A Discrete-Time Stochastic Learning Control Algorithm

Samer S. Saab, Senior Member, IEEE

Abstract—One of the problems associated with iterative learning control algorithms is the selection of a “proper” learning gain matrix for every discrete-time sample and for all successive iterations. This problem becomes more difficult in the presence of random disturbances such as measurement noise, reinitialization errors, and state disturbance. In this paper, the learning gain, for a selected learning algorithm, is derived based on minimizing the trace of the input error covariance matrix for linear time-varying systems. It is shown that, if the product of the input/output coupling matrices is full-column rank, then the input error covariance matrix converges uniformly to zero in the presence of uncorrelated random disturbances. However, the state error covariance matrix converges uniformly to zero in presence of measurement noise. Moreover, it is shown that, if a certain condition is met, then the knowledge of the state coupling matrix is not needed to apply the proposed stochastic algorithm. The proposed algorithm is shown to suppress a class of nonlinear and repetitive state disturbance. The application of this algorithm to a class of nonlinear systems is also considered. A numerical example is included to illustrate the performance of the algorithm.

Index Terms—Iterative learning control, optimal control, stochastic control.

I. INTRODUCTION

ITERATIVE learning control algorithms are reported and applied to fields topping robotics [4]–[7], and [23]–[36], for example, precision speed control of servomotors and its application to a VCR [1], cyclic production process with application to extruders [2], and coil-to-coil control in rolling [3]. Several approaches for updating the control law with repeated trials on identical tasks, making use of stored preceding inputs and output errors, have been proposed and analyzed [4]–[38]. Several algorithms assume that the initial condition is fixed; that is, at each iteration the state is initialized always at the same point. For a learning procedure that automatically accomplishes this task, the reader may refer to [28] and [17]. For algorithms that employ more than one history data, refer to [15] and [16]. A majority of these methodologies are based on contraction mapping requirements to develop sufficient conditions for convergence and/or boundedness of trajectories through repeated iterations. Consequently, a condition on the learning gain matrix K is assumed in order to satisfy a necessary condition for contraction. Based on the nature of the control update law, conditions on K, such as \(|I-KL| < 1\) or \(|I-LK| < 1\), are imposed (where the matrix L is a function of input/output coupling functions to the system) [4]–[20]. Several methods are proposed to find an “appropriate” value of the gain matrix K. For example, the value of K is found based on gradient methods to minimize a quadratic cost error function between successive trials [35], another is based on providing a steepest-descent minimization of the error at each iteration [37], an additional is achieved by choosing larger values of the gain matrix until a convergence property is attained [38]. Another methodology, based on frequency response technique, is also presented in [25]. On the other hand, for task-level learning control, an on-line numerical multivariate parametric nonlinear least square optimization problem is formulated and applied to a two-link flexible arm [24]. Then again, these methods are based on deterministic approach. To the best of the author’s knowledge, this paper presents the first attempt to formulate a computational algorithm for the learning gain matrix, which stochastically minimizes, in a least-square sense, trajectory errors in presence of random disturbances.

This paper presents a novel stochastic optimization approach, for a selected learning control algorithm, in which the learning gain minimizes the trace of the input error covariance matrix. A linear discrete time-varying system, with random state reinitialization, additive random state disturbance and biased measurement noise, is initially considered. The selected control update law \(u(t, k+1)\), at a sampled time t and iterative index \(k+1\), consists of the previous stored control input added to weighted difference of the stored output error, that is \(u(t, k+1) = u(t, k) + K(t, k)[e(t+1, k) - e(t, k)]\). Although this update law possesses a derivative action, the proposed algorithm is shown to become insensitive to measurement noise throughout iterations. The problem can be interpreted as follows: assume that the random disturbances are white Gaussian noise and uncorrelated, then, at time t and iterative index k, find a learning gain matrix K\((t, k)\), such that the trace of the input error covariance matrix at \((t, k+1)\) is minimized. It is shown [11] that if K\((t, k)\) is chosen such that \(|I-K(t,k)C(t+1)B(t)| \leq \rho < 1\), then boundedness of trajectories of this control law is guaranteed. A choice of this learning gain matrix requires the matrix C\((t+1)B(t)\) to be full-column rank. In addition, in absence of all disturbances, uniform convergence of all trajectories is also guaranteed if and only if C\((t+1)B(t)\) is full-column rank. This latter scenario is not realistic because there are always random disturbances in the system in addition to measurement noise. Unfortunately, deterministic learning algorithms cannot learn random behaviors. That is, after a finite number of iterations, the repetitive errors are attenuated and the dominating errors will be due to those random disturbances and measurement noise, which cannot be learned or mastered in a deterministic approach. In this paper, we show that if C\((t+1)B(t)\) is...
full-column rank, then the algorithm that generates the learning gain matrix, which minimizes the trace of the input error covariance matrix automatically assures the contraction condition, that is \(|I - K(t, k)C(t + 1)B(t)| < 1\), which ensures the suppression of repeatable or deterministic errors [11], [14]. Moreover, it is shown that the input error covariance matrix converges uniformly to zero in presence of random disturbances, where the state error covariance matrix converges uniformly to zero in presence of biased measurement noise. Obviously, if the input error is zero and in the case where reinitialization errors are nonzero, then one should not expect this type of learning control to suppress such errors. Furthermore, if a zero-convergence of the state error covariance matrix is desired in presence of state disturbance, then the controller (if any) is expected to “heavily” employ full knowledge of the system parameters, which makes the use of learning algorithms unattractive. On the other hand, it is shown that if a certain condition is met, then knowledge of the state matrix is not required. Application of this algorithm to a class of nonlinear system is also examined. In one case where the nonlinear state function does not require the Lipschitz condition, unlike the results in [5]–[9], [12]–[15], and [24], it is shown that the input error covariance matrix is nonincreasing. In addition, if a certain condition is met, then the results for this class of systems become similar to the case of linear time-varying systems. A class of nonlinear systems satisfying Lipschitz condition is also considered.

The outline of the paper is as follows. In Section II, we formulate the problem and present the assumptions of the disturbance characteristics. In Section III, we develop the necessary learning gain that minimizes the trace of the input error covariance matrix, and present the propagation formula of this covariance matrix through iterations. In Section IV, we assume that \(C(t + 1)B(t)\) is full-column rank and present the convergence results. In Section V, we present a pseudo-code for computer application, and give a numerical example. We conclude in Section VI. In Appendix A, we show that if a certain condition is met, then the knowledge of the state matrix is not required. In Appendix B, we apply this type of stochastic algorithm to a class of nonlinear systems.

II. PROBLEM FORMULATION

Consider a discrete-time-varying linear system described by the following difference equation:

\[
x(t + 1, k) = A(t)x(t, k) + B(t)u(t, k) + w(t, k)
\]
\[
y(t, k) = C(t)x(t, k) + v(t, k) + \nu_b(k)
\]

where \(t \in [0, n_t]; \quad x(t, k) \in \mathbb{R}^n; \quad u(t, k) \in \mathbb{R}^m; \quad y(t, k) \in \mathbb{R}^p; \quad \nu_b(k) \in \mathbb{R}^q; \quad v(t, k) \in \mathbb{R}^q. \quad \nu_b(k) \in \mathbb{R}^q.

The learning update is given by

\[
u(t, k + 1) = u(t, k) + K(t, k)[c(t + 1, k) - c(t, k)]
\]

where \(K(t, k)\) is the \((p \times q)\) learning control gain matrix, and \(c(t, k)\) is the output error; i.e., \(c(t, k) = y(t) - y(t, k)\) where \(y(t)\) is a realizable desired output trajectory. It is assumed that for any realizable output trajectory and an appropriate initial condition \(x_d(0)\), there exists a unique control input \(u_d(t) \in \mathbb{R}^m\) generating the trajectory for the nominal plant. That is, the following difference equation is satisfied:

\[
x_d(t + 1) = A(t)x_d(t) + B(t)u_d(t)
\]
\[
y_d(t) = C(t)x_d(t).
\]

Note that if \(C(t + 1)B(t)\) is full-column rank and \(y_d(t)\) is a realizable output trajectory, then a unique input generating the output trajectory is given by

\[
u_d(t) = [(C(t + 1)B(t))^TC(t + 1)B(t)]^{-1}(C(t + 1)B(t))^TC_d(t + 1) - C(t + 1)A(t)x_d(t)\]

Define the state and the input error vector by \(\delta x(t, k) \equiv x(t, k) - x(t),\) and \(\delta u(t, k) \equiv u(t) - u(t, k),\) respectively.

Assumptions: Denote \(E\) and \(E_k\) to be the expectation operators with respect to time domain, and iteration domain, respectively. Let \(\{u(t, k)\}\) and \(\{v(t, k)\}\) be sequences of zero-mean white Gaussian noise such that \(E(u(t, k)u^T(t, k)) = Q_t\) is a positive–semidefinite matrix, \(E(v(t, k)v^T(t, k)) = R_t\) is a positive–definite matrix for all \(k\), and \(E(u(t_i, k_m)v(t_j, k_q) = E_k(u(t_i, k_m)v(t_j, k_q)) = 0\) for all indices \(i, j, m,\) and \(l, i.e., the mean measurement error sequence \(v(\cdot, \cdot)\) have zero crosscorrelation with \(w(\cdot, \cdot)\) at all times. The initial state error \(\delta x(0, k)\) and the initial input error \(\delta u(0, 0)\) are also assumed to be zero-mean white noise such that \(E(\delta x(0, k)\delta x(0, k)^T) = P_x,0\) is a positive–semidefinite matrix, and \(E(\delta u(0, 0)\delta u(0, 0)^T) = P_u,0\) is a symmetrical positive–definite matrix. One simple scenario is to set \(\delta u(t, 0) \equiv 0 \forall t.\) Moreover, \(\delta x(0, k)\) is uncorrelated with \(\delta u(t, 0), u(0, k),\) and \(v(0, k).\) The main target is to find a proper learning gain matrix \(K(t, k)\) such that the input error variance is minimized. Note that since the disturbances are assumed to be Gaussian processes, then uncorrelated disturbances are equivalent to independent disturbances.

III. STOCHASTIC LEARNING CONTROL ALGORITHM DEVELOPMENT

In this section, we develop the appropriate learning gain matrix, which minimizes the trace of the input/state error covariance matrix. In the following, we combine the system given by (1) and (2) in a regressor form. We first write expressions for the state and input errors. The state error at \(t + 1\) is given by

\[
\delta x(t + 1, k) = A(t)\delta x(t, k) + B(t)\delta u(t, k) - w(t, k)
\]

and the input error for iterate \(k + 1\) is given by

\[
\delta u(t, k + 1)
\]
\[
= \delta u(t, k) - K(t, k)
\cdot \{C(t + 1)[x_d(t + 1) - x(t + 1, k) - w(t + 1, k)] + K(t, k)[C(t)[x_d(t) - x(t, k)] - v(t, k)]\}.
\]
Substituting the values of the state errors in the previous equation, we have

\[
\delta u(t, k + 1) = \delta u(t, k) - K(t, k)C(t + 1)B(t)\delta u(t, k)
\]

\[
+ K(t, k)[C(t) - C(t + 1)A(t)]\delta x(t, k)
\]

\[
+ K(t, k)[v(t + 1, k) - v(t, k) + C(t + 1)w(t, k)].
\]

Collecting terms, the input error yields to

\[
\begin{align*}
\delta u(t, k + 1) &= [I - K(t, k)C(t + 1)B(t)]\delta u(t, k) \\
&+ K(t, k)[C(t) - C(t + 1)A(t)]\delta x(t, k) \\
&+ K(t, k)[v(t + 1, k) - v(t, k) + C(t + 1)w(t, k)].
\end{align*}
\]

(5)

Combining (4) and (5) in the two-dimensional Roesser Model [39] as shown in (6) at the bottom of the page. The following is justified because the input vector of difference equation (6) has zero mean. Note that if disturbances are nonstationary or colored noise, then the system can easily augmented to cover this class of disturbances. Writing (6) in a compact form, we have

\[
X^+ = \Phi X + \Gamma Z
\]

(7)

where

\[
\begin{bmatrix}
\delta u(t, k + 1) \\
\delta x(t + 1, k)
\end{bmatrix}
\]

are \((n + p) \times 1\) vectors

\[
Z =
\begin{bmatrix}
u(t, k) \\
v(t + 1, k) - v(t, k)
\end{bmatrix}
\]

is an \((n + q) \times 1\) vector

\[
\Phi =
\begin{bmatrix}
[I - K(t, k)C(t + 1)B(t)] & K(t, k) & [C(t) - C(t + 1)A(t)] & A(t)
\end{bmatrix}
\]

is an \((n + p) \times (n + p)\) matrix, and

\[
\Gamma = 
\begin{bmatrix}
K(t, k)C(t + 1) & K(t, k) \\
0 & -I
\end{bmatrix}
\]

is an \((n + p) \times (n + q)\) matrix.

Next, we attempt to find a learning gain matrix \(K(t, k)\) such that the trace of the error covariance matrix \(P^+ \triangleq E(X^+X^+^T)\) is minimized. It is implicitly assumed that the input and state errors have zero mean, so it is proper to refer to \(P^+\) as a covariance matrix. Note that the mean-square error is used as the performance criterion because the trace is the summation of error variances of the elements of \(X^+\). Toward this end, we first form the error covariance matrix

\[
P^+ = E[(\Phi X + \Gamma Z)(\Phi X + \Gamma Z)^T]
\]

\[
= \Phi E[XX^T]\Phi^T + \Phi E[XZ^T]\Gamma^T + \Gamma E[ZX^T]\Phi^T + \Gamma E[ZZ^T]\Gamma^T.
\]

Since the initial error state \(\delta x(0, k)\) is uncorrelated with the disturbances \(u(t, k)\) and \(v(t, k)\), then at iterate \(k\),

\[
E[\delta u(t, k)u(t, k)^T] = 0, \quad \text{and} \quad E[\delta x(t, k)v(t + 1, k) - v(t, k)^T] = 0.
\]

Thus, \(E[XX^T] = 0, \quad \text{and} \quad E[ZX^T] = 0\). The last equation is now reduced to

\[
P^+ = \Phi E[XX^T]\Phi^T + \Gamma E[ZZ^T]\Gamma^T.
\]

(8)

Let

\[
P = E[XX^T] \triangleq \begin{bmatrix} P_{uu, k} & P_{ux, k} \\ P_{ux, k}^T & P_{xx, k} \end{bmatrix}
\]

where

\[
P_{uu, k} = E[\delta u(t, k)\delta u(t, k)^T]
\]

\[
P_{ux, k} = E[\delta u(t, k)\delta x(t, k)^T]
\]

and

\[
P_{xx, k} = E[\delta x(t, k)\delta x(t, k)^T],
\]

As a consequence, \(P_{uu, k}\) and \(P_{xx, k}\) are symmetrical and positive–semidefinite matrices.

\[
E[ZZ^T] = \begin{bmatrix} Q_k & 0 \\ 0 & R_k + R_{k+1} \end{bmatrix}
\]

where the zero submatrices are due to zero crosscorrelation between \(u(\cdot, \cdot)\) and \(v(\cdot, \cdot)\). For compactness, we denote

\[
K_k \triangleq K(t, k), \quad A \triangleq A(t), \quad B \triangleq B(t), \quad C \triangleq C(t + 1), \quad C \triangleq C(t), \quad \Phi_1 \triangleq I - K_kC^+B, \quad \Phi_2 \triangleq K_k(C - C^+A).
\]

Expanding on the left hand terms of (8), we get the equation shown at the bottom of the next page. Consequently, the trace of \(P^+\) is equivalent to the following:

\[
\text{trace} (P^+)
\]

\[
= \text{trace} \left\{ (I - K_kC^+B)P_{uu, k} (I - K_kC^+B)^T \ight. \\
+ (I - K_kC^+B)P_{ux, k}K_k(C - C^+A)^T \\
+ [K_k(C - C^+A)]P_{xx, k}[K_k(C - C^+A)]^T \\
+ [K_k(C - C^+A)]P_{xx, k}K_k(C - C^+A)^T \\
+ [K_kC^+B]P_{uu, k}B^T + AP_{uu, k}B^T + BP_{uu, k}A^T \\
+ AP_{xx, k}A^T + K_kC^+Q_kC^+K_k^T \\
+ K_k(R_k + R_{k+1})K_k^T + Q_k \right\}.
\]
Expanding and rearranging the terms on the right-hand side of the last equation, we get

\[
\text{trace } (P^+) = \text{trace } \left\{ K_k \left[ C^+ B P_{uk} B^T C^{+T} - (C - C^+ A)^T P_{ux, k} C^{+T} + C^+ B P_{uk} (C - C^+ A)^T + (C - C^+ A) P_{x, k} (C - C^+ A)^T \right. \right.
\]
\[
\left. \left. + C^+ Q_t C^{+T} + R_t + R_{t+1} \right] \right\}
\]

\[
+ K_k \left[ -C^+ B P_{uk} + (C - C^+ A) P_{ux, k} \right] K_k^T
\]
\[
+ P_{uk} + BP_{uk} B^T + AP_{ux, k} B^T
\]
\[
+ BP_{ux, k} A^T + AP_{x, k} A^T + Q_t \right\}. \tag{9}
\]

Setting

\[
H \triangleq \begin{bmatrix} -C^+ B & C - C^+ A \end{bmatrix}, \quad P1_k \triangleq \begin{bmatrix} P_{uk} & P_{ux, k} \end{bmatrix}
\]
and
\[
M1_k \triangleq P_{uk} + BP_{uk} B^T + AP_{ux, k} B^T
\]
\[
+ BP_{ux, k} A^T + AP_{x, k} A^T + Q_t \]
then the above equation is reduced to

\[
\text{trace } (P^+) = \text{trace } \left\{ K_k \left( H P H^T + C^+ Q_t C^{+T} + R_t + R_{t+1} \right) K_k^T \right. \right.
\]
\[
\left. \left. + K_k H P1_k^T + P1_k H K_k^T + M1_k \right\}. \tag{10}
\]

Since the learning gain, \( K(t, k) \) is employed to update the control law at \( (t, k + 1) \), and in addition, this gain does not affect the state error at \( (t + 1, k) \), that is

\[
\frac{d}{dK_k} (\text{trace } (P^+)) = \frac{d}{dK_k} \left( E \left[ \delta u(t, k+1) \delta u(t, k)^T \right] \right)
\]

Next, we show that \( P_{ux, k} \) is positive–semidefinite matrices, and \( R_t \) is positive definite, then \( H PH^T + C^+ Q_t C^{+T} + R_t + R_{t+1} \) in nonsingular. The following solution of \( K_k \) is the optimal learning gain which minimizes the trace of \( P^+ \)

\[
K_k = -P1_k H^T \left( H PH^T + C^+ Q_t C^{+T} + R_t + R_{t+1} \right)^{-1}.
\]

Define

\[
\Phi_{1, k} \triangleq I - K(t, k) C(t + 1) B(t)
\]
\[
\Phi_{2, k} \triangleq K(t, k) [C(t) - C(t + 1) A(t)]
\]
and
\[
g_{t, k} \triangleq K(t, k) [v(t + 1, k) - v(t, k) + C(t + 1) w(t, k)]
\]
Similarly, iterating the \( k \) argument of (5), we get for \( k > 1 \)

\[
\delta u(t, k) = \left( \prod_{i=0}^{k-1} \Phi_{t,i}^T \right)^T \delta u(t, 0) + \sum_{j=0}^{k-1} \left( \prod_{i=j}^{k-1} \Phi_{t,i+1}^T \right)^T \left[ \Phi_{2,j} \delta x(t, j) + g_{x,t} \right].
\]

(12)

Correlating the right hand terms of (11) and (12), it can be readily concluded that \( \delta x(t, 0), \delta u(t, 0), w(t), 0 \leq i \leq t - 1, \) \( v(t + 1), 0 \leq j \leq k - 1, \) \( v(t, 0) \leq j \leq k - 1, \) and \( w(t, 0) \leq j \leq k - 1 \), are all uncorrelated. In addition, these terms are also uncorrelated with \( \delta u(0 \leq i \leq t - 1, k), \) and \( \delta x(t, 0 \leq j \leq k - 1). \) At this point, the correlation of (11) and (12) is equivalent to correlating \( \delta x(t, k) = \sum_{t=0}^{k-1} M_t \delta u(i, k) \) and \( \delta u(t, k) = \sum_{j=0}^{k-1} N_j \delta x(t, j) \), where

\[
M_t \triangleq \left( \prod_{i=0}^{k-2} A(j + 1)^T \right) B(i), \quad N_j \triangleq \left( \prod_{i=0}^{k-2} \Phi_{t,1+m+1}^T \right)^T \Phi_{2,j}.
\]

Since the terms \( \delta u(0 \leq i \leq t - 1, k), \) and \( \delta x(t, 0 \leq j \leq k - 1), \) cannot be represented as a function of one another, hence these terms are considered uncorrelated. Therefore, \( P_{\delta x_k} = E[\delta u(t, k) \delta x(t, k)^T] = 0, \) and consequently \( P_{\delta x_k} = 0. \)

Then, \( K_k \) is reduced to

\[
K_k = P_{x_k} (C^+ B)^T \left[ \left( C^+ B \right) P_{x_k} (C^+ B)^T + (C - C^+ A) P_{x_k} \left( C - C^+ A \right)^T + C^+ Q_t C^+ + R_t + R_{t+1} \right]^{-1}
\]

and the input error covariance matrix becomes

\[
P_{\delta x_k+1} \triangleq E[\delta u(t, k + 1) \delta u(t, k + 1)^T]
= (I - K_k C^+ B) P_{x_k} (I - K_k C^+ B)^T
+ K_k \left( (C - C^+ A) P_{x_k} (C - C^+ A)^T + C^+ Q_t C^+ + R_t + R_{t+1} \right) K_k^T.
\]

(13)

IV. CONVERGENCE

In this section, we show that “stochastic” convergence is guaranteed in presence of random disturbances if \( C^+ B \) is full-column rank. In particular, it is shown that the input error covariance matrix converges uniformly to zero in presence of random disturbances (Theorem 3), where the state error covariance matrix converges uniformly to zero in presence of biased measurement noise (Theorem 5). In the following, some useful results are first derived.

**Proposition 1:** If \( C^+ B \) is a full-column rank matrix, then \( K_k = 0 \) if and only if \( P_{x_k} = 0. \)

**Proof:** Sufficient condition is the trivial case [set \( P_{x_k} = 0 \) in (13)]. Necessary condition. Since \( C^+ B \) is full-column rank, then \( (C^+ B)^T \) is a full-row rank

\[
M_{2_k} \triangleq \left( C^+ B \right)^T \left[ \left( C^+ B \right) P_{x_k} (C^+ B)^T + (C - C^+ A) P_{x_k} \left( C - C^+ A \right)^T + C^+ Q_t C^+ + R_t + R_{t+1} \right]^{-1}
\]

is also full-row rank. Multiplying both sides of (13), from the right-hand side, by \( M_{2_k} (M_{2_k} M_{2_k}^T)^{-1} \) and letting \( K_k = 0, \) implies that \( P_{\delta x_k} = 0. \)

The input error covariance matrix associated with the learning gain (13) may now be derived. Note that if the initial condition of \( P_{x_k} \) is symmetric, then, by examining (14), \( P_{x_k} \) is symmetric for all \( k. \) For compactness, we define \( N \triangleq C^+ B, \) and \( S_{1,k} \triangleq (C - C^+ A) P_{x_k} (C - C^+ A)^T + C^+ Q_t C^+ + R_t + R_{t+1}. \)

Expanding (14), we have

\[
P_{x_k+1} = P_{x_k} - K_k N P_{x_k} - P_{x_k} N P_{x_k}^T K_k^T + K_k (N P_{x_k} N^T + S_{1,k}) K_k^T.
\]

(15)

Substitution of (13) into (15) yields to

\[
P_{x_k+1} = P_{x_k} - P_{x_k} N^T (N P_{x_k} N^T + S_{1,k})^{-1} N P_{x_k} - P_{x_k} N^T (N P_{x_k} N^T + S_{1,k})^{-1} N P_{x_k} - P_{x_k} N P_{x_k} N^T (N P_{x_k} N^T + S_{1,k})^{-1} N P_{x_k}
= (I - K_k N) P_{x_k}.
\]

(16)

**Claim 1:** Assuming that \( P_{x_k} \) is symmetric positive-definite, then \( \forall k, t \in [0, \eta_t], \)

\[
I - K_k C^+ B = \left[ I + P_{x_k} (C^+ B)^T S_{1,k}^{-1} C^+ B \right]^{-1}.
\]

(17)

**Proof:** Substituting (13) in \( I - K_k N, \) we get

\[
I - K_k N = I - P_{x_k} N^T (N P_{x_k} N^T + S_{1,k})^{-1} N.
\]

One way to show the equality claimed, we multiply the left-hand side of (17) by the inverse of the right-hand side, and show that the result is nothing but the identity matrix

\[
[I - P_{x_k} N^T (N P_{x_k} N^T + S_{1,k})^{-1} N] \left[ I + P_{x_k} N^T S_{1,k}^{-1} N \right] = I + P_{x_k} N^T S_{1,k}^{-1} N - P_{x_k} N^T (N P_{x_k} N^T + S_{1,k})^{-1} N
- P_{x_k} N P_{x_k} N^T S_{1,k}^{-1} N.
\]

(18)

Using a well-known matrix inversion lemma [40], we get

\[
(N P_{x_k} N^T + S_{1,k})^{-1}
= S_{1,k}^{-1} - S_{1,k}^{-1} N \left( N^T S_{1,k}^{-1} N + P_{x_k}^{-1} \right)^{-1} N^T S_{1,k}^{-1} N.
\]

(19)

Substituting (19) into the right-hand side of (18), we have

\[
[I - P_{x_k} N^T (N P_{x_k} N^T + S_{1,k})^{-1} N] \left[ I + P_{x_k} N^T S_{1,k}^{-1} N \right] = I + P_{x_k} N^T S_{1,k}^{-1} N - P_{x_k} N^T S_{1,k}^{-1} N
+ P_{x_k} N^T S_{1,k}^{-1} N \left( N^T S_{1,k}^{-1} N + P_{x_k}^{-1} \right)^{-1} N^T S_{1,k}^{-1} N
- P_{x_k} N P_{x_k} N^T S_{1,k}^{-1} N
+ P_{x_k} N^T S_{1,k}^{-1} N \left( N^T S_{1,k}^{-1} N + P_{x_k}^{-1} \right)^{-1} N^T S_{1,k}^{-1} N.
\]
Canceling two terms and rearranging others, the above equation yields to
\[
\left[ I - P_{u,k} N^T (N P_{u,k} N^T + S_{1,k})^{-1} N \right] \left( I + P_{u,k} N^T S_{1,k}^{-1} N \right) - I
\]
\[
= I - P_{u,k} N^T S_{1,k}^{-1} N P_{u,k} N^T S_{1,k}^{-1} N + P_{u,k} N^T S_{1,k}^{-1} N \left( N^T S_{1,k}^{-1} N + P_{u,k} \right)^{-1}
\]
\[
\cdot \left( I + N^T S_{1,k}^{-1} N P_{u,k} \right) N^T S_{1,k}^{-1} N.
\]

We may rewrite the above equation as
\[
\left[ I - P_{u,k} N^T \left( N P_{u,k} N^T + S_{1,k} \right)^{-1} N \right] \left( I + P_{u,k} N^T S_{1,k}^{-1} N \right) - I
\]
\[
= I + P_{u,k} N^T S_{1,k}^{-1} N \left( N^T S_{1,k}^{-1} N + P_{u,k} \right)^{-1}
\]
\[
\cdot \left( I + N^T S_{1,k}^{-1} N P_{u,k} - \left( N^T S_{1,k}^{-1} N + P_{u,k} \right) P_{u,k} \right)
\]
\[
\cdot N^T S_{1,k}^{-1} N = I.
\]

**Lemma 2:** If \(C(t + 1)B(t)\) is full-column rank, then the learning algorithm, presented by (2), (13), and (16), guarantees that \(P_{u,k} \triangleq E[\delta u(t, k) \delta u(t, k)^T]\) is a symmetric positive-definite matrix \(\forall k\), and \(t \in [0, \eta_k]\). Moreover, the eigenvalues of \((I - K_t C(t) B)\) are positive and strictly less than one, i.e., \(0 < \lambda(I - K_t C(t) B) < 1 \forall k\), and \(t \in [0, \eta_k]\).

**Proof:** The proof is proceeded by induction with respect to the iteration index \(k\) for \(t \in [0, \eta_k]\). By examining (14), since \(P_{u,0}\) is assumed to be a symmetric positive-definite matrix, then \(P_{u,1} \triangleq \frac{1}{2} (I - K_t C(t) B)\) is symmetric and nonnegative definite. Define \(D_k \triangleq I - K_t C(t) B\). Equation (17) implies that \(D_0 \triangleq \frac{1}{2} (I + P_{u,0} N^T S_{1,k}^{-1} N)^{-1}\). Since \(N^T S_{1,k}^{-1} N\) is symmetric positive-definite, \(N^T S_{1,k}^{-1} N\) is symmetric positive-definite. In addition, having \(P_{u,0}\) symmetric and positive-definite implies that all eigenvalues of \(P_{u,0} N^T S_{1,k}^{-1} N\) are positive. Therefore, the eigenvalues of \(I + P_{u,0} N^T S_{1,k}^{-1} N\) are strictly greater than one, which is equivalent to conclude that the eigenvalues of \(D_0\) are positive and strictly less than one.

This implies that \(D_0\) is nonsingular. Equation (16) implies that \(P_{u,1} \triangleq \frac{1}{2} (I - D_0 P_{u,0})\) is nonsingular. Thus, \(P_{u,1}\) is a symmetric positive-definite matrix. We may now assume that \(P_{u,k}\) symmetric positive definite. Equation (14) implies that \(P_{u,k+1}\) is symmetric and nonnegative definite. Using a similar argument of \(D_0\) for \(D_k\), implies that \(0 < \lambda(D_k) < 1\), and consequently \(D_k\) with strictly positive eigenvalues, is nonsingular and \(P_{u,k+1}\) is symmetric and positive-definite.

**Remark:** Since all the eigenvalues of \((I - K_t C(t) B)\) are strictly positive and strictly less than one \(\forall k\), and \(t \in [0, \eta_k]\) (Lemma 2), then there exists a consistent norm \(\| \cdot \|\) such that \(\forall k\), and \(t \in [0, \eta_k]\)

\[
\|I - K_t C(t) B\| < 1.
\]

**Theorem 3:** If \(C(t + 1)B(t)\) is full-column rank, then the learning algorithm, presented by (2), (13), and (16), guarantees that \(\|P_{u,k+1}\| < \|P_{u,k}\|\forall k\). In addition, \(P_{u,k} \rightarrow 0\) and \(K_k \rightarrow 0\) uniformly in \([0, \eta_k]\) as \(k \rightarrow \infty\).

**Proof:** Equation (16) along with (20), imply

\[
\|P_{u,k+1}\| < \|I - K_t N\|\|P_{u,k}\| < \|P_{u,k}\|
\]

where again \(N \triangleq C(t + 1)B(t)\). We now show that \(\lim_{k \rightarrow \infty} P_{u,k} = 0\) by a contradiction argument. Since \(\|P_{u,k}\|\) is strictly decreasing sequence and bounded from below by zero, then the \(\lim_{k \rightarrow \infty} \|P_{u,k}\|\) exists. Assume that \(\lim_{k \rightarrow \infty} \|P_{u,k}\| \neq 0\). Consider the following ratio test for the sequence \(\|P_{u,k}\|\):

\[
\frac{\|P_{u,k+1}\|}{\|P_{u,k}\|} < \|I - K_t N\| = \|D_k\| < 1.
\]

Note that

\[
\lim_{k \rightarrow \infty} \sup \|D_k\| = \lim_{k \rightarrow \infty} \sup \left\| \left( I + P_{u,k} N^T S_{1,k}^{-1} N \right)^{-1} \right\|.
\]

Since \(\lim_{k \rightarrow \infty} \|P_{u,k}\|\) is bounded in \([0, \eta_k]\), then \(\lim_{k \rightarrow \infty} \sup \|P_{u,k}\|\) is also bounded in \([0, \eta_k]\), where \(P_{u,k} \triangleq E[\delta u(t, k) \delta u(t, k)^T]\). This can be seen by employing (11). Define

\[
A_1 \triangleq \left[ \prod_{m=0}^{t-1} A(m)^T \right]^T, \quad A_2, i \triangleq \left[ \prod_{j=i}^{t-2} A(j+1)^T \right]^T
\]

and

\[
P_{u,k,i,j} \triangleq E[\delta u(i, k) \delta u(j, k)^T]
\]

then

\[
P_{u,k} = A_1 A_2, 0 A_1^T + \sum_{i=0}^{t-1} A_2, i B(i) P_{u,k,i,j}
\]

\[
\cdot \left( \sum_{j=i}^{t-1} A_2, j B(j) \right)^T + \sum_{i=0}^{t-1} A_2, i Q_i \left( \sum_{i=0}^{t-1} A_2, i \right)^T.
\]

Since \(\lim_{k \rightarrow \infty} \|P_{u,k}\|\) is bounded, then \(\lim_{k \rightarrow \infty} \sup \|P_{u,k}\|\) is symmetric positive-definite and bounded. Therefore, \(\lim_{k \rightarrow \infty} \sup \|P_{u,k}\|\) is also bounded. The latter together with the assumption that \(\lim_{k \rightarrow \infty} \|P_{u,k}\| \neq 0\), then one may conclude that \(\lim_{k \rightarrow \infty} \sup \|D_k\| < 1\), or

\[
\lim_{k \rightarrow \infty} \sup \frac{\|P_{u,k+1}\|}{\|P_{u,k}\|} < 1.
\]

Since \(\|P_{u,k}\| > 0\), then the ratio test implies that \(\lim_{k \rightarrow \infty} \|P_{u,k}\| = 0\), which is equivalent to \(\lim_{k \rightarrow \infty} P_{u,k} = 0\). Employing Proposition 1, we get \(\lim_{k \rightarrow \infty} K = 0\). Since \(t\) takes on a finite values \((t \in [0, \eta_k])\), then uniform convergence is equivalent to pointwise convergence.

**Lemma 4:** If \(\lim_{k \rightarrow \infty} E[\delta u(t, k) \delta u(t, k)^T] = 0\), \(\forall t\), then \(\lim_{k \rightarrow \infty} E[\delta u(t, k) \delta u(m, k)^T] = 0\) for \(i \neq m\).

**Proof:** Let \(\delta u \triangleq \delta u(i, k)\), and \(\delta u \triangleq \delta u(m, k)\). Then,

\[
E[(\delta u - \zeta) (\delta u - \zeta)^T] \geq 0\] (positive-semidefinite matrix), and

\[
E[(\delta u + \zeta) (\delta u + \zeta)^T] \geq 0\]

Taking the limit as \(k \rightarrow \infty\), we have \(\lim_{k \rightarrow \infty} E[(\delta u - \zeta) (\delta u - \zeta)^T] = -2 \lim_{k \rightarrow \infty} E[\delta u(t, k)^T] \geq 0\). Sim-
Similarly, \( \lim_{k \to \infty} E[(\varepsilon + \zeta)(\varepsilon + \zeta)^T] = 2 \lim_{k \to \infty} E[\varepsilon^2] \geq 0 \). Therefore, \( \lim_{k \to \infty} E[\varepsilon_k^2] = 0 \).

**Theorem 5:** If \( C(t+1)B(t) \) is a full-column rank matrix, then in absence of state disturbance and reinitialization errors (excluding biased measurement noise \( \leftrightarrow R_k \) is positive definite), the learning algorithm, presented by (2), (13), and (16), guarantees that \( \|P_{u,k+1}\| < \|P_{u,k}\| \) \( \forall k \). In addition, \( P_{u,k} \to 0, K_k \to 0 \), and the state error covariance matrix \( E(\delta x(t, k)\delta x(t, k)^T) \to 0 \) uniformly in \( [0, n] \) as \( k \to \infty \).

**Proof:** Obviously, the results of the previous theorem still applies. Setting \( \delta x(0, k) = 0 \), and \( u(\cdot, k) = 0 \). (11) is reduced to

\[
\delta x(t, k) = \sum_{i=0}^{t-1} \prod_{j=i}^{t-2} A^T(j+1) B(i) \delta u(i, k).
\]

Consequently, taking the expected value of \( \delta x(t, k)\delta x(t, k)^T \), and defining \( A^{ij} = \prod_{j=i}^{t-2} A^T(j+1) \), we have

\[
E[(\delta x(t, k)\delta x(t, k)^T)] = \sum_{i=0}^{t-1} A^{ij} B(i) \sum_{m=0}^{t-1} \delta u(i, k) \delta u(m, k) B(m)^T A^{mT} T^T.
\]

Taking the limit as \( k \to \infty \), we get

\[
\lim_{k \to \infty} E[(\delta x(t, k)\delta x(t, k)^T)] = \sum_{i=0}^{t-1} A^{ij} B(i) \sum_{m=0}^{t-1} \lim_{k \to \infty} \delta u(i, k) \delta u(m, k) B(m)^T A^{mT} T^T.
\]

From previous results, \( \lim_{k \to \infty} P_{u,k} = \lim_{k \to \infty} E[\delta u(t, k)\delta u(t, k)^T] = 0. \forall t \). Therefore, \( \lim_{k \to \infty} E[\delta x(t, k)\delta x(t, k)^T] = 0. \forall 0 \leq i, m \leq t - 1 \). Consequently, \( \lim_{k \to \infty} E[(\delta x(t, k)\delta x(t, k)^T)] = 0 \). Again, uniform convergence for \( t \in [0, n] \) is straightforward.

**V. APPLICATION**

In this section, we go over the steps involved in the application of the proposed algorithm. A pseudo code is presented for both cases where the term \( (C - C^+ A)P_{u,k}(C - C^+ A)^T \) is considered, and neglected. In addition, a numerical example is presented to illustrate the performance of the proposed algorithm.

**A. Computer Algorithm**

In order to apply the proposed learning control algorithm, the state error covariance matrix needs to be available. From Section III, one can extract the following

\[
P_z(t+1, k) = E[(\delta x(t+1, k)\delta x(t+1, k)^T) = AP_z(t+1, k)A^T + BP_{u,k}B^T + Q_k.
\]

**TABLE I**

<table>
<thead>
<tr>
<th>Performance of Proposed Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>( z = u_1/z_2 )</td>
</tr>
<tr>
<td>( z(t) = z(t, k = 100)^T )</td>
</tr>
<tr>
<td>( \text{avg} [(z(t) - z(t, k = 100)^T]^2 )</td>
</tr>
<tr>
<td>( \text{max}_{n=100} \text{avg} [(z(t) - z(t, k = 100)^T]^2 )</td>
</tr>
<tr>
<td>( \text{max}<em>{n=100} \text{max}</em>{n=100} [(z(t) - z(t, k = 100)^T]^2 )</td>
</tr>
</tbody>
</table>

We assume that \( \delta u(t) \). \( (t, k) \). \( P_u \), \( Q_t \), \( R_k \), and \( P_z(k, 0) \) are all available. Starting with \( k = 0 \), then the pseudocode is as follows:

Step 1: For \( t \in [0, n] \)

a) apply \( u(t, k) \) to the system described by (1), and find the output \( e(t, k) \),

b) employing (22), compute \( P_z(t, k) \);

c) using (13), compute learning gain \( K(t, k) \);

d) using (2), update the control \( u(t, k+1) \);

e) using (16), update \( P_u, k+1 \).

Step 2: \( k = k + 1 \), go to Step 1.

In the case where \( (C - C^+ A)P_{u,k}(C - C^+ A)^T \) may be neglected, then the pseudocode is the same as the previous one excluding the second numbered item, and using (24) instead of (13), to compute learning gain \( K(t, k) \).

**B. Numerical Example**

**System Description:** In this section, we apply the proposed learning algorithm to a discrete-time varying linear system, and compare the performance with a similar algorithm where the gain is computed in a deterministic approach. The latter is referred to as “generic” algorithm. The system considered is given by

\[
x_1(t+1, k) = x_1(t, k) + T_S \left\{ \left( \delta x_1(t, k) + u(t, k) \right) \right\} + w_1(t, k)
\]

\[
x_2(t+1, k) = x_2(t, k) + T_S \left\{ \left( \delta x_2(t, k) + u(t, k) \right) \right\} + w_2(t, k)
\]

where “the integration period” \( T_S \) \( = 0.01 \) second, \( w_1(t, k) \) and \( w_2(t, k) \) are normally distributed white random processes with variances \( \delta T \), \( 1 \times 10^{-2} \), \( 1 \times 10^{-2} \), \( 1 \times 10^{-3} \), \( 1 \times 10^{-3} \), and 1, respectively. All the disturbances are assumed to be unbiased except for the measurement noise with mean \( 1 \). The desired trajectories with domain \( T \in [0, 100] \) (second interval) are the trajectories associated with \( u(t) = 30 \sin(\Delta t) \). The initial input for the generic and proposed algorithms \( u(t, 0) = 0. \forall t \in [0, 100] \). The initial input error covariance scalar (single input) \( P_{u, 0} = 1 \). \forall t \in [0, 100] \). Next, the generic algorithm requires the control gain to satisfy \( ||I - K_k|| \leq 3 \). It is noted that if \( K_k \) is chosen such that \( ||I - K_k|| \) is close to zero, then, in the iterative domain, fast transient response is noticed but at the cost of larger “steady-state” errors. However, if \( K_k \) is chosen such that \( ||I -
is close to one, then, smaller steady-state errors is noticed at the cost of very slow transient response. Consequently, the “generic” gain $K_k$ is chosen such that $\left\| I - K_kN \right\| = 0.5$, for all $t \in [0, 100]$.

**Performance:** In order to quantify the statistical and deterministic size of input and state errors, after a fixed number of iterations, we use three different measures. For statistical evaluation, we measure the variance and the mean square error of each variable. For deterministic examination, the infinite norm is measured. Since the convergence rate, as the number of iteration increases, is also of importance, then the accumulation of each of these measures (variance, mean square, and infinite norm) are also computed. The superiority of the proposed algorithm can be detected by examining both Table I and Fig. 1. In Fig. 1, the solid, and dashed lines represent the employment of the proposed and generic gains, respectively. The top plot of Fig. 2 shows the evolution of $K_k$, whereas the bottom plot shows the conformation of Proposition 1. In addition, an examination of Table I and Fig. 2 (top) shows the close conformity of the computed input error variance and the estimated one generated by the proposed algorithm. In particular, at $k = 100$, $\left\{ \text{stddev}(u_d(t) - u(t, k)) \right\}_k^2 = 16.1$, and $\text{avg}_{t \in [0, 100]} P_{u,k} = 18.1$, with $12.6 \leq P_{u,k} \leq 21.6$.

**VI. CONCLUSION**

This paper has formulated a computational algorithm for the learning gain matrix, which stochastically minimizes, in a least-squares sense, trajectory errors in presence of random disturbances. It is shown that if $C(t+1)B(t)$ is full-column rank, then the algorithm automatically assures the contraction condition, that is $\left\| I - K(t, k)C(t+1)B(t) \right\| < 1$, which ensures the suppression of repeatable errors and boundedness of trajectories in presence of bounded disturbances. The proposed algorithm is shown to stochastically reject uncorrelated disturbances from the control end. Consequently, state trajectory tracking becomes stochastically insensitive to measurement noise. In particular, it is shown that, if the matrix $C(t+1)B(t)$ is full-column rank, then the input error covariance matrix converges uniformly to zero in presence of uncorrelated random state disturbance, reinitialization errors, and measurement noise. Consequently, the state error covariance matrix converges uniformly to zero in presence of measurement noise. Moreover, it is shown that, if a certain condition is met, then the knowledge of the state coupling matrix is not needed to apply the proposed stochastic algorithm. Application of this algorithm to class of nonlinear systems is also examined. A numerical example is added to conform with the theoretical results.

**APPENDIX A**

**MODEL UNCERTAINTY**

Unfortunately, the determination of the proposed learning gain matrix, specified by (13), requires the update of the state error covariance matrix, which in turns requires full knowledge of the system model. This makes the application of the proposed learning control algorithm somehow unattractive. Note that in many applications, only the continuous-time statistical and deterministic models are available, furthermore, the discrete matrices $A(t)$ and $B(t)$, and the system and measurement noise...
covariance matrices depend on the size of the sampling period (while discretizing). Suppose that a continuous-time linear time-invariant system is given by $\dot{x}(t') = A'x(t') + B'u(t')$, and $y(t') = C'x(t')$ where $t'$ is the time argument in continuous domain. In order to evaluate the discrete-state matrix $A$, we may use

$$A = e^{A'T_S} = I + A'T_S + \frac{1}{2!}(A'T_S)^2 + \frac{1}{3!}(A'T_S)^3 + \cdots$$

($C = C'$) with $T_S$ the sampling period. For a small sampling period, one may approximate $A = e^{A'T_S} \approx 0$. There are several other mathematical applications where $C \in C^+ A = 0$ and the sampling period does not have to be small, e.g., if the rows of the state matrix $A$ corresponding to $i$th output is equal to the associated row of the output coupling matrix $C$. For example, if $C = [0 \ 1 \ 0]$, and only the second row of $A$ is equal to $C$, then $C = C^+ A = 0$. Therefore, the learning gain matrix is reduced to

$$K_k = P_{u,k} (C+B)^T \left[(C+B) P_{u,k} (C+B)^T \
+C'Q_tC+C'+R_t + R_{t+1}\right]^{-1}. \quad (24)$$

Note that, by using the learning gain of (24) [instead of (13)], all the results in Sections IV and V still apply as presented. Consequently, the computer algorithm does not require the update of the state error covariance matrix, hence knowledge of the state matrix is no longer needed. In the case where $C - C^+ A \neq 0$, and in absence of plant knowledge, then a parameter identification scheme, similar to the one presented in [34], may be integrated with the proposed algorithm.

**APPENDIX B**

**APPLICATION TO NONLINEAR SYSTEMS**

A. Application to a Class of Nonlinear Systems

Consider the state-space description for a class of nonlinear discrete-time varying systems given by the following difference equation:

$$x(t + 1, k) = f(x(t, k), t) + B(t)u(t, k) + w(t, k)$$
$$y(t, k) = C(t)x(t, k) + v(t, k) + \eta(k) \quad (25)$$

where $f(\cdot, \cdot)$ is a vector-valued function with range in $\mathbb{R}^n$. Define $f_{x,k} \triangleq f(x(t, k), t)$, and $f_{t,d} \triangleq f(x_d(t), t)$. We assume that the initial state $x_d(0)$ and a desired trajectory $y_d(t)$ are given and assumed to be realizable that is, $\exists$ bounded $x_d(t) \in \mathbb{R}^n$, $f_{t,d} \in \mathbb{R}^{n \times p}$, and $u_d(t) \in \mathbb{R}^n$, such that $y_d(t) = C(t)x_d(t)$, and $x_d(t + 1) = f_{t,d} + B(t)u_d(t) \forall t \in [0, \eta_t]$. The restrictions on the noise inputs $w(\cdot, \cdot)$ and $v(\cdot, \cdot)$ are as assumed in Section II. Again, we define $C \triangleq C(t), C^+ \triangleq C(t+1), N \triangleq C+B(t), K_k \triangleq K(t, k), \delta f_{t,k} \triangleq f_{t,d} - f_{t,k}$. The state error vector $\delta x(t, k) \triangleq x_d(t) - x(t, k)$, and the input error vector

---

**Fig. 2.** Evolution of learning gain function $K_k, t \in [0, 100]$ for $k = 1, 2, 5,$ and $10$ (top), $\text{avg}_{t \in [0, 100]} K_k$, and $\text{avg}_{t \in [0, 100]} P_{u,k}$ versus $k, 1 \leq k \leq 100$ (bottom).
Using (25), and the learning algorithm presented in (2), we obtain
\[ \delta u(t, k) = u_d(t) - u(t, k) \]
Using (25), the learning algorithm presented in (2), and since \( f(t, x(t)) \) is a function of \( x(t) \), then \( f(t, x(t)) \).

Borrowing the results from Section III, we find that the learning gain that minimizes the trace of \( P_{u,k} \) is similar to the one presented by (13). In particular, defining
\[ S_{1,k} = (C \delta x(t, k) - C^+ \delta f_{t,k}) \cdot (C \delta x(t, k) - C^+ \delta f_{t,k})^T + C^+ Q C^+ \gamma + R_t + R_{t+1} \]
and
\[ K_k = P_{u,k} N^T \left[ N P_{u,k} N^T + S_{1,k} \right]^{-1} \]
Consequently, the input error covariance matrix becomes
\[ P_{u,k+1} = E \left[ \delta u(t, k + 1) \delta u(t, k + 1)^T \right] = (I - K_k N) P_{u,k} (I - K_k N)^T \]
Theorem 6: Consider the system presented in (25). If \( f(t, x(t)) \) is uniformly globally Lipschitz in \( x \) and for \( t \in [0, n_k] \), then \( f(t, x(t), k) \) is also uniformly globally Lipschitz in \( x \) and for \( t \in [0, n_k] \). The error equation corresponding to (27) is given by
\[ \delta x(t + 1, k) = A(t) \delta x(t, k) + B(t) \delta u(t, k) + f(t, x(t), k) \]
where \( f(t, x(t), k) \) may represent the repetitive state disturbance. It is assumed that the function \( f(t, x(t)) \) is uniformly globally Lipschitz in \( x \) and for \( t \in [0, n_k] \). Let \( \mathcal{T} (x(t), k) = A(t) x(t, k) + f(t, x(t), k) \), then \( f(t, x(t)) \) is also uniformly globally Lipschitz in \( x \) and for \( t \in [0, n_k] \). The error equation corresponding to (27) is given by
\[ \delta x(t + 1, k) = A(t) \delta x(t, k) + B(t) \delta u(t, k) + f(t, x(t), k) \]
Substituting \( \mathcal{T} (x(t)) \) in place of \( f(t, x(t)) \) in the definition of the \( S_{1,k} \) matrix of (26), the following results are concluded.

Theorem 8: If \( C(t + 1) B(t) \) is a full-column rank matrix, then for any positive constant \( m \), the learning algorithm, presented by (2), (26), and (16), will generate a sequence of inputs such that the infinite-norms of the input, output, and state errors given by (28) decrease exponentially for \( n_k - 1 < k < m \) in \( [0, n_k] \).

Theorem 7: Consider the system presented in (25). If \( C(t + 1) B(t) \) is full-column rank and \( C \delta x(t, k) - C^+ \delta f_{t,k} \) is Lipschitz in \( x \) and for \( t \in [0, n_k] \), then \( \|P_{u,k+1}\| \leq \|P_{u,k}\| \), in addition, \( P_{u,k} \rightarrow 0 \) uniformly in \( [0, n_k] \) as \( k \rightarrow \infty \).

Theorem 7: Consider the system presented in (25). If \( C(t + 1) B(t) \) is full-column rank and \( f(t, x(t)) \) is uniformly globally Lipschitz in \( x \) and for \( t \in [0, n_k] \), then the learning algorithm, presented by (2), (26), and (28), guarantees that \( \|P_{u,k} \| \leq \|P_{u,k} \| \), in addition, \( P_{u,k} \rightarrow 0 \) uniformly in \( [0, n_k] \) as \( k \rightarrow \infty \).

Proof: Define \( D_k = I - K_k N \). Consider \( |D_k| < 1 \) in all \( k \), in particular, for \( k \leq m \), and \( R = \max_{0 \leq k \leq m} \|D_k\| \), then \( \|I - K(t, k) C(t + 1) B(t)\| \leq R < 1 \) for \( k \leq m \) and \( t \in [0, n_k] \). Borrowing results of [14, Corollary 2], then the infinite-norms of the input, output, and state errors given by (28) decrease exponentially for \( n_k - 1 < k < m \) in \( [0, n_k] \). The results related to the input error covariance matrix can be concluded from the previous subsection.

REFERENCES


