Monotonic Convergent Iterative Learning Controller Design Based on Interval Model Conversion

Hyo-Sung Ahn, Kevin L. Moore, and YangQuan Chen

***Abstract*—This note presents a robust iterative learning controller design method for plants subject to interval model uncertainty in the A-matrix of their state–space description. First-order perturbation theory is used to find bounds on the eigenvalues and eigenvectors of the powers of A when A is an interval matrix. These bounds are then used for calculation of the interval uncertainty of the Markov matrix. The bounds on the Markov matrix are then used to design an iterative learning controller that ensures monotonic convergence for all systems in the interval plant.**

***Index Terms*—Interval conversion, iterative learning control, matrix perturbation, monotonic convergence.**

I. Introduction

Iterative learning control (ILC) is a control strategy for systems that execute the same trajectory, motion, or operation over and over. ILC exploits the repetitiveness inherent in such processes to improve their performance from trial to trial [1], [2]. Many existing ILC works have focused on optimality and analytical solution for a nominal plant under the assumption that there is no uncertainty in the system. In the existing literature, with respect to model uncertainties or noise considerations, one can find H_∞-based ILC [3], [4], frequency-domain ILC [5], [6], and stochastic-control-based ILC [7], [8] design techniques. However, most of these works focus on time-domain analysis and do not consider monotonic convergence. In this note, our concern is to relax the assumption that the plant has no uncertainty. In particular, we consider design of the learning gain matrix using the super-vector framework [9] to guarantee the monotone convergence of the output response on the iteration axis for linear plants in which the A-matrix of the plant's state space description is an interval matrix. The advantage of the super-vector notation is that the two-dimensional problem of ILC is changed into a one-dimensional multiple-input–multiple-output (MIMO) problem. Using such a framework, most discrete-time ILC problems can be expressed in the form $Y_k = HU_k$ where k is the iteration index, $Y_k, U_k \in R^n$, where n is the trial length, and H is a lower-triangular Toeplitz matrix. Note that in [9], a monotonically-convergent ILC algorithm has been algebraically designed without considering model uncertainty. The contribution of this note over [9] is to add model uncertainty in the nominal plant model. It is believed that the suggested algorithm is more effective than existing robust/frequency-domain ILC algorithms in the point of transient response and practical implementation, because, for example, H_∞ ILC and stochastic ILC do not guarantee monotonic convergence. The practical importance of monotonic convergence has been well-studied (for more details, refer to [10]). We also note that the ILC algorithm considered in this note includes causal/noncausal and Arimoto-like, band-limited,[1] time-varying learning gains. However, the approach here handles only linear uncertain ILC system and is not applicable to nonlinear systems such as given in [12].

The approach we present for robust ILC design is to use first-order perturbation theory to find bounds on the eigenvalues and eigenvectors of the powers of A when A is an interval matrix. These bounds are then used for calculation of the interval uncertainty of the Markov matrix. The bounds on the Markov matrix are then used to design an iterative learning controller that ensures monotonic convergence for all systems in the interval plant. The note is organized as follows. In Section II, we describe the notion of interval model conversion, i.e., converting from an interval uncertainty description in the state–space to an interval uncertainty description for the system Markov matrix. In Section III, matrix perturbation methods are used to find bounds on perturbed eigenpairs associated with powers of A, and in Section IV bounds on the Markov parameters are computed. ILC design and test results are given in Section V and in Section VI, respectively.

II. Interval Model Conversion in ILC

In the super-vector framework, Markov parameters are used for ILC design. Thus, to accomplish the robust ILC design that we seek to demonstrate in this note, the interval uncertainty of the nominal plant, either state space or transfer function, has to be converted into interval uncertainties in the Markov parameters. This process is called "interval model conversion." In this note, we focus on the nominal discrete system model given in the following single-input–single-output (SISO) state–space form:

$$x(t+1) = Ax(t) + Bu(t) \quad y(t) = Cx(t) \tag{1}$$

where A, B, and C are the matrices describing the system in the state space and $x(t)$, $u(t)$, and $y(t)$ are the state, input, and output variables, respectively. In particular, we call A the *nominal plant matrix*. Without loss of generality, the relative degree of the system is assumed to be 1. The Markov parameters of the plant are defined as $h_k = CA^{k-1}B$. These Markov parameters form the Markov matrix, a lower-triangular Toeplitz matrix whose first column is the vector $[h_1, h_2, \ldots, h_N]^T$, which is denoted as H. Then, the interval conversion process is to find bounds on h_k from the bounds on uncertain A, B, and C using $h_k = CA^{k-1}B$. To simplify our presentation, it is assumed that the model uncertainty exists in A only. It is possible to extend our approach to uncertain B and C matrices also. However, since the key issue in interval model conversion is to find the power of interval A matrix (as commented in [13], it is NP-hard to find exact boundaries of power of interval matrix system), this note only focuses on systems with an uncertain A-matrix.

Throughout this note, the underline $\underline{(\cdot)}$ represents the (element-wise) minimum value of $(\cdot)$, the overline $\overline{(\cdot)}$ represents the (element-wise) maximum value of $(\cdot)$, the symbol := represents set or algebraic definition, and the matrix A is represented by $A = [a_{ij}]$, where a_{ij} are elements of the matrix. The superscript I is used such as A^I to include the model uncertainty in A. Thus, *interval model conversion* can be explicitly defined as a process to find the uncertain boundaries of $h_k \in [\underline{h_k}, \overline{h_k}]$ from A^I, B and C.

[1]In this note, "Arimoto-like algorithm" means an ILC update rule with the standard learning form for a linear system with relative degree equal one: $u_{k+1}(t) = u_k(t) + \gamma e_k(t+1)$, where t is the discrete time index, k is the iteration index, and γ is the learning gain. By "band-limited" we mean that Γ is not fully populated; rather it has partially populated causal/noncausal bands. For a more detailed explanation, refer to [1] and [11].

Manuscript received December 8, 2004; revised June 1, 2005 and October 23, 2005. Recommended by Associate Editor F. Bullo.

H.-S. Ahn and Y. Chen are with the Center for Self-Organizing and Intelligent Systems (CSOIS), Department of Electrical and Computer Engineering, Utah State University, Logan, UT 84322-4160 USA (e-mail: hyosung@cc.usu.edu; yqchen@ece.usu.edu).

K. L. Moore is with the Division of Engineering, Colorado School of Mines, Golden, CO 80401 USA (e-mail: kmoore@mines.edu).

Digital Object Identifier 10.1109/TAC.2005.863498

To clarify our interval concepts, we introduce the following definitions.

Definition 2.1: The $n \times n$ interval matrix A^I is defined as $A^I := \left\{A = \left[a_{ij} : a_{ij} \in [\underline{a_{ij}}, \overline{a_{ij}}]\right]\right\}$, where $\underline{a_{ij}}$ and $\overline{a_{ij}}$ are the lower and upper bounds associated with the elements a_{ij} of the interval uncertain A matrix.

Definition 2.2: The nominal matrix A_o is defined to be $A_o = \left[a_{ij}^o : a_{ij}^o = (\underline{a_{ij}} + \overline{a_{ij}})/2\right]$.

Definition 2.3: The interval perturbation matrix ΔA^I is defined as

$$\Delta A^I := \left\{\Delta A = \left[\Delta a_{ij} : \Delta a_{ij} \in [\underline{a_{ij}} - a_{ij}^o, \overline{a_{ij}} - a_{ij}^o]\right]\right\}.$$

Note that we can also write $\Delta A^I := \left\{\Delta A : \Delta A = A - A_o \quad \forall\, A \in A^I\right\}$.

Definition 2.4: The vertex matrix set A^v is a subset of the interval matrix A^I that is defined as $A^v := \left\{A = \left[a_{ij} : a_{ij} \in \{\underline{a_{ij}}, \overline{a_{ij}}\}\right]\right\}$ and $(\Delta A)^v$ is the vertex matrix set of ΔA^I.

Note that in general the interval matrix A^I is a set with an infinite number of elements. On the other hand, the vertex set A^v contains a finite number of elements.

Next, to proceed we make two assumptions. The first is a technical assumption. The second is more practical, as will be described in a later remark.

Assumption 2.1: Every matrix $A \in A^I$ is diagonalizable and thus can be written as $A = X\Lambda X^{-1}$, with $\Lambda = \mathrm{diag}(\lambda_i)$, where λ_i are the eigenvalues of A.

Assumption 2.2: Every matrix $A \in A^I$ is Schur stable.

Now, based on the previous assumptions, we develop the following theorem to estimate the boundary of the powers of the interval matrix A^I.

Theorem 2.1: Let A^I be an $n \times n$ interval matrix and let Λ^I, X^I, Y^I be $n \times n$ interval matrices such that for each $A \in A^I$ there exist a diagonal matrix $\Lambda \in \Lambda^I$ and a matrix $X \in X^I$ such that $A = X\Lambda X^{-1}$ holds and $X^{-1} \in Y^I$. Then, for each $k \geq 1$ we have

$$\left\{A^k : A \in A^I\right\} \subseteq X^I \otimes (\Lambda^I)^k \otimes Y^I \tag{2}$$

where the operation $\otimes$ is the multiplication operator performed in interval arithmetic (refer to [14] and [15]) and $(\Lambda^I)^k = \underbrace{\Lambda^I \otimes \Lambda^I \otimes \cdots \otimes \Lambda^I}_{k \text{ times}}$.

Proof: The proof is straightforward since any $A \in A^I$ can be written, according to Assumption 2.1, as $A = X\Lambda X^{-1}$ for some $\Lambda \in \Lambda^I$ and $X \in X^I$ which additionally satisfies $X^{-1} \in Y^I$. Then $A^k = X\Lambda^k X^{-1} \in X^I \otimes (\Lambda^I)^k \otimes Y^I$ due to the basic "inclusion isotony" of interval arithmetic operations (see, e.g., [15]), which proves the inclusion (2). ∎

From now on, for convenience, we use the notation $(A^I)^k := \left\{A^k : A \in A^I\right\}$ and $A_k^I := X^I \otimes (\Lambda^I)^k \otimes Y^I$. Thus, Theorem 2.1 states that $(A^I)^k \subseteq A_k^I$.

Remark 2.1: From a computational perspective, the boundary of $(A^I)^k$ can be estimated by multiplying the interval matrix using interval calculation software such as Intlab [16], but the result will be quite conservative as k increases and it requires a huge amount of computation. Further, notice that the exact boundary of $(A^I)^k$ cannot be calculated mathematically or analytically. On the other hand, since $(A^I)^k \subseteq A_k^I$, we can estimate the boundary of $(A^I)^k$ by estimating the boundary of A_k^I, which is also easily done in Intlab, using the boundaries of the three interval matrices Λ^I, X^I, and Y^I, which is explained in Section IV.

Remark 2.2: Observe that if the maximum absolute eigenvalue of $A \in A^I$ is less than 1, for all $\mathcal{A}_k \in A_k^I$, $\mathcal{A}_k$ will converge to zero as $k \to \infty$ because Λ^k, for $\Lambda \in \Lambda^I$, converges to zero as $k \to \infty$. However, if the maximum absolute eigenvalue is bigger than 1, $\mathcal{A}_k \in A_k^I$ could diverge, and then our bound on the the uncertain interval boundary of h_k will become bigger as k increases. For this reason, we assumed A is Schur stable.

In this section, we showed that we can contain our original interval system inside a "bigger" interval system according to: $(A^I)^k \subseteq A_k^I$. Therefore, the remaining work is to estimate the boundaries of Λ^I, X^I, and Y^I and from these estimates to compute bounds on A_k^I which then bounds A^k. The Section III suggests an analytical method to estimate the bounds of Λ^I, X^I, and Y^I from A^I using first-order perturbation theory [17].

III. Interval Matrix Eigenpair Bounds

In this section, we briefly summarize first-order perturbation theory and then suggest two lemmas to obtain analytical solutions of the boundaries of X^I, Λ^I, and Y^I from A^I to be used for estimating the boundary of A_k^I. Let us suppose that the $n \times n$ nominal matrix $A_o = X_o\Lambda_o X_o^{-1} = X_o\Lambda_o Y_o$ has n different nominal eigenvalues λ_{0i}, a specific set of nominal left eigenvectors x_{0i}, and a specific set of nominal right eigenvectors y_{0i}, where $i = 1, \ldots, n$, satisfying $x_{0i}^T A_o = \lambda_{i0} x_{0i}^T$ and $A_o y_{0i} = \lambda_{i0} y_{0i}$. Further define the specific eigenvalue matrix, Λ_o, the specific nominal left eigenvector matrix, X_o, and the specific nominal right eigenvector matrix, Y_o by $\Lambda_o = \mathrm{diag}(\lambda_{o,i})$, $X_o = [x_{01}, x_{02}, \ldots, x_{0n}]^T$, and $Y_o = [y_{01}, y_{02}, \ldots, y_{0n}]$, with $Y_o^{-1} = X_o$. Next, assume a small perturbation in A_o such as $A = A_o + \Delta A$, where $A \in A^I$ and $\Delta A \in \Delta A^I$ based on Definition 2.1 and Definition 2.3. Denote the eigenvalues and a specific set of eigenvectors associated with the perturbation ΔA as: λ_{1i}, x_{1i}, and y_{1i}, respectively. In other words, when λ_i, x_i, and y_i represent eigenvalues and a specific set of eigenvectors of $A \in A^I$, the following relationships are satisfied: $\lambda_i = \lambda_{0i} + \lambda_{1i}$, $x_i = x_{0i} + x_{1i}$, and $y_i = y_{0i} + y_{1i}$.[2] Observe that the set of all possible λ_{1i} forms scalar interval variable, and the set of all possible x_{1i} and y_{1i} form vector intervals. In this section, we are interested in finding the boundaries of λ_{1i}, x_{1i}, and y_{1i}, from which we can estimate the boundaries of λ_i, x_i, and y_i.

From [17], the following formulae are adopted for the perturbed eigenvalues:

$$\lambda_{1i} = x_{0i}^T \Delta A y_{0i} \qquad \forall\, \Delta A \in \Delta A^I \tag{3}$$

and for the perturbed eigenvectors

$$x_{1i} = \sum_{k=1}^{n} \gamma_{ik} x_{0k} \quad y_{1i} = \sum_{k=1}^{n} \varepsilon_{ik} y_{0k} \tag{4}$$

where $\gamma_{ik} = (y_{0k}^T \Delta A x_{0i}/(\lambda_{0i} - \lambda_{0k}))$; $\varepsilon_{ik} = (x_{0k}^T \Delta A y_{0i}/(\lambda_{0i} - \lambda_{0k}))$, $i, k = 1, \ldots, n$ for $i \neq k$, and $\gamma_{ik} = \varepsilon_{ik} = 0$ for $i = k$, $\forall\, \Delta A \in \Delta A^I$. Notice that $\Delta A \in \Delta A^I$, where ΔA^I is an interval perturbation matrix, so it is quite messy to calculate λ_{1i}, x_{1i}, and y_{1i} in (3) and (4). To give an analytical calculation of (3) and (4), the maximum absolute values of the real and imaginary parts are considered separately. Let us denote the maximum of λ_{1i} on the real axis by $\lambda_{1i}|_{\mathrm{real}}^{\max}$, defined as $\lambda_{1i}|_{\mathrm{real}}^{\max} = \max\Big\{\lambda_{1i}|_{\mathrm{real}} : \lambda_{1i}|_{\mathrm{real}} = |\mathrm{Re}(\lambda_{1i})|,\ \lambda_{1i} =$

[2]Note that we do not imply that the eigenvalues of the sum of two matrices are equal to the sum of the individual matrices. Rather, the eigenpairs λ_{1i}, x_{1i}, and y_{1i} are "perturbation" eigenpairs and represent what would be added to the nominal values to obtain equivalent eignepairs of the perturbed matrix. Also note, in [17, p. 103], λ_{1i}, x_{1i}, and y_{1i} are called the first-order perturbation eigensolution.

$x_{0i}^T \Delta A y_{0i},\ \forall\ \Delta A\ \in\ \Delta A^I \Big\}$, and the maximum of λ_{1i} on the imaginary axis by $\lambda_{1i}|_{\mathrm{imag}}^{\max}$, defined as $\lambda_{1i}|_{\mathrm{imag}}^{\max} = \max\Big\{ \lambda_{1i}|_{\mathrm{imag}} : \lambda_{1i}|_{\mathrm{imag}} = |\mathrm{Im}(\lambda_{1i})|,\ \lambda_{1i} = x_{0i}^T \Delta A y_{0i},\ \forall\ \Delta A \in \Delta A^I \Big\}$, where Re means the real part and Im means the imaginary part. Then, we have the following lemma.

Lemma 3.1: For each $i = 1, \ldots, n$

$$\lambda_{1i}|_{\mathrm{real}}^{\max} = \max_{(\Delta A)^v} \{\mathrm{Re}[x_{0i}^T (\Delta A)^v y_{0i}]\};$$
$$\lambda_{1i}|_{\mathrm{imag}}^{\max} = \max_{(\Delta A)^v} \{\mathrm{Im}[x_{0i}^T (\Delta A)^v y_{0i}]\}. \tag{5}$$

The proof of Lemma 3.1 is very similar to that of Lemma 3.2. We show the proof of Lemma 3.2 instead of the proof of Lemma 3.1 because the former is a more comprehensive derivation.

The radii of the perturbed eigenvectors can also be estimated using the vertex matrix finite set. First, let us consider the left eigenvectors and let us denote the jth element of x_{1i} by $(x_{1i})_j$. Then, denoting the maximum of $(x_{1i})_j$ on the real axis by $(x_{1i})_j|_{\mathrm{real}}^{\max}$, defined as $(x_{1i})_j|_{\mathrm{real}}^{\max} = \max\Big\{ (x_{1i})_j|_{\mathrm{real}} : (x_{1i})_j|_{\mathrm{real}} = |\mathrm{Re}((x_{1i})_j)|,\ (x_{1i})_j = \sum_{k=1}^n (y_{0k}^T \Delta A x_{0i} / (\lambda_{0i} - \lambda_{0k}))(x_{0k})_j,\ \forall\ \Delta A \in \Delta A^I \Big\}$, and denoting the maximum of $(x_{1i})_j$ on the imaginary axis by $(x_{1i})_j|_{\mathrm{imag}}^{\max}$, defined as $(x_{1i})_j|_{\mathrm{imag}}^{\max} = \max\Big\{ (x_{1i})_j|_{\mathrm{imag}} : (x_{1i})_j|_{\mathrm{imag}} = |\mathrm{Re}((x_{1i})_j)|,\ (x_{1i})_j = \sum_{k=1}^n (y_{0k}^T \Delta A x_{0i} / (\lambda_{0i} - \lambda_{0k}))(x_{0k})_j,\ \forall\ \Delta A \in \Delta A^I \Big\}$, we provide the following lemma (the radii of the perturbed right-eigenvectors are calculated in the same way).

Lemma 3.2: For each i and j (i.e., at a fixed element of a fixed eigenvector), the maximum perturbations of the real and imaginary parts of $(x_{1i})_j$ are given by

$$(x_{1i})_j|_{\mathrm{real}}^{\max} = \max_{(\Delta A)^v} \left\{ \mathrm{Re}\left[\sum_{k=1}^n \frac{y_{0k}^T (\Delta A)^v x_{0i}}{\lambda_{0i} - \lambda_{0k}} (x_{0k})_j \right] \right\}$$
$$(x_{1i})_j|_{\mathrm{imag}}^{\max} = \max_{(\Delta A)^v} \left\{ \mathrm{Im}\left[\sum_{k=1}^n \frac{y_{0k}^T (\Delta A)^v x_{0i}}{\lambda_{0i} - \lambda_{0k}} (x_{0k})_j \right] \right\}$$

where $(x_{0k})_j$ is the jth element of the kth eigenvector x_{0k}.

Proof: From (4), we have

$$\begin{aligned} x_{1i} &= \gamma_{i1} x_{01} + \cdots + \gamma_{in} x_{0n} \\ &= \frac{y_{01}^T \Delta A x_{0i}}{\lambda_{0i} - \lambda_{01}} x_{01} + \cdots + \frac{y_{0n}^T \Delta A x_{0i}}{\lambda_{0i} - \lambda_{0n}} x_{0n}. \end{aligned} \tag{6}$$

In (6), since the denominators $\lambda_{0i} - \lambda_{01}, \ldots, \lambda_{0i} - \lambda_{0n}$ are nonzero scalars and $y_{01}, \ldots, y_{0n},\ x_{0i}, x_{01}, \ldots, x_{0n}$ are vectors, (6) can be rewritten as

$$x_{1i} = \xi^1 \Delta A \eta^1 x_{01} + \cdots + \xi^n \Delta A \eta^n x_{0n} \tag{7}$$

where the substitutions $(x_{0i}/(\lambda_{0i} - \lambda_{01})) = \eta^1, \ldots, (x_{0i}/(\lambda_{0i} - \lambda_{0n})) = \eta^n$, and $y_{01}^T = \xi^1, \ldots, y_{0n}^T = \xi^n$ are used. Observing that $\xi^1 \Delta A \eta^1, \ldots, \xi^n \Delta A \eta^n$ are scalars, we can write the jth element of the vector x_{1i} as

$$(x_{1i})_j = \xi^1 \Delta A \eta^1 (x_{01})_j + \cdots + \xi^n \Delta A \eta^n (x_{0n}) j. \tag{8}$$

Expanding ξ^i (a row vector) and η^i (a column vector) into their components, we rewrite (8) as

$$\begin{aligned} (x_{1i})_j &= \left\{ \sum_{k=1}^n \sum_{l=1}^n \left((\xi^1)_k (\Delta A)_{kl} (\eta^1)_l \right) \right\} (x_{01})_j + \cdots \\ &\quad + \left\{ \sum_{k=1}^n \sum_{l=1}^n ((\xi^n)_k (\Delta A)_{kl} (\eta^n)_l) \right\} (x_{0n})_j \\ &= \sum_{p=1}^n \left\{ \sum_{k=1}^n \sum_{l=1}^n ((\xi^p)_k (\Delta A)_{kl} (\eta^p)_l) \right\} (x_{0p})_j \\ &= \sum_{k=1}^n \sum_{l=1}^n \left\{ \sum_{p=1}^n ((\xi^p)_k (\eta^p)_l) (x_{0p})_j \right\} (\Delta A)_{kl}. \end{aligned}$$

Therefore, since $\sum_{p=1}^n ((\xi^p)_k (\eta^p)_l) (x_{0p})_j$ is a complex number, simply by denoting $\alpha_{kl} + \beta_{kl} i := \sum_{p=1}^n ((\xi^p)_k (\eta^p)_l) (x_{0p})_j$, we have

$$(x_{1i})_j = \sum_{k=1}^n \sum_{l=1}^n (\alpha_{kl} + \beta_{kl} i)(\Delta A)_{kl}. \tag{9}$$

Now, in order to find the maximum absolute magnitude of $(x_{1i})_j$, we separate the real part and the imaginary part. Let us first investigate the maximum absolute value of the real part. The real part is $\sum_{k=1}^n \sum_{l=1}^n \alpha_{kl} (\Delta A)_{kl}$, where $\alpha_{kl} \in \Re$ and $(\Delta A)_{kl}$ are scalar intervals. Observe that $\sum_{k=1}^n \sum_{l=1}^n \alpha_{kl} (\Delta A)_{kl}$ is a scalar interval, which ranges within $[\underline{\delta}, \overline{\delta}]$, where

$$\underline{\delta} = \min_{(\Delta A)_{kl} \in (\Delta A)_{kl}^I} \left\{ \sum_{k=1}^n \sum_{l=1}^n \alpha_{kl} (\Delta A)_{kl} \right\} \text{ and}$$
$$\overline{\delta} = \max_{(\Delta A)_{kl} \in (\Delta)_{kl}^I} \left\{ \sum_{k=1}^n \sum_{l=1}^n \alpha_{kl} (\Delta A)_{kl} \right\}$$

where scalar interval $(\Delta A)_{kl}^I := \{ (\Delta A)_{kl} : (\Delta A)_{kl} \in [\underline{a_{ij}} - a_{ij}^o, \overline{a_{ij}} - a_{ij}^o] \}$. Thus, $\underline{\delta}$ and $\overline{\delta}$ are defined as the minimum and maximum of a sum of scalar intervals. From the argument given in [18], it can be shown that $\underline{\delta}$ and $\overline{\delta}$ are computed vertex points of $(\Delta A)_{kl}^I$ depending on the sign of α_{kl}. In other words, if $\alpha_{kl} \geq 0$, then the minimum of δ occurs at $a_{ij}^o - \underline{a_{ij}}$; else if $\alpha_{kl} < 0$, then the minimum of δ occurs at $\overline{a_{ij}} - a_{ij}^o$. In the same way, $\overline{\delta}$ occurs at a vertex point of $(\Delta A)_{kl}$ depending on the sign of α_{kl}. If $\alpha_{kl} \geq 0$, then the maximum of δ occurs at $\overline{a_{ij}} - a_{ij}^o$; else if $\alpha_{kl} < 0$, then the maximum of δ occurs at $a_{ij}^o - \underline{a_{ij}}$. Next, the same procedure can be repeated in the imaginary part as the procedure performed in the real part. Thus, the maximum and minimum boundaries of eigenvectors can be checked by investigating the vertex matrices of the interval perturbation matrix. ■

Lemmas 3.1 and show how the maximum magnitude of the perturbed eigenvalues and eigenvectors can be calculated, respectively. Thus, since the perturbed eigenpairs are calculated by $\lambda_i = \lambda_{0i} + \lambda_{1i}$, $x_i = x_{0i} + x_{1i}$ and $y_i = y_{0i} + y_{1i}$, we have effectively computed the bounds on the interval matrices Λ^I, X^I, and Y^I.

IV. Markov Parameter Bounds

In Section II, the interval model conversion method was developed and in Section III, an analytical method for finding the maximum magnitudes of the perturbed eigenpairs was suggested. That is, Section II showed that the interval boundaries of A^k, where $A \in A^I$, can be bounded using the inequality: $(A^I)^k \subseteq A_k^I$, which provides the following relationship:

$$\underline{\mathcal{A}^k} \leq \underline{A^k} \leq A^k \leq \overline{A^k} \leq \overline{\mathcal{A}^k} \tag{10}$$

where $A^k \in (A^I)^k$ and $\mathcal{A}^k \in A_k^I$. Then, the interval boundaries of the interval plant's Markov parameters (i.e., $h_{k+1} = CA^kB$) can be estimated such as: $\underline{h_{k+1}} = \underline{CA^kB}$; $\overline{h_{k+1}} = \overline{CA^kB}$, where C, B are constant vectors describing the system (1) and A^k is a matrix which is lower-bounded and upper-bounded by $\underline{\mathcal{A}^k} \le A^k \le \overline{\mathcal{A}^k}$ from (10). Finally, these bounds, $\underline{\mathcal{A}^k}$ and $\overline{\mathcal{A}^k}$, can be computed using Lemmas 3.1 and 3.2 of Section III, which showed that the analytical solution for estimating the boundaries of the interval matrix A_k^I can be obtained using the vertex matrices of interval perturbation matrix ΔA^I.

V. Robust ILC Design

At this point, we turn our discussion to iterative learning control. Using the standard supervector notation for the system (1), we define the plant as $Y_k = H^I U_k$, where H^I is an interval Markov matrix that is a lower triangular Toeplitz matrix derived from the Markov parameters $h_k \in [\underline{h_k}, \overline{h_k}]$, with the bounds on h_k determined from the bounds on A^I as described in the previous section. We assume an Arimoto-like ILC law of the form $U_{k+1} = U_k + \Gamma E_k$, where Γ is the learning gain and $E_k = Y_d - Y_k$ is the error on iteration k relative to the desired trajectory Y_d. Before we continue, let us make the following definitions:

$$s_{ij}^1 := \overline{a_{ij}} \text{ if } i = j; \quad s_{ij}^1 := \max\{|\underline{a_{ij}}|, |\overline{a_{ij}}|\} \text{ if } i \ne j;$$
$$s_{ij}^2 := \underline{a_{ij}} \text{ if } i = j; \quad s_{ij}^2 := \min\{-|\underline{a_{ij}}|, -|\overline{a_{ij}}|\} \text{ if } i \ne j$$

where s_{ij}^1 is the ith row and jth column element of the matrix S^1; s_{ij}^2 is an element of matrix S^2, and $a_{ij}^I = [\underline{a_{ij}}, \overline{a_{ij}}]$ is an element of the general interval matrix A^I. The following lemma is adopted from the literature (for a proof, see [19, Th. 2]).

Lemma 5.1: For the interval matrix $A \in A^I$, if $\beta = \max\{\rho(S^1), \rho(S^2)\} < 1$, where ρ is the spectral radius, then the interval matrix A^I is Schur stable.

Monotonic convergence of the interval ILC system in the two-norm topology can be checked by Lemma 5.1 after small modifications. For the ILC update law we have given, the evolution of the error is given by the following error vector update law: $E_{k+1} = (I - H^I\Gamma)E_k$, where the singular value of $(I - H^I\Gamma) = T^I$ is calculated as

$$\overline{\sigma}\left(T^I\right) = \rho\left[(T^I)^T T^I\right] = \sqrt{\rho\begin{bmatrix} 0 & (T^I)^T \\ T^I & 0 \end{bmatrix}} = \sqrt{\rho\left(\mathbf{P}^I\right)}. \quad (11)$$

Then, the monotonic convergence of the interval ILC system is checked by analyzing the Schur stability of $\mathbf{P}^I$ in the two-norm topology using Lemma 5.1. Additionally, monotonic convergence conditions in the 1 and ∞ norm topologies can be adopted from [18] as follows.

Lemma 5.2: Given $h_i \in [\underline{h_i}, \overline{h_i}]$, the interval ILC system is monotonically convergent if $\|I - H^v\Gamma\|_k < 1$, where k is 1 or ∞; and H^v are vertex Markov matrices associated the interval Markov matrix H^I.

The main advantage of Lemma 5.2 is that the monotone convergence property of the interval ILC system is checked just using vertex Markov matrices of H^I without using vertex matrices of T^I. This characteristic enables us to save computation time and to reduce conservativeness. For interval ILC design, we consider both asymptotical stability and monotone convergence. For achieving asymptotic stability, if Arimoto-like gains are selected such that $|1 - \gamma\underline{h_1}| < 1$ and $|1 - \gamma\overline{h_1}| < 1$ [18], the asymptotic stability of the interval ILC system is achieved. To increase robustness, the following scheme is recommended:

$$\gamma = \begin{cases} \frac{1}{\overline{h_1}} & \text{if } h_1 \ge 0 \text{ for all } h_1 \in h_1^I \\ \frac{1}{\underline{h_1}}, & \text{if } h_1 < 0, \text{ for all } h_1 \in h_1^I. \end{cases} \quad (12)$$

The reason for (12) is that the asymptotic stability condition $|1 - \gamma\overline{h_1}|$ is guaranteed with $\gamma = 1/\overline{h_1}$ if $h_1 \ge 0$ for all $h_1 \in h_1^I$ and with $\gamma = 1/\underline{h_1}$ if $h_1 < 0$ for all $h_1 \in h_1^I$. For the monotonically convergent ILC gain design in the two-norm topology, we use (11). For the learning gain matrix design, the following optimization is suggested based on Lemma 5.1.

Suggestion 5.1: Let $\mathbf{p}_{ij}^I$ be the ith row and jth column element of $\mathbf{P}^I$. If we define a matrix $\mathbf{M}$ whose elements are given as: $\mathbf{m}_{ij} = \max\{|\underline{\mathbf{p}_{ij}}|, |\overline{\mathbf{p}_{ij}}|\}$, then we solve the following optimization problem to design Γ:

$$\min_{\Gamma} \rho(\mathbf{M}) \text{ s.t. } h_k \in [\underline{h_k}, \overline{h_k}]. \quad (13)$$

Remark 5.1: To explain why the matrix $\mathbf{M}$ is used in (13), define $\mathbf{s}_{ij}^1$ ($\mathbf{S}^1 = [\mathbf{s}_{ij}^1]$) and $\mathbf{s}_{ij}^2$ ($\mathbf{S}^2 = [\mathbf{s}_{ij}^2]$) as

$$\mathbf{s}_{ij}^1 := \overline{\mathbf{p}_{ij}} \text{ if, } i = j; \quad \mathbf{s}_{ij}^1 := \max\{|\underline{\mathbf{p}_{ij}}|, |\overline{\mathbf{p}_{ij}}|\} \text{ if } i \ne j$$
$$\mathbf{s}_{ij}^2 := \underline{\mathbf{p}_{ij}} \text{ if, } i = j; \quad \mathbf{s}_{ij}^2 := \min\{-|\underline{\mathbf{p}_{ij}}|, -|\overline{\mathbf{p}_{ij}}|\} \text{ if } i \ne j.$$

Then, since the matrix $\mathbf{P}^I$ is symmetric, $\mathbf{S}^1 = -\mathbf{S}^2$. Hence, using the fact that $\rho(\mathbf{S}^1) = \rho(\mathbf{S}^2)$ and the diagonal terms of $\mathbf{P}^I$ are all zeros, we only need to check the spectral radius of the matrix composed of the off-diagonal terms of $\mathbf{S}^1$.

If Lemma 5.2 is used, another optimization scheme can be suggested without constraints.

Suggestion 5.2: If k is 1 or ∞, the following optimization is straightforward:

$$\min_{\Gamma} \|I - H^v\Gamma\|_k. \quad (14)$$

Several remarks follow.

Remark 5.2: The optimization is to minimize $\|I - H^v\Gamma\|_k$ using Γ, where Γ is a band-fixed learning gain matrix. There is a tradeoff. In a small band size, it is possible that there could not exist an optimization solution such that $\|I - H^v\Gamma\|_k < 1$. In this case, the band size should be increased until the optimization algorithm finds Γ such that $\|I - H^v\Gamma\|_k < 1$. However, as band size increases, more causal and noncausal learning gains are required. This is practically undesirable because we need to store more data into memory for the current control update. The optimization in Suggestion 5.1 is a nonlinear constraint minimization problem and the optimization in Suggestion 5.2 is a nonlinear nonconstrained minimization problem. These problems can be easily solved using the Matlab optimization toolbox.

Remark 5.3: Depending on the interval ILC system, the optimization scheme suggested above may not find the optimization solution even with the fully populated learning gain matrix. In this case, the following control update law could be used: $U_{k+1} = Q(U_k + \Gamma E_k)$, where Q is a time-invariant diagonal matrix. Then, since the error vector is updated by the following formula: $E_{k+1} = Q(I - H\Gamma)E_k + (I - Q)Y_d$, it is easy to make $\|Q(I - H\Gamma)\| < 1$ by Q and Γ. However, this approach will result in a nonzero steady-state error. This is a tradeoff.

VI. Simulation Illustration

For demonstration purposes, let us use the following simple discrete servo system model, whose nominal plant was identified from the Quanser SRV02 system[3]:

$$x_1(k+1) = a_{11}x_1(k) + a_{12}x_2(k) + 2u(k);$$
$$x_2(k+1) = a_{21}x_1(k) + a_{22}x_2(k) + 0.5u(k)$$
$$y(k) = x_1(k) \quad (15)$$

where interval parameters are bounded as: $-0.74 \le a_{11} \le -0.66$, $-0.53 \le a_{12} \le -0.47$, $0.95 \le a_{21} \le 1.05$, and $0.19 \le a_{22} \le 0.21$,

[3]http://www.quanser.com/english/html/challenges/fs_chall_rotary_flash.htm

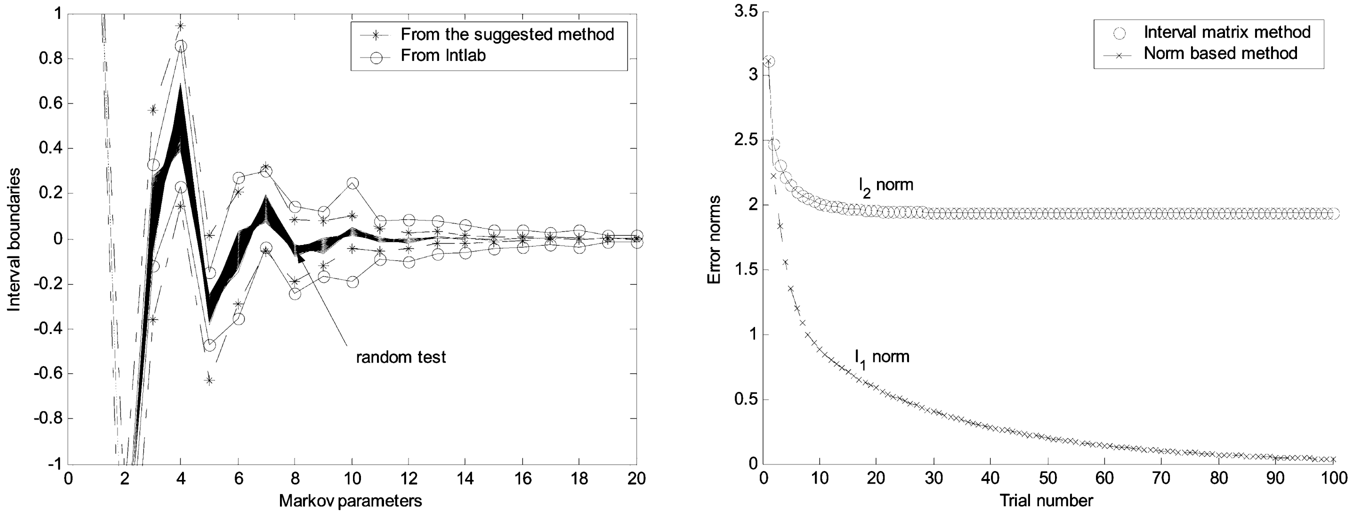

Fig. 1. Left: Calculated interval uncertain boundaries of Markov parameters. Right: ILC convergence test.

and u is the control force. Using the results of foregoing sections, a simulation test is performed with the following reference sinusoidal signal: $Y_d = \sin(8.0j/n)$, where $n = 20$ and $j = 1, \ldots, n$. The band size is fixed to be 3. The left plot of Fig. 1 shows the calculated interval boundaries of the Markov parameters using the computations given in Sections III and IV. The circle-marked line represents the maximum/minimum boundaries of the Markov parameters calculated from the Intlab software [16]; and the $*$-marked line represents the Markov parameter boundaries calculated from the method suggested in this note. For verification of the suggested method, a Monte-Carlo type random test was also performed as shown in figure (identified by "random test" in the figure). Observe that the suggested method gives reliable bounds of the interval ranges of the Markov parameters. We see that our suggested method gives less conservative bounds than Intlab after h_6; but, from h_1 to h_5, Intlab is slightly less conservative. However, we note that we have found in other experiments that in the case of a marginally stable system our method is less conservative than Intlab for all Markov parameters. So the suggested method is particularly suitable for monotonic convergent ILC design. We should further stress that our technique gives an analytical computation of the bounds for all h_k, unlike the bounds found from Intlab. To see the convergence of the ILC system for a single plant in the interval system, refer to the right-hand plot of Fig. 1. The circle-marked line (called the "interval matrix method") is the maximum l_2 norm error of the ILC system, whose learning gain matrix was designed by (13); and the $\times$-marked line (called the "norm-based method") is the maximum l_1 norm error when the learning gain matrix was designed by (14). In the case of the interval matrix method, there is a steady state error, because we fixed $Q = 0.9$ to guarantee the monotone convergence as commented in Remark 5.3) (when $Q = 1$, the optimization did not find the optimal solution such that the norm is less than 1).

VII. Concluding Remarks

In this note, a robust iterative learning controller was designed with consideration of interval uncertainty in the plants A-matrix. The interval uncertainty of the system was converted into the super-vector iterative learning control system (i.e., into an interval Markov matrix) using first-order perturbation theory. Optimization schemes were also suggested based on an interval matrix stability analysis method and a norm-based method. In the case of the norm based method, the ILC learning gain matrix guaranteed the monotonic convergence of the uncertain ILC system with zero steady state error. However, the interval matrix method only guaranteed the monotonic convergence with nonzero steady error. From these results, we conclude that the norm-based method is less conservative than the interval matrix method. However, the norm-based method requires much more computational time than the interval matrix method. Future work will consider ILC gain matrix design for optimizing convergence speed for interval plants.

Acknowledgment

The authors would like to thank the four anonymous reviewers for their constructive comments which improved the presentation of this note.

References

[1] K. L. Moore, "Iterative learning control for deterministic systems," in *Advances in Industrial Control*. New York: Springer-Verlag, 1993.
[2] Y. Chen and C. Wen, *Iterative Learning Control: Convergence, Robustness and Applications*. London, U.K.: Springer-Verlag, 1999, vol. LNCIS-248, Lecture Notes series on Control and Information Science.
[3] W. Paszke, K. Galkowski, E. Rogers, and D. H. Owens, "H_∞ control of discrete linear repetitive processes," in *Proc. 42nd IEEE Conf. Decision and Control*, Maui, HI, Dec. 2003, pp. 628–633.
[4] K. L. Moore, Y. Chen, and H.-S. Ahn, "Algebraic H_∞ design of higher-order iterative learning controllers," in *Proc. 2005 IEEE Int. Symp. Intelligent Control*, Limassol, Cyprus, 2005, pp. 1207–1212.
[5] D. de Roover and O. H. Bosgra, "Synthesis of robust multivariable iteraive learning controller with application to a wafer stage motion system," *Int. J. Control*, vol. 73, no. 10, pp. 968–979, 2000.
[6] M. Norrlöf, "Iterative learing control: Analysis, design, and experiments," Ph.D. dissertation, Linköping University, Linköping, Sweden, 2000.
[7] S. S. Saab, "A discrete-time stochastic learning control algorithm," *IEEE Trans. Autom. Control*, vol. 46, no. 6, pp. 877–887, Jun. 2001.
[8] H. F. Chen and H. T. Fang, "Output tracking for nonlinear stochastic systems by iterative learning control," *IEEE Trans. Autom. Control*, vol. 49, no. 4, pp. 583–588, Apr. 2004.
[9] K. L. Moore, Y. Chen, and V. Bahl, "Monotonically convergent iterative learning control for linear discrete-time systems," *Automatica*, vol. 41, no. 9, pp. 1529–1538, 2005.
[10] R. W. Longman, "Iterative learning control and repetitive control for engineering practice," *Int. J. Control*, vol. 73, no. 10, pp. 930–954, 2000.
[11] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *J. Robot. Syst.*, vol. 1, no. 2, pp. 123–140, 1984.
[12] J. X. Xu and J. Xu, "On iterative learning from different tracking tasks in the presence of time-varying uncertainties," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 1, pp. 589–597, Jan. 2004.

[13] O. Kosheleva, V. Kreinovich, G. Mayer, and H. T. Nguyen, "Computing the cube of an interval matrix is NP-hard," Comp. Sci. Dept. Univ. Texas El Paso, El Paso, TX, Tech. Rep. UTEP-CS-04-28, 2004.

[14] R. E. Moore, *Interval Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1966.

[15] G. Alefeld and J. Herzberger, *Introduction to Interval Computations*. New York: Academic, 1983.

[16] G. I. Hargreaves, "Interval analysis in MATLAB," Univ. Manchester, Manchester, U.K., Numerical Analysis Rep. 416, 2002.

[17] L. Meirovitch, *Computational Methods in Structural Dynamics*. Rockville, MD: Sijthoff and Noordhoff, 1980.

[18] H.-S. Ahn, K. L. Moore, and Y. Chen, "Stability analysis of iterative learning control system with interval uncertainty," in *Proc. 16th IFAC World Congr.*, Prague, Czech Republic, Jul. 4–8, 2005, pp. 1–6.

[19] J. J. D. Delgado-Romero, R. S. Gonzalez-Garza, J. A. Rojas-Estrada, G. Acosta-Villarreal, and F. Delgado-Romero, "Some very simple Hurwitz and Schur stability tests for interval matrices," in *Proc. 30th Conf. Decision and Control*, Kobe, Japan, Dec. 1996, pp. 2980–2981.

Parameter Identifiability of Nonlinear Systems With Time-Delay

J. Zhang, X. Xia, and C. H. Moog

***Abstract*—In this note, various parameter identifiability concepts for nonlinear systems with time-delay are defined, complete characterizations of these concepts as well as easily checkable criteria are provided. It is proved that geometric identifiability is equivalent to identifiability with known initial conditions, algebraic identifiability implies geometric identifiability. As for identifiability with partially known initial conditions, an easy characterization is also provided.**

***Index Terms*—Identifiability, linear algebraic approach, nonlinear systems, time-delay.**

I. Introduction

In this note, we consider the parameter identifiability problem of a nonlinear system with time-delay

$$\Sigma_\theta : \begin{cases} \dot{x} &= f(x(t-i), \theta, u(t-j) : i, j \in S_-) \\ y &= h(x(t-i), \theta, u(t-j) : i, j \in S_-) \\ x(t) &= x_0(t) \qquad \forall t \in [-s, 0] \end{cases} \tag{1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^p$, and $\theta \in R^q$ is the parameter. This problem grows out of the same problem for systems without time-delay (see [9]–[11] for historical account and some recent results), and it has major applications for time-delay systems, for example, the identification of the death rate in the SIS epidemic model with maturation delay (see [2]) as displayed in Section IV.

The study of the identifiability of control systems with time-delay has been scarce, and limited to linear systems with time-delay. In [8], [7], [1], aspects of identifiability of linear time-delay system such as system parameters, transfer function coefficients as well as time-delays are brought forward. Our results in this note are only concerned about parameter identifiability of nonlinear systems with time-delay. As discussions in [7] and [1] reveal, the parameter identifiability of a linear time-delay system is itself an intricate problem. For nonlinear time-delay systems, the problem remains largely open and difficult. On the other hand, a generic version of the problem lends itself to complete characterizations in light of the algebraic framework developed in [12] and the rigorous approach taken in [11] for nonlinear systems without time-delay.

In this note, various parameter identifiability concepts for nonlinear systems with time-delay are defined, complete characterizations of these concepts and easily checkable criteria for all of them are obtained.

Specifically, the results of [11] are generalized to nonlinear systems with time-delay. Similar results as in [11] are established. That is, we prove that geometric identifiability is equivalent to identifiability with known initial conditions, algebraic identifiability implies geometric identifiability, and some easy criteria for the two kinds of identifiability. As for identifiability with partially known conditions, an easy characterization is also provided.

It is worthy to note that there are fundamental differences between the systems with and without time-delay. For example, Theorems 2 and 3 are not direct generalizations of the corresponding results in [11], since extra operations depending the delay operator have to be done.

The note is organized as follows. In Section II, we give some definitions. The main results are established in Section III. Section IV is devoted to examples. The last section offers some concluding remarks.

II. Definitions

To make things more precise, assume that in the system (1), the functions f and h are meromorphic functions which are defined as the quotients of convergent power series with real coefficients. The integer s is nonnegative, and the set $S_- := \{0, 1, \ldots, s\}$ is a finite set of constant time delays, and

$$\begin{aligned} &f(x(t-i), \theta, u(t-j), i, j \in S_-) \\ &:= f(x(t), x(t-1), \ldots, x(t-s), \theta, u(t), u(t-1), \ldots, u(t-s)). \end{aligned}$$

The function x_0 denotes a continuous function of initial condition. Assume that $\operatorname{rank}(\partial h/\partial x) = p$, that is, for any fixed θ, and u in some open sets, the p components of h are independent functions of x and its shifts. The variable θ is the parameter to be identified and it is assumed to belong to $\mathcal{P}$ which is an open subset of $\mathbb{R}^q$. Moreover, without loss of generality, x_0 is assumed to be independent of θ and u. Denote by $\mathcal{M} := C[-s, 0]$ the set of initial functions on $[-s, 0]$.

For any open subset $U \subseteq \mathbb{R}^m$, an admissible input function $u(t) : [-s, T] \to U$ is defined to be an input on $[-s, T]$ such that the differential equation in (1) admits a unique (local) solution. For any initial function x_0 and an admissible input $u(t)$ on $[-s, T]$, there exists a parameterized solution $x(t, \theta, x_0, u)$ on some interval $[-s, \bar{T}]$, $\bar{T} \leq T$. Denote the corresponding output by $y(t, \theta, x_0, u)$. The following definitions are generalizations of the corresponding ones in [11].

Definition 1: The system Σ_θ is said to be x_0-identifiable at θ through an admissible input u (on $[-s, T]$) if there exists an open set $\mathcal{P}^0 \subset \mathcal{P}$ containing θ such that for any two distinct $\theta_1, \theta_2 \in \mathcal{P}^0$, the solutions $x(t, \theta_1, x_0, u)$ and $x(t, \theta_2, x_0, u)$ exist on $[-s, \epsilon]$, $0 < \epsilon \leq T$, and their corresponding outputs satisfy $y(t, \theta_1, x_0, u) \neq y(t, \theta_2, x_0, u)$ on $t \in [-s, \epsilon]$.

Now, consider the generic property of identifiability. The same topology for the input function spaces as in [11] is used in this note,

Manuscript received December 23, 2004; revised June 23, 2005 and October 7, 2005. Recommended by Associate Editor L. Xie.

J. Zhang and C. H. Moog are with IRCCyN, Institut de Recherche en Communications et Cybernétique de Nantes, 44321 Nantes Cedex 3, France (e-mail: Claude.Moog@irccyn.ec-nantes.fr).

X. Xia is with the Department of Electrical, Electronic, and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa (e-mail: xxia@postino.up.ac.za).

Corresponding author: X. Xia

Digital Object Identifier 10.1109/TAC.2005.863497