Comparison of Soft-Computing and Conventional Methodologies in Control of Servo Systems

M. Önder Efe
Boğaziçi University
Electrical and Electronics Eng.
80815 Bebek, Istanbul, Turkey
efemond@boun.edu.tr

A. Buğra Koku,
Boğaziçi University
Systems and Control Eng.
80815 Bebek, Istanbul, Turkey
koku@boun.edu.tr

Okyay Kaynak
UNESCO Chair on Mechatronics
Boğaziçi University
80815 Bebek, Istanbul, Turkey
kaynak@boun.edu.tr

***Abstract* - In this study, the performance of soft-computing methodologies and the conventional controllers are compared for the control of a servo system. A number of approaches in both domains are simulated and their performances are assessed in terms of several performance measures. It is seen that the use of soft computing methodologies result in some desirable characteristics.**

I. INTRODUCTION

This study investigates the soft-computing techniques and the learning algorithms prescribed by these techniques in comparison with the conventional design methodologies from the system control point of view.

In many control applications soft-computing algorithms seem to be preferable when they are compared to the traditional design methodologies. What make them so attractive are the following: Real systems are, in general, complex and have nonlinear structure. Therefore, understanding the operation of a system and estimating its future behavior emerge as formidable problems when traditional methods based on analytical techniques are used. In fact, the problem stems from the intricacy in the definition of the physical problem and the categoricity of the computational environment.

When the control problem in hand is considered with structural or unstructural uncertainties, nonlinearities and demanding control objectives, the degree of freedom in the analytical design domain decreases. This clearly entails an order of adaptiveness which takes care of the above mentioned difficulties. Recent studies strongly stipulated that the learning controllers are the best candidates when the analytical design is tedious or is not satisfactory enough. In other words, if the control system possesses an intelligent controller, it may become easier to alleviate the adverse effects of time-varying behavior, noisy measurements or unmodeled dynamics. Narendra and Parthasarathy [1] have introduced four plant models and systematically analyzed the identification and control of systems by utilizing neural networks. The model adopted by them for a neuron is the conventional one. On the other hand, Gupta and Rao [2] have considered the same problem with a dynamical neuron model which has a memory and which can easily handle the demanding objectives of the control problem by exploiting the form of the past history of the outputs of the plant. Another way of injecting intelligence into a control mechanism is the use of Fuzzy Logic which was originally proposed by Zadeh in early 1960s. In the literature, many successful approaches can be found which describe the integration of fuzzy logic in various control schemes, especially in manipulator control. Some of these are based on inverse dynamics obtained by the use of fuzzy identifiers [3]. This is also the approach adopted in this work.

In this paper, a cart driven on a linear track by a d.c. servo motor is used as the test bed. In conventional methodologies, estimated plant parameters are used in the design of the controller, whereas, in the soft-computing methodologies, the plant parameters are assumed to be unknown.

In the next section, the servo model used in simulations is introduced. The following section explains the details of the two conventional control techniques used, namely, PD and LQ controls. Section IV is devoted to Soft Computing Methodologies. Dynamical Neural Units (DNU) and Fuzzy Logic controller architectures are explained. Finally, the comparison of these methods are assessed with respect to several performance criteria based on the given simulation results.

II. MOTOR MODEL

As is mentioned earlier, a cart driven on a linear track by a d.c. servo motor is used in the simulations. The general form of the model for this type of a system is given by (1).

$$\frac{x(s)}{u(s)} = \frac{1}{s(\alpha s + \beta)} \qquad (1)$$

In (1), x(s) is the position of the cart, and u(s) is the voltage applied to the d.c. motor (control input) in volts.

The plant model seems simple but in real implementations, uncertainty in plant parameters, and the effect of friction are likely to make the design of a controller difficult.

III. CONVENTIONAL CONTROL STRATEGIES

If the plant and the performance specifications are defined well, it is relatively a simple matter to design a suitable controller. In this study, the following parameters are considered as the true parameter values of the servo system.

0-7803-4503-7/98/$10.00 1998 IEEE

$$[\alpha \quad \beta] = [0.2600 \quad 4.4700] \tag{2}$$

The classical controllers are designed based on these values, whereas the simulations are carried out by the use of parameters given by (3).

$$[\alpha \quad \beta] = [0.30 \quad 5.00] \tag{3}$$

In the next subsections, a PD and an LQ control scheme for the linear servo system are developed, a "model-plant mismatch" of about 10-15% is therefore introduced using the nominal values given by (2).

A. PD Control Strategy

A proportional and derivative controller is able to achieve a critically damped closed-loop behavior in response to a step input. If the control given by (4) is applied to the system, closed loop transfer function is obtained as given by (5).

$$u(t) = K_p(x_d(t) - x(t)) - K_d\dot{x}(t) \tag{4}$$

$$\frac{x(s)}{x_d(s)} = \frac{K_p}{\alpha s^2 + (\beta + K_d)s + K_p} \tag{5}$$

where K_p, K_d are proportional and derivative gains respectively and x_d is the desired position trajectory. In order to obtain a critically damped response to a step input with a peak time of 0.5 seconds, the PD gains are set as follows:

$$[K_p \quad K_d] = [8\pi^2\alpha \quad 4\pi\alpha - \beta] \tag{6}$$

The desired position trajectory is illustrated in Fig. 1, the tracking error and the control signal graphs for PD control scheme are given in Figs. 2 and 3 respectively.

B. Linear Quadratic Controller

This section concerns the design of a linear quadratic controller. The reason why this method is used here is the fact that the LQ regulator design methodology allows the designer to assign different weights to tracking ability and the control effort. This has a practical importance because the boundaries of the admissible controls are determined by physical limitations.

The motor model can be expressed in state-space form as follows;

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{7}$$

where;

$$A = \begin{bmatrix} -\beta/\alpha & 0 \\ 1 & 0 \end{bmatrix},\ B = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ C = [0 \quad 1/\alpha] \tag{8}$$

$$J = \frac{1}{2}\int_{to}^{tf}\left(\|x - r\|_Q^2 + \|u\|_R^2\right)dt \tag{9}$$

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1e5 \end{bmatrix},\ R = 10 \tag{10}$$

Minimization of the cost function in (9) with the weights given in (10) yields the following control which is obtained from the solution of the Riccatti equation for the steady state case.

$$u = -Kx + Sr \tag{11}$$

where;

$$K = [5.0715\ 100] \text{ and } S = 100 \tag{12}$$

The simulation results for LQR design are given in Figs. 4 and 5.

IV. SOFT COMPUTING METHODOLOGIES

Recent studies have shown that the use of a controller that utilizes neural networks and fuzzy logic can be attractive for several reasons. Firstly, the adaptive behavior of these approaches leads to the learning of dynamical properties of the system under control. Consequently, better fulfillment of the performance specifications can be achieved.

Fuzzy logic processes the data obtained from the control system operating on linguistic variables, whereas in the neural network approach, the data processing is carried out on the physical values of the system parameters. The most striking aspect of these methods is that the former reflects the experience of an expert and the latter imitates the human brain activity. In both cases a mathematically tractable system model and an admissible controller are sought.

In this section, two different approaches for the control of the DC motor drive introduced in the second section are presented.

A. Dynamical Neural Networks for Control

Dynamical neural networks are composed of dynamical neural units (DNU) which possess a second order discrete system and an output sigmoidal nonlinearity. The neuron model is comprised of synaptic and somatic parts and adaptation is carried out on the coefficients of this second order block (synaptic part) and the on the slope of its nonlinear activation function (somatic part).

The topology of a single dynamical neural unit consists of delay elements, feedforward and feedback synaptic weights and a nonlinear somatic operator. The architecture of the DNU model is illustrated in Fig 6. The difference equation which describes the behavior of the second order dynamical structure is given in (13) in which $v_1(k)$, $x(k) \in R^1$. Similarly, the pulse transfer function of this part can be given by (14).

$$v_1(k) = - b_1v_1(k-1) - b_2v_1(k-2) + a_0x(k) + a_1x(k-1) + a_2x(k-2) \quad (13)$$

$$T(z) = \frac{a_0 + a_1z^{-1} + a_2z^{-2}}{1 + b_1z^{-1} + b_2z^{-2}} \quad (14)$$

The output of the DNU is evaluated as follows;

$$v(k) = g_s v_1(k) \quad (15)$$

$$u(k) = \Psi(v(k)) = \tanh(g_s v_1(k)) \quad (16)$$

The control objective is based on the minimization of the instantaneous error evaluated at the output of the system. The cost function which is to be minimized is defined by (18) in which E denotes the expectation operator. If w denotes the parameters of a single DNU, the gradient descent based update rule can be given by:

$$w_{k+1} = w_k - \mu \frac{\partial J}{\partial w} \quad (17)$$

$$J = \frac{1}{2} E\left(e^2(k)\right) \quad (18)$$

$$e(k) = y_d(k) - u(k) \quad (19)$$

More compactly, the parameter update rule is given by (20) through (22) where i = 0,1,2 and j = 1,2 ;

$$\Delta a_i(k+1) = \mu_{a_i} E\left(e(k)\, g_s^2(k)\, \mathrm{sech}^2[v_1(k)]\, x(k-i)\right) \quad (20)$$

$$\Delta b_j(k+1) = -\mu_{b_j} E\left(e(k)\, g_s^2(k)\, \mathrm{sech}^2[v_1(k)]\, v_1(k-j)\right) \quad (21)$$

$$\Delta g_s(k+1) = g_s(k)\mu_{g_s} E\left(e(k)\, \mathrm{sech}^2[v_1(k)]\, v_1(k)\right) \quad (22)$$

In the parameter update equations, the coefficients μ_{ai}, μ_{bj} and μ_{gs}, denote the step size for the corresponding parameter and are chosen to be constant throughout a simulation. In Fig. 7, DNU layer includes the desired number of individual DNU blocks whose inputs are connected together and whose outputs are added to form the control u(k). Depending on the magnitude of the output error, the algorithm updates the feedforward and feedback weights and the gain of the nonlinear activation function of each dynamical neural unit in the DNU layer. The derivation of the algorithm is given in [2] in detail. In the simulations, three DNUs are used and the results are presented in Figs. 8 and 9.

B. Fuzzy Control of Electrical Drives

Fuzzy logic based systems are well known for their property of reflecting the expert knowledge to a system. In terms of control, this is done via exporting our intuitive feeling or experience into the controller.

Fuzzy logic has been proved to be very convenient for designing controllers where the designer has some experience or information about the plant to be controlled. However, the major problem of parameter tuning is still there, due to the fact that, the conversion of the qualitative information into a quantitative control action is not very easy.

In this paper, a fuzzy logic identifier is used, which is similar to a neural network structure. To overcome the above mentioned problem of parameter tuning, the well known backpropagation algorithm is utilized.

A fuzzy logic system, with product inference rule, singleton fuzzifier and Gaussian membership function given in (23) can be represented as a three layer feedforward network and can serve as a universal approximator [4],

$$f(\underline{x}) = \frac{\sum_{k=1}^{M} \bar{y}^k \left(\prod_{i=1}^{n} \mu_{F_i^k}(x_i) \right)}{\sum_{k=1}^{M} \left(\prod_{i=1}^{n} \mu_{F_i^k}(x_i) \right)} \quad (23)$$

where, k=1,2,...,M and i=1,2,...,n. $\underline{x}$ is the input vector of size n and $\bar{y}$ is a vector of size M containing the center values of the center average defuzzifier. There are M rules governing this fuzzy logic system. Finally, $\mu_{F_i^k}$ is the Gaussian membership function for the i^{th} input in the k^{th} rule.

The above mentioned three layer feedforward representation of the fuzzy logic system is illustrated in Fig. 10. This method of representing the fuzzy system enables us to use the gradient descent methodology to update the two parameters c_i^k and σ_i^k of the Gaussian and the centers of defuzzifier vector $\bar{y}$.

Let the pair $(\underline{x}^D, f^D)$ represent a desired input output pair. Then, the error can be defined as:

$$e = \frac{1}{2}\left[f(\underline{x}^D) - f^D\right]^2 \quad (24)$$

The error backpropagation results in the following update procedure:

$$\bar{y}^k(t+1) = \bar{y}^k(t) - \alpha \left.\frac{\partial e}{\partial \bar{y}^k}\right|_k = \bar{y}^k(t) - \alpha \frac{f - f^D}{b} z^k \quad (25)$$

$$\underline{x}_i^k(t+1) = \underline{x}_i^k(t) - \alpha \left.\frac{\partial e}{\partial \underline{x}_i^k}\right|_k$$

$$\underline{x}_i^k(t+1) = \underline{x}_i^k(t) - \alpha \frac{f - f^D}{b}(\bar{y}^k - f) z^k \frac{2(x_i^D - \bar{x}_i^k(t))}{\sigma_i^{k^2}(t)} \quad (26)$$

$$\sigma_i^k(t+1) = \sigma_i^k(t) - \alpha \left. \frac{\partial e}{\partial \underline{x}_i^k} \right|_k$$

$$\sigma_i^k(t+1) = \sigma_i^k(t) - \alpha \frac{f - f^D}{b} (\bar{y}^k - f) z^k \frac{2(x_i^D - \bar{x}_i^k(t))^2}{{\sigma_i^k}^3(t)} \qquad (27)$$

where $t=0,1,\ldots$ and α is the learning rate.

The control system is designed to behave as an inverse plant as shown in Fig. 11, and the simulation results are given in the Figs. 12 and 13. As can be seen, the control signal generated by the fuzzy logic control strategy has a lot of chatter, due to the discontinuities in the reference signal, requiring infinite acceleration. In an actual implementation, this should not pose a difficulty because this signal is the voltage input to the d.c. motor and it will be smoothed out by the actuator dynamics.

V. CONCLUSIONS

In order to evaluate the capabilities of the four control schemes described above, a series of simulations have been performed. In all studies, a "plant-model mismatch" is introduced as stated previously.

A comparison of the simulation results is given in Table 1. Each row of this table corresponds to a different performance objective frequently encountered in control engineering. The first row evaluates the most visible measure; the tracking performance. All control schemes are capable of achieving high degree of tracking performance, except the PD scheme. As mentioned earlier, the philosophy of the LQC design is based on the relative importance of low control effort or high output tracking ability but not the both. In this study the tracking performance is considered as the major objective for LQC design without violating the bounds of controls. The second row is responsible for the applicability of the approaches considered to different types of plants. Soft computing methodologies are comparably successful in this sense because they require less a priori knowledge about the plant under control. Another comparison metric is the robustness of the approaches under perturbations. If there is a high degree of plant model mismatch, conventional methodologies collapse due to ambiguities in the values of the plant parameters. The fourth row explains the performance under noisy observations. In the simulations, plant outputs are corrupted by Gaussian random noise which degrades the true outputs up to 10% of their magnitudes. Fuzzy logic controller has shown the best performance in the case of noisy observations. The last row considers the fault tolerance capability. If the plant parameters are changed abruptly, the best compensation is observed in dynamical neural control strategy. This is due to the fact that the scheme operates on-line learning strategy with highly parallel, memory oriented nonlinear structure.

Based on the comparison measures discussed above, it can be stated that the performance of soft-computing methodologies seem to have more desirable characteristics.

It is obvious that PD and LQ control strategies are strictly dependent on the system parameters and cannot compensate the deficiencies caused either by poor parameter estimation or by time varying parameters. This can be observed in Figs 2. and 4. There is no tendency to reduce the tracking error in time. However, this is not the case for the controllers designed by the use of Soft Computing Methodologies.

It is also to be noted that, learning, which is generally introduced as a very desirable property, has a transient phase which may result in instantaneously large controls or a period of ringing in the control signal.

Work is in progress having the goal of improving the transient response of the controller by means of adapting the learning strategy depending on the past history of the controls and system response.

TABLE I
COMPARISON OF CONTROL STRATEGIES

	PD	LQC	DNU	FLC
Tracking performance	M	H	H	H
Applicability to different plants	L	M	H	H
Robustness under perturbations	L	L	M	M
Noise reduction	M	L	M	H
Capability of fault tolerance	L	M	H	M

(H: High, M: Medium, L: Low)

REFERENCES

[1] Narendra, K. S., Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, March 1990, pp. 4-27.

[2] Gupta, M. M., Rao, D. H., "Dynamical Neural Units with Applications to the Control of Unknown Nonlinear Systems", *Journal of Intelligent and Fuzzy Systems*, vol. 1, no. 1 1993, pp. 73-92.

[3] Erbatur K., Kaynak, O., Rudas I., "An Inverse Dynamics Based Robot Control Method Using Fuzzy Identifiers", in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 16-20 June, 1997, Tokyo, Japan

[4] Wang, L., *Adaptive Fuzzy Systems and Control*, Design and Stability Analysis, PTR Prentice Hall, 1994, pp. 29-31.

[5] Zadeh, L., A., "Fuzzy Sets", *Information and Control*, Vol. 8, 1965

[6] Efe, M. Ö., "Identification and Control of Nonlinear Dynamical Systems Using Neural Networks," M.S. Thesis, Boğaziçi University, 1996

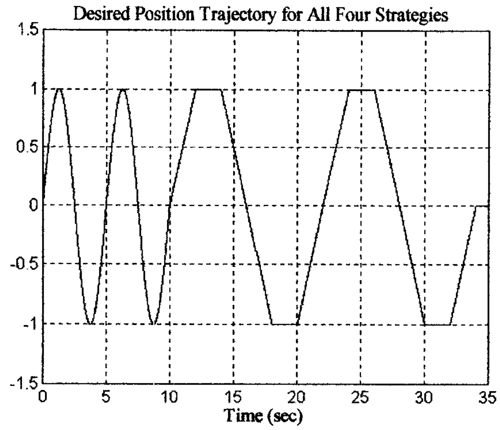

Figure 1.Desired Position Trajectory

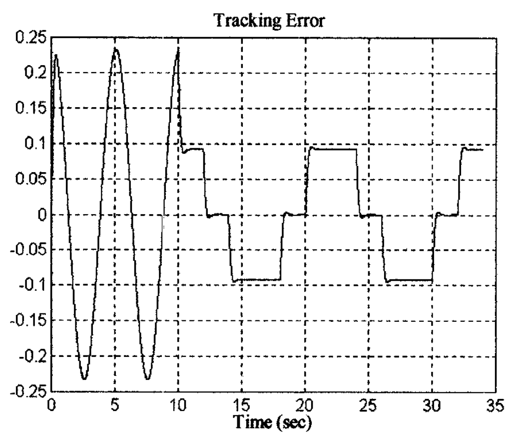

Figure 2.Tracking Error for PD Control Strategy

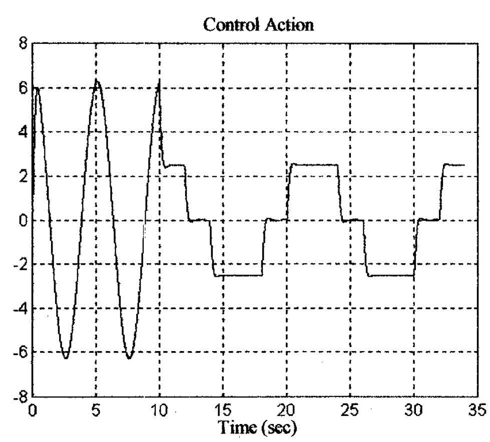

Figure 3.Control Signal Generated by PD Control Strategy

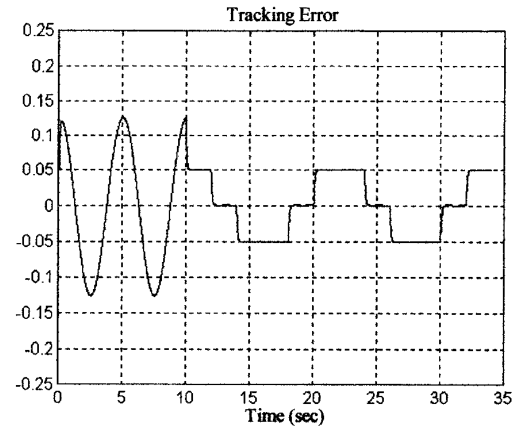

Figure 4.Tracking Error for LQ Control Strategy

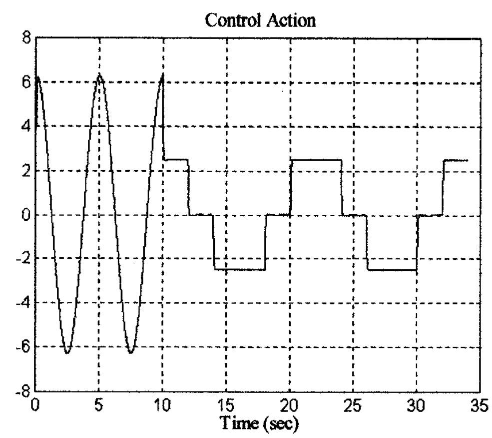

Figure 5.Control Signal Generated by LQ Control Strategy

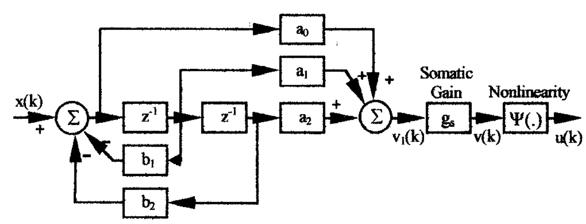

Figure 6. Structure of Dynamical Neural Unit

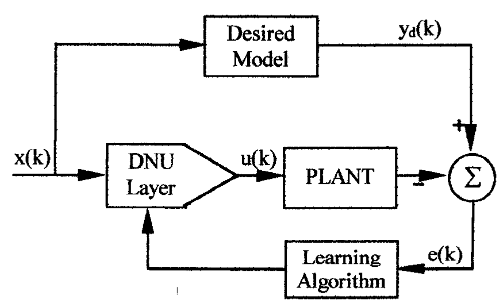

Figure 7. Control System Architecture with Dynamical Neural Unit based Controller

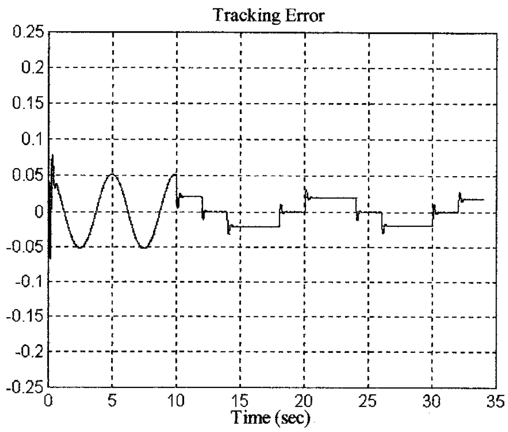

Figure 8.Tracking Error for DNU Control Strategy

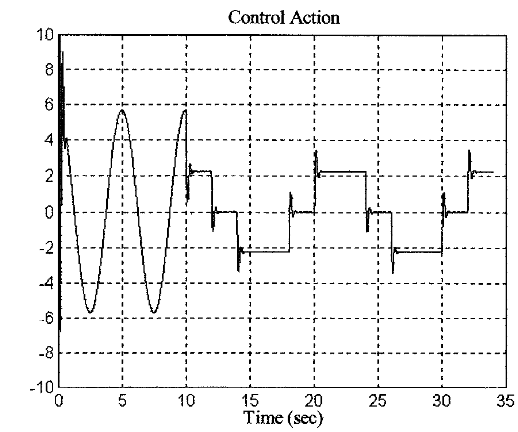

Figure 9.Control Signal Generated by DNU Control Strategy

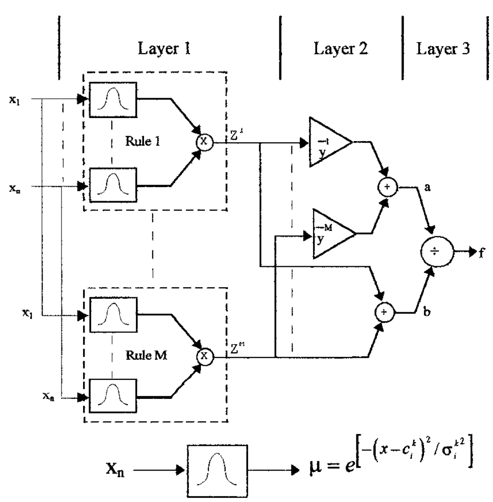

Figure 10. The 3 Layer Network Representation of the Fuzzy Logic System.

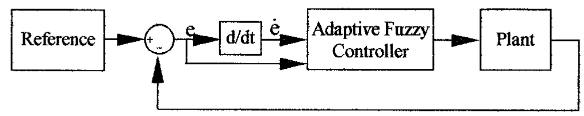

Figure 11. Architecture of Fuzzy Control Strategy

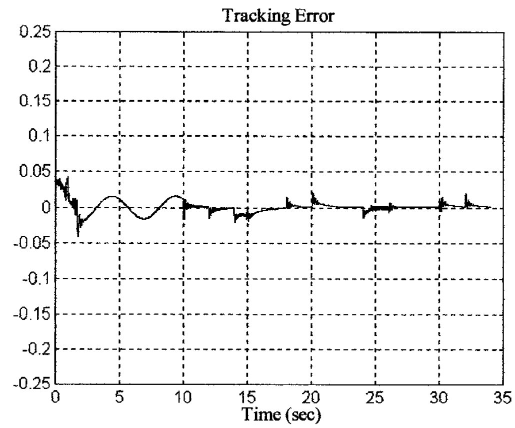

Figure 12.Tracking Error for DNU Control Strategy

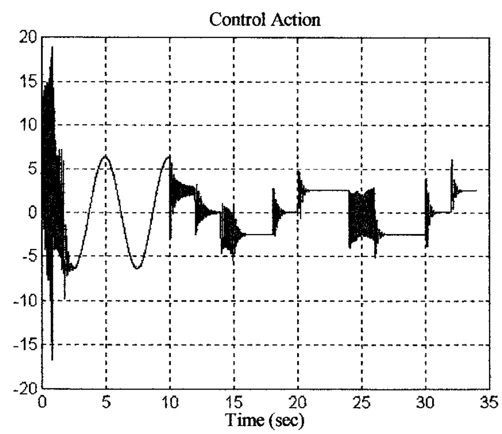

Figure 13. Control Signal Generated by Fuzzy Logic Control Strategy