Trajectory Control of DC Servo Using OS-ELM Based Controller

Vikas Kumar
Div. of ICE, NSIT
Dwarka, New Delhi-110078
E-mail sainivika@gmail.com

Prerna gaur
Div. of ICE, NSIT
Dwarka, New Delhi-110078
E-mail prernagaur@yahoo.com

A.P.Mittal
Div. of ICE, NSIT
Dwarka, New Delhi-110078
E-mail mittalap@gmail.com

***Abstract*— In this work implementation of On Line Sequential Extreme Learning Machine (OS-ELM) based trajectory control strategy of DC servo motor is investigated. Extreme Learning Machine (ELM) is a novel method for training of the forward neural networks with single hidden layer is extremely fast and accurate compared to gradient based learning methods. Further to minimize the effect of machine parameters variations, ELM is trained on line using sequential recursive least square method. Comparative study of artificial neural network based controller, ELM based controller, and OS-ELM based controller has been done for DC servo motor for non linear fan or impeller type of load.**

Keywords- ***Artificial Neural Network, Extreme Learning Machine, Back Propagation, Single Hidden Layer Feed forward networks, OS-ELM***

I. INTRODUCTION

Servo motor is an Electro-mechanical device which forms a part of servomechanism used to control the motion and/or position of load precisely. These are high performance, small size and low-inertia motors that respond quickly to excitation-voltage changes [1]. Very low armature inductance and low electrical time constant in servo motors further sharpens servo motor response to the command signals.

DC servo motors are normally used in applications where starts and stops are made quickly and accurately, computer numerically controlled machines and robotics for precise positioning of the end effectors. Various control schemes are available for the DC motor speed control. Among the commonly used schemes the conventional PID control, Sliding Mode Control (SMC) are the most dominant as these controls are simple and easy to implement However these control techniques rely on the mathematical dynamics of the plant and cannot deal with the nonlinearity in the process and uncertain problems [2, 3, 4]. To overcome these problems the adaptive control and model based control are proposed. These methods can deal with the model uncertainties and but their performance in the presence of unstructured and unmodeled uncertainty is not guaranteed [5]. Intelligent control develops the control that attempt to emulate important characteristics of human intelligence [6, 7, and 8]. Fuzzy logic control is an example of a rule based representation of human knowledge and deductive process [9, 10] Using fuzzy logic systems (FLS) any non-linear function can be approximated to a given accuracy (Wang) but for the implementation of fuzzy logic controller for a given process, the large number of parameters which depends on the knowledge of the process, needs to be tuned prior to the implementation. An artificial neural network (ANN) is an emulation of biological neural system. Artificial neural network possesses the learning and adaptation capabilities similar to the biological neural system and are applied successfully in wide engineering and non-engineering applications. ANN can be trained using some input-output data such that the application of arbitrary set of inputs produces the desired set of outputs however ANN when trained with traditional gradient based learning methods generally trains very slowly and can converge to local minima. For the faster and efficient training of FFNN [11] numerous training algorithms are available in literature.

Huang *et al.* proposed a novel method of training the single hidden layer forward neural networks (SLFNs) i.e. Extreme Learning Machine (ELM). It has been well documented in literature that a SLFN with activation function 'tansig' in hidden layer and 'purelin' in the output layer can approximate almost every non linear function [12]. In ELM, the input weights are assumed randomly and the weights connecting the hidden layer and output layer are determined analytically. Using ELM the training time can be reduced by a factor of hundreds to thousands compared to the traditional gradient based learning methods [13, 14 and 15]. For the SLFNs, the training methods usually available are of batch leaning type, which trains very slowly and the training parameters like learning rate, no of epochs and the stopping criteria must be chosen prior to the training in order to ensure the convergence of the training algorithm. Huang *et. al.* proposed a training method for on line sequential [16] training of the SLFNs, which can take data chunk by chunk basis or block by block basis for training and needs almost no parameters to be determined prior to the training. The training time for OS-ELM and ELM is very small compared to the traditional gradient based methods [16].

In the present work mathematical modelling of DC motor is done for impeller driven or fan type of load. The modelling is used in MATLAB to implement the reference trajectory control using ANN, ELM and OS-ELM based controller and it has been found that the performance of the OS-ELM is comparable to that of ELM and is better compared to the ANN based controller in terms of training time and training accuracy.

978-1-4673-0934-9/12/$31.00 ©2012 IEEE

II. MATHEMATICAL MODELING OF DC MOTOR

Mathematically the dynamics of the DC motor is described by the following equations [17].

$$v_a(t) = R_a i_a(t) + L_a \frac{di_a}{dt} + e_b(t) \quad (1)$$

$$e_b(t) = K_b \omega(t) \quad (2)$$

$$T_M(t) = K_T\, i_a = J\frac{d\omega(t)}{dt} + B\omega(t) + T_L(t) + T_F \quad (3)$$

where

$v_a(t)$ = applied armature voltage, $\omega(t)$=angular velocity of the motor rotor (rad/sec), $i_a(t)$= armature current (amps), R_a = armature winding resistance, L_a =armature winding inductance, $T_M(t)$ = torque developed by the motor, K_T=torque constant, K_b=back emf constant, T_F = Frictional torque. $e_b(t)$ = back emf (volts), J=sum of moment of inertia of the motor rotor and mechanical load, B= sum of viscous friction constant of the motor rotor and mechanical load, $T_L(t)$= disturbance load torque,.

The load torque can be gives as:

$$T_L(t) = \Psi(\omega) \quad (4)$$

Where $\Psi(.)$ depends on the nature of the load. For most propeller driven or fan type loads the function $\Psi(.)$ Takes the following form

$$T_L(t) = \mu\omega^2(t)[sgn\omega(t)] \quad (5)$$

In order to model DC motor as single input and single output system the equations (1)-(5) are combined and discrete time model is derived

$$L_aJ\left[\frac{\omega(k+1)-2\omega(k)+\omega(k-1)}{T^2}\right] + (R_aJ + L_aB)\left[\frac{\omega(k+1)-\omega(k)}{T}\right] + (R_aB + K_bK_T)\omega(k) + L_a\frac{T_L(k)-T_L(k-1)}{T} + R_aT_L(k) + R_aT_F + K_Tv_a(k) = 0 \quad (6)$$

$$T_L(k) = \mu\omega^2(k)[sgn\omega(k)] \quad (7)$$

$$T_L(k-1) = \mu\omega^2(k-1)[sgn\omega(k-1)] \quad (8)$$

Where T= sampling period

$$\omega(k) \triangleq \omega(t = kT);\ k=0,1,2,\ldots \quad (9)$$

Using (8), (6) can be rewritten as:

$$\omega(k+1) = K_1\omega(k) + K_2\omega(k-1) + K_3[sgn\omega(k)]\omega^2(k) + K_4[sgn\omega(k)]\omega^2(k-1) + K_5v_a(k) + K_6 \quad (10)$$

Where

$$K_1 = \frac{2L_aJ + T(R_aJ + L_aB) - T^2((R_aB + K_bK_T)}{L_aJ + T(R_aJ + L_aB}$$

$$K_2 = -\frac{L_aJ}{L_aJ + T(R_aJ + L_aB)}$$

$$K_3 = -\frac{T(\mu L_a + \mu R_aT)}{L_aJ + T(R_aJ + L_aB)}$$

$$K_4 = \frac{T\mu L_a}{L_aJ + T(R_aJ + L_aB)}$$

$$K_5 = \frac{K_TT^2}{L_aJ + T(R_aJ + L_aB)}$$

$$K_6 = \frac{T_FR_aT^2}{L_aJ + T(R_aJ + L_aB)}$$

Following numerical values are the parameters of 1 HP, 220V, 550rad/min DC motor are used in the simulation:

$J = 0.068\ Kg\,.m^{-2}$, $B = 0.03475\ N.m.A^{-1}$, $R_a = 7.56\Omega$ $L_a = 0.055H$, $\mu = 0.0039M.\Omega.m.s^{-2}$, $K_b = 3.475$ Volt-sec/rad. $K_T = 3.475$ Volt-sec/rad. $T_F = 3.475$ N-m , $T = 0.04$ sec.

Using above parameters for DC motor the numerical value of the constants is as follows:

$K_1 = 0.03466$, $K_2 = -0.1534069$, $K_3 = -2.286928e-3$, $K_4 = 3.5193368e-4$, $K_5 = 0.2280595$, $K_6 = -0.105284$

Equations (1)-(10) are used to model the DC motor with propeller driven or fan type load. The line daigram of the DC servo motor is shown in fig. 1.

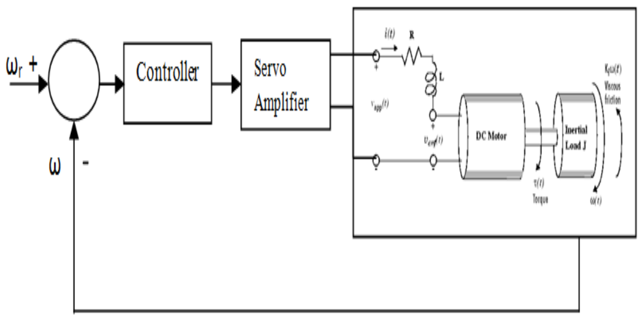

Fig. 1 Line Diagram for the DC Servo Drive

III. MATHEMATICAL MODELING OF ELM

Extreme learning machine [12] is discussed in this section.

A. Single hidden layer Feed forward networks (SLFNS):

The structure of single hidden layer feed forward network is as shown in fig. 2.

For N distinct input-output data samples (p_i, t_i)

Where $p_i = [p_{i1}, p_{i2}, \ldots\ldots\ldots\ldots p_{in}]^T$ and $t_i = [t_{i1}, t_{i2}, \ldots\ldots\ldots\ldots t_{in}]^T \epsilon R^n$

For $\tilde{N}$ number of hidden nodes and activation function activation function g(p) in the hidden layer, The SLFNs are mathematically written as

$$\sum_{i=1}^{n} \gamma_i\, g(w_i\, p_j + b_i) = o_j$$

J= 1, 2..........., N

where $w_i = [w_{i1}, w_{i2}, \ldots\ldots\ldots\ldots w_{in}]^T$ is the weight vector connecting the i^{th} hidden node and the input node and

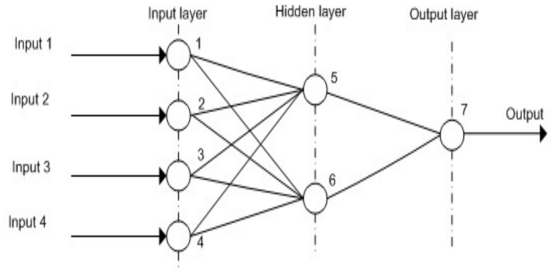

Fig. 2: Single hidden layer feed forward network

$\gamma_i = [\gamma_{i1}, \gamma_{i2}, \ldots \ldots \ldots \ldots . \gamma_{in}]^T$ is the weight vector connecting the i^{th} hidden node and the output node. b_i is the threshold of the i^{th} hidden node.

With activation function g(p) and Ñ hidden nodes, SLFNs can approximate these n samples with zero error $\sum_{j=1}^{\tilde{N}} || o_j - t_j || = 0$ such that $\sum_{i=1}^{n} \gamma_i \; g(w_i \, p_j + b_i) = t_j$, j= 1, 2 ...N.

$\alpha\gamma = T$ where

$$\alpha(w_1, \ldots w_{\tilde{N}}, b_1, \ldots \ldots \ldots b_{\tilde{N}}, p_1, \ldots p_{\tilde{N}}) = \begin{bmatrix} g(w_1 \, p_1 + b_1) \ldots \ldots \ldots g(w_{\tilde{N}} \, p_{\tilde{N}} + b_{\tilde{N}}) \\ \ldots \ldots \ldots \ldots \qquad \ldots \ldots \ldots \ldots \\ \ldots \ldots \ldots \ldots \qquad \ldots \ldots \ldots \ldots . \\ \ldots \ldots \ldots \ldots \qquad \ldots \ldots \ldots \ldots . \\ g(w_1 \, p_N + b_1) \qquad g(w_{\tilde{N}} \, p_N + b_{\tilde{N}}) \end{bmatrix}$$

$$\gamma = \begin{bmatrix} \gamma_1^T \\ . \\ . \\ . \\ . \\ \gamma_{\tilde{N}}^T \end{bmatrix} \text{ and } t = \begin{bmatrix} t_1^T \\ . \\ . \\ . \\ . \\ t_{\tilde{N}}^T \end{bmatrix}$$

H is hidden layer output matrix of the neural network.

Training of SLFN requires the determination of $\left|\left|\alpha(\hat{w}_1, \ldots \ldots \hat{w}_{\tilde{N}}, b_1 \ldots \ldots \ldots \ldots b_{\tilde{N}})\gamma - T\right|\right| = \min_{w_i, b_i, \gamma} \left|\left|\alpha(\hat{w}_1, \ldots \ldots \hat{w}_{\tilde{N}}, b_1 \ldots \ldots \ldots \ldots b_{\tilde{N}})\gamma - T\right|\right|$ which is equivalent to minimizing the function.

$$E = \sum_{j=1}^{N} \left(\sum_{i=1}^{\tilde{N}} g(w_i \, p_j + b_i) - t_j \right)^2$$

Gradient based learning algorithm used to find the minimum value of ||Hγ –T||. For minimization of ||Hγ –T|| vector the set of weights$(w_i \, \gamma_i)$ and biases b_i are adjusted iteratively using back propagation of error as follows:

$$w_k = w_{k-1} - \eta \frac{\partial E(w)}{\partial w}$$

where η is the learning rate. The value of the learning rate should be chosen carefully, as the too small value of learning rate can make training process very slow and too large a value of learning rate makes the training algorithm unstable. The BP algorithm can converge to local minima even if it is located above global minima. The gradient based learning methods takes very large to converge in most applications.

B ELM Algorithm

Given a training set (p_i, t_i) where

$p_i = [p_{i1}, p_{i2}, \ldots \ldots \ldots \ldots . p_{in}]^T$ and

$t_i = [t_{i1}, t_{i2}, \ldots \ldots \ldots \ldots . t_{in}]^T$

Standard SLFNs with Ñ hidden nodes and activation function g(p).

Step 1: The input weights w_i and bias b_i for i = 1.......... Ñ are randomly assumed

Step 2: Hidden layer output matrix α is determined using α.

Step 3: calculate the output weight matrix γ using $\gamma = \alpha^{-1}T$

Where α^{-1} is the Moore-Penrose inverse of matrix H.

C. Online Sequential-ELM

OS-ELM consists of boosting phase and sequential learning phase. In the boosting phase the batch of training data samples equal to the number of hidden nodes (Ñ) is used to initialize the ELM. In the sequential learning phase, training data is presented on one by one basis [16].

step 1 **Initialization Phase** : For a small training set (p_i, t_i) The learning algorithm is initialized as follows:

(a) Assign random input weight W_i and bias b_i for the hidden layer

(b) The initial hidden layer output matrix is calculated as : $\alpha_0 = [\alpha_1 \ldots \ldots \ldots \alpha_{\tilde{N}}]^T$

where

$\alpha_i = [g(w_1 \, p_i + b_1) \ldots \ldots \ldots g(w_{\tilde{N}} \, p_{\tilde{N}} + b_{\tilde{N}}), \ldots . g(w_{\tilde{N}} \, p_1 + b_1) \ldots \ldots \ldots g(w_{\tilde{N}} \, p_{\tilde{N}} + b_{\tilde{N}})]^T$ i-1, 2…N.

(c) Initial output weight matrix is given by:

$\gamma^0 = \eta_0 \alpha_0^T T_0$ where $\eta_0 = (\alpha_0^T \alpha_0)^{-1}$ and $T_0 = [t_1 \ldots \ldots . . t_{\tilde{N}}]^T$

(d) Set k=0

step 2 **On-Line Learning Phase:** for each further coming observation (p_i, t_i) and i= $\tilde{N} + 1, \tilde{N} + 2, \tilde{N} + 3 \ldots \ldots \ldots$,do

(a) Calculate the hidden layer output vector $h_{(k+1)} = [g(w_1 \, p_i + b_1) \ldots \ldots \ldots g(w_{\tilde{N}} \, p_{\tilde{N}} + b_{\tilde{N}}), \ldots . g(w_{\tilde{N}} \, p_1 + b_1) \ldots \ldots \ldots g(w_{\tilde{N}} \, p_{\tilde{N}} + b_{\tilde{N}})]^T$.

(b) Output weight are updated using Recursive Least Square (RLS) algorithm:

$$\eta_{k+1} = \eta_k - \frac{\eta_k \alpha_{k+1} {\eta_{k+1}}^T \eta_k}{1 + h_{k+1}^T \eta_k \alpha_{k+1}}$$

$$\gamma^{(k+1)} = \gamma^{(k)} + \eta_{k+1} \alpha_{k+1} (T_i^T - \alpha_{k+1}^T \gamma^{(k)})$$

(c) Set k=k+1

IV. RESULTS AND DISCUSSIONS

The trajectory of the motor is controlled using ANN, ELM and OS-ELM based controller. The response of the ELM based controller for the trajectory control is shown in fig. 3, in which the blue line shows the reference trajectory and the red line shows the controlled trajectory. For the ANN based controller the training time was 19s and the training accuracy achieved is 2.51e-3. The response of the ELM based controller to the reference trajectory is shown in fig. 4. The training time for the ELM based is 0.157s and the training accuracy achieved is 6.04e-6. The response of ELM based controller is better compared to ANN based controller. In OS-ELM, the ELM learned the data one by one in sequential manner, the training data is discarded once the learning is complete. The response of the OS-ELM to the reference trajectory is shown in Fig. 5. After the completion of the training, the generalization performance is evaluated, the results shows, OS-ELM and ELM has the very small training time, high training accuracy and better generalization performance. The performance of and OS-ELM and ELM is better compared to the ANN based controller. The comparison of the implemented controllers is shown in fig. 6. The blue line shows the reference trajectory, the red line shows the response of the ANN based controller, green line shows the response of the ELM based controller and the black line shows the response of the OS-ELM based controller. The OS-ELM based controller offered the best tracking response compared to the ANN based and ELM based controller.

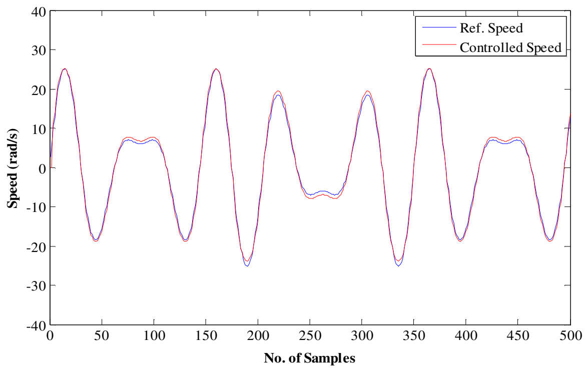

Fig. 3: Response of ANN based controller to a reference trajectory

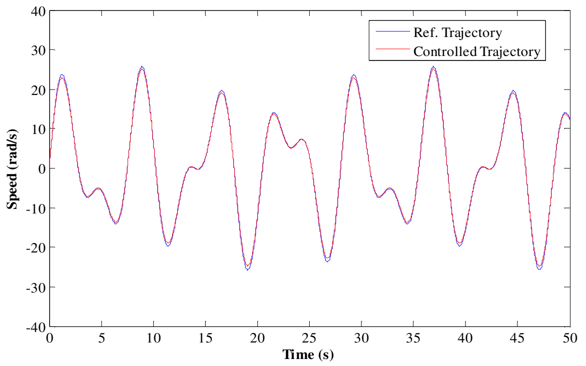

Fig. 4: Response of ELM based controller to a reference trajectory

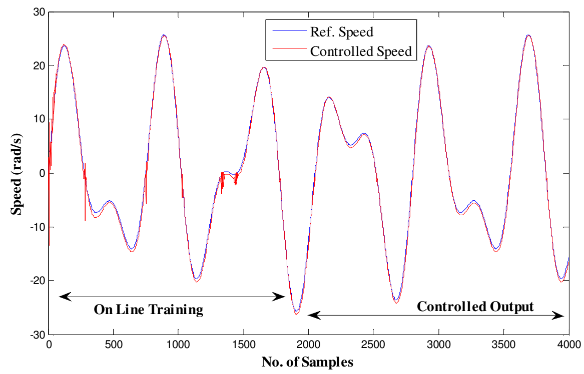

Fig. 5: Response of OS-ELM based controller to a reference trajectory

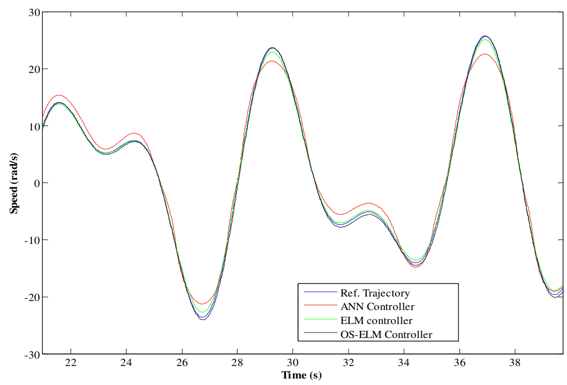

Fig. 6: Comparison of the response of implemented controllers

CONCLUSION:

The trajectory of DC servo motor is controlled through ANN, ELM and OS-ELM based controller. The MSE for the ANN based controller is 0.3934. The response of ELM based controller for the trajectory control is better as compared to the ANN based controller, the MSE for ELM based controller is 0.1959. The response of the OS-ELM based trajectory controller is better compared to ELM based controller. The training time for OS-ELM is also very small and the MSE is 0.1892. OS-ELM can find applications in the rapid changing environments where data is available on one by one basis. OS-ELM can suppress the effects of parameters changes as it is trained online that is the added advantage. Similar to ELM, OS-ELM can approximate any non linear function.

REFRENCES:

[1] G. Mccomb and M. Predko, "Robot Builder's Bonanza", pp.393-413, 2006

[2] J. Zumberge and K.M., Passino, **“**A case study in intelligent vs. conventional control for a process control experiment**”,** Proceedings of the IEEE International Symposium on Intelligent Control, 15-18 Sept. pp 37 – 42, 1996.

[3] K. F. Man, K. S. Tang, and S. Kwong, “GAs: Concepts and applications,” *IEEE Trans. Ind. Electron.*, vol. 43, pp. 519–533, Oct. 1996.

[4] H.-B. Shin, “New antiwindup PI controller for variable-speed motor drives,” *IEEE Trans. Ind. Electron.*, vol. 45, pp. 445–450, June 1998.

[5] L. A. Dessaint, M. Saad, B. Hebert, and K. Al-Haddad, “An adaptive controller for a direct-drive SCARA robot,” *IEEE Trans. Ind. Electron.*, vol. 39, pp. 105–111, April, 1992.

[6] B. K. Bose, “Expert System, fuzzy Logic, and Neural Network Applications in Power Electronics and Motion Control “Proceedings of the IEEE, Vol.82, pp.1303–1323, Aug 1994.
[7] K. Passino, “*Intelligent control*”, in The Control Handbook Boca Raton: CRC Press, 1996, pp. 999–1001.
[8] Y.F. Li and C.C. Lau "Development of fuzzy algorithms for servo systems" EEE Control System Magazine pp 65-71, April 1989.
[9] P.J. King and E.H. Mamdani "The application of fuzzy control systems to industrial processes" Automatica. 1977 vol. 13 pp 235-242.
[10] D. Driankov, H. Hellendoorn, and M. Reinfrank, “*An Introduction to Fuzzy Control*”, N Y: Springer-Verlag, 1993.
[11] C.S. Leung, Ah-Chung Tsoi, and L. W. Chan, “Two Regularizers for Recursive Least Squared Algorithms in Feed forward Multilayered Neural Networks,” *IEEE Transactions Neural Networks*, vol. 12, no. 6, pp. 1314-1332, Nov. 2001.
[12] G. B. Huang, Q. Y. Zhu, and C. K. Siew, “Extreme learning machine: Theory and applications,” *Neurocomputing*, vol. 70, pp. 489–501, 2006.
[13] Y. Lan, Y. C. Soh, and G. B. Huang, “Ensemble of online sequential extreme learning machine,” *Neurocomputing*, vol. 72, pp. 3391–3395, 2009.
[14] Nan Liu and Han Wang, ‘Ensemble Based Extreme Learning Machine,’ *IEEE Signal Processing Letters,* vol. 17, no. 8, August 2010.
[15] G.B. Huang, N.Y. Liang, H. J. Rong, P. Saratchandran, and N. Sundararajan, “On Line Sequential Extreme Learning Machine,” IASTED-2005, July 4-6, 2005.
[16] N.Y. Liang, G.B. Huang, P. Saratchandran, and N. Sundararajan, “A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks,” *IEEE Transactions Neural Networks*, vol. 17, no. 6, pp. 1313-1332, Nov. 2006.
[17] M. Gopal, “Digital Control and State Variables Methods”, 2003.