Iterative Learning Control with Forgetting Factor for Linear Distributed Parameter Systems with Uncertainty

Xisheng Dai, 1 Senping Tian, 2 Wenguang Luo, 1 and Yajun Guo 1

1 School of Electrical and Information Engineering, Guangxi University of Science and Technology, Liuzhou 545006, China
2 School of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China

Correspondence should be addressed to Senping Tian; ausptian@scut.edu.cn

Received 1 November 2014; Accepted 2 December 2014; Published 18 December 2014

Academic Editor: Wuneng Zhou

Iterative learning control is an intelligent control algorithm which imitates human learning process. Based on this concept, this paper discussed iterative learning control problem for a class parabolic linear distributed parameter systems with uncertainty coefficients. Iterative learning control algorithm with forgetting factor is proposed and the conditions for convergence of algorithm are established. Combining the matrix theory with the basic theory of distributed parameter systems gives rigorous convergence proof of the algorithm. Finally, by using the forward difference scheme of partial differential equation to solve the problems, the simulation results are presented to illustrate the feasibility of the algorithm.

1. Introduction

Iterative learning control (ILC) is an intelligent control method for systems which perform tasks repetitively over a finite time interval. Since the original work by Arimoto et al. [1], in the last three decades, ILC has been constantly studied. Currently, the iterative learning control (ILC) problem for lumped parameter systems which describes ordinary differential equations has been studied continuously, and very rich theory results are obtained [2–5]. Meanwhile, owning to the ILC method is also suitable to some systems which possess model uncertainty and nonlinear characteristics, so in practice ILC has also been widely applied; for example, ILC has been applied successfully in industrial robots, intelligent transportation systems, injection process, biomedical engineering, aspects of steelmaking, and tobacco fermentation systems and has obtained great economic benefit [6–10]. The main benefit of ILC is that for the design of control law not much information about the plant is required and it may even be completely unknown, it only requires the tracking references and input/output signals. However, the algorithm is simple and effective [11–13].

Recently, iterative learning control problem of distributed parameter systems described by partial differential equations has become a hot research. In [14], the ILC method was used on temperature control of the nonisothermal turbulence chemical reactor which has a first-order hyperbolic distributed parameter systems characteristics. Qu further applied iterative learning control to a flexible system described by a class of second order hyperbolic equations at in [15]; Zhao and Rahn discussed the ILC of distributed parameter systems control problems in the material transportation system [16], and its learning control laws act to the system boundary. Based on the operator semigroup theory, Chao and his coauthors give a sufficient condition of P-type and D-type learning algorithm for a class of parabolic distributed parameter systems [17]. In [18], Cichy and his coauthors used Crank-Nicholson discretization; ILC for parabolic distributed parameter systems is proposed. Dai and his coauthors discussed nonlinear learning algorithms of distributed parameter systems by using vector diagram theory and studied ILC of distributed parameter systems learning control problems with state time delays [19–21]. Lately, Huang and his coauthors studied a kind of steady-state iterative learning control problem for a class of nonlinear distributed parameter systems [22, 23].

On the other hand, ILC is a branch of the intelligent control with strict mathematical description, where one of its basic research issues is to design learning algorithm and...
analyze convergence of the algorithm. By analyzing these research results of related ILC of distributed parameter systems, it is not hard to find, not as lump parameter systems, that there are few ILC algorithms concerning distributed parameter systems. The main algorithms are P-type ILC or D-type ILC. Because the state space of distributed parameter systems is infinite dimensional function space, it is difficult to analyze the convergence of ILC algorithm. Generally, to obtain the condition of algorithm convergence, we must simultaneously consider three domains, for example, time, space, and iterative domain. According to the characteristics of forgetting among people in the learning process, some scholars have designed a forgetting factor learning algorithm for lumped parameter system [24–26]. Motivated by these facts, in this paper, we design an ILC algorithm with forgetting factor for class linear distributed parameter systems and present and prove the convergence condition of the ILC algorithm based on the \( L^2, L \) norm. This result has enriched the contents of iterative learning control algorithms of distributed parameter systems theoretically. The numerical examples also illustrate the feasibility of the algorithm.

2. Problem Statement and New Algorithm

Consider the following forms of linear distributed parameter systems with uncertainty:

\[
\frac{\partial Q(x,t)}{\partial t} = D\Delta Q(x,t) + A(t)Q(x,t) + B(t)u(x,t),
\]

where \( \Omega \) is a bounded open subset with smooth boundary \( \partial \Omega \) in \( \mathbb{R}^n \), \( Q \in \mathbb{R}^m \), and \( y \in \mathbb{R}^l \) represent the state vector and input and output vector of the system, respectively. \( A, B, C \) are uncertain bounded matrices, and \( A \in \mathbb{R}^{s\times m}, B \in \mathbb{R}^{s\times n}, C \in \mathbb{R}^{l\times n} \); \( D \) is a positive bounded diagonal matrix; that is \( D = \text{diag}(d_1, d_2, \ldots, d_n) \), where \( 0 < p_1 < d_i < \infty, i = 1,2,\ldots,n \), and \( p_1 \) is known. \( \Delta = \sum_{i=1}^{n}(\partial^2/\partial x_i^2) \) is the Laplace operator on the area \( \Omega \).

The corresponding initial and boundary conditions of the system (1) are

\[
aQ(x,t) + \beta \frac{\partial Q(x,t)}{\partial v} = 0, \quad (x,t) \in \partial \Omega \times [0,T],
\]

where \( \partial / \partial v \) is unit outward normal derivative on \( \partial \Omega \), \( \alpha = \text{diag}(\alpha_1, \alpha_2, \ldots, \alpha_n) \), and \( \alpha_i \geq 0 \); and \( \beta = \text{diag}(\beta_1, \beta_2, \ldots, \beta_n) \) (\( \beta > 0 \)) are diagonal matrices.

According to the controlled object described by the system (1), let desired output be \( y_d(x,t) \). Now, the aim is to seek a corresponding desired input \( u_d(x,t) \), such that the actual output of the system (1)

\[
y'_d(x,t) = C(t)Q_d(x,t) + G(t)u_d(x,t)
\]

will approximate to the desired output \( y_d(x,t) \). Since the system contains uncertainties, the desired control input \( u_d(x,t) \) can not be obtained directly; therefore, we will gradually gain the control sequence \( u_k(x,t) \), by using learning control method, such that

\[
\lim_{k \to \infty} u_k(x,t) = u_d(x,t).
\]

System (1) offers the following iterative learning control algorithm suitable for distributed parameters systems with forgetting factor:

\[
u_{k+1}(x,t) = \alpha u_k(x,t) + (1 - \alpha) u_{k-1}(x,t) + \Gamma(t) \epsilon_{k+1}(x,t),
\]

where \( \epsilon_{k+1}(x,t) = y_d(x,t) - y_{k+1}(x,t) \), \( \Gamma(t) \) is the gain matrix in learning process still to be sought, and \( \alpha \) is forgetting factor, satisfying \( 0 < \alpha \leq 1 \). \( u_k(x,t) \) and \( u_{k-1}(x,t) \) are the actual input of the \( k \) time and \( k - 1 \) time, respectively.

For the dynamic system (1), we make the following assumptions.

Assumption 1. With the repetitive operation on a finite time interval \([0,T]\), the systems can be written as

\[
\frac{\partial Q_k(x,t)}{\partial t} = D\Delta Q_k(x,t) + A(t)Q_k(x,t) + B(t)u_k(x,t),
\]

where \( Q_k(x,t) = Q_k(x,t) + \Phi(t) \), \( u_k(x,t) \),

\[
y_k(x,t) = C(t)Q_k(x,t) + G(t)u_k(x,t),
\]

where \( u_k(x,t) \) and \( u_{k-1}(x,t) \) are the actual output of the \( k \) time and \( k - 1 \) time, respectively.

Assumption 2. In the iterative learning process, the boundary value and initial value conditions of the system (1) which satisfy

\[
Q_k(x,0) = \varphi_k(x), \quad x \in \Omega, \quad k = 1,2,3,\ldots
\]

\[
\|\varphi_{k+1}(x) - \varphi_k(x)\|_2^2 \leq I r^k,
\]

where \( \varphi_k(x) \in \mathbb{R}^n \cap L^2(\Omega) \), \( I > 0, 0 < r < 1 \).

Remark. Assumption 1 complies with certain intermittent industrial processes with repeated characteristics, such as the temperature control of the reaction diffusion system; Assumption 2 is similar to human learning process; that is, the greater the number of learning iteration \( k \) is, the more accurate the initial position is.

Here are some general mathematical symbols used in this paper.

For the \( n \) dimensional vector \( W = (w_1, w_2, \ldots, w_n)^T \), where the norm of that is defined as \( ||W|| = \sqrt{\sum_{i=1}^{n} w_i^2} \). Corresponding to \( n \times n \), the norm of matrix \( A \) is \( ||A|| = \sqrt{\lambda_{\max}(A^TA)} \), and \( \lambda_{\max}(\cdot) \) represents the largest eigenvalue. \( L^2(\Omega) \) (or short in \( L^2 \)) represents a kind of function space consisted by all measurable functions, satisfying \( ||Q||_2^2 = \int_{\Omega} ||Q(x)||^2 dx \); if \( Q(x) \in L^2(\Omega), i = 1,2,\ldots,n, then
\( Q(x) = (Q_1(x), \ldots, Q_n(x)) \in \mathbb{R}^n \cap L^2(\Omega), \) and \( \|Q\|_{L^2} = \left\{ \int_{\Omega} (Q(x)^T Q(x))^2 \, dx \right\}^{1/2}. \) For the function \( f(x, t) : \Omega \times [0, T] \to \mathbb{R}^n, f(x, t) \in \mathbb{R}^n \cap L^2(\Omega), t \in [0, T], \) define its \((L^2, \lambda)\) norm as follows
\[
\|f\|_{(L^2, \lambda)} = \sup_{0 \leq t \leq T} \left\{ \|f(t, \cdot)\|_{L^2}^2 \right\}^{1/2}.
\]

3. Convergence Analysis of the Algorithm

Before giving the main conclusions of this paper, we first give some lemmas as follows.

**Lemma 3** (see [27]). Suppose constant \( 0 \leq \delta < 1; \) real sequences \( \{V_k\}_{k=0}^\infty \) and \( \{Z_k\}_{k=0}^\infty \) satisfy

1. \( \lim_{k \to \infty} V_k = 0, \)
2. \( Z_{k+1} \leq \delta Z_k + V_k. \)

Then we have
\[
\lim_{k \to \infty} Z_k = 0.
\]

**Lemma 4.** If one writes \( \bar{u}_k(x, t) = u_{k+1}(x, t) - u_k(x, t), E(t) = \left[ \begin{array}{cc} E(t) & f(t) \\ f^T(t) & -G(t) \end{array} \right], \) \( H_k(x, t) = \left[ \begin{array}{c} \epsilon_k(x, t) \\ \epsilon_k^T(x, t) \end{array} \right], \) \( F_k(x, t) = \left[ \begin{array}{cc} E(t) & F_k(x, t) \end{array} \right], \) then, based on system (1) and Algorithm (6), the following equation holds:
\[
H_{k+1}(x, t) = E(t) H_k(x, t) + F_k(x, t).
\]

Proof. According to Algorithm (6), we have
\[
u_{k+1}(x, t) - u_k(x, t) = (\alpha - 1) u_k(x, t) + (1 - \alpha) u_{k-1}(x, t) + \Gamma(t) \epsilon_k(x, t) = (\alpha - 1)(u_k(x, t) - u_{k-1}(x, t)) + \Gamma(t) \epsilon_k(x, t) = (\alpha - 1) u_k(x, t) + \Gamma(t) \epsilon_k(x, t).
\]

Moreover, because
\[
\epsilon_{k+1}(x, t) = \epsilon_k(x, t) + y_k(x, t) - y_{k+1}(t) = \epsilon_k(x, t) - C(t) (Q_{k+1}(x, t) - Q_k(x, t)) - G(t)(u_{k+1}(x, t) - u_k(x, t))
\]

thus substituting (12) in (13), we have
\[
\epsilon_{k+1}(x, t) = (I - G(t) \Gamma(t)) \epsilon_k(x, t) + (1 - \alpha) G(t) \bar{u}_k(t)
\]

while making matrix forms of (12) and (14), we can obtain (11). Lemma 4 is proved.

**Lemma 5.** For the state of the system \((1) Q_{k+1}(x, t), \) let \( \bar{Q}_{k+1}(x, t) = Q_{k+1}(x, t) - Q_k(x, t), \) \( g = \max_{0 \leq t \leq T} (\lambda_{\max}(B(t))), \)
\[
h = \max_{0 \leq t \leq T} [\lambda_{\max}(A(t)) + A(t)] \lambda_{\max}(B(t)).
\]

Then for a suitable large positive number \( \lambda, \) there is the following estimation:
\[
\|\bar{Q}_{k+1}(x, t)\|_{(L^2, \lambda)} \leq t^\lambda + \frac{g}{\lambda - \lambda_1} \|u_{k+1} - u_k\|_{(L^2, \lambda)}.
\]

Proof. From repetitive learning system (1), we have
\[
\frac{\partial}{\partial t} (Q_{k+1}(x, t) - Q_k(x, t)) = D \Delta (Q_{k+1}(x, t) - Q_k(x, t))
\]

Moreover, because \( Q_{k+1}(x, t) - Q_k(x, t) \) to left-multiply both sides of (16), it becomes
\[
\frac{1}{2} \frac{d}{dt} \left( \bar{Q}_k^T(x, t) \bar{Q}_k(x, t) \right) = \bar{Q}_k^T(x, t) D \bar{Q}_k(x, t)
\]

Integrating two sides of (17) about \( x \) on \( \Omega, \) we have
\[
\frac{d}{dt} \left( \|\bar{Q}_k(x, t)\|_{L^2}^2 \right) = 2 \int_{\Omega} \bar{Q}_k^T(x, t) D \bar{Q}_k(x, t) \, dx
\]

By Green formula, for the first term right side of (18),
\[
2 \int_{\Omega} \bar{Q}_k^T(x, t) D \bar{Q}_k(x, t) \, dx
\]

Let \( d = \min\{d_1, d_2, \ldots, d_n\}; \) by boundary conditions \((2), \) one obtains
\[
\left. \frac{\partial \bar{Q}_k^T(x, t)}{\partial y} \right|_{x \in \partial \Omega} = \bar{Q}_k^T(x, t) \frac{D^{\alpha}}{\beta} \bar{Q}_k(x, t) \left|_{x \in \partial \Omega} \right.
\]

(20)
Note $\alpha \geq 0$, $\beta > 0$, and $D$ is positive diagonal matrix; from (18) we get
\[
\frac{d}{dt} \left( \| \overline{Q}_k(t) \|^2_{L^2} \right) \leq \int_{\Omega} \overline{Q}_k^T(x, t) \left( A^T(t) + A(t) \right) \overline{Q}_k(x, t) \, dx \\
+ \int_{\Omega} \overline{Q}_k^T(x, t) B(t) \bar{u}_k(x, t) \, dx.
\]  
(21)

Using Hölder inequality and $ab \leq (1/2)(a^2 + b^2)$, (21) yields
\[
\frac{d}{dt} \left( \| \overline{Q}_k(t) \|^2_{L^2} \right) \leq \lambda_{\max} \left( A^T(t) + A(t) \right) \int_{\Omega} \overline{Q}_k^T(x, t) \overline{Q}_k(x, t) \, dx \\
+ \lambda_{\max} (B(t)) \left( \| \overline{Q}_k(t) \|^2_{L^2} + \| \bar{u}_k(t) \|^2_{L^2} \right) \\
\leq h \| \overline{Q}_k(t) \|^2_{L^2} + g \| \bar{u}_k(t) \|^2_{L^2},
\]  
where
\[
g = \max_{0 \leq t \leq T} \left( \lambda_{\max} (B(t)) \right),
\]
\[
h = \max_{0 \leq t \leq T} \left( \lambda_{\max} \left( A^T(t) + A(t) \right) + \lambda_{\max} (B(t)) \right).
\]  
(23)

Integrating (22) two sides from 0 to $t$, we have
\[
\| \overline{Q}_k(t) \|^2_{L^2} \leq \| \overline{Q}_k(0) \|^2_{L^2} + h \int_0^t \| \overline{Q}_k(s) \|^2_{L^2} \, ds \\
+ g \int_0^t \| \bar{u}_k(s) \|^2_{L^2} \, ds.
\]  
(24)

Using Bellman–Gronwall inequality to (24), we get
\[
\| \overline{Q}_k(t) \|^2_{L^2} \leq \| \overline{Q}_k(0) \|^2_{L^2} e^{ht} + g \int_0^t e^{(t-s)} \| \bar{u}_k(s) \|^2_{L^2} \, ds \\
\leq lr^k e^{ht} + g \int_0^t e^{(t-s)} \| \bar{u}_k(s) \|^2_{L^2} \, ds.
\]  
(25)

Choosing sufficiently large positive $\lambda$, such that $\lambda > h$, and using $e^{-\lambda t}$ to multiply both sides of (25)
\[
\| \overline{Q}_k(t) \|^2_{L^2} e^{-\lambda t} \leq lr^k + g \int_0^t e^{-(\lambda-h)(t-s)} \| \bar{u}_k(s) \|^2_{L^2} e^{-\lambda s} \, ds \\
\leq lr^k + g \int_0^t e^{-(\lambda-h)(t-s)} \| \bar{u}_k(s) \|^2_{L^2} \, ds \\
\leq lr^k + \frac{g}{\lambda-h} \| u_{k+1} - u_k \|^2_{L^2(A)}.
\]  
(26)

This is (15). Lemma 5 is proof complete.

**Theorem 6.** If the matrix $E(t) = \frac{1-G(t)}{\Gamma(t)} \frac{(1-a)G(t)}{(a-1)I}$ satisfies $2\lambda_E < 1$, where $\lambda_E = \max_{0 \leq t \leq T} \| E(t) \|^2$, $\Gamma(t)$ is the gain matrix of Algorithm (6). Then Algorithm (6) can achieve the desired trajectory tracking entirely; namely, the tracking error in the $[0, T]$ uniformly converges; that is
\[
\lim_{k \to \infty} \| e_k(\cdot, t) \|^2_{L^2} = 0, \quad \forall t \in [0, T].
\]  
(27)

**Proof.** In Lemma 4, both sides of (11) are left multiplied by $H^T_{k+1}(x, t)$, and we get
\[
H^T_{k+1}(x, t) H_{k+1}(x, t) \\
= (E(t) H_k(x, t) + F_k(x, t))^T (E(t) H_k(x, t) + F_k(x, t)) \\
\leq 2 (E(t) H_k(x, t))^T (E(t) H_k(x, t)) + 2F^T_k (F_k(x, t)) \\
\leq 2 \lambda_E H^T_k (H_k(x, t) + 2 \lambda_C \overline{Q}_{k+1}(t) \overline{Q}_{k+1}(t) .
\]  
(28)

Here $\lambda_C = \max_{0 \leq t \leq T} \| C(t) \|^2$; integrating by $x$ both sides of (28) on $\Omega$, we get
\[
\| H_{k+1}(\cdot, t) \|^2_{L^2} \leq 2 \lambda_E \| H_k(\cdot, t) \|^2_{L^2} + 2 \lambda_C \| \overline{Q}_{k+1}(\cdot, t) \|^2_{L^2}.
\]  
(29)

Select the appropriate large $\lambda$; multiplying both sides of (29) by $e^{-\lambda t}$, we have
\[
\| H_{k+1}(\cdot, t) \|^2_{L^2} e^{-\lambda t} \leq 2 \lambda_E \| H_k(\cdot, t) \|^2_{L^2} e^{-\lambda t} + 2 \lambda_C \| \overline{Q}_{k+1}(\cdot, t) \|^2_{L^2} e^{-\lambda t}
\]  
(30)

\[2 \lambda_E \| H_k(\cdot, t) \|^2_{L^2(A)} + 2 \lambda_C \| \overline{Q}_{k+1}(\cdot, t) \|^2_{L^2(A)}. \]

(31)

Note that the right side of (30), it is unrelated with $t$, there is
\[
\| H_{k+1}(\cdot, t) \|^2_{L^2(A)} \leq 2 \lambda_E \| H_k(\cdot, t) \|^2_{L^2(A)} + 2 \lambda_C \| \overline{Q}_{k+1}(\cdot, t) \|^2_{L^2(A)}. \]

(32)

According to Lemma 5 and (31),
\[
\| H_{k+1}(\cdot, t) \|^2_{L^2(A)} \leq 2 \lambda_E \| H_k(\cdot, t) \|^2_{L^2(A)} + 2 \lambda_C \left( \frac{g}{\lambda-h} \| u_{k+1} - u_k \|^2_{L^2(A)} \right).
\]  
(32)

From (32), in order to estimate $H_{k+1}(x, t)$, we need to estimate $\overline{u}_{k+1}(x, t)$.

By multiplying both sides of (13) by $\overline{u}_{k+1}(x, t)$, we obtain
\[
\overline{u}_{k+1}(x, t) \overline{u}_{k+1}(x, t) \\
= \left( (a-1) \overline{u}_k(x, t) + \Gamma(t) e_k(x, t) \right)^T \\
\times \left( (a-1) \overline{u}_k(x, t) + \Gamma(t) e_k(x, t) \right) \\
\leq 2 \left( a \overline{u}_k(x, t) \right)^T \left( (a-1) \overline{u}_k(x, t) \right) \\
+ 2 \left( \Gamma(t) e_k(x, t) \right)^T \left( \Gamma(t) e_k(x, t) \right) \\
\leq 2 (a-1)^2 \overline{u}_k^T(x, t) \overline{u}_k(x, t) + 2 \lambda_F e_k^T(x, t) e_k(x, t); \]

(33)

here $\lambda_F = \max_{0 \leq t \leq T} \| \Gamma(t) \|^2$. 


Integrating two sides of (33) above on \( \Omega \) and multiplying both sides by \( e^{-\lambda t} \), we have

\[
\| \mathbf{u}_{k+1} \|_{(L^2, \lambda)} \leq 2(\alpha - 1)^2 \| \mathbf{u}_k \|_{(L^2, \lambda)} + 2\lambda \Gamma e_k \| H_k \|_{(L^2, \lambda)}. \tag{34}
\]

By (31) and (33), we can obtain

\[
\| H_{k+1} \|_{(L^2, \lambda)} \leq 2\lambda E \| H_k \|_{(L^2, \lambda)} + 2\lambda [2(\alpha - 1)^2 + \lambda \Gamma] \leq 2(\alpha - 1)^2 + 2\lambda \Gamma \leq \rho. \tag{35}
\]

According to the condition of Theorem 6 \( 2\lambda E < 1 \), the real continuity shows selecting the appropriate large \( \lambda \) such that

\[
\rho = 2\lambda E + 2\lambda C \frac{2g}{\lambda - h} [(\alpha - 1)^2 + \lambda \Gamma] < 1. \tag{36}
\]

Therefore, from (36), we obtain

\[
\| H_{k+1} \|_{(L^2, \lambda)} \leq 2\lambda C e_k + \rho \| H_k \|_{(L^2, \lambda)}. \tag{37}
\]

It can be seen from (37) and Lemma 3 that we can get

\[
\lim_{k \to \infty} H_{k+1} \|_{(L^2, \lambda)} = 0 \tag{38}
\]

because

\[
\| e_k (\cdot, t) \|_{L^2} \leq \| H_k (\cdot, t) \|_{L^2} e^{-\lambda t} \leq \| H_k \|_{(L^2, \lambda)} e^{\lambda T}. \tag{39}
\]

Thus according to (38) the following can be obtained

\[
\lim_{k \to \infty} \| e_k (\cdot, t) \|_{L^2} = 0. \tag{40}
\]

Theorem 6 is proved completely.

4. Numerical Simulation

To illustrate the effectiveness of the algorithm, contrasting system (1), we set the system state and the control input and output to

\[
Q(x, t) = \begin{bmatrix} q_1(x, t) \\ q_2(x, t) \end{bmatrix}, \quad u(x, t) = \begin{bmatrix} u_1(x, t) \\ u_2(x, t) \end{bmatrix}, \quad y(x, t) = \begin{bmatrix} y_1(x, t) \\ y_2(x, t) \end{bmatrix}, \tag{41}
\]

where the space and time variables \((x, t) \in [0, 1] \times [0, 1]\) and the coefficient matrices and the gain matrices of system (1) are as follows:

\[
D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} e^{-2t} & 0.2 \\ 0 & e^{-3t} \end{bmatrix}, \quad B = \begin{bmatrix} e^{-t} & 0.5 \\ 0 & e^{-1.5t} \end{bmatrix},
\]

\[
C = \begin{bmatrix} 1.5 & 1 \\ 0.9 & 2 \end{bmatrix}, \quad G = \begin{bmatrix} 1.9 & 0.1 \\ 0 & 1.9 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 0.45 & 0 \\ 0 & 0.46 \end{bmatrix}. \tag{42}
\]

The following are the desired trajectory:

\[
y_{1d}(x, t) = 3 \left( e^{-t} - 1 \right) \sin 2 \pi x, \tag{43}
\]

\[
y_{2d}(x, t) = -4 \left( \cos \left( \frac{1}{2} \pi t \right) - 1 \right) \sin 2 \pi x.
\]

And the two initial inputs \( u_1(x, t) \) and \( u_2(x, t) \) of the initial state and Algorithm (6) are all zeros, and \( r = 0 \) and \( \alpha = 0.95 \) which easily calculate \( \lambda E < 0.5 \) to meet the conditions of Theorem 6.

Figures 1 and 2 are the desired surfaces and Figures 3 and 4 are output surfaces of 8th iteration. Figures 5 and 6
are the corresponding error surface of iteration 8. Figure 7 is the varying trajectories where $L^2$ norm of error changes with iterations and the maximum values of the eighth iteration errors are $1.2563 \times 10^{-3}$, $9.7165 \times 10^{-4}$, respectively. Therefore, the iterative learning algorithm (6) is effective for system (1).

5. Conclusions

In this paper, a class of parabolic distributed parameter system with coefficient uncertain but bounded is studied by using iterative learning algorithm with adjustable forgetting factor. The sufficient conditions of convergence $L^2$-norm of the tracking error are given. The theoretical analysis and simulation results show that the proposed ILC algorithm is effective. We will look forward to more learning algorithms applied to the distributed parameter systems in the future.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.
Acknowledgments

The work was supported by the National Natural Science Foundation of China (nos. 61364006 and 61374104) and was supported by the Project of Outstanding Young Teachers’ Training in higher education institutions of Guangxi.

References


