step size of the reference input. Figure 17 shows the system results when the system reference input is a square wave. It is obvious that the results of the portions in which the signal is stepping from zero to one is the same as that of step input of figures 16. At the instants of transitions of the reference input from one to zero, the armature current has high values, due to the fact that when the reference input is one, the response, which is the speed, stabilized at one. Consequently, the back *emf* is also fixed to a value according to step(3). At the instant of transition from one to zero, the reference input becomes zero, whereas the back *emf* is not zero because, the speed is not zero due to existence of inertia. So, the armature current becomes too large in the opposite direction. So, the steps of transition from higher value to another lower one should be also taken into consideration and not exceed the allowable step size.

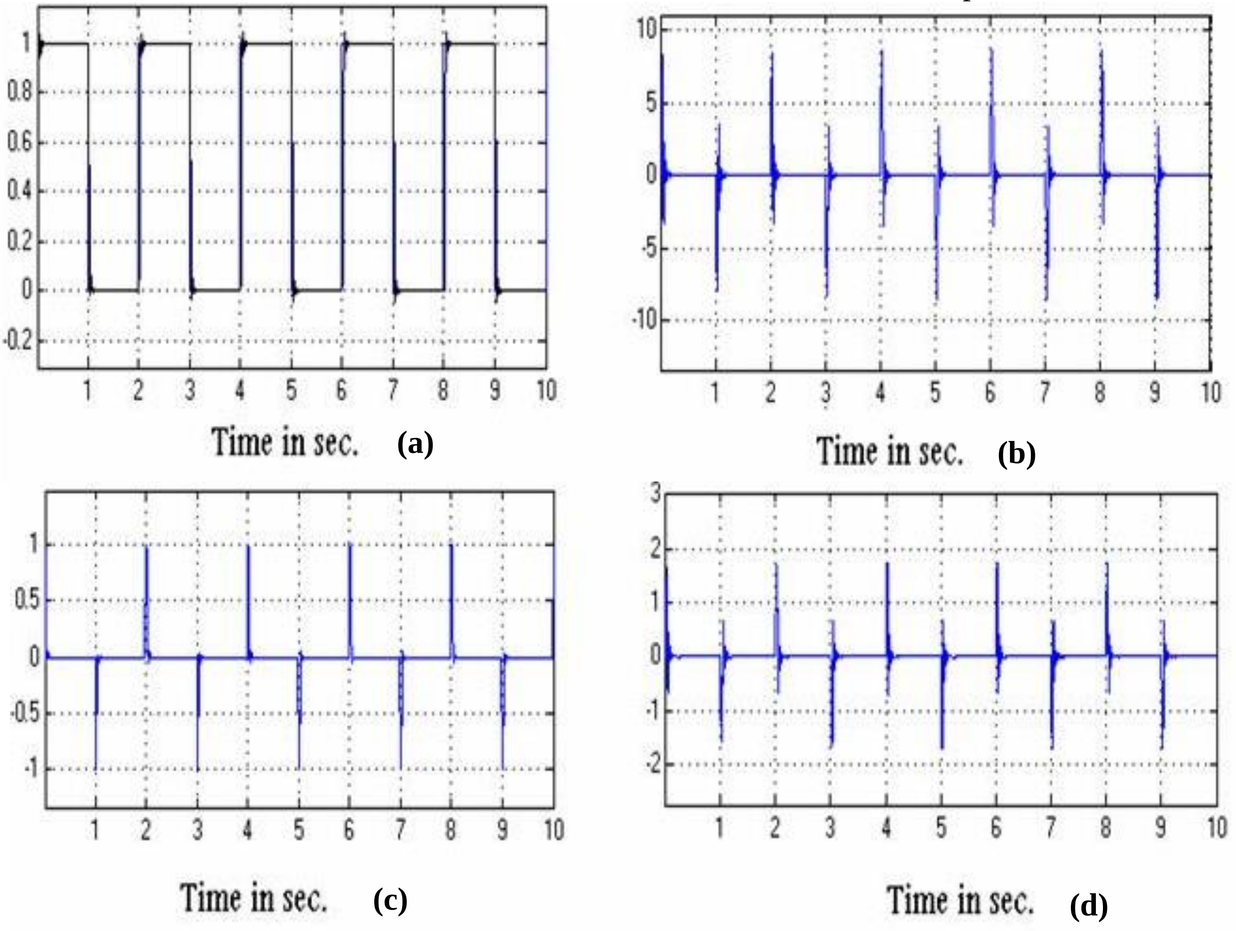


Figure (16) system results for a square wave reference input with neural fuzzy controller: a. system response. b. armature current (A).c. normalized system error signal. d. torque (N.m).

As mentioned previously, fuzzy controllers have the great ability to control systems that its mathematical model is unknown. Exactly, the system general behavior is required. However, the fuzzy logic parameters, i.e. the membership functions and consequent parts parameters, have the great effect on the performance of a fuzzy logic controller. So, when the fuzzy logic parameters are initialized to give a certain performance to a plant, the performance would not remain when the plant is varied with time. Figure (16) shows the system when a step load is applied. It is apparent that the time required for the response to return to its final value is relatively large, of about 1-second. The rising time is also relatively large, of 430 msec. On the other hand, a fuzzy neural controller has the great ability to adjust the fuzzy logic parameters to give an optimal output, i.e. to make the plant output as close as possible to the reference input. Small rising time, of 34.2 msec, had been obtained when using fuzzy neural controller as a speed controller. The time required to return to the original fixed value, when load is applied, is also too small which make the system more robust against unknown load applications. It is obvious that the starting current in the case of fuzzy logic controller is smaller than the corresponding value when a fuzzy neural controller is utilized, because of poor step response of fuzzy logic controller and satisfactory step response of fuzzy neural controller .

In many applications, motor load variations must not be neglected; the conventional (PD) controller cannot take into account this variation that results in a poor control scheme. The cause of poorness of conventional PD controller is that during the controller design, only the parameters of the case under study are to be considered. Another drawback on the conventional controller is that the exact plant mathematical model is required. Fuzzy logic controllers can overcome the problem of knowledge of the exact mathematical model of the plant to be controlled via the utilization the expertise of a human operator to control the plant. So, only the general behavior of the plant to be controlled is required. However, the fuzzy logic controllers have a performance limitation of constant fuzzy inference parameters. So, the fuzzy neural controller can perform the job perfectly due to the fact that in case of load variation, also in cases of variable reference input, the controller parameters are adjusted to make the plant output as near as possible to the reference input, which would make the overall system more robust. The powerful gradient descent is utilized as an optimization method to produce an optimal output. Rising and settling times, and peak over shoot are highly reduced, which result in good tracking behavior. Good tracking behaviors had been obtained under the insertion of various loads, which would reinforce the utility of fuzzy neural controller for even unknown load characteristics. Due to the high performance of the controller under study, the system error signal e(t) is highly reduced. Armature current and torque measurement also had been considered. High starting currents, in cases of high input values, may be avoided by reducing the step size, i.e. using the soft starting, which would result in larger time to get to the final value of the reference input step and sine wave as shown in fig. 17. However, many applications require too small time to get to the final value of the input. So, soft starting is avoided in such cases and high armature current motors are used. Other types of applications do not require small time to get to the final value. So, soft starting is important, in this case, to make the motor operating with safety even with relative small armature current withstanding.

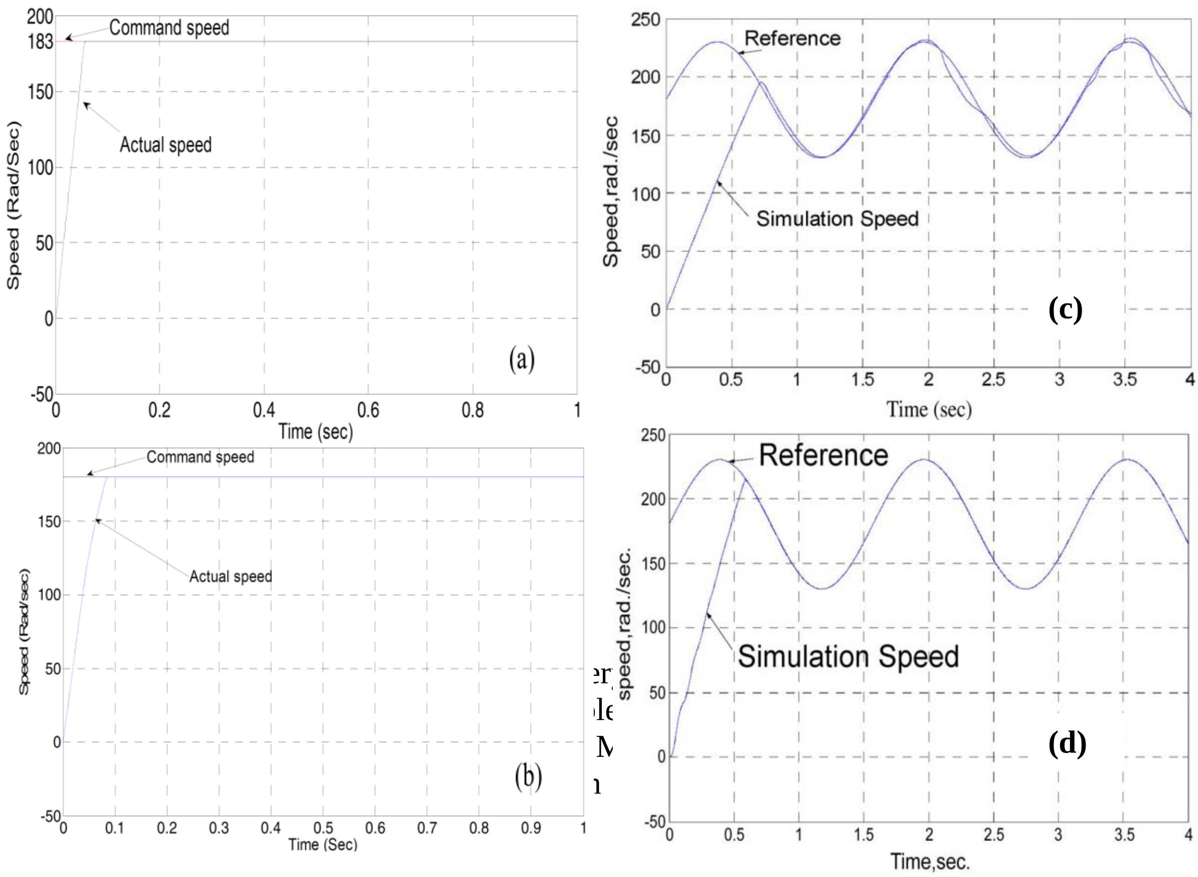


Figure (17) system simulation speed with: a. neural fuzzy controller system step response. b. fuzzy logic step response .c. neural fuzzy controller system sine response. d. fuzzy logic sine response.

butter response had been produced but with increasing in epochs, so it is better to use try and error method to get the best learning rate. However, number of hidden nodes in MRNN is essential in designing a NN since the larger number of hidden nodes led to simulation of complex nonlinear functions, but this argument will lead to sufficient increasing in calculation time of MRNN and hence will effect on the ability be used in the real time control systems. MRNN is very important, because it can used along with FLC to produce a precise rule table ,hence FLC was connected with MRNN after learning and try and error procedure is implemented to choose the best rules for FLC ,i.e. the rules of FLC cold be changed safely without destroying plant environments. The FLC could control SS and crop the changes of the input very fast (within 30 samples) and with overshoot less than 30% which is acceptable in most cases and if disturbance exist in the control system, FLC will over comes this and continue working. The new form of fuzzy rules and FNN employs the new form of fuzzy rules with linear state-space equations as the consequences. The linear state-space equations are convenient to human to represent dynamic systems, and in common sense such representation can grasp the inherent essential of the system. Natural and simple as the fuzzy set and fuzzy inference mechanism are to define the different aspects of a complex system, the nonlinearity of membership functions enables the network to simulate a nonlinear system well. In the simulation of the pendulum system, FNN shows a very strong generalization on different initial states and force inputs.

References

Ahmed R., Abdul R. Ofoli and Donatus C. 2007 “DSP-Based Real-Time Implementation of a Hybrid *H∞* Adaptive Fuzzy Tracking Controller for Servo-Motor Drives” IEEE Transactions On Industry Applications, VOL. 43, NO. 2, march/april.

Bimal K. Bose 2006 “Power Electronics and Motor Drives Advances and Trends” Elsevier Inc.

Cheng Qi-ming, 2007"The Control System of Multi-variable PID Neural Network and Its Application in Deaerator Water Level Control", Journal of Shanghai University of Electric Power, Vol.23, No.1, pp.33-38, January.

Crockett, K , Bandar, Z. Fowdar, J, O'Shea, 2006 “On Producing Balanced Fuzzy Decision Tree Classifiers", IEEE World Congress in Computational Intelligence 2006, Vancouver, Canada. IEEE, pp 8415- 8421.

Dongmei Yu, Qing Hu and Fei Zhao,2009 " Synchronous Motion Control of Biaxial Driving System with Linear Servo Motors", Proceedings of the IEEE International Conference on Automation and Logistics Shenyang, China.

Feng G., 2006 “A Survey on analysis and design of model-based fuzzy control systems,” *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 5, pp. 676–697, Oct.

Golob M. 2001"Decomposed Fuzzy Proportional-Integral-Derivative Controllers", Applied Soft Computing: Engineering Design and Manufacturing, Maribor, Slovenia. http://www.au.feri.uni-mb.si/marjan/

Grabowski P. Z., Kazmierkowski M. PBose., B. K., and F. Blaabjerg, 2000 “A simple direct-torque neuro-fuzzy control of PWM-inverter-fed induction motor drive,” *IEEE Trans. Ind. Electron.*, vol. 47, pp. 863–870, August.

Mohan B.M., Sinha A., (2008) “Analytical structures for fuzzy PID controllers”, IEEE Trans. Fuzzy Syst. 16 (1) 52–60.

Nicholas B., 2008 “A Simple Multi-Chromosome Genetic Algorithm Optimization of a Proportional-plus-Derivative Fuzzy Logic Controller” *IEEE Transactions on Fuzzy Systems*, vol. 16, pp. 611-616.

Paulo G., Jorge H., Dourado A. and Ramos H. D., 2000"Application of A Recurrent Neural Network in on-line Modeling of Real Time System" CISUC Informatics Engineering Department, Coimbra –Portugal.

Pham D. T. and X. Liu, 1997 "Neural Networks for Identification, Prediction and Control", Springer-Verlag, New York.

Reznik L., 1997 "Fuzzy Controllers", Victoria University of Technology, Melbourne, Australia Newnes,.

Radu E. P., Stefan P., EmilM P., Jozsef K. T., Marius L. T., Claudiu P. 2009 “Generic two-degree-of-freedom linear and fuzzy controllers for integral processes” Journal of the Franklin Institute, Published by Elsevier Ltd.

Shahnazi R., Shanechi H.M,. Pariz N, (2008) “Position control of induction and DC servomotors: a novel adaptive fuzzy PI sliding mode control”, IEEE Trans. Energy Conv. 23 (1) 138–147.

Vengerov, D. Bambos, N. Berenji, H 2005 “A fuzzy reinforcement learning approach to power control in wireless transmitters”. IEEE Transactions on Systems, Man, and Cybernetics, Part B 35(4): 768-778.

Zurada J. M, 1996., "Introduction to Artificial Neural Systems", Jaico Publishing House, 2nd edition. India.