Proceedings of the American Control Conference
Philadelphia, Pennsylvania • June 1998

Adaptive Tracking Using ARMARKOV/Toeplitz Models

Ravinder Venugopal and Dennis S. Bernstein *
Department of Aerospace Engineering
The University of Michigan
Ann Arbor, MI 48109-2118
{ravinder, dsbaero}@umich.edu

Abstract

An adaptive algorithm is developed for the MIMO tracking problem. The MIMO system and controller are represented as ARMARKOV/Toeplitz models, and the parameter matrix of the compensator is updated on-line by means of a gradient algorithm. The algorithm does not require any knowledge of the plant. Simulation results on a fourth order system are presented.

Notation

I_l	$l \times l$ identity matrix
$0_{l\times m}$	$l \times m$ zero matrix
$1_{l\times m}$	$l \times m$ ones matrix
$\mathcal{I}_{l\times m}$	$[I_l \ \cdots \ I_l] \in \mathcal{R}^{l\times ml}$

1. Introduction

In feedback control applications the problems of disturbance rejection and command following, also known as tracking, can be viewed as dual. For disturbance rejection problems the objective is to maintain an equilibrium state in the presence of external disturbances, while for command following problems the objective is to follow specified command signals. It is common practice to assume that in disturbance rejection problems the disturbance signal is unmeasured, while in command following problems the command is generated by an external system and thus is assumed to be known. We note that in certain applications such as active noise control it is sometimes assumed that the disturbances are measured. Such feedforward cancellation methods are discussed in [1].

Various theories have been developed for both of these problems under a variety of assumptions. For example, LQG theory for disturbance rejection assumes unknown and generally unmeasured disturbances with white spectra. If the disturbance spectrum is known and is nonwhite, then appropriate filters can be embedded into the plant and thus play a role in the Riccati equations during synthesis. In certain disturbance rejection and command following problems, it may be assumed that the disturbances or commands are generated by an exogenous system with known dynamics. In this case, a compensator can be designed that provides asymptotic rejection or tracking using an internal model controller [2, 3, 4, 5]. If, for example, the disturbance or command is sinusoidal, then this approach requires knowledge of the natural frequency of the exogenous system. This approach requires that neither the amplitude nor the phase of the disturbance need be known.

In a recent paper [6, 7], an alternative approach to disturbance rejection was developed using the ARMARKOV/Toeplitz framework [8, 9, 10]. This approach was based upon a recursive update of the controller gains that was shown to have the ability to adapt to changes in the disturbance spectrum. Thus, if the disturbance is sinusoidal, then the approach of [6, 7] requires neither the amplitude nor the phase nor the frequency of the disturbance. In addition, the method requires identification of only the transfer function from the control input to the performance variable. Experimental application of this algorithm was reported in [6, 7] for a noise control application.

In the present paper we develop an approach to adaptive tracking using the ARMARKOV/Toeplitz framework. Since the command is assumed to be known, this approach to tracking requires no modeling of the plant dynamics. Consequently, the modeling requirements for the tracking problem are less burdensome than for the disturbance rejection problem.

The approach given in this paper for command following problems can be viewed as an alternative to model reference adaptive control methods [11]. In the model reference approach, the system is required to follow the output of a reference model to a prescribed command, however, the proposed algorithm drives the system such that the output of the system follows the command. Future work will involve structural and performance based comparisons to model reference adaptive control systems.

2. ARMARKOV Models of Systems

In this section we derive the ARMARKOV representation of a state space model. Consider the nth-order discrete-time finite-dimensional linear time-invariant system

$$x(k+1) = Ax(k) + Bu(k), \quad (1)$$
$$y(k) = Cx(k) + Du(k), \quad (2)$$

*This research was supported in part by the Air Force Office of Scientific Research under grant F49620-95-1-0019 and the University of Michigan Office of the Vice President for Research.

where $u(k) \in \mathcal{R}^{m_u}$ and $y(k) \in \mathcal{R}^{l_y}$. The Markov parameters $H_j \in \mathcal{R}^{l_y \times m_u}$ of this system are defined as

$$\begin{aligned} H_j &\triangleq D, \quad && j=-1, && (3)\\ &\triangleq CA^jB, \quad && j \geq 0, && (4) \end{aligned}$$

and satisfy

$$G(z) \triangleq C(zI-A)^{-1}B + D = \sum_{j=-1}^{\infty} H_j z^{-(j+1)}. \tag{5}$$

The μ-*ARMARKOV* model [6, 7, 8, 9] or μ step ahead predictor [11], pp. 169-179, [12], pp. 136-139, of (1) and (2) is given by

$$\begin{aligned} y(k) &= \sum_{j=1}^{n} -\alpha_j y(k-\mu-j+1) + \qquad (6)\\ &\quad \sum_{j=1}^{\mu} H_{j-1}u(k-j+1) + \sum_{j=1}^{n} \mathcal{B}_j u(k-\mu-j+1), \end{aligned}$$

where $\alpha_j \in \mathcal{R}$ and $\mathcal{B}_j \in \mathcal{R}^{l_y \times m_u}$, $j = 1, \ldots, n$.

Equation (6) is an input-output relation that explicitly involves μ Markov parameters. For $\mu = 1$, (6) specializes to the usual ARMA model.

Now, let p be a positive integer and define the *extended measurement vector* $Y(k) \in \mathcal{R}^{lp}$ and the *ARMARKOV regressor vector* $\Phi_{yu}(k) \in \mathcal{R}^{l_y(p+n-1)+m_u(\mu+p+n-1)}$ by

$$\begin{aligned} Y(k) &\triangleq \begin{bmatrix} y(k) & \cdots & y(k-p+1) \end{bmatrix}^{\mathrm{T}}, && (7)\\ \Phi_{yu}(k) &\triangleq [y(k-\mu) \ \cdots \ y(k-\mu-p-n+2) \\ &\qquad u(k) \ \cdots \ u(k-\mu-p-n+2)]^{\mathrm{T}}. && (8) \end{aligned}$$

Using (6), $Y(k)$ and $\Phi_{yu}(k)$ are related by

$$Y(k) = W_{yu}\Phi_{yu}(k), \tag{9}$$

where the block-Toeplitz *ARMARKOV weight matrix* W_{yu} is defined by

$$W_{yu} \triangleq \left[\begin{matrix} -\alpha_1 I_{l_y} & \cdots & -\alpha_n I_{l_y} & 0_{l_y} & \cdots & 0_{l_y} \\ 0_{l_y} & \ddots & & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & & \ddots & 0_{l_y} \\ 0_{l_y} & \cdots & 0_{l_y} & -\alpha_1 I_{l_y} & \cdots & -\alpha_n I_{l_y} \end{matrix}\right. \qquad (10)$$

$$\left.\begin{matrix} H_{-1} & \cdots & H_{\mu-2} & \mathcal{B}_1 & \cdots & \mathcal{B}_n & 0_{l_y \times m_u} & \cdots & 0_{l_y \times m_u} \\ 0_{l_y \times m_u} & \ddots & & \ddots & \ddots & & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & & \ddots & \ddots & & \ddots & 0_{l_y \times m_u} \\ 0_{l_y \times m_u} & \cdots & 0_{l_y \times m_u} & H_{-1} & \cdots & H_{\mu-2} & \mathcal{B}_1 & \cdots & \mathcal{B}_n \end{matrix}\right],$$

We note that a state space realization of the system from (6) either by constructing a canonical form or by using the eigenvalue realization algorithm (ERA) [8, 9].

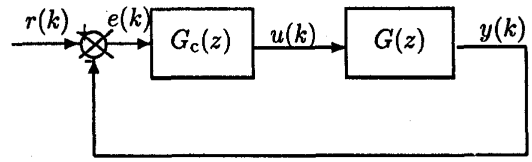

Figure 1: Block diagram of closed-loop system for tracking

3. Adaptive Tracking Algorithm

3.1. Case 1: Known Plant Parameters

Consider the closed-loop system shown in Figure 1. The objective here is to make the output of the plant $y(k)$ follow the prescribed reference trajectory $r(k) \in \mathcal{R}^{l_y}$, that is, to drive the error signal $e(k) = r(k) - y(k)$ to zero. Using ARMARKOV models to represent the plant $G(z)$ and the controller $G_c(z)$, it follows from (9) that

$$Y(k) = A_y \Phi_y(k) + B_{yu}U(k), \tag{11}$$

where $Y(k)$ is defined as in (8) and

$$\begin{aligned} \Phi_y(k) &\triangleq \begin{bmatrix} y(k-\mu) & \cdots & y(k-\mu-n-p+2) \end{bmatrix}^{\mathrm{T}}, && (12)\\ U(k) &\triangleq \begin{bmatrix} u(k) & \cdots & u(k-\mu-n-p+2) \end{bmatrix}^{\mathrm{T}}, && (13) \end{aligned}$$

$$A_y \triangleq \begin{bmatrix} -\alpha_1 I_{l_y} & \cdots & -\alpha_n I_{l_y} & 0_{l_y} & \cdots & 0_{l_y} \\ 0_{l_y} & \ddots & & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & & \ddots & 0_{l_y} \\ 0_{l_y} & \cdots & 0_{l_y} & -\alpha_1 I_{l_y} & \cdots & -\alpha_n I_{l_y} \end{bmatrix} \tag{14}$$

$$B_{yu} \triangleq \left[\begin{matrix} H_{yu,-1} & \cdots & H_{yu,\mu-2} & \mathcal{B}_{yu,1} & \cdots & \mathcal{B}_{yu,n} \\ 0_{l_y \times m_u} & \vdots & \ddots & & \ddots & \vdots \\ \vdots & \ddots & \ddots & & \ddots & \ddots \\ 0_{l_y \times m_u} & \cdots & 0_{l_y \times m_u} & H_{yu,-1} & \cdots & H_{yu,\mu-2} \end{matrix}\right.$$

$$\left.\begin{matrix} 0_{l_y \times m_u} & \cdots & 0_{l_y \times m_u} \\ \ddots & \ddots & \vdots \\ \ddots & \ddots & \vdots \\ \mathcal{B}_{yu,1} & \cdots & \mathcal{B}_{yu,n} \end{matrix}\right]. \tag{15}$$

Next, we use a strictly proper controller in ARMARKOV form of order n_c with μ_c Markov parameters, so that, analogous to (6), the control $u(k)$ is given by

$$\begin{aligned} u(k) &= \sum_{j=1}^{n_c} -\alpha_{c,j}(k)u(k-\mu_c-j+1) \\ &\quad + \sum_{j=1}^{\mu_c-1} H_{c,j-1}(k)e(k-j+1) \\ &\quad + \sum_{j=1}^{n_c} \mathcal{B}_{c,j}(k)e(k-\mu_c-j+1), \qquad (16) \end{aligned}$$

where $H_{c,j} \in \mathcal{R}^{m_u \times l_y}$ are the Markov parameters of the controller. Next, define the *controller parameter block vector*

$$\begin{aligned}\theta(k) \triangleq [\, -\alpha_{c,1}(k)I_{m_u} & \cdots & -\alpha_{c,n_c}(k)I_{m_u} \\ H_{c,0}(k) \ \cdots \ H_{c,\mu_c-2}(k) & \mathcal{B}_{c,1}(k) & \cdots \ \mathcal{B}_{c,n}(k) \,].\end{aligned} \tag{17}$$

Now from (13) and (16) it follows that $U(k)$ is given by

$$U(k) = \sum_{i=1}^{p_c} L_i\theta(k-i+1)R_i\Phi_{uy}(k), \tag{18}$$

where

$$\begin{aligned}\Phi_{ue}(k) \triangleq [\, u(k-\mu_c) \ \cdots \ u(k-\mu_c-n_c-p_c+2) \\ e(k-1) \ \cdots \ e(k-\mu_c-n_c-p_c+2)\,]^{\mathrm{T}},\end{aligned} \tag{19}$$

and

$$L_i \triangleq \begin{bmatrix} 0_{(i-1)m_u \times m_u} \\ I_{m_u} \\ 0_{(p_c-i)m_u \times m_u} \end{bmatrix}, \tag{20}$$

$$\tag{21}$$

$$R_i \triangleq \begin{bmatrix} 0_{q_1\times(i-1)m_u} & I_{q_1\times q_1} & 0_{q_1\times(p_c-i)m_u} & 0_{q_1\times(i-1)l_y} & 0_{q_1\times q_2} & 0_{q_1\times(p_c-i)l_y} \\ 0_{q_2\times(i-1)m_u} & 0_{q_2\times q_1} & 0_{q_2\times(p_c-i)m_u} & 0_{q_1\times(i-1)l_y} & I_{q_2\times q_2} & 0_{q_2\times(p_c-i)l_y} \end{bmatrix}, \tag{22}$$

with $q_1 \triangleq n_c m_u$ and $q_2 \triangleq (n_c+\mu_c-1)l_y$. Thus, from (11) and (18) it follows that

$$Y(k) = A_y\Phi_y(k) + B_{yu}\sum_{i=1}^{p_c} L_i\theta(k-i+1)R_i\Phi_{ue}(k). \tag{23}$$

We first obtain an update law for the controller parameter block parameter for the case in which B_{yu} is known. To do this, we consider a cost function that evaluates the performance of the current value of $\theta(k)$ based upon the behavior of the system during the previous p steps. Therefore, we define the *controller based extended output* $\tilde{Y}(k)$ by

$$\tilde{Y}(k) \triangleq A_y\Phi_y(k) + B_{yu}\sum_{i=1}^{p_c} L_i\theta(k)R_i\Phi_{ue}(k), \tag{24}$$

which has the same form as (23) but with $\theta(k-i+1)$ replaced by the current block parameter vector $\theta(k)$. Note from (18), (23) and (24) that

$$\tilde{Y}(k) = Y(k) + B_{yu}\left(\sum_{i=1}^{p_c} L_i\theta(k)R_i\Phi_{ue}(k) - U(k)\right). \tag{25}$$

We now define the *tracking performance cost function*

$$J(k) \triangleq \frac{1}{2}(R(k)-\tilde{Y}(k))^{\mathrm{T}}(R(k)-\tilde{Y}(k)), \tag{26}$$

where

$$R(k) \triangleq [\, r(k) \ \cdots \ r(k-p+1)\,]^{\mathrm{T}}. \tag{27}$$

Using (23) and (26), the gradient of $J(k)$ with respect to $\theta(k)$ is given by

$$\frac{\partial J(k)}{\partial\theta(k)} = -\sum_{i=1}^{p_c} L_i^{\mathrm{T}}B_{yu}^{\mathrm{T}}(R(k)-\tilde{Y}(k))\Phi_{ue}^{\mathrm{T}}(k)R_i^{\mathrm{T}}. \tag{28}$$

This gradient is used in the update law

$$\theta(k+1) = \theta(k) - \eta(k)\frac{\partial J(k)}{\partial\theta(k)}, \tag{29}$$

where the adaptive step size $\eta(k)$ is chosen as in [7] to be

$$\eta(k) = \frac{1}{\|B_{yu}\|_{\mathrm{F}}^2\|\Phi_{ue}(k)\|_2^2}. \tag{30}$$

From (25), (28) and (30) we note that the matrix B_{yu} must be known to implement the update law (29).

3.2. Case 2: Unknown Plant Parameters

We now consider the case in which B_{yu} is not known. The controller parameter update law (29) is modified by replacing B_{yu} by an estimate that is updated at each time step using the recursive time domain identification algorithm of [8].

Define the *estimated output of the system* $\hat{Y}(k)$ by

$$\hat{Y}(k) \triangleq \hat{A}_y(k)\Phi_y(k) + \hat{B}_{yu}(k)U(k), \tag{31}$$

where $\hat{A}_y(k)$ and $\hat{B}_{yu}(k)$ are estimates at the time step k of A_y and B_{yu} respectively. Further define

$$\hat{W}_{yu}(k) \triangleq [\, \hat{A}_y(k) \quad \hat{B}_{yu}(k)\,], \tag{32}$$

to rewrite (31) as

$$\hat{Y}(k) = \hat{W}(k)\Phi_{yu}(k), \tag{33}$$

where $\Phi_{yu}(k)$ is defined as in (8). Next, the *output estimate error* $\varepsilon(k)$ is defined by

$$\varepsilon(k) \triangleq Y(k) - \hat{Y}(k), \tag{34}$$

and the *identification cost function* $J_{\rm ID}(k)$ by

$$J_{\rm ID}(k) = \frac{1}{2}\varepsilon^{\rm T}(k)\varepsilon(k). \tag{35}$$

We now use the system identification algorithm of [8] to update our estimate $\hat{B}_{yu}$ by updating $\hat{W}(k)$ as

$$\hat{W}(k+1) = \hat{W}(k) - \frac{1}{\|\Phi_{yu}(k)\|_2^2}\frac{\partial J_{\rm ID}(k)}{\partial \hat{W}(k)}. \tag{36}$$

The gradient of $J_{\rm ID}(k)$ with respect to $\hat{W}(k)$ is given by

$$\frac{\partial J_{\rm ID}(k)}{\partial \hat{W}(k)} = -U_{\rm c} \circ \varepsilon(k)\Phi_{yu}^{\rm T}(k), \tag{37}$$

where "$\circ$" denotes the Hadamard product of two matrices and $U_{\rm c}$ is a constraint matrix that preserves the structure of $\hat{W}(k)$ in the update law.

We now use the estimate of B_{yu} obtained as described above in an update law for the controller parameter block vector. First, we define the *current controller based extended output estimate* $\tilde{\hat{Y}}(k)$ by

$$\tilde{\hat{Y}}(k) \triangleq \hat{A}_y(k)\Phi_y(k) + \hat{B}_{yu}(k)\sum_{i=1}^{p_c} L_i\theta(k)R_i\Phi_{ue}(k), \tag{38}$$

and the *estimated tracking cost function*

$$\hat{J}(k) \triangleq \frac{1}{2}(R(k) - \tilde{\hat{Y}}(k))^{\rm T}(R(k) - \tilde{\hat{Y}}(k)). \tag{39}$$

From (38) and (39) it follows that the gradient of $\hat{J}(k)$ with respect to $\theta(k)$ is given by

$$\frac{\partial \hat{J}(k)}{\partial \theta(k)} = -\sum_{i=1}^{p_c} L_i^{\rm T}\hat{B}_{yu}^{\rm T}(k)(R(k) - \tilde{\hat{Y}}(k))\Phi_{ue}^{\rm T}(k)R_i^{\rm T}. \tag{40}$$

We utilize this gradient in the update law for $\theta(k)$,

$$\theta(k+1) = \theta(k) - \hat{\eta}(k)\frac{\partial \hat{J}(k)}{\partial \theta(k)}, \tag{41}$$

with the step size $\hat{\eta}(k)$ is chosen analogous to (30) to be

$$\hat{\eta}(k) = \frac{1}{\|\hat{B}_{yu}(k)\|_2^2\|\Phi_{ue}(k)\|_2^2}. \tag{42}$$

4. Algorithm Implementation

- Update the vectors $Y(k)$, $U(k)$, $\Phi_{yu}(k)$ and $\Phi_{ue}(k)$.
- Calculate $u(k)$ using (18).
- Extract the estimates $\hat{A}_{yu}(k)$ and $\hat{B}_{yu}(k)$ from $\hat{W}_{yu}(k)$.
- Calculate $\tilde{\hat{Y}}$ using (38).
- Use the estimate $\hat{B}_{yu}(k)$ in (40) and (42) to update $\theta(k)$ using (41).
- Update $\hat{W}(k)$ using (36).

5. Numerical Example

A numerical simulation of the algorithm on an example chosen from [13] pp. 235-236 is presented in this section. The objective here is to make the stick angle of an excavator follow a prescribed reference angle. The stick is connected to the bucket of the excavator and is driven by a hydraulic actuator. The dynamics of the system are described by the s-domain transfer function

$$G(s) = \frac{1000}{s(s+10)(s^2+1.2s+144)}. \tag{43}$$

Compensation is effected using the adaptive tracking algorithm described in the previous section running at a sampling frequency of 50 Hz. The normalized reference trajectory is a unit step and the initial value of the normalized angle is 0.05. The initial controller parameter vector is chosen to be a random vector and the initial value of the matrix $\hat{W}(k)$ is chosen to be the zero matrix. The time history of the closed-loop system output and the control signal are shown in Figures 2 and 3 respectively.

References

[1] S. M. Kuo and D. R. Morgan, *Active Noise Control Systems.* New York: Wiley, 1996.

[2] E. Davison and A. Goldenberg, "Robust Control of a General Servomechanism Problem: The Servo Compensator," *Automatica,* pp. 461 - 471, Vol. 11, No. 5, 1975.

[3] E. Davison and I. Ferguson, "The Design of Controllers for the Multivariable Robust Servomechanism Problem Using Parameter Optimization Methods," *IEEE Transactions on Automatic Control,* pp. 93 - 110, Vol. 26, No. 1, 1981.

[4] C. Desoer and Y. Wang, "Linear Time-Invariant Robust Servomechanism Problem: A Self Contained Exposition," *Control and Dynamics Systems,* pp. 81-129, Academic Press, 1980.

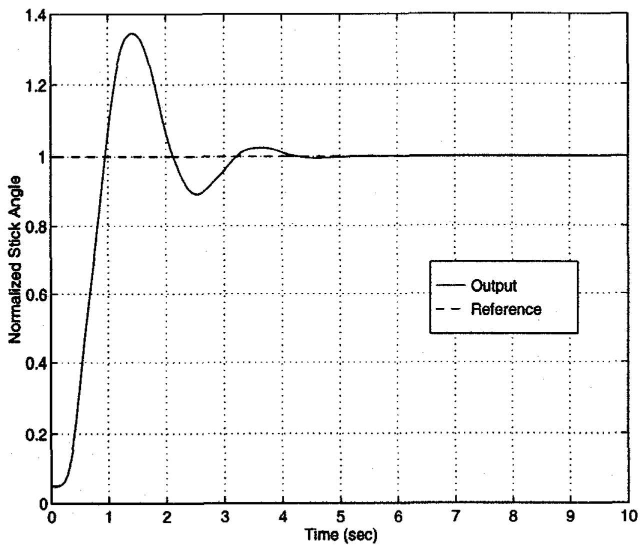

Figure 2: Unit step tracking

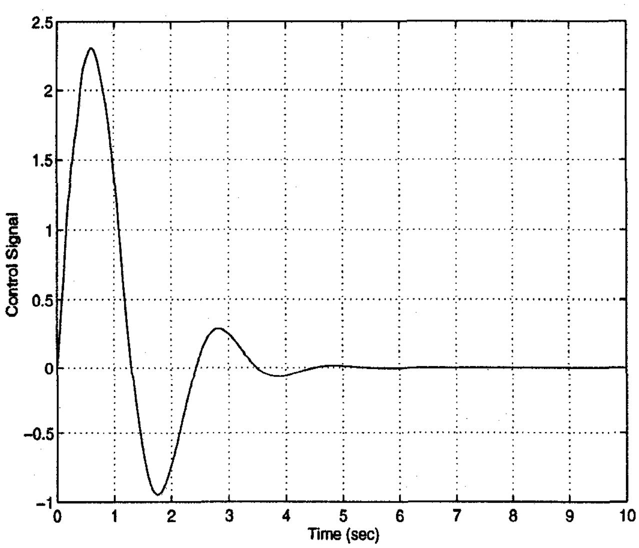

Figure 3: Control input for unit step tracking

[5] A. G. Sparks and D. S. Bernstein, "Optimal Rejection of Stochastic and Deterministic Disturbances," *ASME J. Dyn. Sys. Meas. Contr.*, pp. 140-143, Vol. 119, 1997.

[6] R. Venugopal and D. S. Bernstein, "Adaptive Disturbance Rejection Using ARMARKOV/Toeplitz Models," *Amer. Contr. Conf.*, pp. 1657-1661, Albuquerque, NM, June 1997.

[7] R. Venugopal and D. S. Bernstein, "Adaptive Disturbance Rejection Using ARMARKOV System Representations," *Conf. Dec. Contr.*, San Diego, CA, December 1997.

[8] J. C. Akers and D. S. Bernstein, "Time-Domain Identification Using ARMARKOV/Toeplitz Models," *Amer. Contr. Conf.*, pp. 191-195, Albuquerque, NM, June 1997.

[9] J. C. Akers and D. S. Bernstein, "ARMARKOV Least-Squares Identification," *Amer. Contr. Conf.*, pp. 186-190, Albuquerque, NM, June 1997.

[10] D. C. Hyland, E. G. Collins, Jr., W. M. Haddad and D. L. Hunter, "Neural Network System Identification for Improved Noise Rejection," *Proceedings of the American Control Conference, Seattle*, pp. 345 - 349, June 1995.

[11] K. J. Astrom and B. Wittenmark, *Adaptive Control*, second edition, Reading, MA: Addison-Wesley, 1995.

[12] P. P. Kanjilal, *Adaptive Prediction and Predictive Control*, Herts, U. K.:Peter Peregrinus, 1995.

[13] G. F. Franklin, J. D. Powell and M. L. Workman, *Digital Control of Dynamic Systems*, second edition, Reading, MA: Addison-Wesley, 1992.