12/09/98 13 SEL

ACC

1997 American Control Conference

June 4-6, 1997

Albuquerque, New Mexico

NMU

2

Welcome

Getting Started

Conference Information

Sessions

Authors

Proceedings of the American Control Conference
Albuquerque, New Mexico June 1997
0-7803-3832-4/97/$10.00 © 1997 AACC

Reinforcement Learning Schemes for Control Design

E. F. Costa, R. Tinós, V. A. Oliveira and A. F. R. Araújo
Departamento de Engenharia Elétrica - USP/EESC
Caixa Postal 359, 13560-970, São Carlos, SP, Brazil
eduardoc@sel.eesc.sc.usp.br

ABSTRACT

In this paper we consider the use of associative search and adaptive critic elements and artificial neural network for control of non-linear and unstable plants. The reinforcement learning schemes we propose are used in the design of different controllers. An example of a magnetic suspension system is presented to illustrate the effectiveness of these controllers. We also include results of a linear optimal controller.

1. INTRODUCTION

Reinforcement learning addresses the problem of on-line learning using a measure of plant performance [1], [2]. This involves determining a function through experience that maps current states of a plant into control actions and constructing a critic to judge which control actions are acceptable (success) and which are not (failure).

The reinforcement learning schemes proposed are based on the Adaptive Critic algorithm by Barto, Sutton and Anderson [1] which is modified in different ways in order to improve learning performance. These schemes have led to the development of three controllers and incorporate: (a) weight update direction sign; (b) failure instant evaluation and; (c) a multilayer perceptron neural network to perform the controller.

The problem of interest here is to construct a control system capable of driving to and holding around an equilibrium point the plant, allowing it an extended operating range.

2. PROBLEM STATEMENT

Conditions that assert the solution of control problems of non-linear systems are not available, except in local terms, as in the neighborhood of a point [3]. Also, design procedures work well only if a number of conditions are satisfied.

Besides, available design procedures generally do not consider frequent practical constraints on the system, such as the limits of the control signals.

We consider a non-linear plant of the form

$$\mathbf{x}(k+1) = \mathbf{f}[k, \mathbf{x}(k), \mathbf{u}(k)] \quad (1)$$

where $\mathbf{x}$ denotes the n-dimensional state vector, $\mathbf{u}$ is the m-dimensional input vector, k denotes discrete time and $\mathbf{f}$ is a function. The n-dimensional linear space over $\Re$ is called the space state Ψ.

In this paper, it will be considered that a state $\mathbf{x}_0$ is an equilibrium point of the system (1) if $\mathbf{f}(k,\mathbf{x}_0,\mathbf{u}_0)=0$ for a given $\mathbf{u}_0$. By definition, a non-linear system is autonomous if $\mathbf{f}$ does not depend on its first argument, k [4].

This paper deals with the class of problem: stabilize and keep a dynamic nonlinear system around an equilibrium point considering constraints on the amplitudes of the control signal.

The use of reinforcement learning provides an interesting alternative to the control of systems. Some of its advantages are the no need of plant parameters or plant model [5], and the facility of incorporating practical constraints in the controller design. As a result, this technique can be applied to a large class of systems.

3. REINFORCEMENT LEARNING CONTROL SCHEMES

The strategy of the control schemes based on Barto, Sutton, Anderson [1] algorithm is as follows. The control problem is split into sub-problems by dividing the space state Ψ in several partitions. Each partition is associated to a specific control signal by adaptive elements. These elements have their weights adjusted by a reinforcement signal, a qualitative measure of plant performance. The adaptive elements are the called adaptive search element (ASE) and associative critic element (ACE).

A decoder is used to partition Ψ. Its output d_i is equal to one when the system is in partition i and zero elsewhere, for $i=1,\ldots,i_T$ where i_T is the total of partitions.

The ASE output $u(k)$ (2) is the control signal that is sent to the plant.

$$u(k) = f\left[\sum_{i=1}^{i_T} w_i(k)d_i(k) + n_s(k)\right] \quad (2)$$

where k is the discrete time, w_i is the weight associated with the partition i, n_s is a real random variable with probability density function h, and f is either a threshold, sigmoid or identity function.

The update weight equation is written as

$$w_i(k+1) = w_i(k) + \alpha\hat{r}(k)e_i(k) \quad (3)$$

where α is the learning rate, $\hat{r}$ is the internal reinforcement signal, described bellow, and e_i is the eligibility term, which associates the responsibility of the actions of the partition i to the failure occurrence.

The eligibility is given by

$$e_i(k+1) = \delta\, e_i(k) + (1-\delta)\, d_i(k) \tag{4}$$

where δ is a decay rate.

The internal reinforcement signal is provided by the ACE. The ACE objective is to predict a failure before it happens, allowing antecipative weight adjust.

In this paper, the equations of the ACE are the same as in Barto, Sutton, and Anderson [1] and are presented here for easy reference.

$$p(k) = \sum_{i=1}^{i_T} v_i(k)\, d_i(k) \tag{5}$$

$$\hat{r}(k) = r(k) + \gamma\, p(k) - p(k-1) \tag{6}$$

$$v_i(k+1) = v_i(k) + \beta\, \hat{r}\, \bar{d}_i(k) \tag{7}$$

$$\bar{d}_i(k+1) = \lambda\, \bar{d}_i(k) + (1-\lambda)\, d_i(k) \tag{8}$$

where $p(k)$ is the output of the ACE, $v_i(k)$ is the weight associated with partition i, $\bar{d}_i$ is analogous to the eligibility for the ACE, r is the external reinforcement signal responsible for adjusting the weights or not, γ and λ are real valued constants and β is the critic learning rate.

The control scheme described in (2)-(8) which we call Reinforcement Learning Control Scheme (RLCS) incorporates the sign (positive or negative) of weight update direction into the external reinforcement criterion. This means that the plant output has to be analyzed to provide an appropriate criterion.

The steps of the learning algorithm are as follows:

```
1. Initialize the ASE and ACE weights;
2. Initialize the state of the dynamic system; set $t=1$;
3. While $t < t_{max}$, where $t_{max}$ is the total of trials defined, do:
    3.1. $t=t+1$, $k=0$, $r=0$;
    3.2. While $r=0$ and $k < k_{max}$, do:
        3.2.1. $k=k+1$;
        3.2.2. Generate $d_i(k)$, $i=1,\dots,i_T$, by decodifying the actual system state vector $x_j(k)$, $j=1,\dots,n$;
        3.2.3. Calculate the element outputs $u(k)$ (2) and $p(k)$ (5);
        3.2.4. Present $u(k)$ to the system and calculate $x_j(k+1)$, $j=1,\dots,n$ (1);
        3.2.5. Calculate $e_i(k+1)$ (4) and $\bar{d}_i(k+1)$ (8), $i=1,\dots,i_T$;
        3.2.6. Determine $r(k+1)$ using a defined criterion and $\hat{r}(k+1)$ (6);
        3.2.7. Adjust the ASE weights $w_i(k+1)$ (3) and the ACE weights $v_i(k+1)$ (7);
    3.3. If $k = k_{max}$, where $k_{max}$ is the total of sampling periods define, finish the algorithm.
```

3.1 Control with Failure Instant Evaluation

The previous control scheme may be modified in order to have a more generic algorithm without the need to analyze the plant output. Thus, the question that arises is how to determine the sign of the weight adjust only by observing its effects on the system performance.

Let us consider that if the failure time instant in the current trial is greater than the previous one the sign is correct, otherwise the sign must change. Hence the new weight update equation is:

$$w_i(k+1) = w_i(k) + \hat{r}(k)\, e_i(k)\, sg(t, k_f(t) - k_f(t-1)) \tag{9}$$

where

$$sg(t,v) = \begin{cases} sg(t-1,v), & \text{if } v \geq 0 \\ -sg(t-1,v), & \text{if } v < 0 \end{cases} \tag{10}$$

with $k_f(t)$ the instant k in which the failure occurs at trial t, and $sg(0,v)=1$.

The reinforcement criterion may be defined as

$$r(k) = \begin{cases} -1, & \text{failure state} \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where the failure state can be defined according to a desired behavior, as in the example given in section 5.

The learning algorithm is basically the same: in step 3.2.7 (3) must be replaced for (9) and in 3.2 the condition $r=0$ must be substituted for the condition $x \in C$ where C is the set of the feasible states. We call this Reinforcement Learning Control Scheme with Failure (RLCSF).

4. CONTROL WITH SUPERVISED ARTIFICIAL NEURAL NETWORK

The main problem of the controllers described in the previous section is that the control signal is abrupt and not continuous. Also, as a specific control signal is associated with a entire region of Ψ, it is not possible to asymptotically stabilize the system, except for particular classes of systems.

In order to solve these problems, we take advantage of the interpolation capability of multilayer perceptron neural networks [6].

In the literature, there are many applications which make use of reinforcement learning as a rule to train neural networks [5], [7]. For these applications, however, the number of trials to train the controller can be decreased about one hundred times if the decodification process we use here is included in the training procedure.

In this paper, we preferred to use the decodification process and make subsequent use of an artificial neural network controller (ANNC) trained by a supervised learning algorithm. The ANNC thus used may include an hybrid scheme, resulting in an adaptive control system.

The neural network is trained by the RLCS or the RLCSF, according to the Figure 1 [8].

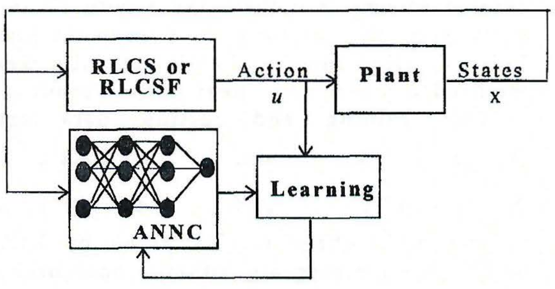

Figure 1. ANNC training.

5. MAGNETIC SUSPENSION SYSTEM EXAMPLE

The control problem is to keep around an operational point a magnetic suspension system, which is nonlinear, autonomous and intrinsically unstable. This system consists of a steel sphere kept at suspension by a magnetic field. This field is generated by a current circulating in a coil (Figure 2). The discrete dynamic equations are described in the Appendix [9].

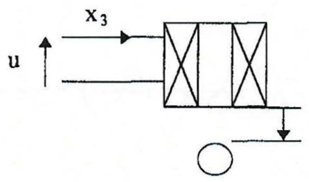

Figure 2. Magnetic suspension system.

This system can be controlled around an operational equilibrium point using linear system techniques. The linear quadratic regulator (LQR) can be used with success to find the control law (u=-**k**x) which minimize a performance function.

The system dynamic performance is close to the predicted by the linearized system if it obeys the constraints inequality:

$$1-\varphi \le \frac{\mathbf{x}_j(k+1)-\mathbf{x}_j(k)}{\mathbf{x}_{jl}(k+1)-\mathbf{x}_{jl}(k)} \le 1+\varphi \qquad (12)$$

where $\mathbf{x}_{jl}$ is the state in the linearized model, and φ is a real valued constant. If the inequality (12) is not satisfied, the difference between the behavior of the real system and the linearized one is significative, and the system may become unstable. Figure 3 shows a region θ_L with φ=0.2, for the suspension magnetic system and Figure 4 shows results obtained for different initial positions, some of them yields unstable dynamics.

5.1 Results

In this section we present simulation results for the control schemes developed.

5.1.1 RLCS and RLCSF Results: The total of partitions of the state space Ψ adopted is 288 for the first two schemes: 12 intervals for x_1, 6 for x_2, and 4 for x_3.

The learning algorithm is executed for different initial states, as follows: we set $X_{ini} = [\mathbf{x}^1,\ldots,\mathbf{x}^{\mu_{e\,max}}]$ where $\mu_{e\,max}$ is the total of initial states; the learning algorithm is run for μ_e=1,..., $\mu_{e\,max}$, where in step 2 the system states are initialized as $\mathbf{x}(0)=X_{ini}(\mu_e)$. This procedure is repeated until r remains in zero for every $\mathbf{x} \in X_{ini}$. The typical number of trials for each initial state was 12 and 37 for the RLCS and RLCSF, respectively. The controller parameters are found in the Appendix.

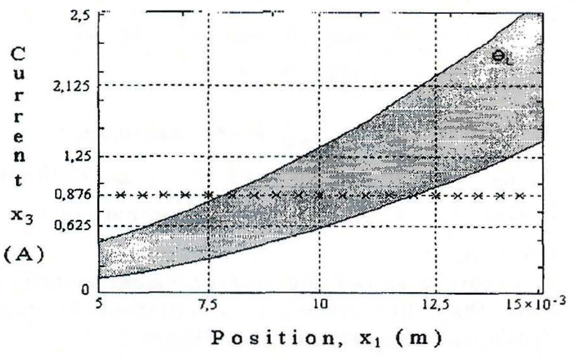

Figure 3. Region of state space with linear behavior for φ=0,2. The states marked by "x" are the initial states of Figure 3.

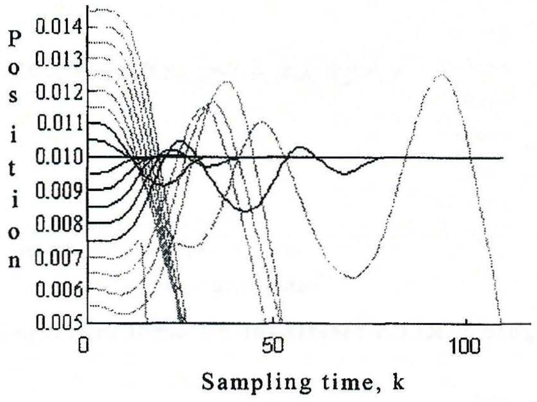

Figure 4. Sphere position for the optimal state feedback controller.

The reinforcement criterion for the RLCS is developed by analyzing the magnetic suspension system. It is expressed by:

$$r=\begin{cases}-1, \text{if } x_1(k)<x_{1c\,max} \text{ and } x_1(k)<x_1(k-1)\\ +1, \text{ if } x_1(k)>x_{1c\,min} \text{ and } x_1(k)>x_1(k-1)\\ 0, \text{ otherwise}\end{cases} \qquad (13)$$

where x_{1cmax} is the superior limit and x_{1cmin} is the inferior limit of the tolerance region of state x_1.

Figure 5 shows simulation results for RLCS for different initial positions of the sphere. Note that the sphere position oscillates in the region defined by external reinforcement criterion.

The reinforcement criteria for the RLCSF was developed by imposing a minimal acceptable performance. Equation (14) shows the reinforcement criterion.

$$r(k)=\begin{cases}0 & \text{if } x_{1\inf}(k)<x_1(k)<x_{1\sup}(k) \text{ and } \\ & x_{2\inf}(k)<x_2(k)<x_{2\sup}(k) \\ -1 & \text{otherwise}\end{cases} \qquad (14)$$

where

$$x_{j\inf}=\begin{cases}(x_{j\,c\,min}-x_{j\,i\,min})/k_c+x_{j\,imin}, & k<k_c \\ x_{j\,cmin}, & k\geq k_c\end{cases}$$

$$x_{j\sup}=\begin{cases}(x_{j\,cmax}-x_{j\,imax})/k_c+x_{j\,imax}, & k<k_c \\ x_{j\,cmax}, & k\geq k_c\end{cases}$$

k_c is an integer, $x_{j\,cmax}$ is the maximum x_j in k_c, $x_{j\,cmin}$ is the minimum x_j in k_c, $x_{j\,imax}$ is the maximum initial x_j, $x_{j\,imin}$ is the minimum initial x_j and $j=1,\ldots,n_s$.

Figure 6 shows the reinforcement criterion for x_1. Note that this criterion is satisfied by the system dynamics, also presented in Figure 6.

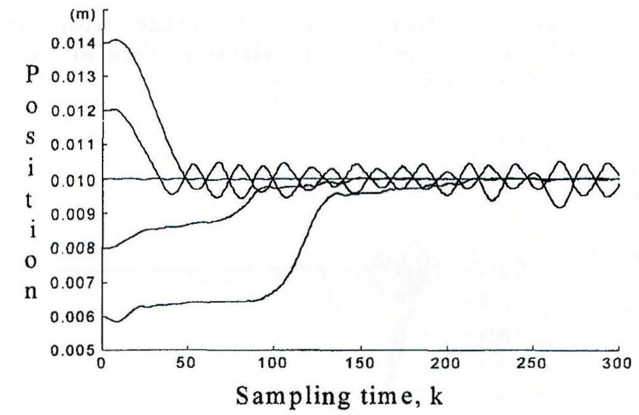

Figure 5. RLCS results for different initial positions.

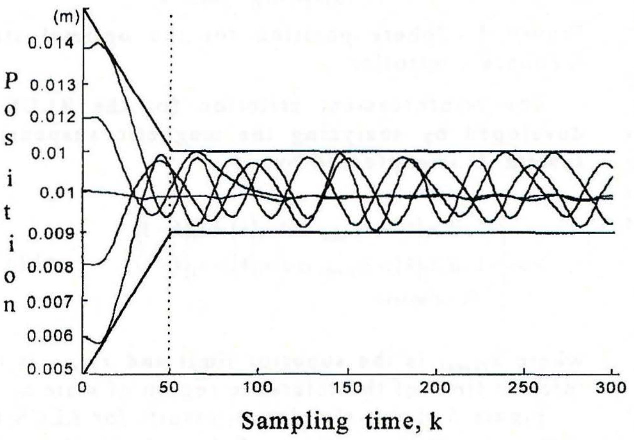

Figure 6. RLCSF results for different initial positions. The criterion utilized is also shown.

5.1.2 ANNC Results: The ANN used in the third control scheme is a three layer with 12, 6, and 1 units, respectively, from the first to the last layer. The activation functions are hyperbolic tangent in the hidden layers and linear in the output layer.

The training and testing data sets are $\mathbf{X}_a=[\mathbf{x}^1,\ldots,\mathbf{x}^{\mu_a}]$ and $\mathbf{u}_a=[u(\mathbf{x}^1),\ldots,u(\mathbf{x}^{\mu_a})]$, $\mathbf{X}_t=[\mathbf{x}^1,\ldots,\mathbf{x}^{\mu_t}]$ and $\mathbf{u}_t=[u(\mathbf{x}^1),\ldots,u(\mathbf{x}^{\mu_t})]$, respectively, where $u(\cdot)$ is given by RLCSF or RLCS. The learning algorithm is back propagation with Levenberg Marquardt method. The typical number of epochs is 20 using μ_a =1400 and μ_t=1200.

Figure 7 shows the ANNC results trained by the RLCSF for different initial positions and Figure 8 shows the control signal for both RLCSF and ANNC for initial position at 0.014 m. Note the behavior of the ANNC signal continuous and smooth.

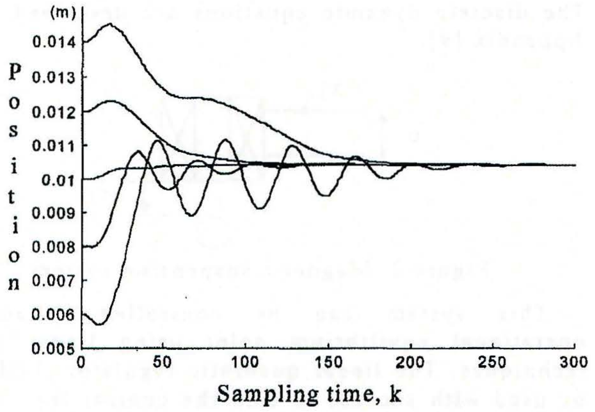

Figure 7. ANNC trained by RLCSF.

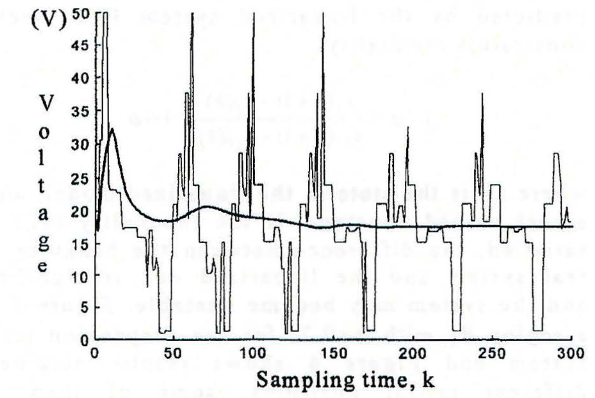

Figure 8. Control signal generated by the RLCSF (gray curves) and by the ANNC (black curves) for initial position at 0,014 m.

6. DISCUSSIONS AND CONCLUSIONS

In this paper, reinforcement learning schemes using adaptive elements are used in the design of different controllers, providing a solution to the control problem described. In addition, an supervised ANN trained by one of the developed schemes is used to perform the controller. Each controller developed has its own characteristics.

The RLSC can be used on-line, is adaptive and presents a fast training process. The RLSCF can also be used on-line, is adaptive, and needs no knowledge of the plant parameters or models. Finally, the ANNC presents a continuous control signal removing the oscillation that may occur when using the other controllers. Hence, each control scheme can be used with success to solve different nonlinear control problems.

The example presented showed that all developed controllers can cope with an operating range of the plant wider than that obtained with the quadratic linear optimal controller presented. Well developed control methods can be successfully applied to suspension systems but they are more dependent on the knowledge of the plant dynamics and uncertainties.

APPENDIX

The system suspension dynamical equations:

$$x_1(k+1) = T_0^2 g + \frac{L_{b0}}{2am}\left(\frac{T_0 x_3(k)}{1+x_1(k-1)/a}\right)^2 + 2x_1(k-1) - x_1(k-2)$$

$$x_2(k+1) = \left(1/T_0\right)\left(x_1(k) - x_1(k-1)\right)$$

$$x_3(k+1) = \left(\frac{1}{T_0 R + L}\right)\left(L x_3(k-1) + T_0 u(k)\right)$$

where x_1 is the sphere position, x_2 the sphere speed, x_3 the coil current, T_0 the sampling period, u the coil voltage, g the gravity, m the mass of steel ball, R the coil resistance, L the coil inductance, L_{b0} the coil inductance when $x_1 = 0$ and a a constant.

Parameters of real system of magnetic suspension: m=0.02255 kg, R=19.9 Ω, a=0.00607, L_b=0.47 H, L_{b0}=0.0245 H and T_0=0.001 s. The equilibrium point adopted is (x_{10}, x_{20}, x_{30}) = (0.01; 0.0; 0.876). The system position and current are constrained by: $\{x_1, x_2, x_3 \in \Re\text{: } 0.005 \le x_1 \le 0.015 \text{ and } 0 \le x_3 \le 2.51\}$.

Intervals for x_1: [0.0050 0.0065]; [0.0065 0.0078]; [0.0078 0.0088]; [0.0088 0.0094]; [0.0094 0.0099]; [0.0099 0.0100]; [0.0100 0.0101]; [0.0101 0.0106]; [0.0106 0.0112]; [0.0112 0.0122]; [0.0122 0.0135]; [0.0135 0.0150]. Intervals for x_2: [-0.40 -0.20]; [-0.20 -0.05]; [-0.05 0.00]; [0.00 0.05]; [0.05 0.20]; [0.20 0.40]. Intervals for x_3: [0.00 0.94]; [0.94 1.26]; [1.26 1.57]; [1.57 2.51].

Parameters of RLSC: t_{max}=100; k_{max}=500; x_{1cmax}=0.0105; x_{1cmin}=0.0095; α=0.9; β=0.3; δ=0.85; γ=0.995, λ=0.95, f is the identity function and h a mean zero Gaussian distribution.

Parameters of RLSCF: t_{max}=100; k_{max}=500; k_c=50; x_{1cmax}=0.0108; x_{1cmin}=0.092; x_{1imax}=0.015; x_{1imin}=0.005; x_{2cmax}=0.1; x_{2cmin}=-0.1; x_{2imax}=0.3; x_{2imin}=-0.3; $x_{3cmax}=x_{3imax}=\infty$; $x_{3cmin}=x_{3imin}=-\infty$; μ_{emax}=9; for X_{ini} we set x_2=0 and x_3=0.876, and varied x_1=0.006 to x_1=0.014 in steps of 0.001; α=3; β=0.2; λ=0.7; γ=1, δ=0.9, again f is the identity function and h a mean zero Gaussian distribution function.

REFERENCES

[1] A. G. Barto, R. S. Sutton and C. W. Anderson, "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems", *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13, pp 834-846, 1983.

[2] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement Learning is Direct Adaptive Optimal Control", *IEEE Control Systems Magazine*, pp 19-22, Apr. 1992.

[3] A. U. Levin, *Neural Networks in Dynamical Systems*, PhD Thesis, Yale University, 1992.

[4] M. Vidyasagar, *Nonlinear Systems Analysis*, Prentice-Hall International Editions, 1993.

[5] J. C. Hoskins and D. M. Himmelblau, "Process Control Via Artificial Neural Networks and Reinforcement Learning", *Computers Chem. Engng.*, Vol.16, No. 4, pp 241-251, 1992.

[6] J. Hertz, A. Krogh and R. G. Palmer, *Introduction to the Theory of Neural Computation.* Addison Wesley Publishing Company, 1991.

[7] C. H. Anderson, "Learning to Control an Inverted Pendulum Using Neural Networks", *IEEE Control Systems Magazine*, Vol. 9, pp 31-36, 1989.

[8] M. Brown and Chars, *Neurofuzzy Adaptive Modeling and Control.* Prentice Hall, 1994.

[9] K. Oguchi and Y. Tomigashi, "Digital Control for a Magnetic Suspension System as an Undergraduate Project", *Int. J. Elect. Enging. Educ.*, Vol. 27, pp 226-236, 1990.