Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2014, Article ID 984742, 11 pages
http://dx.doi.org/10.1155/2014/984742

*Research Article*

# Multiple Model ILC for Continuous-Time Nonlinear Systems

**Xiaoli Li,[1] Kang Wang,[1] and Yang Li[2]**

[1] *School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China*
[2] *School of International Studies, Communication University of China (CUC), Beijing 100024, China*

Correspondence should be addressed to Xiaoli Li; lixiaoli@hotmail.com

Received 13 August 2014; Accepted 12 September 2014; Published 21 October 2014

Academic Editor: Shen Yin

Multiple model iterative learning control (MMILC) method is proposed to deal with the continuous-time nonlinear system with uncertain and iteration-varying parameters. In this kind of control strategy, multiple models are established to cover the uncertainty of system; a switching mechanism is used to decide the most appropriate model for system in current iteration. For system operating iteratively in a fixed time interval with uncertain or jumping parameters, this kind of MMILC can improve the transient response and control property greatly. Asymptotical convergence is demonstrated theoretically, and the control effectiveness is illustrated by numerical simulation.

## 1. Introduction

Data-driven techniques, which can extract and utilize useful information from huge amount of available process data to realize the control of complex system, have been one of the hottest research focuses [1, 2]. As one of the most representative data-driven methods, iterative learning control (ILC) can use the information of measured state error and control input to realize desired control performance through iterative refinement of control signal [3]. Just relaying on the process measurements without prior knowledge of the controlled plant, ILC is an effective control strategy to realize the uniform tracking of desired trajectory, even for systems with uncertainties like imprecise parameters and unmodeled dynamics.

However, it should be noted that, for traditional ILC methods since Arimoto et al. [4], strict iteration-invariant identical conditions on initial states and process dynamics, as well as the desired trajectory, must be satisfied [5]. ILC has been applied to robot manipulator [6], chemical process [7], and semiconductor manufacturing [8] based on these identical conditions. However, in practical control process, the strict conditions are always difficult to be guaranteed and hinder the further applications of ILC.

Recently, how to improve ILC to deal with iteration-varying factors has been an important research focus [9]. As we know, iterative learning control is a 2-dimensional problem working on the time axis and iteration axis; model uncertainty in the iteration domain will increase the difficulty in the 2D scheme, while traditional ILC is based on the iteration-invariant premise. Meanwhile, as the system dynamic will be changed due to iteration-variance in parameters, it is a relatively difficult problem compared with other kinds of nonrepetitiveness [10–12].

The first representative approach dealing with iteration-varying parameters in ILC is the norm-based method. For global Lipschitz continuous system, the 2D problem can be formulated into a contraction mapping problem, and the convergence is analyzed by the norm-based theory. Paper [13] supposes that the freeway system is global Lipschitz continuous about all iteration-varying parameters—free speed and jam density. Further, based on the norm-based theory, the proposed ILC learning law is proved convergent to track the desired traffic density despite the existence of iteration-varying parameters. It should be noted that the global Lipschitz condition is the key condition for norm-based method.

Another representative method is the super-vector analysis approach. For discrete-time interval system, the super-vector scheme can transform the 2D dynamic system into a 1D MIMO static plant with a Markov matrix only in iteration dimension [9]. This framework provides us a unified method to design and analyze ILC for the system with model uncertainty in the iteration domain. The design problem of ILC for system with iteration-varying parameters will be converted into a Markov representation problem [14–18]. In [16], the varying range of Markov parameters was assumed to be known, and it proved that the convergence of ILC can be guaranteed by the vertex point of the Markov parameters. Optimal method was utilized to update the control input to handle the iteration-varying parameter uncertainty in [17]. The super-vector method greatly simplifies the design and analysis of ILC, and the global Lipschitz condition is no longer needed. However, due to the matrix transform, this method is not applicable for continuous-time system.

The composite energy function (CEF) is the third method. Derived from Lyapunov stability theory, the CEF method is a systematic way for learning law design and convergence analysis of the ILC of continuous-time nonlinear system without requiring global Lipschitz condition [19]. For the system with periodically iteration-varying parameter, [20] incorporated adaptive control into ILC and the convergence was proved in the CEF way. The CEF analysis method has also been successfully applied to automatic train operation ILC control despite the existence of iteration-varying parameters such as temperatures [21]. The CEF method does not require the global Lipschitz continuous condition, but it is restricted for continuous-time system. Due to the different applications of the above three design and analysis methods, they are complementary in the design procedure of ILC for the system with iteration-varying parameters.

In this paper, we focus on continuous-time nonlinear systems with uncertain or jumping parameters. In practice, many factors can cause the abrupt changes of system's structure and parameter, for example, faults in system, abrupt disturbances, failures of sensor and actuator, and load changes. These abrupt changes in dynamic parameters cause the operation of controlled plant in multiple environments. Traditional single model ILC can no longer handle this kind of system perfectly. Single fixed controller is established based on the unchanged model, but when parameters jump, it cannot learn from previous operation because the model is no longer the same in the iteration domain and the learning process has to start all over again. As the model is fixed and the plant is changing, mismatch between model and system is inevitable, so that the learning process gets relative long. We incorporate multiple model adaptive control (MMAC) [22, 23] to cover parameter uncertainty with multiple models, detect which situation the plant is in at every iteration, and launch the corresponding ILC controller. Thus, we can improve transient response performance (in this paper, transient response performance means the iteration trails before system output converges to the desired trajectory) and control property for the system with uncertain parameters. Further, by the norm-based method, we prove that the proposed ILC method can achieve asymptotic convergence.

This paper is organized as follows. Section 2 introduces the model algorithmic ILC and formulates the dynamic system with iteration-varying jumping parameters. Section 3 presents the MMILC algorithm in detail and Section 4 analyzes its convergence for system with unknown constant parameters and system with jumping parameters. Numerical simulations are conducted in Section 5 and Section 6 concludes this work.

## 2. Model Algorithmic ILC and Problem Formulation

*2.1. Model Algorithmic ILC.* Consider the following continuous-time nonlinear system:

$$\dot{\mathbf{x}}_k(t) = \mathbf{f}(\mathbf{x}_k(t), t) + \mathbf{B}(\mathbf{x}_k(t), t)\,\mathbf{u}_k(t), \tag{1}$$

where the subscript $k = 0, 1, 2, \ldots$ denotes the $k$th iteration. $\mathbf{u}_k(t) \in \mathbb{R}^r$ and $\mathbf{x}_k(t) \in \mathbb{R}^n$ are accessible control input vector and system state vector, respectively, at the $k$th iteration. The system operates iteratively on the fixed time interval; that is, $t \in [0, T]$. $\mathbf{f}$ and $\mathbf{B}$ are global Lipschitz continuous about $\mathbf{x}$, $\mathbf{B}$ is of full column rank and upper bounded on $\mathbb{R}^n \times [0, T]$; that is, for all $t \in [0, T]$, $\mathbf{x}, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$,

$$\begin{gathered}\left\|\mathbf{f}(\mathbf{x}_1, t) - \mathbf{f}(\mathbf{x}_2, t)\right\| \leqslant k_{\mathbf{f}} \left\|\mathbf{x}_1 - \mathbf{x}_2\right\|, \\ \left\|\mathbf{B}(\mathbf{x}_1, t) - \mathbf{B}(\mathbf{x}_2, t)\right\| \leqslant k_{\mathbf{B}} \left\|\mathbf{x}_1 - \mathbf{x}_2\right\|, \\ \left\|\mathbf{B}(\mathbf{x}, t)\right\| \leqslant b_{\mathbf{B}},\end{gathered} \tag{2}$$

where $k_{\mathbf{f}}$ and $k_{\mathbf{B}}$ are the lipschitz coefficients of functions $\mathbf{f}$ and $\mathbf{B}$, respectively, and $b_{\mathbf{B}}$ is the upperbound of $\mathbf{B}$.

In this paper, we denote $\|\mathbf{a}\|$ as the infinite norm of vector $\mathbf{a} \in \mathbb{R}^m$ with its entries $a_i$; that is, $\|\mathbf{a}\| = \|\mathbf{a}\|_\infty = \max_{1 \leqslant i \leqslant m} |a_i|$. And the notation of $\lambda$-norm of vector function $\mathbf{N}(t) \in \mathbb{R}^n$ is given as

$$\|\mathbf{N}(t)\|_\lambda = \sup_{t \in [0,T]} \left\{ e^{-\lambda t} \|\mathbf{N}(t)\| \right\}, \tag{3}$$

where $\lambda > 0$.

For the nonlinear dynamic plant described by (1), suppose that its model is obtained as

$$\dot{\widetilde{\mathbf{x}}}_k(t) = \widetilde{\mathbf{f}}(\widetilde{\mathbf{x}}_k(t), t) + \widetilde{\mathbf{B}}(\widetilde{\mathbf{x}}_k(t), t)\,\mathbf{u}_k(t), \tag{4}$$

where $\{\widetilde{\mathbf{f}}, \widetilde{\mathbf{B}}\}$ is the model of $\{\mathbf{f}, \mathbf{B}\}$ and $\widetilde{\mathbf{x}}(t) \in \mathbb{R}^n$ is the model state vector.

Paper [24] proposed the model algorithmic ILC learning scheme:

$$\begin{aligned}\mathbf{u}_{k+1}(t) = \mathbf{u}_k(t) &+ \widetilde{\mathbf{B}}^+(\mathbf{x}_d(t), t)\left[\dot{\mathbf{x}}_d(t) - \widetilde{\mathbf{f}}(\mathbf{x}_d(t), t)\right] \\ &- \widetilde{\mathbf{B}}^+(\mathbf{x}_k(t), t)\left[\dot{\mathbf{x}}_k(t) - \widetilde{\mathbf{f}}(\mathbf{x}_k(t), t)\right],\end{aligned} \tag{5}$$

where $\widetilde{\mathbf{B}}^+$ is the generalized inverse of $\widetilde{\mathbf{B}}$; that is, $\widetilde{\mathbf{B}}^+ = [\widetilde{\mathbf{B}}^{\mathrm{T}}\widetilde{\mathbf{B}}]^{-1}\widetilde{\mathbf{B}}^{\mathrm{T}}$. Suppose that $\widetilde{\mathbf{B}}^+$ is bounded and $\widetilde{\mathbf{B}}^+$ and $\widetilde{\mathbf{B}}^+\widetilde{\mathbf{f}}$ are global Lipschitz continuous about $\mathbf{x}$.

As the system state $\mathbf{x}_k(t)$ is accessible, it is reasonable to replace $\widetilde{\mathbf{x}}_k(t)$ with $\mathbf{x}_k(t)$ in ILC learning law (5). If learning law (5) is applied to system (1), we have the following lemma.

**Lemma 1** (see [24, 25]). *If the ILC system and model described by* (1) *and* (4), *respectively, satisfy*

(1) $\left\|\mathbf{I}-\widetilde{\mathbf{B}}^{+}(\mathbf{x}(t),t)\mathbf{B}(\mathbf{x}(t),t)\right\| \leqslant \rho < 1$ $(\mathbf{x}, t \in \mathbb{R}^{n} \times [0,T])$,

(2) $\mathbf{x}_{k}(0)=\mathbf{x}_{d}(0)$ $(k=0,1,2,\ldots)$,

*then, for arbitrarily given and achievable desired trajectory* $\mathbf{x}_{d}(t)$ $(t \in [0,T])$*, ILC learning law (5) can generate a control sequence which guarantees* $\mathbf{x}_{k}(t)$ *to converge to* $\mathbf{x}_{d}(t)$*; that is,*

$$\lim_{k\to\infty}\left\|\mathbf{x}_{d}(t)-\mathbf{x}_{k}(t)\right\|=0 \quad (t\in[0,T]). \tag{6}$$

*The proof of Lemma 1 is attached at the end of this paper as the Appendix. Further, from the proof, we can draw the following conclusion.*

**Corollary 2** (see [24]). *Denote* $\Delta\mathbf{x}_{k}(t) = \mathbf{x}_{d}(t)-\mathbf{x}_{k}(t)$, $\Delta\mathbf{u}_{k}(t) = \mathbf{u}_{d}(t)-\mathbf{u}_{k}(t)$, $\mathbf{L}(\mathbf{x},t) = \widetilde{\mathbf{B}}^{+}(\mathbf{x},t)$*, and* $\mathbf{h}(\mathbf{x},t) = \mathbf{L}(\mathbf{x},t)[\dot{\mathbf{x}}_{d}(t)-\widetilde{\mathbf{f}}(\mathbf{x},t)]$*. As* $\widetilde{\mathbf{B}}^{+}$ *and* $\widetilde{\mathbf{B}}^{+}\widetilde{\mathbf{f}}$ *are Lipschitz continuous about* $\mathbf{x}$*;* $\mathbf{h}$ *satisfies the Lipschitz condition; that is, for all* $t \in [0,T]$*, and* $\mathbf{x}_{1}, \mathbf{x}_{2} \in \mathbb{R}^{n}$*,*

$$\left\|\mathbf{h}(\mathbf{x}_{1},t)-\mathbf{h}(\mathbf{x}_{2},t)\right\| \leqslant k_{\mathbf{h}}\left\|\mathbf{x}_{1}-\mathbf{x}_{2}\right\|, \tag{7}$$

*where* $k_{\mathbf{h}}$ *is the Lipschitz coefficient of* $\mathbf{h}$*.*

*For single model ILC, the proof of Lemma 1 gives that*

$$\left\|\Delta\mathbf{x}_{k}\right\| \leqslant b_{\mathbf{B}}\int_{0}^{t} e^{b_{2}(t-\tau)}\left\|\Delta\mathbf{u}_{k}\right\| d\tau, \tag{8a}$$

$$\left\|\Delta\mathbf{u}_{k+1}\right\|_{\lambda} \leqslant \left(\rho+b_{1}b_{\mathbf{B}}\lambda_{b_{2}}\right)\left\|\Delta\mathbf{u}_{k}\right\|_{\lambda}, \tag{8b}$$

*where* $b_{1} = k_{\mathbf{h}} + b_{\mathbf{L}}(k_{\mathbf{f}} + k_{\mathbf{B}}b_{\mathbf{u}d})$, $b_{2} = k_{\mathbf{f}} + k_{\mathbf{B}}b_{\mathbf{u}d}$, $\lambda_{b_{2}} = (1-e^{(b_{2}-\lambda)T})/(\lambda-b_{2})$, $b_{\mathbf{L}} = \sup_{t\in[0,T]}\|\mathbf{L}(\mathbf{x}(t))\|$, $b_{\mathbf{u}d} = \sup_{t\in[0,T]}\|\mathbf{u}_{d}(t)\|$*, and* $\lambda$ *is a arbitrarily given positive value.*

*Remark 3.* From the above definition of different elements, $b_{\mathbf{B}}$, $b_{2}$, and $\lambda_{b_{2}}$ are only related to the dynamic system $(\mathbf{f},\mathbf{B})$, while $\rho$, $b_{1}$ are decided by both dynamic system $(\mathbf{f},\mathbf{B})$ and model $(\widetilde{\mathbf{f}},\widetilde{\mathbf{B}})$. Consequently, (8a) is related to the dynamic system while (8b) is decided by both dynamic system and model.

*Remark 4.* Most ILC methods just use the information of process measurements, so corresponding control strategies are always date-based and lack mechanism analysis. In practice, a priori knowledge about the process is more or less known. The model algorithmic ILC is not a model free or a pure data-driven method; it is a model-data integrated method which also takes the estimated system model into account, so that the known information can be fully used to perfect the actual performance [26].

*2.2. Problem Formulation.* In this paper, we consider nonlinear system with iteration-varying jumping parameters in the form of

$$\dot{\mathbf{x}}_{k}(t)=\mathbf{f}\left(\mathbf{x}_{k}(t),\alpha_{k}(t),t\right)+\mathbf{B}\left(\mathbf{x}_{k}(t),\beta_{k}(t),t\right), \tag{9}$$

where $\alpha_{k}(t)$ and $\beta_{k}(t)$ are unknown piecewise-continuous parameters which are constant at first and then change abruptly to other constant values; this change may happen at any time and in any iteration. For $k = 0,1,\ldots,\infty$, $t\in[0,T]$, uncertain parameters vary in finite convex set $\Pi$; that is,

$$\begin{aligned}\left(\alpha_{k}(t),\beta_{k}(t)\right)\in\Pi=\Big\{\left(\alpha_{i_{\alpha}},\beta_{i_{\beta}}\right)\mid i_{\alpha}&=1,2,\ldots,d_{\alpha};\\ i_{\beta}&=1,2,\ldots,d_{\beta}\Big\},\end{aligned} \tag{10}$$

where $d_{\alpha}$ and $d_{\beta}$ are finite integers.

The control objective is for system described as (9), to find such an appropriate control input $\mathbf{u}(t)$, that the generated state output $\mathbf{x}(t)$ can be guaranteed to track the desired trajectory $\mathbf{x}_{d}(t)$ as precisely and quickly as possible. In other words, the track error converges to zero as the iteration approaches infinity; that is,

$$\lim_{k\to\infty}\left\|\mathbf{x}_{d}(t)-\mathbf{x}_{k}(t)\right\|\longrightarrow 0. \tag{11}$$

To achieve this target, we suppose that system (9) satisfies the following assumptions.

*Assumption 5.* Specific values of varying parameters $\alpha_{k}(t)$ and $\beta_{k}(t)$ are unknown, but the probable varying scope is known as

$$\widetilde{\Pi}=\left\{(\alpha,\beta)\mid\alpha\in\left(\alpha_{l},\alpha_{u}\right),\beta\in\left(\beta_{l},\beta_{u}\right)\right\}, \tag{12}$$

where $\alpha_{l}$ and $\beta_{l}$ are the lower bound and $\alpha_{u}$ and $\beta_{u}$ are the upper bound of $\alpha$ and $\beta$, respectively.

*Assumption 6.* Consider the time variant system as (9); the parameters of system will be piecewise constant, and the time between two adjacent parameter jumps is relatively long.

*Assumption 7.* Nonlinear function $\mathbf{B}(\mathbf{x},\beta)$ is full column rank and upper bounded on $\mathbb{R}^{n}\times[0,T]$; that is, for all $t\in[0,T]$, $\beta\in\Pi_{\beta}=\{\beta_{i_{\beta}}\mid i_{\beta}=1,2,\ldots,d_{\beta}\}$, $\mathbf{x}\in\mathbb{R}^{n}$,

$$\left\|\mathbf{B}(\mathbf{x},\beta,t)\right\|\leqslant b_{\mathbf{B}}'. \tag{13}$$

*Assumption 8.* Functions $\mathbf{f}(\mathbf{x},\alpha)$ and $\mathbf{B}(\mathbf{x},\beta)$ satisfy the global Lipschitz condition about $\mathbf{x}$, that is, for all $\{\alpha,\beta\}\in\Pi$, $t\in[0,T]$, $\mathbf{x}_{1},\mathbf{x}_{2}\in\mathbb{R}^{n}$,

$$\begin{aligned}\left\|\mathbf{f}(\mathbf{x}_{1},\alpha,t)-\mathbf{f}(\mathbf{x}_{2},\alpha,t)\right\|&\leqslant k_{\mathbf{f}}'\left\|\mathbf{x}_{1}-\mathbf{x}_{2}\right\|,\\ \left\|\mathbf{B}(\mathbf{x}_{1},\beta,t)-\mathbf{B}(\mathbf{x}_{2},\beta,t)\right\|&\leqslant k_{\mathbf{B}}'\left\|\mathbf{x}_{1}-\mathbf{x}_{2}\right\|,\end{aligned} \tag{14}$$

where $k_{\mathbf{f}}'$ and $k_{\mathbf{B}}'$ are Lipschitz parameters.

Considering Lemma 1, model algorithm ILC can somehow handle model imprecision and mismatch; however, this is based on the condition that controlled plant as (1) must be strictly repetitive along the iteration axis. For plant in the form of (9), the dynamic system will change as parameter jumps, model mismatch may get more serious, and the ILC method cannot learn from previous iterations and has to restart the learning process. Consequently, transient response will deteriorate and long time will be needed before the output tracks the desired trajectory for the relearning process.

To handle this problem, we introduce multiple model adaptive control to ILC. Multiple models are established to cover the parameter uncertainty, an adaptive switching scheme will be proposed to accommodate the dynamic system change, so that the transient response can be improved.

## 3. Multiple Model Iteration Learning Control

For system described by (9), the iteratively jumping parameters cause the plant to operate among multiple environments. Our goal is to detect the specific environment at current iteration, serve this environment appropriately, and finally realize fast and precise trajectory tracking in multiple environments. In this section, we discuss the establishment of multiple models according to the iteration-varying parameters, the establishment of ILC controllers, introduction of switching index function and the design of multiple model iteration learning control algorithm.

*3.1. The Establishment of Multiple Models.* Multiple parameter set $\mathbf{P}$ can be established manually to cover the varying scope of $(\alpha, \beta)$ described as (12):

$$\mathbf{P} = \left\{ \left( a_{i_a}, b_{i_b} \right) \mid i_a = 1, 2, \ldots, d_a;\ i_b = 1, 2, \ldots, d_b \right\}, \quad (15)$$

where $d_a$ and $d_b$ are finite integers.

Correspondingly, multiple models can be set up as

$$M = \left\{ M_{\mathbf{i}} = \left\{ \widetilde{\mathbf{f}}_{i_a}\left( \mathbf{x}(t), a_{i_a}, t \right), \widetilde{\mathbf{B}}_{i_b}\left( \mathbf{x}(t), b_{i_b}, t \right) \right\} \mid \mathbf{i} \in \Omega \right\}, \quad (16)$$

where $\Omega = \{\mathbf{i} = (i_a, i_b) \mid i_a = 1, 2, \ldots, d_a;\ i_b = 1, 2, \ldots, d_b\}$.

For notational simplify, let $\widetilde{\mathbf{f}}_{i_a}(\mathbf{x}(t), t) = \widetilde{\mathbf{f}}_{i_a}(\mathbf{x}(t), a_{i_a}, t)$ and $\widetilde{\mathbf{B}}_{i_b}(\mathbf{x}(t), t) = \widetilde{\mathbf{B}}_{i_b}(\mathbf{x}(t), b_{i_b}, t)$. Suppose that multiple models $\{\widetilde{\mathbf{f}}_{i_a}, \widetilde{\mathbf{B}}_{i_b}\}$ satisfy the following assumption.

*Assumption 9.* $\widetilde{\mathbf{B}}^{+}_{i_b}$ is bounded and $\widetilde{\mathbf{B}}^{+}_{i_b}$ and $\widetilde{\mathbf{B}}^{+}_{i_b}\widetilde{\mathbf{f}}_{i_a}$ satisfy the Lipschitz continuous condition about $\mathbf{x}$; that is, for all $t \in [0, T]$, $\mathbf{x}, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$,

$$\begin{gathered} \left\| \widetilde{\mathbf{B}}^{+}_{i_b}(\mathbf{x}, t) \right\| \leqslant b_{i_b}, \\ \left\| \widetilde{\mathbf{B}}^{+}_{i_b}(\mathbf{x}_1, t) - \widetilde{\mathbf{B}}^{+}_{i_b}(\mathbf{x}_2, t) \right\| \leqslant k_{i_b} \left\| \mathbf{x}_1 - \mathbf{x}_2 \right\|, \\ \left\| \widetilde{\mathbf{B}}^{+}_{i_b}(\mathbf{x}_1, t)\, \widetilde{\mathbf{f}}_{i_a}(\mathbf{x}_1, t) - \widetilde{\mathbf{B}}^{+}_{i_b}(\mathbf{x}_2, t)\, \widetilde{\mathbf{f}}_{i_a}(\mathbf{x}_2, t) \right\| \leqslant k_{\mathbf{i}} \left\| \mathbf{x}_1 - \mathbf{x}_2 \right\|, \end{gathered} \quad (17)$$

where $\widetilde{\mathbf{B}}^{+}_{i_b} = [\widetilde{\mathbf{B}}^{\mathrm{T}}_{i_b} \widetilde{\mathbf{B}}_{i_b}]^{-1} \widetilde{\mathbf{B}}^{\mathrm{T}}_{i_b}$, $b_{i_b}$ is the bound of $\widetilde{\mathbf{B}}^{+}_{i_b}$, and $k_{i_b}$ and $k_{\mathbf{i}}$ are the Lipschitz coefficients of $\widetilde{\mathbf{B}}^{+}_{i_b}$ and $\widetilde{\mathbf{B}}^{+}_{i_b}\widetilde{\mathbf{f}}^{+}_{i_a}$, respectively.

Though ILC can deal with system with model uncertainty, the more accurate the model is, the faster the transient response will be. Based on this idea, multiple models are built to cover the parametric uncertainty, so that at each iteration the most approximate model to current system will be selected and finally the performance of transience response can be improved.

*3.2. The Establishment of Controllers.* For model algorithmic ILC, learning law (5) is based on the established model of the controlled plant. Corresponding to established multiple models, we present multiple controllers according to ILC learning law (5); that is,

$$C = \left\{ C_{\mathbf{i}} \mid \mathbf{i} \in \Omega \right\}, \quad (18)$$

where

$$\begin{aligned} C_{\mathbf{i}} : \mathbf{u}_{k+1}(t) = {} & \mathbf{u}_k(t) + \widetilde{\mathbf{B}}^{+}_{i_b}\left( \mathbf{x}_d(t), t \right) \left[ \dot{\mathbf{x}}_d(t) - \widetilde{\mathbf{f}}_{i_a}\left( \mathbf{x}_d(t), t \right) \right] \\ & - \widetilde{\mathbf{B}}^{+}_{i_b}\left( \mathbf{x}_k(t), t \right) \left[ \dot{\mathbf{x}}_k(t) - \widetilde{\mathbf{f}}_{i_a}\left( \mathbf{x}_k(t), t \right) \right]. \end{aligned} \quad (19)$$

*Remark 10.* For system (9), if parameters $\alpha_k(t)$ and $\beta_k(t)$ finally stop jumping at a group of fixed values, every single controller $C_i$ can guarantee output to converge to desired trajectory according to Lemma 1. However, the learning process has to restart as the single model may be far away from the plant when parameters change, and the process to convergent will be relatively long.

In this section, we have established multiple controllers according to the uncertainty of parameters and we hope that the most approximate model and corresponding controller will be switched into control system before every iteration so that the transient response time can be greatly reduced. Though the controller set is based on model algorithmic ILC learning law (5) and every single controller can guarantee system convergence according to Remark 10, how to keep the integrated system convergent with infinite switching times among multiple controllers is essential to MMILC.

*3.3. Switching Index Function.* When parameters jump, original model is no longer suitable. From this perspective, the switching scheme focuses on finding the most appropriate model for current environment from model set, further activating the corresponding controller. The ultimate purpose is to improve the transient response of the adaptive system with the least prior knowledge. The switching index function is given as

$$J_{\mathbf{i},k} = \mu \left\| \mathbf{e}^{\mathbf{i}}_k(T) \right\|_2^2 + \nu \int_0^T e^{-\gamma(T-\tau)} \left\| \mathbf{e}^{\mathbf{i}}_k(\tau) \right\|_2^2 \mathrm{d}\tau, \quad (20)$$

where $\mathbf{e}^{\mathbf{i}}_k(t)$ denotes the error between system state and the state of model $M_{\mathbf{i}}$ at the $k$th iteration:

$$\begin{gathered} \mathbf{e}^{\mathbf{i}}_k(t) = \mathbf{x}_k(t) - \widetilde{\mathbf{x}}^{\mathbf{i}}_k(t), \\ \dot{\widetilde{\mathbf{x}}}^{\mathbf{i}}_k(t) = \widetilde{\mathbf{f}}_{i_a}\left( \widetilde{\mathbf{x}}^{\mathbf{i}}_k(t), t \right) + \widetilde{\mathbf{B}}_{i_b}\left( \widetilde{\mathbf{x}}^{\mathbf{i}}_k(t), t \right) \mathbf{u}_k(t). \end{gathered} \quad (21)$$

$\tau > 0$ is the forgetting factor; $\mu > 0$ and $\nu > 0$ are parameters to adjust the weights of current and past errors in the calculation of switching index.

At the end of $k$th iteration, switching index function $J_{\mathbf{i},k}$ of every model $M_{\mathbf{i}}$ $(\mathbf{i} \in \Omega)$ is calculated and the most suitable model for the $k$th iteration is decided by minimum $J_{\mathbf{i},k}$; that is, model $M_{\mathbf{l}(k+1)}$ will be selected if the following equation is satisfied:

$$\mathbf{l}(k+1) = \arg\min_{i\in\Omega} J_{\mathbf{i},k} \tag{22}$$

and $\mathbf{l}(0) = (l_a(0), l_b(0))$, where $l_a(0) = \text{random}\{1, 2, \ldots, d_a\}$ and $l_b(0) = \text{random}\{1, 2, \ldots, d_b\}$. Corresponding controller $C_{\mathbf{l}(k+1)}$ will be immediately selected as the controller for the next iteration.

*3.4. Multiple Model ILC Control Algorithm.* Based on the above discussion, the main procedures for multiple model ILC control algorithm are provided.

*Algorithm 11.* (1) According to varying scope of $\alpha_k(t)$ and $\beta_k(t)$, multiple model set will be established as follows:

$$M = \left\{M_{\mathbf{i}} = \left\{\widetilde{\mathbf{f}}_{i_a}(\mathbf{x}(t), t), \widetilde{\mathbf{B}}_{i_b}(\mathbf{x}(t), t)\right\} \mid \mathbf{i} \in \Omega\right\}, \tag{23}$$

where $\Omega = \{\mathbf{i} = (i_a, i_b) \mid i_a = 1, 2, \ldots, d_a;\ i_b = 1, 2, \ldots, d_b\}$.

(2) Based on Model Algorithmic Learning law (5), multiple controllers responding to the model set $M$ will be set up as

$$C = \{C_{\mathbf{i}} \mid \mathbf{i} \in \Omega\}, \tag{24}$$

where

$$\begin{aligned} C_{\mathbf{i}}\colon \mathbf{u}_{k+1}(t) = \mathbf{u}_k(t) &+ \widetilde{\mathbf{B}}^{+}_{i_b}(\mathbf{x}_d(t), t)\left[\dot{\mathbf{x}}_d(t) - \widetilde{\mathbf{f}}_{i_a}(\mathbf{x}_d(t), t)\right] \\ &- \widetilde{\mathbf{B}}^{+}_{i_b}(\mathbf{x}_k(t), t)\left[\dot{\mathbf{x}}_k(t) - \widetilde{\mathbf{f}}_{i_a}(\mathbf{x}_k(t), t)\right]. \end{aligned} \tag{25}$$

(3) At the initial iteration, controller $C_{\mathbf{l}(0)}$ is randomly selected,

$$\mathbf{l}(0) = (l_a(0), l_b(0)), \tag{26}$$

where $l_a(0) = \text{random}\{1, 2, \ldots, d_a\}$, $l_b(0) = \text{random}\{1, 2, \ldots, d_b\}$.

(4) At the $k$th iteration, controller $C_{\mathbf{l}(k)}$ is selected as the current controller. At the end of $k$th iteration, switching index function of every model $J_{\mathbf{i},k}$ will be calculated and the $M_{\mathbf{l}(k+1)}$ is selected if

$$\mathbf{l}(k+1) = \arg\min_{i\in\Omega} J_{i,k}. \tag{27}$$

(5) $k = k + 1$; go to step 4.

## 4. Convergence Analysis

Convergence is the essential condition of controller design. Though every single model and corresponding controller can keep the convergence of system, how to guarantee that the entire MMILC is stable even with infinite switching among model set is the key problem for the proposed algorithm. Conditions for convergent MMILC and its detailed proof are given here.

*4.1. System with Unknown Constant Parameter.* For system (9), if parameters $\alpha_k(t)$ and $\beta_k(t)$ are unknown constant values irrelative with $t$ and $k$, we have the following theorem.

**Theorem 12.** *For iteratively operating system depicted as* (9) *but with unknown constant parameters $\alpha_k(t)$ and $\beta_k(t)$ which are irrelative with t and k, if Assumptions 7–9 and the following conditions are satisfied,*

(1) $\forall i_b \in 1, 2, \ldots, d_b$,

$$\left\|\mathbf{I} - \widetilde{\mathbf{B}}^{+}_{i_b}(\mathbf{x}, t)\,\mathbf{B}(\mathbf{x}, t)\right\| \leqslant \rho' < 1, \tag{28}$$

(2) $\forall k$,

$$\mathbf{x}_k(0) = \mathbf{x}_d(0), \tag{29}$$

*then, for any given reachable desired trajectory $\mathbf{x}_d(t)$ $(t \in [0, T])$, the proposed MMILC learning algorithm will generate a control sequence, which can guarantee that the system state is convergent to the desired trajectory; that is,* $\lim_{k\to\infty}\|\mathbf{x}_d(t) - \mathbf{x}_k(t)\| = 0$ $(t \in [0, T])$.

*Proof.* According to the MMILC method, multiple models $M_{\mathbf{i}} = \{\widetilde{\mathbf{f}}_{i_a}(\mathbf{x}(t), t), \widetilde{\mathbf{B}}_{i_b}(\mathbf{x}(t), t)\}$ $(\mathbf{i} = (i_a, i_b) \in \Omega)$ will be established based on the parameters' varying scope. For notational simplify, we denote $\mathbf{L}_{i_b}(\mathbf{x}, t) = \widetilde{\mathbf{B}}^{+}_{i_b}(\mathbf{x}, t)$ and $\mathbf{h}_{\mathbf{i}}(\mathbf{x}, t) = \mathbf{L}_{i_b}(\mathbf{x}, t)[\dot{\mathbf{x}}_d(t) - \widetilde{\mathbf{f}}_{i_a}(\mathbf{x}, t)]$. As $\widetilde{\mathbf{B}}^{+}_{i_b}$ and $\widetilde{\mathbf{B}}^{+}_{i_b}\widetilde{\mathbf{f}}_{i_a}$ are Lipschitz continuous about $\mathbf{x}$ by Assumption 8, $\mathbf{h}_{\mathbf{i}}$ satisfies the Lipschitz condition; that is, $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$, $t \in [0, T]$,

$$\left\|\mathbf{h}_{\mathbf{i}}(\mathbf{x}_1, t) - \mathbf{h}_{\mathbf{i}}(\mathbf{x}_2, t)\right\| \leqslant k_{\mathbf{h}_{\mathbf{i}}}\left\|\mathbf{x}_1 - \mathbf{x}_2\right\|. \tag{30}$$

From the switch index function (27) of MMILC algorithm, model $M_{\mathbf{l}(k)} = (\widetilde{\mathbf{f}}_{l_a(k)}, \widetilde{\mathbf{B}}_{l_b(k)})$ will be selected at the $k$th iteration. So focusing on every single iteration, the proposed MMILC can be viewed as single model ILC. As parameters $\alpha_k(t)$ and $\beta_k(t)$ are both constant, dynamic system is strictly iteration-time invariant, and the system Assumptions 7 and 8 can be easily guaranteed. As model Assumption 9 is also satisfied, according to Corollary 2, at the $k$th iteration, for every single element model $M_{\mathbf{i}}$, $\mathbf{i} \in \Omega$, we have

$$\left\|\Delta\mathbf{x}_k\right\| \leqslant b'_{\mathbf{B}}\int_0^t e^{b'_2(t-\tau)}\left\|\Delta\mathbf{u}_k\right\| \mathrm{d}\tau, \tag{31a}$$

$$\left\|\Delta\mathbf{u}_{k+1}\right\|_\lambda \leqslant \left(\rho' + b_{1\mathbf{i}} b'_{\mathbf{B}}\lambda_{b'_2}\right)\left\|\Delta\mathbf{u}_k\right\|_\lambda, \tag{31b}$$

where $b'_2 = k'_{\mathbf{f}} + k'_{\mathbf{B}} b_{\mathbf{u}d}$, $b_{1\mathbf{i}} = k_{\mathbf{h}_{\mathbf{i}}} + b_{ib} b'_2$, and $\lambda_{b'_2} = (1 - e^{(b'_2-\lambda)T})/(\lambda - b'_2)$.

Further, if we denote $k'_{\mathbf{h}} = \max_{i\in\Omega} k_{\mathbf{h}_i}$, $b'_{\mathbf{L}} = \max_{i_b\in\{1,2,\ldots,d_b\}} b_{i_b}$, and $b'_1 = k'_{\mathbf{h}} + b'_{\mathbf{L}} b'_2$, for arbitrary model switching among the model set $M$, (31b) can be written as

$$\left\|\Delta\mathbf{u}_{k+1}\right\|_\lambda \leqslant \left(\rho' + b'_1 b'_{\mathbf{B}}\lambda_{b'_2}\right)\left\|\Delta\mathbf{u}_k\right\|_\lambda. \tag{32}$$

Since $\rho' < 1$, it is possible to find $\lambda > 0$ large enough so that $\rho' + b_1' b_{\mathbf{B}}' \lambda_{b_2}' < 1$. Therefore, $\|\Delta \mathbf{u}_{k+1}\|_\lambda \rightarrow 0$ as $k \rightarrow \infty$. Since

$$\begin{aligned} \sup_{t\in[0,T]} \|\Delta \mathbf{u}_k(t)\| &= \sup_{t\in[0,T]} \left\{ e^{\lambda t} e^{-\lambda t} \|\Delta \mathbf{u}_k(t)\| \right\} \\ &\leqslant e^{\lambda T} \sup_{t\in[0,T]} \left\{ e^{-\lambda t} \|\Delta \mathbf{u}_k(t)\| \right\} \qquad (33) \\ &= e^{\lambda T} \|\Delta \mathbf{u}_k(t)\|_\lambda, \end{aligned}$$

therefore, we have $\sup_{t\in[0,T]} \|\Delta \mathbf{u}_k(t)\| \rightarrow 0$ as $k \rightarrow \infty$; that is to say, $\mathbf{u}_k(t) \rightarrow \mathbf{u}_d(t)$ for $t \in [0,T]$ when $k \rightarrow \infty$. Furthermore, combing (31a), we have $\mathbf{x}_k(t) \rightarrow \mathbf{x}_d(t)$ uniformly on $[0,T]$. □

### *4.2. System with Unknown Jumping Parameter*

**Corollary 13.** *For iteratively operating system with jumping parameters described as* (9)*, in addition to system Assumptions 5–8 and model Assumption 9, if the following conditions are also satisfied,*

(1) *for* $i_b \in \{1, 2, \ldots, d_b\}$,

$$\left\| \mathbf{I} - \widetilde{\mathbf{B}}_{i_b}^{+}(\mathbf{x}, t)\, \mathbf{B}(\mathbf{x}, t) \right\| \leqslant \rho'' < 1, \qquad (34)$$

(2) *for all k,*

$$\mathbf{x}_k(0) = \mathbf{x}_d(0), \qquad (35)$$

*then, for any given reachable desired trajectory* $\mathbf{x}_d(t)$ ($t \in [0,T]$)*, the proposed MMILC learning algorithm will generate a control sequence, which can improve the control property for this kind of system.*

We just consider the time variant system with piecewise constant parameters. According to Assumption 6, the jumping parameters will be kept for relatively long time, and the system in two adjacent parameter jumpings can be regarded as a time-invariant system. Due to the existence of multiple model, the transient response will be improved greatly, so the state of system will converge to the desired trajectory rapidly, and the control property of piecewise-constant jumping system will be improved greatly.

## 5. Simulation Results

Consider a single-link robotic manipulator with the following dynamic system:

$$I\ddot{\theta}_k + d(t)\,\dot{\theta}_k + mgl\cos\theta_k = \tau_k \qquad (36)$$

which operates iteratively on the fixed time interval; that is, $t \in [0, 10]$, and $k$ denotes the iterations. $\theta$, $\dot{\theta}$, and $\ddot{\theta}$ are the joint angle, angular velocity, and accelerated velocity, respectively. $I = 0.75ml^2\ \mathrm{kg\cdot m^2}$ is the moment of inertia of the manipulator, $l = 1.25\ \mathrm{m}$ is the length of the manipulator, $m = 1.0\ \mathrm{kg}$ is the mass of the joint, $g = 9.8\ \mathrm{m/s^2}$ is the gravity, $d_k(t)\ \mathrm{N\cdot m\cdot s/rad}$ is the viscous friction coefficient which is decided by the shape and size of the link and the viscidity of the lubricant, and $d_k(t)$ is unknown parameter varying in $[0, 5]$. $\tau_k$ is the control input.

Let $x_{1k}(t) = \theta$ and $x_{2k}(t) = \dot{\theta}$; the above dynamic system can be rewritten in the form of state equation:

$$\begin{aligned} \begin{bmatrix} \dot{x}_{1k}(t) \\ \dot{x}_{2k}(t) \end{bmatrix} &= \begin{bmatrix} x_{2k}(t) \\ -\dfrac{d_k(t)}{I} - \dfrac{mgl}{I}\cos x_{1k}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{1}{I} \end{bmatrix} \tau_k(t) \\ &= \begin{bmatrix} x_{2k}(t) \\ -\dfrac{4d_k(t)}{3} - \dfrac{49}{3}\cos x_{1k}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{4}{3} \end{bmatrix} \tau_k(t). \end{aligned} \qquad (37)$$

The manipulator is controlled to fulfil repeated transfer task. Correspondingly, the control target is to manipulate the control input $\tau_k(t)$ to guarantee the control state to track the desired trajectory:

$$\mathbf{x}_d(t) = \begin{bmatrix} \cos(\pi t) \\ -\pi\sin(\pi t) \end{bmatrix}. \qquad (38)$$

In the simulation, we consider two cases in which parameter $d_k(t)$ is unknown constant or jumping varying in the prior known scope $[0, 5]$. First, 6 different values $d_i = i - 5/6$ $(i = 1, 2, \ldots, 6)$ are selected to cover the varying scope; that is,

$$\begin{aligned} d_1 &= \frac{1}{6}; & d_2 &= \frac{7}{6}; & d_3 &= \frac{13}{6}; \\ d_4 &= \frac{19}{6}; & d_5 &= \frac{25}{6}; & d_6 &= \frac{31}{6}. \end{aligned} \qquad (39)$$

Consequently, 6 models will be established as

$$M_i = \left\{ \widetilde{\mathbf{f}}_i, \widetilde{\mathbf{B}}_i \right\}, \qquad (40)$$

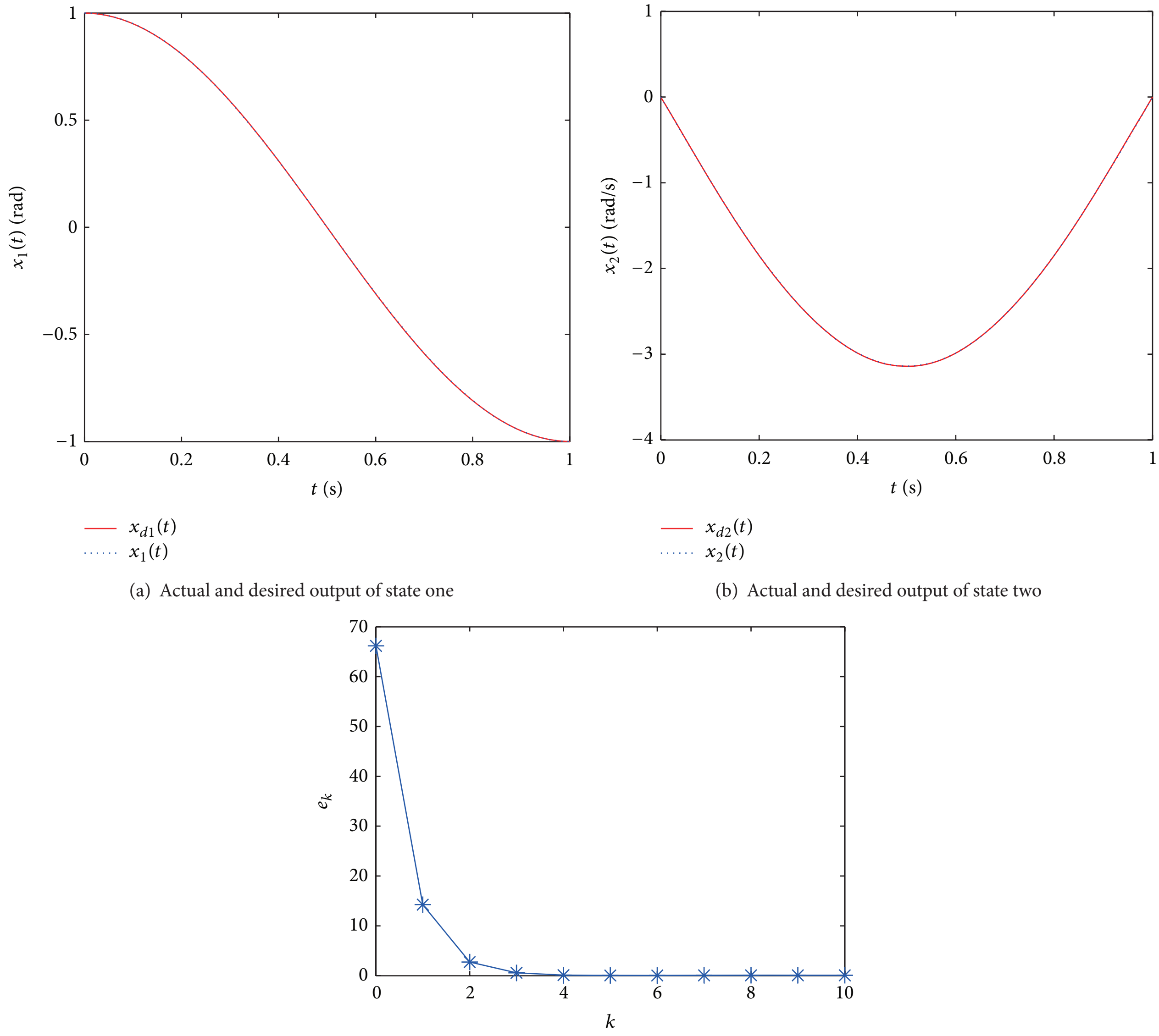
where $\widetilde{\mathbf{f}}_i = \left[\begin{smallmatrix} x_{2k}(t) \\ -4d_i/3-(49/3)\cos x_{1k}(t) \end{smallmatrix}\right]$, $\widetilde{\mathbf{B}}_i = \left[\begin{smallmatrix} 0 \\ 4/3 \end{smallmatrix}\right]$, and $i = 1, 2, \ldots, 6$.

As an illustrative example, there is jumping parameter only in the $\mathbf{f}$ part; the following can be verified:

$$\frac{\mathbf{df}}{\mathbf{dx}^{\mathrm{T}}} = \begin{bmatrix} 0 & 1 \\ \dfrac{49\sin \mathbf{x}_{1k}(t)}{3} & 0 \end{bmatrix} \leqslant \begin{bmatrix} 0 & 1 \\ \dfrac{49}{3} & 0 \end{bmatrix} \qquad (41)$$

so it can been seen apparently that the system and established multiple models satisfy all the Lipschitz and norm bounded conditions. So the proposed MMILC algorithm can be applied to the controlled system according to Theorem 12 and Corollary 13.

*5.1. Plant with Unknown Time-Invariant Parameter.* If plant (37) contains unknown time-invariant parameter, control results by using single model ILC algorithm (5) and proposed multiple model ILC Algorithm 11 can be obtained as follows.



(a) Actual and desired output of state one

(b) Actual and desired output of state two

(c) State error with iteration

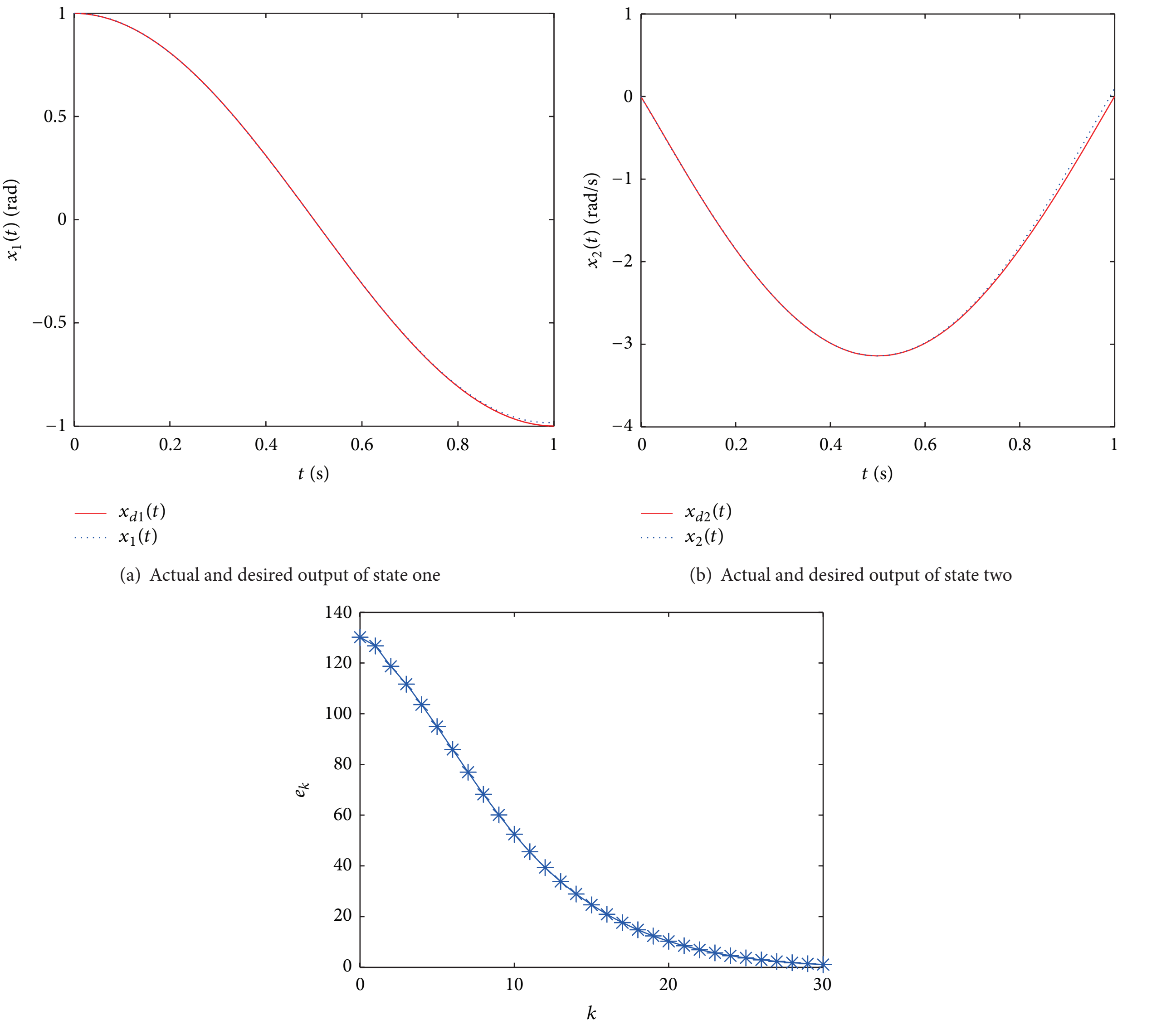
Figure 1: Single model ILC when $d_k(t) = 3/7$.

*5.1.1. Single Model ILC.* Suppose that the established model of plant (37) is $\{\tilde{\mathbf{f}}, \tilde{\mathbf{B}}\}$, where $\tilde{\mathbf{f}} = \left[\begin{smallmatrix} x_{k2} \\ -(\tilde{d}/I)x_{k_2} - (mgl/I)\cos x_{k1} \end{smallmatrix}\right]$, $\tilde{\mathbf{B}} = \left[\begin{smallmatrix} 0 \\ 1/I \end{smallmatrix}\right]$, and $\tilde{d} = 1/6$. When the unknown system parameter is $d_k(t) = 3/7$ and $d_k(t) = 33/7$, respectively, corresponding control results are shown in Figures 1 and 2, where $e_k = \int_0^1 x_{k1}^2(t)\mathrm{d}t + \int_0^1 x_{k2}^2(t)\mathrm{d}t$.

When model parameter $\tilde{d} = 1/6$ is close to actual unknown plant parameter $d_k(t) = 3/7$, single model ILC can track the desired trajectory with about 6 iterations shown as Figure 1(c). However, when there is relative great mismatch between model parameter $\tilde{d} = 1/6$ and plant parameter $d_k(t) = 33/7$, it takes more than 30 iterations before the desired state output is obtained as shown in Figure 2.

*5.1.2. Multiple Model ILC.* In this part, when unknown system parameter is $d_k(t) = 33/7$, multiple models are established as (40), and the multiple model ILC control Algorithm 11 is applied to the plant, control result can be found in Figure 3.

As in Figure 2, when there is relatively great mismatch between model parameter and system parameter, single model ILC cannot track the desired trajectory within reasonable iterations. However, in multiple model ILC control scheme, multiple models are established to cover the parameter's varying scope; in Figure 3(b), MMILC can identify that $M_1$ with $d_1 = 1/6$ is not the best one for plant, and the most appropriate model $M_6$ with model parameter $d_6 = 31/6$ is selected immediately since the second iteration. Since the introduced multiple model set and switching mechanism,



(a) Actual and desired output of state one

(b) Actual and desired output of state two

(c) State error with iteration

FIGURE 2: Single model ILC when $d_k(t) = 33/7$.

iterations are decreased greatly to about 6 as shown in Figure 3(a) compared with single model ILC with $\tilde{d} = 1/6$ shown in Figure 2.

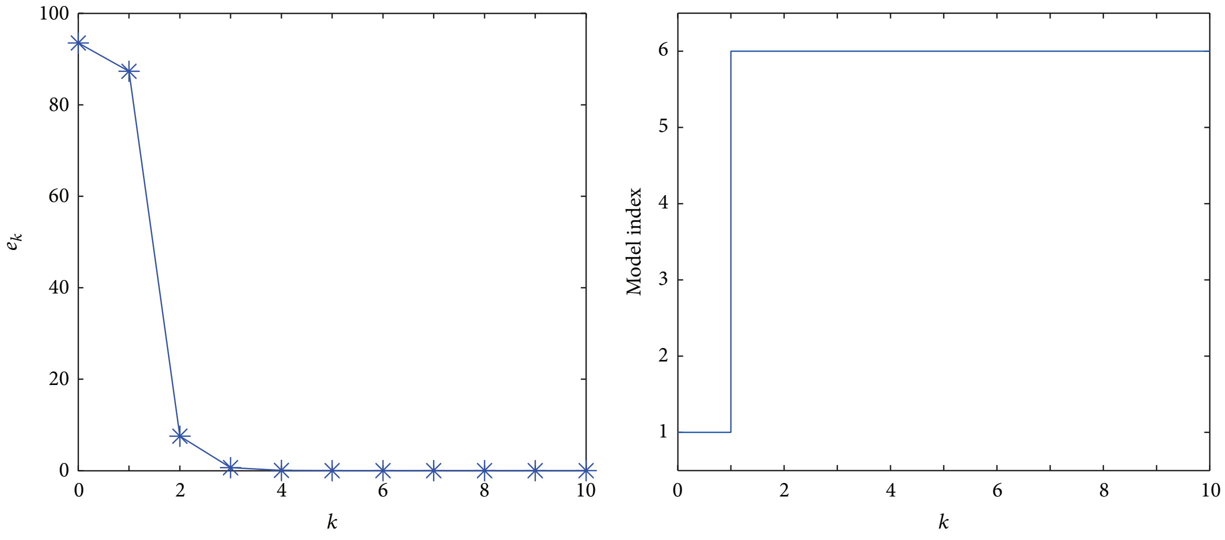
*5.2. Plant with Jumping Parameter.* For plant (37), if $d_k(t)$ is a piece-wise constant jumping parameter in the form

$$d_k(t) = \begin{cases} \dfrac{3}{7} & \text{if } k < 30,\ 0 \leqslant t \leqslant 10,\ k = 30,\ 0 \leqslant t < 6 \\ \dfrac{33}{7} & \text{if } k = 30,\ 6 \leqslant t \leqslant 10, \\ & \quad 30 < k < 60,\ 0 \leqslant t \leqslant 10, \\ & \quad k = 60,\ 0 \leqslant t < 3 \\ \dfrac{23}{7} & \text{if } k = 60,\ 3 \leqslant t \leqslant 10,\ k < 90,\ 0 \leqslant t \leqslant 10, \end{cases} \tag{42}$$

This parameter jumping process is shown as in Figure 4(b) in which the length of each iteration is equal to the fixed time interval.

If single model ILC with parameter $\tilde{d} = 1/6$ and multiple model ILC with multiple models as (40) are adopted to control the plant, respectively, state errors respective to each iteration of these two methods are shown in Figure 4(a).

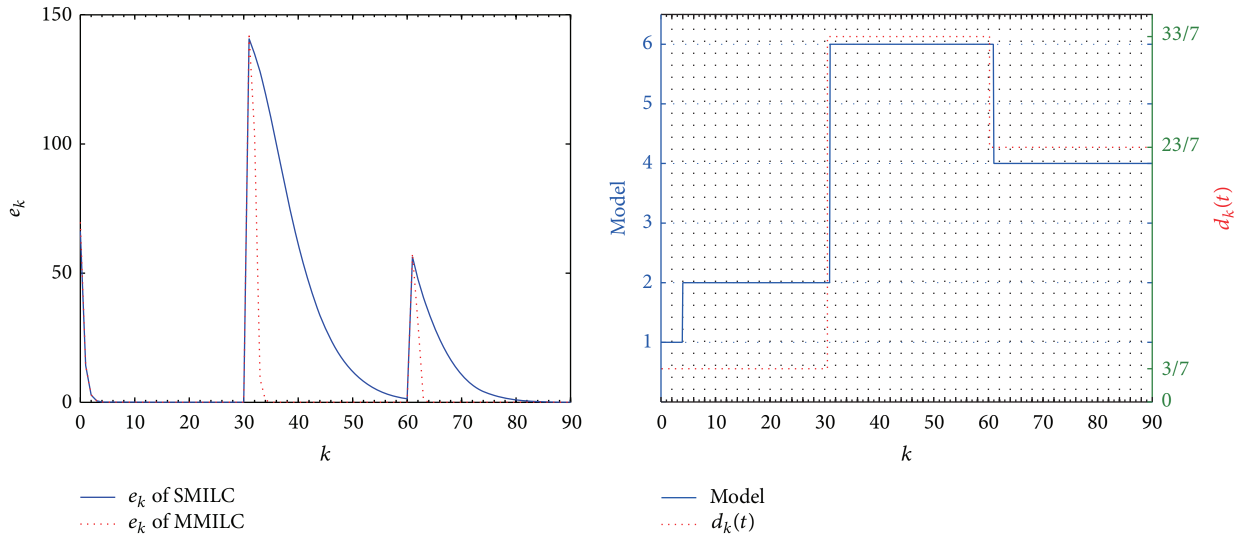
When parameter jumping happens, as the dynamic system will change at the same time, no matter single model or multiple model ILC, the learning process has to restart. In the first 30 iterations, $\tilde{d} = 1/6$ in single model ILC is close to system parameter $d(k) = 3/7$, so the control effect looks as well as the effect of MMILC. However, for single model ILC, when the parameter deviates far from model parameter, the approaching process gets relatively long, as in Figure 4(a); it takes about more than 30 and 20 iterations before the state



(a) State error with iteration

(b) Model switching process

FIGURE 3: Multiple model ILC when $d_k(t) = 33/7$.



(a) State errors of single and multiple model ILC

(b) Parameter jumping and model switching process

FIGURE 4: Control results of single and multiple model ILC dealing with plant with jumping parameter.

output converges to the desired track after the second and third parameter jumpings.

According to the MMILC strategy, multiple models are established to cover the parameter uncertainty; at the end of each iteration, the index functions showing the matching degree between each model and the current plant are calculated. In the next iteration, the closest matching model and corresponding controller will be selected. The parameter jumping and model switching process are shown in Figure 4(b). In the 30th iteration, there is a parameter jumping, consequently, model $M_6$ is selected at the 31st iteration; parameter change in the 60th iteration is also identified and the most closest model $M_4$ is switched to system. Taking this model switching scheme, convergent processes after each parameter jumping are reduced greatly to about 5 iterations as shown in Figure 4(a).

## 6. Conclusion

In this paper, we have incorporated multiple model adaptive control into model algorithm iterative learning control for nonlinear repetitive operating system with jumping parameters. In order to reduce the iterations before the control state converges to the desired trajectory when there is model mismatch, multiple models are set up to cover the parameter uncertainty; model index function has been established to evaluate the matching degree between each model and plant in current iteration, a switching mechanism has also been proposed so that the closest model and corresponding controller will be selected at every iteration. Further, the multiple model ILC algorithm is given and it has been proved that the generated control signal by MMILC can guarantee the output convergent to desired trajectory. Two numerical simulations show that the proposed MMILC can improve the transient response greatly.

## Appendix

## The Proof of Lemma 1 [24, 25]

*Proof.* For notational simplify, let

$$\begin{gathered}\mathbf{L}(\mathbf{x},t)=\widetilde{\mathbf{B}}(\mathbf{x},t),\qquad \mathbf{h}(\mathbf{x},t)=\mathbf{L}(\mathbf{x},t)\left[\dot{\mathbf{x}}_d(t)-\widetilde{\mathbf{f}}(\mathbf{x},t)\right]\\ \mathbf{x}_k=\mathbf{x}_k(t),\qquad \widetilde{\mathbf{x}}_k=\widetilde{\mathbf{x}}_k(t)\\ \mathbf{f}_k=\mathbf{f}\left(\mathbf{x}_k(t),t\right),\qquad \widetilde{\mathbf{f}}_k=\widetilde{\mathbf{f}}\left(\mathbf{x}_k(t),t\right)\\ \mathbf{f}_d=\mathbf{f}\left(\mathbf{x}_d(t),t\right),\qquad \widetilde{\mathbf{f}}_d=\widetilde{\mathbf{f}}\left(\mathbf{x}_d(t),t\right).\end{gathered}\tag{A.1}$$

For (5), let $\Delta\mathbf{u}_k=\mathbf{u}_d-\mathbf{u}_k$, $\Delta\mathbf{x}_k=\mathbf{x}_d-\mathbf{x}_k$; we have

$$\begin{aligned}\mathbf{u}_{k+1}&=\mathbf{u}_k-\mathbf{L}_k\left(\dot{\mathbf{x}}_k-\widetilde{\mathbf{f}}_k\right)+\mathbf{h}_d\\ &=\mathbf{u}_k+\mathbf{L}_k\left(\dot{\mathbf{x}}_d-\dot{\mathbf{x}}_k\right)+\mathbf{h}_d-\mathbf{h}_k\\ &=\mathbf{u}_k+\mathbf{L}_k\left\{\mathbf{f}_d-\mathbf{f}_k+\left(\mathbf{B}_d-\mathbf{B}_k\right)\mathbf{u}_d+\mathbf{B}_k\Delta\mathbf{u}_k\right\}\\ &\quad+\mathbf{h}_d-\mathbf{h}_k.\end{aligned}\tag{A.2}$$

Further,

$$\begin{aligned}\Delta\mathbf{u}_{k+1}={}&\left[\mathbf{I}-\mathbf{L}_k\mathbf{B}_k\right]\Delta\mathbf{u}_k-\mathbf{L}_k\left\{\mathbf{f}_d-\mathbf{f}_k+\left(\mathbf{B}_d-\mathbf{B}_k\right)\mathbf{u}_d\right\}\\ &-\left(\mathbf{h}_d-\mathbf{h}_k\right).\end{aligned}\tag{A.3}$$

For above equation, taking norms of both sides gives

$$\begin{aligned}\left\|\Delta\mathbf{u}_{k+1}\right\|&\leqslant\left\|\mathbf{I}-\mathbf{L}_k\mathbf{B}_k\right\|\left\|\Delta\mathbf{u}_k\right\|+\left\|\mathbf{h}_d-\mathbf{h}_k\right\|\\ &\quad+\left\|\mathbf{L}_k\right\|\left(\left\|\mathbf{f}_d-\mathbf{f}_k\right\|+\left\|\mathbf{B}_d-\mathbf{B}_k\right\|\left\|\mathbf{u}_d\right\|\right)\\ &\leqslant\rho\left\|\Delta\mathbf{u}_k\right\|+\left\{k_{\mathbf{h}}+b_{\mathbf{L}}\left(k_{\mathbf{f}}+k_{\mathbf{B}}b_{\mathbf{u}d}\right)\right\}\left\|\Delta\mathbf{x}_k\right\|\\ &=\rho\left\|\Delta\mathbf{u}_k\right\|+b_1\left\|\Delta\mathbf{x}_k\right\|,\end{aligned}\tag{A.4}$$

where $b_{\mathbf{L}}$ is the norm bound of matrix $\mathbf{L}$, $b_{\mathbf{u}d}=\sup_{t\in[0,T]}\|\mathbf{u}_d(t)\|$, and $b_1=k_{\mathbf{h}}+b_{\mathbf{L}}(k_{\mathbf{f}}+k_{\mathbf{B}}b_{\mathbf{u}d})$.

Now, since $\mathbf{x}_k(0)=\mathbf{x}_d(0)$, for all $k$, from (1), we have

$$\begin{aligned}\left\|\Delta\mathbf{x}_k\right\|&=\left\|\int_0^t\left\{\mathbf{f}_d+\mathbf{B}_d\mathbf{u}_d-\left(\mathbf{f}_k+\mathbf{B}_k\mathbf{u}_k\right)\right\}\mathrm{d}\tau\right\|\\ &\leqslant\int_0^t\left(\left\|\mathbf{f}_d-\mathbf{f}_k\right\|+\left\|\mathbf{B}_d-\mathbf{B}_k\right\|\left\|\mathbf{u}_d\right\|+\left\|\mathbf{B}_k\right\|\left\|\Delta\mathbf{u}_k\right\|\right)\mathrm{d}\tau\\ &\leqslant\int_0^t\left\{\left(k_{\mathbf{f}}+k_{\mathbf{B}}b_{\mathbf{u}d}\right)\left\|\Delta\mathbf{x}_k\right\|+b_{\mathrm{B}}\left\|\Delta\mathbf{u}_k\right\|\right\}\mathrm{d}\tau,\end{aligned}\tag{A.5}$$

where $b_{\mathbf{B}}$ is the norm bound of matrix $\mathbf{B}$. Considering the Bellman-Gronwall Lemma, we have

$$\left\|\Delta\mathbf{x}_k\right\|\leqslant b_{\mathbf{B}}\int_0^t e^{b_2(t-\tau)}\left\|\Delta\mathbf{u}_k\right\|\mathrm{d}\tau,\tag{A.6}$$

where $b_2=k_{\mathbf{f}}+k_{\mathbf{B}}b_{\mathbf{u}d}$. Combing (A.4) with (A.6), we get

$$\left\|\Delta\mathbf{u}_{k+1}\right\|\leqslant\rho\left\|\Delta\mathbf{u}_k\right\|+b_1b_{\mathbf{B}}\int_0^t e^{b_2(t-\tau)}\left\|\Delta\mathbf{u}_k\right\|\mathrm{d}\tau.\tag{A.7}$$

Multiplying the above equation with the positive function $e^{-\lambda t}$, we get

$$\begin{aligned}e^{-\lambda t}\left\|\Delta\mathbf{u}_{k+1}\right\|&\leqslant\rho e^{-\lambda t}\left\|\Delta\mathbf{u}_k\right\|\\ &\quad+b_1b_{\mathbf{B}}\int_0^t e^{(b_2-\lambda)(t-\tau)}e^{-\lambda\tau}\left\|\Delta\mathbf{u}_k\right\|\mathrm{d}\tau.\end{aligned}\tag{A.8}$$

From the notation of $\lambda$-norm vector, we have

$$\left\|\Delta\mathbf{u}_{k+1}\right\|_\lambda\leqslant\left(\rho+b_1b_{\mathbf{B}}\lambda_{b_2}\right)\left\|\Delta\mathbf{u}_k\right\|_\lambda,\tag{A.9}$$

where $\lambda_{b_2}=(1-e^{(b_2-\lambda)\mathrm{T}})/(\lambda-b_2)$. Since $\rho<1$, it is possible to find that $\lambda>0$ is large enough so that $\rho+b_1b_{\mathbf{B}}\lambda_{b_2}<1$. Therefore, $\|\Delta\mathbf{u}_{k+1}\|_\lambda\to 0$ as $k\to\infty$. By the definition of $\lambda$-norm (3), we know that

$$\begin{aligned}\sup_{t\in[0,T]}\left\|\Delta\mathbf{u}_k(t)\right\|&=\sup_{t\in[0,T]}\left\{e^{\lambda t}e^{-\lambda t}\left\|\Delta\mathbf{u}_k(t)\right\|\right\}\\ &\leqslant e^{\lambda T}\sup_{t\in[0,T]}\left\{e^{-\lambda t}\left\|\Delta\mathbf{u}_k(t)\right\|\right\}\\ &=e^{\lambda T}\left\|\Delta\mathbf{u}_k(t)\right\|_\lambda.\end{aligned}\tag{A.10}$$

Therefore, we have $\sup_{t\in[0,T]}\|\Delta\mathbf{u}_k(t)\|\to 0$ as $k\to\infty$; that is to say, $\mathbf{u}_k(t)\to\mathbf{u}_d(t)$ for $t\in[0,T]$ when $k\to\infty$. Furthermore, combing equation (A.6), we have $\mathbf{x}_k(t)\to\mathbf{x}_d(t)$ uniformly on $[0,T]$. □

## Conflict of Interests

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

## Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities under Grant [FRF-TP-12-005B]; the Program for New Century Excellent Talents in Universities under Grant [NCET-11-0578]; and the Specialized Research Fund for the Doctoral Program of Higher Education (SRFDP) under Grant [20130006110008].

## References

[1] S. Yin, X. Yang, and H. R. Karimi, "Data-driven adaptive observer for fault diagnosis," *Mathematical Problems in Engineering*, vol. 2012, Article ID 832836, 21 pages, 2012.

[2] S. Yin, G. Wang, and H. R. Karimi, "Data-driven design of robust fault detection system for wind turbines," *Mechatronics*, vol. 24, no. 4, pp. 298–306, 2014.

[3] S. Yin, X. W. Li, H. J. Gao, and O. Kaynak, "Data-based techniques focused on modern industry: an overview," *IEEE Transactions on Industrial Electronics*, no. 99, pp. 1–11, 2014.

[4] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *Journal of Robotic Systems*, vol. 1, no. 2, pp. 123–140, 1984.

[5] Y. Chen and K. L. Moore, "Iterative learning control with iteration-domain adaptive feedforward compensation," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, pp. 4416–4421, Maui, Hawaii, USA, December 2003.

[6] M. Gautier, A. Jubien, and A. Janot, "New iterative learning identification and model based control of robots using only actual motor torque data," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics: Mechatronics for Human Wellbeing (AIM '13)*, pp. 1436–1441, July 2013.

[7] M. Mezghani, G. Roux, M. Cabassud, M. V. Le Lann, B. Dahhou, and G. Casamatta, "Application of iterative learning control to an exothermic semibatch chemical reactor," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 6, pp. 822–834, 2002.

[8] J. H. Chen, N. Cheng, and Y.-C. Cheng, "Hybrid model based iterative learning control for semiconductor processes with uncetain metrology delay," in *Proceedings of the 6th IEEE Conference on Industrial Electronics and Applications (ICIEA '11)*, pp. 1519–1524, June 2011.

[9] K. L. Moore, Y. Q. Chen, and H. S. Ahn, "Iterative learning control: a tutorial and big picture view," in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 2352–2357, 2006.

[10] R. H. Chi, Z. S. Hou, S. T. Jin, and D. W. Wang, "Discrete-time adaptive ILC for non-parametric uncertain nonlinear systems with iteration-varying trajectory and random initial condition," *Asian Journal of Control*, vol. 15, no. 2, pp. 562–570, 2013.

[11] X.-D. Li, T. W. Chow, and L. L. Cheng, "Adaptive iterative learning control of non-linear MIMO continuous systems with iteration-varying initial error and reference trajectory," *International Journal of Systems Science. Principles and Applications of Systems and Integration*, vol. 44, no. 4, pp. 786–794, 2013.

[12] E. A. Omer and K. L. Moore, "Adaptive reduced-order control of discrete repetitive processes with iteration-varying reference signals," in *Proceedings of the IEEE International Symposium on Intelligent Control (ISIC '12)*, pp. 1238–1243, 2012.

[13] Z. Hou, J. Yan, J.-X. Xu, and Z. Li, "Modified iterative-learning-control-based ramp metering strategies for freeway traffic control with iteration-dependent factors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 606–618, 2012.

[14] H. S. Ahn, K. L. Moore, and Y. Q. Chen, "Monotonic convergent iterative learning controller design with iteration varying model uncertainty," in *Proceedings of the IEEE International Conference on Mechatronics and Automation*, vol. 2, pp. 572–577, 2005.

[15] H.-S. Ahn, K. L. Moore, and Y. Q. Chen, "Monotonic convergent iterative learning controller design based on interval model conversion," *IEEE Transactions on Automatic Control*, vol. 51, no. 2, pp. 366–371, 2006.

[16] H. S. Ahn, K. L. Moore, and Y. Chen, "Stability analysis of discrete-time iterative learning control systems with interval uncertainty," *Automatica*, vol. 43, no. 5, pp. 892–902, 2007.

[17] D. H. Nguyen and D. Banjerdpongchai, "Robust iterative learning control for linear systems with iteration-varying parametric uncertainties," in *Proceedings of the 7th Asian Control Conference (ASCC '09)*, pp. 716–721, August 2009.

[18] J. Yan and Z. Hou, "Iterative learning control based ramp metering tracking iteration-varying trajectories with internal model," in *Proceedings of the Chinese Control and Decision Conference (CCDC '09)*, pp. 1713–1718, Guilin, China, June 2009.

[19] J.-X. Xu and Y. Tan, "A composite energy function-based learning control approach for nonlinear systems with time-varying parametric uncertainties," *IEEE Transactions on Automatic Control*, vol. 47, no. 11, pp. 1940–1945, 2002.

[20] C. K. Yin and Z. S. Hou, "Adaptive iterative learning control for a class of iteration-varying nonlinear systems with known periodicity," in *Proceedings of the International Conference on Test and Measurement*, vol. 2, pp. 433–437, December 2009.

[21] Z. X. Li, C. K. Yin, S. T. Jin, and Z. S. Hou, "Iterative learning control based automatic train operation with iteration-varying parameter," in *Proceedings of the 10th IEEE International Conference on Control and Automation (ICCA '13)*, pp. 51–56, Hangzhou, China, June 2013.

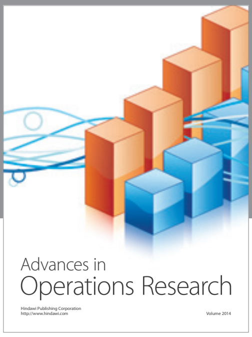
[22] K. S. Narendra and J. Balakrishnan, "Adaptive control using multiple models," *IEEE Transactions on Automatic Control*, vol. 42, no. 2, pp. 171–187, 1997.

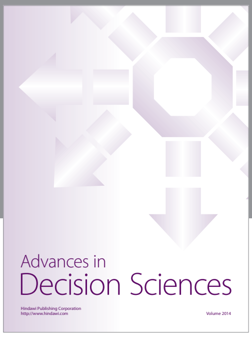
[23] X.-L. Li, C. Jia, D.-X. Liu, and D.-W. Ding, "Nonlinear adaptive control using multiple models and dynamic neural networks," *Neurocomputing*, vol. 136, pp. 190–200, 2014.

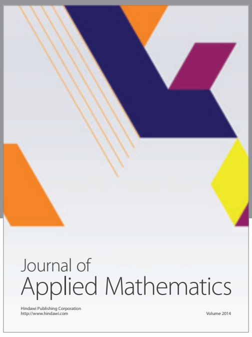
[24] Z. Bien, D.-H. Hwang, and S.-R. Oh, "Nonlinear iterative learning method for robot path control," *Robotica*, vol. 9, no. 4, pp. 387–392, 1991.

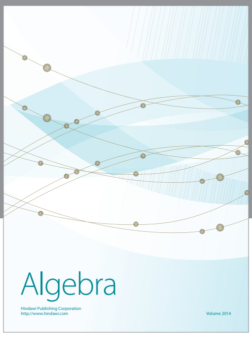
[25] M. X. Sun and B. J. Huang, *Iterative Learning Control*, National Defense Industry Press, Beijing, China, 1999.

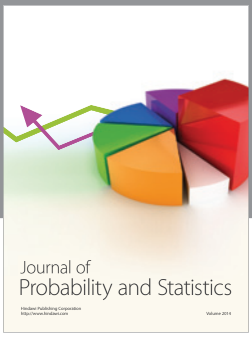
[26] S. Yin, S. X. Ding, X. C. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6418–6428, 2014.


Advances in
Operations Research

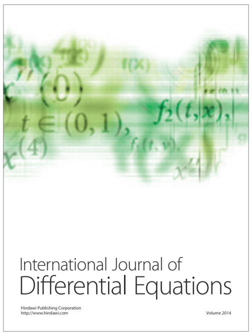

Advances in
Decision Sciences


Journal of
Applied Mathematics

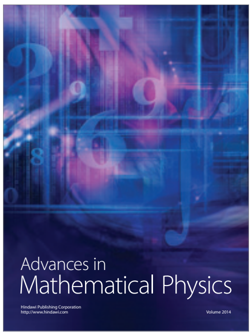

Algebra

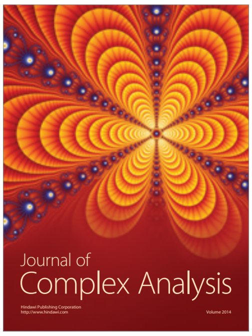

Journal of
Probability and Statistics

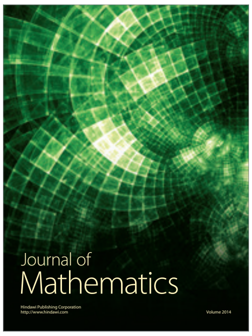
The Scientific
World Journal

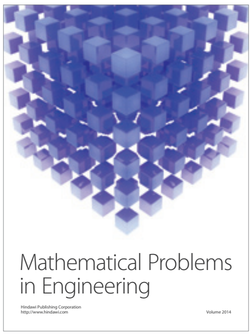

International Journal of
Differential Equations

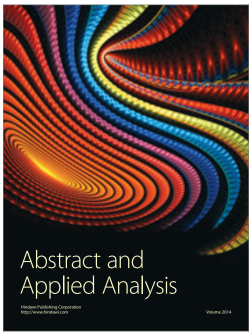
International Journal of
Combinatorics

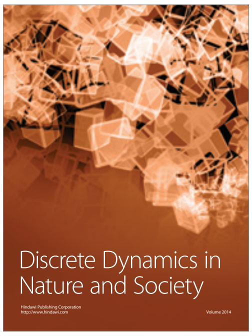

Advances in
Mathematical Physics

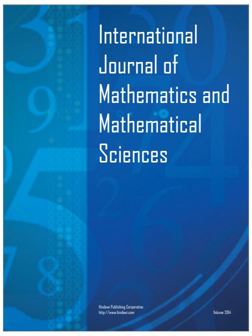

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

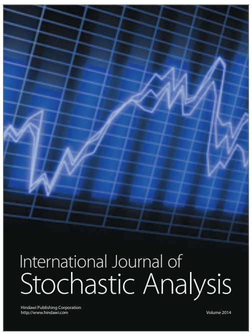

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization