T-81-43.

Robotica (1990) volume 8, pp 31–36

# A model algorithmic learning method for continuous-path control of a robot manipulator

Sang–Rok Oh,* Zeungnam Bien† and Il Hong Suh‡

(Received in Final Form May 6, 1989)

## SUMMARY

A new type of an iterative learning control method is proposed for dynamic systems with uncertain parameters. The method, which employs the model algorithmic control concept in the iteration sequence, is shown to be convergent for linear time-varying systems. Then the method is shown to be applicable for continuous-path control of a robot manipulator.

KEYWORDS: Algorithmic learning; Continuous path; Control; Robot.

## 1. INTRODUCTION

It is well known that dynamic control of a robot manipulator[1] is difficult because of coupling effect and high nonlinearities. In particular, precise continuous-path control of the robot becomes a complicated task if the modelled dynamics contains inexact, or rather, uncertain parameters such as backlash, friction or non-rigidity. In order to expose some of the difficulties in continuous-path control of a robot manipulator with uncertain parameters, let us suppose that the robot dynamics is given in the following form:

$$\dot{x}(t) = f(x(t), u(t); \lambda), \quad x(0) = \xi^0. \tag{1}$$

Here $x(t)$ is an $n$-dimensional state vector and $u(t)$ is an $m$-dimensional input vector of the system. The parameter vector $\lambda$ with appropriate dimension represents the uncertain parameters. Then the continuous-path control problem $(P)$ may be stated as follows:

(P): Given the desired state trajectory $x^D(t)$, $0 \le t \le T$, and the tolerance bound $\varepsilon > 0$, find an input function $u(t)$, $0 \le t \le T$, which steers the system in equation (1) such that,

$$\|x(t) - x^D(t)\|_n \le \varepsilon, \quad 0 \le t \le T \tag{2}$$

where $\|\cdot\|_n$ denotes an $n$-dimensional norm.

It may be observed that the combination of the uncertainty vector in equation (1) and the state inequality constraint in equation (2) makes the above problem quite difficult to solve. In fact, due to the unknown parameter vector $\lambda$, an optimal trajectory tracking control method[2] can not be applied for the problem.

On the other hand, no technique of adaptive control may be successful without violating the constraint in equation (2) all along the trajectory $x^D(t)$, $0 \le t \le T$. It is recalled that the adaptive control, either MRAC-type or STR-type, usually results in asymptotic convergence of error to null, if successful, and as such, the terminal state may be approximately close to the desired state but in general the intermediary behavior of the system may not be strictly controlled to satisfy the constraint in equation (2).

In this note, an iterative learning control method is proposed as a means of effectively handling the above type of problem (P). The proposed method requires that the control function $u_k(t)$, $0 \le t \le T$ applied at $k$th trial be memorized and be utilized as the information to generate the next trial control $u_{k+1}(t)$, $0 \le t \le T$.

In the literature, there are available several iterative learning control methods. In particular, Raibert[3] proposed a learning control method using a parameterization technique. The parameter learning technique in the Raibert's method shows limited application, however, due to difficulty in frequently computing the inverse of a matrix with rank-deficiency. Also Arimoto and his research group[4,5] proposed simple iterative learning control strategies with a proof of convergency. But their formulations require a rather specific knowledge of the manipulator dynamics in determining the weighting factor, and thus the method may fail if the weighting factor is not properly chosen.

The proposed method in this paper employs the structure of the "model algorithmic control"[6] or "model predictive heuristic control"[7] in constituting the iterative sequence with the time frozen. We will show that especially in the case of linear periodic time varying system, the proposed learning method is convergent under a rather weak condition and contains the result of Arimoto et al.[5] To show the validities of the proposed method, two numerical examples are illustrated, one for a linear periodic time varying system and one for a robot manipulator with nonlinear dynamics.

In the sequel, given a matrix $B$, $B^T$ denotes the transpose of $B$, and $I_n$ denotes the $n \times n$ identity matrix. Also an identity operator on a linear vector space will be represented simply by $I_L$. $C^n[0, T]$ is used to represent the normed linear space of all continuous $n$-vector functions defined on $[0, T]$ with sup-norm topology. Also

* Power Controls Lab., Kaist P.O. Box 131, Chongyangni, Seoul 136–791 (Korea).
† Dept. of Electrical Engineering, Kaist, Seoul 130–650 (Korea).
‡ Dept. of Electronic Engineering, Hanyang Univ., Seoul (Korea).

the normed linear space $D^n[0, T]$ consists of all $n$-vector functions on $[0, T]$, which are continuous and have continuous derivatives with norm defined as[8]

$$\|x\| = \sup_{0\le t\le T} \|x(t)\|_n + \sup_{0\le t\le T} \|\dot{x}(t)\|_n .$$

## II. AN ITERATIVE LEARNING CONTROL METHOD

Consider the problem (P) stated in the previous section. We will seek a solution for the problem by a repetitive try-and-learn method. For this, let $\lambda^M$ be a modelled parameter vector of $\lambda$. Also suppose that the function $f(x, u; \lambda)$ possesses the inverse function with respect to the second argument $u$,[9] that is, assume that there exists a functional relationship

$$u = g(x, z; \lambda) \tag{3}$$

such that

$$z = f(x, g(x, z; \lambda); \lambda). \tag{4}$$

In the sequel, we shall use the suffix-indexing as the iteration sequence numbering so that, for example, $x_k(t)$ denotes the state vector at time $t$ for the $k$th iteration.

The proposed iterative learning control algorithm for the nonlinear system in equation (1) is as follows:

Algorithm: For the system in equation (1) with $x(0) = \xi^0$, let there be prespecified the desired trajectory $x^D(t)$, $0 \le t \le T$, and the tolerance bound $\varepsilon > 0$.

Let $u_0(t)$, $0 \le t \le T$, be an initial control function and $x_0(t)$, $0 \le t \le T$, be the resulting trajectory of the system in equation (1).

Let

$$x_0^L(t) = x_0(t), \quad 0 \le t \le T.$$

Let $S$ be an $n \times n$ diagonal positive definite matrix.

A1. Let $k = 0$.

A2. Let

$$e_k(t) = x^D(t) - x_k(t), \quad 0 \le t \le T. \tag{5}$$

A3. If $\|e_k(t)\|_n \le \varepsilon$, stop. In this case $u_k(t)$, $0 \le t \le T$, is the desired control. Otherwise, go to next step A4.

A4. Let

$$x_{k+1}^L(t) = x_k^L(t) + Se_k(t) \tag{6}$$

$$u_{k+1}(t) = g(x_{k+1}^L(t), \dot{x}_{k+1}^L(t); \lambda^M). \tag{7}$$

A5. Apply $u_{k+1}(t)$ to system in equation (1) to obtain $x_{k+1}(t)$.

A6. Let $k = k + 1$ in computer algorithmic sense, that is, let $u_k(t)$, $x_k(t)$ and $x_k^L(t)$ be replaced by $u_{k+1}(t)$, $x_{k+1}(t)$ and $x_{k+1}^L(t)$, respectively.

Go to step A2.

*Remark 1:* The algorithm proposed in equations (5)–(7) can be thought of as a special case of an extended form of the model algorithmic control (MAC).[6] That is, if we identify equation (6) with equation (5) as the control signal generator, then the algorithm in this note becomes a 'MAC' algorithm with one-step closed-loop prediction, applied for a nonlinear dynamic system. Note, however, that the algorithm in this paper utilizes the inverse dynamic system structure (see equation (3) and equation (7)) as the explicit control signal generator instead of the explicit system model with on-line computation of optimizing control signal.[6] Also the input to the inverse dynamic system is chosen as one-step prediction trajectory to reach to the desired trajectory. Further, the 'MAC' scheme of the proposed algorithm is performed in the domain of iteration sequence with the time frozen, not as in the time-domain and hence this controller has the learning capability in the sense that the current input to be applied utilizes the memorized information of the past input of the previous iteration.

*Remark 2:* Note that in adaptive control schemes, the error signal $e_k(t)$ of the type in equation (5) is utilized in generating the current input $u_k(t)$, while, rather differently, the scheme presented here utilizes $e_k(t)$ in generating $u_{k+1}(t)$, $0 \le t \le T$, and thus requires memorization of the whole segment $e_k(t)$, $0 \le t \le T$. It is remarked that other differences between 'MAC'-algorithm and MRAC scheme are also discussed in the reference.[10]

Now suppose the system in equation (1) is linear and time-varying but periodic;

$$\dot{x}(t) = A(t; \lambda(t))x(t) + B(t; \lambda(t))u(t), \qquad x(0) = \xi^0, \quad 0 \le t \le T \tag{8}$$

where the $n \times n$ matrix function $A(t; \lambda(t))$ and the $n \times m$ matrix function $B(t; \lambda(t))$ are periodic such that $A(t + T; \lambda(t + T)) = A(t; \lambda(t))$ and $B(t + T; \lambda(t + T)) = B(t; \lambda(t))$, and in advance, denoted as $A$ and $B$ for brevity. Also, let $A_M$ and $B_M$ are defined to be equal to $A(t; \lambda^M(t))$ and $B(t; \lambda^M(t))$, respectively, where $\lambda^M(t), 0 \le t \le T$, is a model of the time-varying parameter vector $\lambda(t)$. Then under the assumption that the rank of the $n \times m$ matrix $B$ is $m$ for each $t \in [0, T]$, the control signal generator in equation (7) is given as

$$u_{k+1}(t) = [B_M^T B_M]^{-1} B_M^T [\dot{x}_{k+1}^L(t) - A_M x_{x+1}^L(t)] \tag{9}$$

with

$$x_{k+1}^L(t) = x_k^L(t) + Se_k(t), \quad x_0^L(t) = x_0(t). \tag{10}$$

In this case, we can show that the proposed learning control algorithm for the linear time-varying system is convergent. For this, define the linear operator[13] $L: C^m[0, T] \to D^n[0, T]$ by

$$[L(u)](t) = \int_0^t \psi(t)\psi^{-1}(\tau)Bu(\tau)\, d\tau, \tag{11}$$

where $\psi(t)$ is the unique solution of the fundamental matrix differential equation described by

$$\dot{\psi}(t) = A\psi(t), \quad \psi(0) = I_n \tag{12}$$

Similarly, introduce the operator $P^m: D^n[0, T] \to C^m[0, T]$ for the inverse dynamic model in equation (9) in such a way that

$$[P^M(x^L)](t) = [B_M^T B_M]^{-1} B_M^T [\dot{x}^L(t) - A_M x^L(t)] \tag{13}$$

Then the convergency of the proposed algorithm can be shown as follows:

Theorem 1: Consider the linear periodically time-varying system in equation (8). If $x_k(0) = \xi^0$ for every $k$ and the inequality condition

$$\|I_L - LP^M S\| < 1 \tag{14}$$

holds for some $n \times n$ matrix $S$, then the iterative learning controller yields

$$\lim_{k\to\infty} \|e_k\| = 0.$$

*Proof.* From the definition of $L$ and $P^M$, we can obtain

$$[L(P^M(x_k^L))](t) = x_k(t) - \psi(t)\xi^0.$$

Now, observe that

$$\begin{aligned}\|x_{k+1} - x^D\| &= \|L(P^M x_{k+1}^L) - x^D + \psi\xi^0\| \\ &= \|x_k - \psi\xi^0 - x^D + L(P^M S e_k) + \psi\xi^0\| \\ &\le \|(I_L - LP^M S\| \, \|x_k - x^D\|\end{aligned}$$

Defining

$$\rho \triangleq \|I_L - LP^M S\|,$$

we conclude from the assumption of $0 \le \rho \le 1$, that

$$\|e_{k+1}\| \le \rho \, \|e_k\|.$$

This completes the proof.

*Remark 3:* The weighting matrix $S$ in equation (10) and equation (14) may be considered as a compensation factor for model discrepancy between the system parameters and the model parameters. Experiences show that even when the modelling error is not small, the proposed algorithm can be convergent by taking small values of $s_{ij}$ in $S$. In this case the convergence speed may be slow. The weighting matrix $S$ may be chosen as a diagonal matrix whose elements satisfy $0 < s_{ii} \le 1$, for $i = 1, 2, \ldots, n$. It is also remarked that the condition in equation (14) for some $S$ is roughly equivalent to a 'passivity' condition on the operator $LP^M$[11].

*Remark 4:* If we let $S$ be equal to identity matrix and $A_M$ be equal to null matrix, then our algorithm is equivalent to the learning algorithm proposed by Arimoto et al.[4] Hence we may say that our algorithm includes the Arimoto's algorithm.

*Remark 5:* In 'indirect' adaptive control of systems with uncertainty, the control signal is generated based on estimation and updated value of the uncertain parameter $\lambda$.[12] But if the linear system contains time-varying uncertain parameter $\lambda(t)$ the conventional adaptive control cannot be effective in reducing error all along the trajectory as the proposed algorithm works.

## III. EXAMPLES OF APPLICATION FOR ROBOT MANIPULATORS

The iterative learning control algorithm presented in the previous section can be applied for continuous-path control of a robot manipulator. To show the effectiveness of the proposed method, digital computer simulations are performed for a linear time-varying system[13] representing the linearized dynamics of a robot manipulator with respect to a nominal trajectory[1] and for a two link robot manipulator with nonlinear dynamics.

*Example 1:* A Linear Time-Varying System.

Consider a linear time-varying plant whose dynamics is described by

$$\begin{bmatrix}\dot{x}^1(t)\\ \dot{x}^2(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ -(2+5t) & -(3+2t)\end{bmatrix}\begin{bmatrix}x^1(t)\\ x^2(t)\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix}u(t),$$
$$\begin{bmatrix}x^1(0)\\ x^2(0)\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix}, \quad 0 \le t \le 1 \tag{15}$$

Let the desired state trajectory $x^D(t)$ for $t \in [0, 1]$ be given by

$$x^D(t) = \begin{bmatrix} t^3(4-3t)\\ 12t^2(1-t)\end{bmatrix}, \tag{16}$$

and the tolerance bound $\varepsilon = 0.005$. To apply the learning control method proposed in previous section, let $A$ and $B$ be modelled as

$$A^M = \begin{bmatrix}0 & 1\\ -(1+2.5t) & -(1.5+t)\end{bmatrix}, \quad B^M = \begin{bmatrix}0\\0.5\end{bmatrix}. \tag{17}$$

Also, let the initial input $u_0(t)$ for $t \in [0, 1]$ and $S$ in equation (10) be given as

$$u_0(t) = 0, \quad 0 \le t \le 1,$$
$$S = \begin{bmatrix}0.6 & 0\\ 0 & 0.6\end{bmatrix}, \tag{18}$$

Then we can obtain $x_k^i(t)$, $i = 1, 2$, for $k = 1, 2, \ldots$ as shown in Figure 1. It is noted from Figure 1 that even in the case of relatively large parameter discrepancies as assumed in equation (15) and equation (17), we would observe that five iterations were sufficient to get within $\varepsilon$ bound for each $t \in [0, 1]$ to the desired state trajectory $x^D(t)$.

For comparison, we applied the learning control method of Arimoto et al.[4] to the above system, and found that with the learning control of the form

$$u_{k+1}(t) = u_k(t) + \Gamma \dot{e}_k(t), \quad 0 \le t \le T, \tag{19}$$

the maximum absolute output error between the desired trajectory and the system trajectory was 0.06 even with 10 iterations as shown in Figure 2. In equation (19), the design parameter $\Gamma$ was chosen to be $[0\,1]$ which is the best parameter minimizing $\|I - CB\Gamma\|$.

Consider the dynamics of an $n$-joint manipulator described by the vector differential equation of the form:

$$\tau = D(q;\lambda)\ddot{q} + H(q,\dot{q};\lambda) + G(q;\lambda), \tag{20}$$

where $\tau$ is an $n \times 1$ applied torque vector to the joint

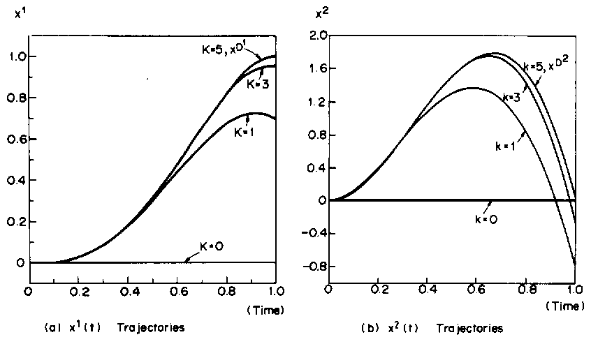



Fig. 1. State Trajectories by the Proposed Learning Algorithm. (a) $X^1(t)$ Trajectories, (b) $X^2(t)$ Trajectories.

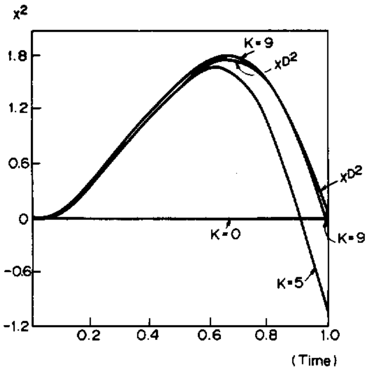



Fig. 2. Output Trajectories by the Arimoto's Algorithm.

actuators, and $q$, $\dot{q}$ and $\ddot{q}$ are the $n \times 1$ angular position, velocity and acceleration vectors, respectively. Let

$$x(t) \triangleq (q^T(t), \dot{q}^T(t))^T,$$
$$u(t) \triangleq (\tau_1(t), \tau_2(t), \ldots, \tau_n(t))^T. \tag{21}$$

Then equation (20) can be expressed in state space representation as

$$\dot{x}(t) = f(x(t), u(t); \lambda).$$

In this way, we can apply the algorithm in sec. 2 for the continuous-path control of a robot manipulator. For simplicity, we will demonstrate the effectiveness via 2-dof manipulator.

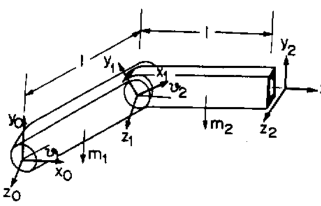



Fig. 3. A Two-Link Robot Arm.

*Example 2:* A Two Link Robot Manipulator.

Consider a two link robot manipulator with revolute joints as shown in Figure 3, whose dynamic equations of motion are given in the following:[1]

$$\begin{bmatrix} \tau^1 \\ \tau^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{3}m_1l^2 + \frac{4}{3}m_2l^2 + m_2c_2l^2 & \frac{1}{3}m_2l^2 + \frac{1}{2}m_2c_2l^2 \\ \frac{1}{3}m_2l^2 + \frac{1}{2}m_2c_2l^2 & \frac{1}{3}m_2l^2 \end{bmatrix} \times \begin{bmatrix} \ddot{\theta}^1 \\ \ddot{\theta}^2 \end{bmatrix} + \begin{bmatrix} -\frac{1}{2}m_2s_2l^2\dot{\theta}^{2^2} - m_2s_2l^2\dot{\theta}^1\dot{\theta}^2 \\ \frac{1}{2}m_2s_2l^2\dot{\theta}^{1^2} \end{bmatrix} + \begin{bmatrix} -\frac{1}{2}m_2glc_1 - \frac{1}{2}m_2glc_{12} - m_2glc_1 \\ -\frac{1}{2}m_2glc_{12} \end{bmatrix} \tag{22}$$

Here $c_i$, $s_i$ and $c_{ij}$ are defined by $\cos\theta^i$, $\sin\theta^i$ and $\cos(\theta^i + \theta^j)$, respectively. In equation (24), $\theta^1$ and $\theta^2$ are joint variables, and the link masses and lengths for the simulation are given in Table I. Let the desired trajectory for $t \in [0, 1]$ and tolerance bound $\varepsilon$ be given by

$$\theta^D(t) = \begin{bmatrix} -\frac{5}{3}t^3 + \frac{5}{2}t^2 \\ -\frac{5}{3}t^3 + \frac{5}{2}t^2 \end{bmatrix}, \tag{23}$$

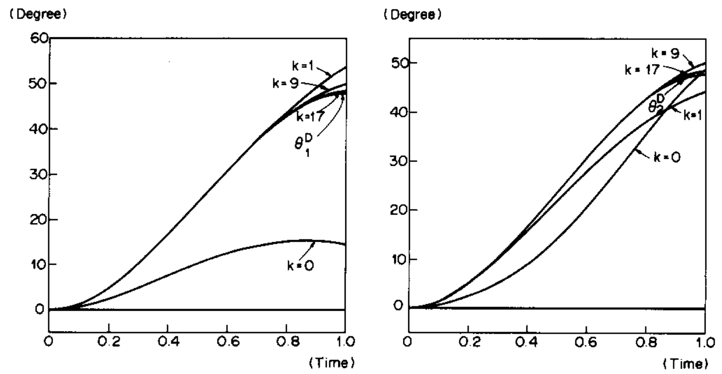
and

$$\varepsilon = 1.0\,[\text{Degree}] = 0.017\,[\text{Radian}]. \tag{24}$$

To generate the control signal in equation (7) for the two link robot manipulator, it may be necessary to obtain the explicit model. But the perfect modelling process may be very difficult and time-consuming. Thus to avoid such complexities, we adopt a rather simple explicit model

Table I. Input Data for Simulation

| value / Parameter | True Value | Modelled Value |
|---|---|---|
| mass of link 1 | $m_1 = 2$ kg | $m_1^M = 1.5$ kg |
| mass of link 2 | $m_2 = 2$ kg | $m_2^M = 1.5$ kg |
| length | $l = 0.5$ m | $l^M = 0.4$ m |




(a) Joint Angle 1 (b) Joint Angle 2

Fig. 4. Joint Angle Trajectories by the Proposed Learning Algorithm. (a) Joint Angle 1, (b) Joint Angle 2.

considering only inertial force term given by

$$\begin{bmatrix} \tau^1 \\ \tau^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{3}m_1^M l^{M^2} + \frac{4}{3}m_2^M l^{M^2} + m_2^M c_2 l^{M^2} & \frac{1}{3}m_2^M l^{M^2} + \frac{1}{2}m_2^M l^{M^2} \\ \frac{1}{3}m_2^M l^{M^2} + \frac{1}{2}m_2^M c_2 l^{M^2} & \frac{1}{3}m_2^M l^{M^2} \end{bmatrix} \times \begin{bmatrix} \ddot{\theta}^{M^1} \\ \ddot{\theta}^{M^2} \end{bmatrix} \quad (25)$$

where in equation (25) the link masses and lengths for the explicit model are given in Table I. Also, let the initial input $\tau_0(t)$ for $t \in [0, 1]$ and $S$ in equation (6) be given as

$$\tau_0(t) = D(\theta^D; \lambda^M)\ddot{\theta}^D(t), \quad 0 \le t \le 1,$$

$$S = \begin{bmatrix} 0.95 & 0 \\ 0 & 0.95 \end{bmatrix}. \quad (26)$$

Then we can obtain $\theta_k^i(t)$, for $i = 1, 2$ for $k = 1, 2, \ldots,$ as shown in Figure 4. It is observed from Figure 4 that even in the case of large parameter uncertainties as in Table I and moreover even with the poor explicit model as in equation (25), seventeen iterations make the maximum absolute error be less than $\varepsilon$ bound for $i = 1, 2$ and for every $t \in [0, 1]$.

It is finally remarked that the proposed algorithm was applied for continuous-path control of a robot manipulator and tested in real-time successfully.[14]

## IV. CONCLUDING REMARKS

A new iterative learning control algorithm for a class of nonlinear dynamic systems was proposed by employing the model algorithmic control method in the domain of iteration sequence with the time frozen. Especially, in the case of a linear periodic time varying system, convergency of the proposed algorithm was shown. The proposed control method results in an iterative learning process in the sense that the subsequent system performance is improved by using the process data of the previous iteration. Also by the computer simulations of a linear time-varying system and a two link robot manipulator, the proposed controller was proven to be a learning process as the number of iterations were increased.

It is remarked that the sufficient condition for convergency in the Theorem 1 may need to be more refined together with further study on the system stability and convergence property of the algorithm for the nonlinear dynamic systems.

## References

1. C.S.G. Lee, R.C. Gonzalez and K.S. Fu, *Tutorial on Robotics* (IEEE Computer Society Press, Los Angeles, 1983).
2. K. Makowski and L.W. Neustadt, "Optimal Control Problems with Mixed Control-Phase Variable Equality and Inequality Constraints" *SIAM J. Control* **12,** 184–228 (1974).
3. M.H. Raibert, "Motor Control and Learning by the State Space Model" *Ph.D. Dissertation* (Dept. of Psychology, MIT 1977).
4. S. Arimoto, S. Kawamura and F. Miyazaki, "Bettering Operation of Robots by Learning" *J. Robotics Systems* **1,** No. 2, 123–140 (1984).
5. S. Arimoto, S. Kawamura and F. Miyazaki, "Can Mechanical Robots Learn by Themselves?" *Proc. 2nd Int. Symp. Robotics Res.* 127–134, Kyoto, Japan (August, 1984).
6. R. Rouhani and R.K. Mehra, "Model Algorithmic Control (MAC): Basic theoretical Properties" *Automatica* **18,** No. 4, 401–414 (1982).
7. J. Richalet, A. Rault, J.L. Testud and J. Papon, "Model Predictive Heuristic Control: Application to Industrial Processes" *Automatica* **14,** 413–428 (1978).
8. D.G. Luenberger, *Optimization by Vector Space Methods* (John Wiley & Sons, New York, 1969).
9. W.H. Fleming, *Functions of Several Variable* (Addison-Wesley, Massachusetts, 1965).
10. R.K. Mehra, R. Rouhani and L. Praly, "New Theoretical Developments in Multivariable Predictive Algorithmic Control" *Proc. JACC* **FA9-B** (1980).
11. C.A. Desoer and M. Vidyasagar, *Feedback Systems: Input–Output Properties* (Academic Press, New York, 1975).

12. A.J. Koivo and T.H. Guo, "Adaptive Linear Controller for Robotic Manipulators" *IEEE Trans. Automat. Contr.* **AC-28,** NO. 2, 162–171 (1983).
13. C.T. Chen, *Introduction to Linear System Theory* (Holt, Rinehart and Winston, New York, 1970).
14. D.S. You, M.J. Chung and Z. Bien, "Real-time implementation and evaluation of dynamic control algorithms for industrial manipulators," *Proc. IECON*87, *SPIE* **853,** 26–31, Cambridge, Massachusetts, USA (November, 1987).