Robust Monotonic Convergent Iterative Learning Control Design: an LMI-based Method

Lanlan Su

Abstract

This work investigates robust monotonic convergent iterative learning control (ILC) for uncertain linear systems in both time and frequency domains, and the ILC algorithm optimizing the convergence speed in terms of $l_2$ norm of error signals is derived. Firstly, it is shown that the robust monotonic convergence of the ILC system can be established equivalently by the positive definiteness of a matrix polynomial over some set. Then, a necessary and sufficient condition in the form of sum of squares (SOS) for the positive definiteness is proposed, which is amendable to the feasibility of linear matrix inequalities (LMIs). Based on such a condition, the optimal ILC algorithm that maximizes the convergence speed is obtained by solving a set of convex optimization problems. Moreover, the order of the learning function can be chosen arbitrarily so that the designers have the flexibility to decide the complexity of the learning algorithm.

Index Terms

Iterative Learning Control, Robust Monotonic Convergence, Convex Optimization, LMI.

I. INTRODUCTION

Iterative learning control (ILC) is a useful control strategy to improve tracking performance over repetitive trials ([1], [2]). The basic idea is to incorporate the error signals from previous iterations into updating the control signal for subsequent iterations. Stability is naturally the most fundamental topic in ILC which is widely studied in both the time and frequency domain (see [3] and the references therein for more details). However, in some stable ILC systems the error can grow very large before converging to the desired output trajectory, which is undesirable in most practical applications [4]. Hence, the objective of monotone convergence is of crucial importance to improve tracking performance from trial to trial (see, for instance [5], [6]).

A relatively new research line in the existing literature regarding ILC is robust monotone convergence in the presence of uncertain models. Several approaches have been proposed by researches for analysis and synthesis of robust ILC. See, e.g., [7] which proposes the inverse model-based iterative learning control, [8] which designs robust monotonic convergent ILC controller based on interval model conversion methods, [9] wherein the gradient-based robust monotone ILC algorithm is provided, and the norm-optimal design with a quadratic cost function is considered in [10], [11]. Since the $H_\infty$ method offers a common approach to robust feedback controller design, there exists a trend of applying it to robust ILC control. For instance, [12] studies the problem of robust convergent ILC by solving $\mu$-synthesis problem; [13] presents sufficient conditions for robust monotonic convergence analysis based on $\mu$ analysis; [14] provides $H_\infty$-based design method to synthesize high-order ILCs. It is generally known that $H_\infty$-based robust controller computation will inevitably encounter the notoriously difficulty caused by $\mu$-synthesis, and consequently only sub-optimal controllers can be derived.

This work investigates robust monotonic convergent ILC for uncertain linear systems in both time and frequency domains, and the optimal ILC algorithm which optimizes the convergence speed in terms of $l_2$ norm of error signals is derived. Specifically, we first show that the robust monotonic convergence of the ILC system can be reformulated as positive definiteness of a matrix polynomial over some set. Then we provide a necessary and sufficient condition for the positive definiteness of the matrix polynomial over the
For a matrix polynomial in two sets of variables $P(x,y)$, we use $\text{deg}(P(x,y))$ to denote the maximum degree of the entries of $P$ in $x$ (respectively, $y$). The acronym SOS stands for sum of squares of matrix polynomials.

\section{Time-domain results with uncertain plant}

In this section, we focus on robust ILC design with finite-time interval per trial denoted as $N$. Consider the uncertain plant

$$P(q,\lambda) = p_1(\lambda)q^{-1} + p_2(\lambda)q^{-2} + \cdots + p_N(\lambda)q^{-N}, \forall \lambda \in \Lambda$$

(1)

where the Markov parameters $p_1,\ldots,p_N$ depend rationally on the uncertainty vector $\lambda \in \Lambda$ with $p_1 \neq 0, \forall \lambda \in \Lambda$, and $q$ is the forward time-shift operator. The Markov parameters can be obtained from uncertain linear time invariant systems represented by state-space equations or transfer functions.

We suppose that $\lambda \in \mathbb{R}^n$ is constrained into a simplex $\Lambda$, which is defined as follows

$$\Lambda = \left\{ \lambda \in \mathbb{R}^n : \sum_{i=1}^n \lambda_i = 1, \lambda_i \geq 0, \forall i = 1,\ldots,n \right\}.$$ 

Remark 1. We note that the formulation of the uncertain plant (1) can be equivalently reformulated into the case where the uncertainty vector is constrained into a generic bounded convex polytope. In fact, consider the plant described as

$$P(q,\theta) = p_1(\theta)q^{-1} + p_2(\theta)q^{-2} + \cdots + p_N(\theta)q^{-N}, \forall \theta \in \Theta$$

(2)

where $p_1,\ldots,p_N$ are rational function in the uncertainty vector $\theta$, and $\Theta$ belongs to a bounded convex polytope $\Theta$. Then the vector $\theta$ in $\Theta$ can be replaced by a linear function $l(\lambda)$ over $\Lambda$ by the change of variable $\theta = l(\lambda)$. Hence, by substituting $\theta$ with $l(\lambda)$, (2) can be rewritten as (1). It should be mentioned that the formulation (1) has also included the uncertain interval plant model represented by

$$P(q) = p_1q^{-1} + p_2q^{-2} + \cdots + p_Nq^{-N}$$

(3)

$\frac{1}{\lambda} \in [\underline{P}_i, \overline{P}_i], \ i = 1,\ldots,N$.
as a special case, as the uncertain coefficients \( p_1, p_2, \ldots, p_N \) are constrained into a hyper-rectangle which is a bounded convex polytope.

The lifted system representation is given by

\[
\begin{pmatrix}
  y_j(1) \\
  y_j(2) \\
  \vdots \\
  y_j(N)
\end{pmatrix} =
\begin{pmatrix}
  p_1(\lambda) & 0 & \cdots & 0 \\
  p_2(\lambda) & p_1(\lambda) & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  p_N(\lambda) & p_{N-1}(\lambda) & \cdots & p_1(\lambda)
\end{pmatrix}
\begin{pmatrix}
  u_j(0) \\
  u_j(1) \\
  \vdots \\
  u_j(N-1)
\end{pmatrix} +
\begin{pmatrix}
  d(1) \\
  d(2) \\
  \vdots \\
  d(N)
\end{pmatrix}
\]

\( (3) \)

where \( y_j, u_j \) denote the \( N \)-dimensional vectors containing output and input signals during the \( j \)-th trial, respectively, and \( d \) is an exogenous signal which repeats at each trial.

We denote by \( y_d \in \mathbb{R}^N \) the desired output trajectory, and the output error of the \( j \)-th trial, \( e_j \), is defined as

\[
\begin{pmatrix}
  e_j(1) \\
  e_j(2) \\
  \vdots \\
  e_j(N)
\end{pmatrix} =
\begin{pmatrix}
  y_d(1) \\
  y_d(2) \\
  \vdots \\
  y_d(N)
\end{pmatrix} -
\begin{pmatrix}
  y_j(1) \\
  y_j(2) \\
  \vdots \\
  y_j(N)
\end{pmatrix}.
\]

\( (4) \)

We adopt the widely used ILC learning algorithm as follows \[4, 15\]:

\[
u_{j+1}(k) = Q(q) [u_j(k) + L(q)e_j(k+1)]
\]

\( (5) \)

wherein the Q-filter \( Q(q) \) and learning function \( L(q) \) are allowed to be non-causal, described respectively by

\[
Q(q) = q_{-(N-1)}q^{-(N-1)} + \cdots + q_{-1}q^1 + q_0 + q_1q^{-1} + \cdots + q_{N-1}q^{N-1}
\]

\( (6) \)

and

\[
L(q) = l_{-(N-1)}q^{-1} + \cdots + l_{-1}q + l_0 + l_1q^{-1} + \cdots + l_{N-1}q^{N-1}
\]

\( (7) \)

For brevity, denote the vector \( (l_{-(N-1)}, \ldots, l_{-1}, l_0, l_1, \ldots, l_{N-1})^T \) as \( l \).

While most existing works do not include the Q-filter, i.e., \( Q = I \), for ensuring perfect tracking, the incorporation of the Q-filter enables us to improve transient learning behavior and robustness (See \[4\] and the references therein). Without loss of generality, we include Q-filter in this work.

Equations \((5)-(7)\) can be rewritten into the lifted form:

\[
\begin{pmatrix}
  u_{j+1}(0) \\
  u_{j+2}(1) \\
  \vdots \\
  u_{j+1}(N-1)
\end{pmatrix} =
\begin{pmatrix}
  q_0 & q_{-1} & \cdots & q_{-(N-1)} \\
  q_1 & q_0 & \cdots & q_{-(N-2)} \\
  \vdots & \vdots & \ddots & \vdots \\
  q_{N-1} & q_{N-2} & \cdots & q_0
\end{pmatrix}
\begin{pmatrix}
  u_j(0) \\
  u_j(1) \\
  \vdots \\
  u_j(N-1)
\end{pmatrix} +
\begin{pmatrix}
  e_j(1) \\
  e_j(2) \\
  \vdots \\
  e_j(N)
\end{pmatrix}
\]

\( (8) \)

In this section we aim at designing the learning algorithm \((8)\) to ensure the robust monotonic convergence defined as follows.
Definition 2. (Robust Monotonic Convergence) The uncertain system (3) with the ILC algorithm (8) is robustly monotonically convergent if there exists $0 \leq \gamma < 1$ such that
\[
\|e_\infty - e_{j+1}\|_2 < \gamma \|e_\infty - e_j\|_2, \forall j \in \mathbb{N}, \forall \lambda \in \Lambda
\]
where $\gamma$ is called the convergence rate.

From the ILC system dynamics (3), (8) and the error (4), it can be derived by some algebraic manipulations that
\[
e_\infty - e_{j+1} = P(\lambda)Q(I - LP(\lambda))P^{-1}(\lambda)(e_\infty - e_j).
\]
Hence, the uncertain system (3) with the ILC algorithm (8) is robustly monotonically convergent if and only if
\[
\gamma \triangleq \max_{\lambda \in \Lambda} \sigma \left( P(\lambda)Q(I - LP(\lambda))P^{-1}(\lambda) \right) < 1.
\]  
(9)

Problem 3. Given a fixed $Q$, find the learning matrix $L$ such that the convergent rate $\gamma$ defined in (9) is minimized, i.e.,
\[
\min \sup_{\lambda \in \Lambda} \sigma \left( P(\lambda)Q(I - LP(\lambda))P^{-1}(\lambda) \right).
\]  
(10)

Let us observe that the optimization problem in (10) can be transformed into minimizing $\gamma$ over $L$ subject to
\[
\gamma^2 I - (P(\lambda)Q(I - LP(\lambda))P^{-1}(\lambda))^T (P(\lambda)Q(I - LP(\lambda))P^{-1}(\lambda)) > 0, \forall \lambda \in \Lambda.
\]  
(11)

Now let us express
\[
P(\lambda)Q(I - LP(\lambda))P^{-1}(\lambda) = \frac{W(L, \lambda)}{a(\lambda)}
\]
where $a(\lambda) = \det(P(\lambda))$ and $W(L, \lambda) = P(\lambda)Q(I - LP(\lambda))\text{adj}(P(\lambda))$. Then, the constraint (11) transforms to
\[
\gamma^2 a^2(\lambda) I - WT(L, \lambda)W(L, \lambda) > 0, \forall \lambda \in \Lambda
\]
which can be rewritten into
\[
\begin{pmatrix}
\gamma^2 a^2(\lambda) I & WT(L, \lambda) \\
W(L, \lambda) & I
\end{pmatrix} > 0, \forall \lambda \in \Lambda.
\]  
(12)

Before continuing, we denote the operator that returns a matrix homogeneous polynomial $\bar{G}(\lambda)$ from a matrix polynomial $G(\lambda)$ in the variable $\lambda \in \Lambda$ while satisfying $\bar{G}(\lambda) = G(\lambda), \forall \lambda \in \Lambda$ as
\[
\bar{G}(\lambda) = \text{hom}(G(\lambda), \lambda).
\]  
(13)

Since $\sum_{i=1}^n \lambda_i = 1$, such an operation can be done by simply multiplying each monomial of $G(\lambda)$ by a suitable power of $\sum_{i=1}^n \lambda_i$.

Now we are ready to define
\[
M(\gamma^2, L, \lambda) = \text{hom} \left( \begin{pmatrix}
\gamma^2 a^2(\lambda) I & WT(L, \lambda) \\
W(L, \lambda) & I
\end{pmatrix}, \lambda \right)
\]  
(14)

and denote by deg($M$) the degree of matrix polynomial $M$ with respect to $\lambda$.

Lemma 4. The condition (12) holds if and only if there exists a scalar $\epsilon > 0$ and an integer $k \in \mathbb{N}$ such that
\[
\left( M(\gamma^2, L, \lambda^2) - \frac{\epsilon}{2} \|\lambda\|^{2 \text{deg}(M)} \right) \|\lambda\|^{2k} \text{ is SOS.}
\]  
(15)

Proof: “$\Leftarrow$” Suppose there exists a scalar $\epsilon > 0$ and an integer $k \in \mathbb{N}$ satisfying (15). It follows that
\[
M(\gamma^2, L, \lambda^2) > 0, \forall \lambda \in \mathbb{R}_0^n.
\]
By exploiting Theorem 11 in [16], we have that $M(\gamma^2, L, \lambda) > 0, \forall \lambda \in \Lambda$. This is the same with [12] due to

$$M(\gamma^2, L, \lambda) = \begin{pmatrix} \gamma^2 a^2(\lambda) I & W^T(L, \lambda) \\ W(L, \lambda) & I \end{pmatrix}, \forall \lambda \in \Lambda.$$  

“⇒” Now assume the condition (12) holds. Since $\Lambda$ is a compact, there exists a small enough scalar $\epsilon > 0$ such that

$$\left( \gamma^2 a^2(\lambda) I \right) \left( W(T(L, \lambda)) - \epsilon I \right) > 0, \forall \lambda \in \Lambda,$$

which is equivalent to $M(\gamma^2, L, \lambda) - \epsilon \left( \sum_{i=1}^{n} \lambda_i \right)^{\text{deg}(M)} I > 0, \forall \lambda \in \Lambda$. Next, let us apply the extended Polya’s Theorem 3 in [17], and obtain that there exists $k \in \mathbb{N}$ such that all coefficients of

$$M(\gamma^2, L, \lambda) - \epsilon \left( \sum_{i=1}^{n} \lambda_i \right)^{\text{deg}(M)} I \left( \sum_{i=1}^{n} \lambda_i \right)^k$$

with respect to $\lambda$ are positive definite. Rewrite

$$\left[ M(\gamma^2, L, \lambda) - \epsilon \left( \sum_{i=1}^{n} \lambda_i \right)^{\text{deg}(M)} I \right] \left( \sum_{i=1}^{n} \lambda_i \right)^k = \sum_{\xi \in \Xi} C_\xi \lambda^\xi
$$

where $\Xi = \{ \xi \in \mathbb{N}^n : \sum_{i=1}^{n} \xi_i = \text{deg}(M) + k \}$, and it follows that all $C_\xi > 0$. Hence, we can express $C_\xi$ as $C_\xi = D_\xi^T D_\xi$ by Cholesky decomposition. Now let us replace $\lambda$ with $\lambda^2$, and it can be obtained that

$$\left( M(\gamma^2, L, \lambda^2) - \epsilon \| \lambda \|_2^{2\text{deg}(M)} \right) \| \lambda \|_2^{2k} = \sum_{\xi \in \Xi} \left( D_\xi \lambda^\xi \right)^T \left( D_\xi \lambda^\xi \right),$$

which shows that the condition (15) holds.

Remark 5. It is worth mentioning that the condition (15) is necessary when $k$ is sufficiently large. In fact, the upper bounds of $k$ required for achieving necessity in Lemma [4] has been investigated in [17].

Theorem 6. The optimization problem (10) can be solved by the following SOS program:

$$\min_{\eta, \epsilon > 0} \eta$$

$$k \in \mathbb{N}, l$$

s.t. $$\left( M(\gamma^2, L, \lambda^2) - \epsilon \| \lambda \|_2^{2\text{deg}(M)} \right) \| \lambda \|_2^{2k}$$

is SOS

where $\gamma^* = \sqrt{\eta^*}$ and $l^*$ is given by the optimal solution of (16).

Remark 7. Theorem 6 provides an approach to address Problem 3 via solving convex optimization problems. Specifically, for fixed $k \in \mathbb{N}$, the condition in (16) can be verified through an SDP for the reason that the condition for a matrix polynomial depending linearly on some decision variables to be SOS is equivalent to an LMI feasibility test [16]. For each fixed $k$, the SOS program in Theorem 6 provides an upper bound for $\gamma^*$, and this upper bound becomes strict when $k$ is sufficiently large. To sum up, to solve (16), one can start solving the SDP with small fixed degree $k$, and then repeat for increased $k$ until the optimal solution $\eta^*$ converges. Observe that $\eta^*$ will decrease as $k$ increases, and Theorem 6 is guaranteed to be nonconservative for sufficiently large $k$ based on Remark 5.

III. Z-DOMAIN RESULTS

When the interval per trial $N$ is large, it is obvious that the matrix $\mathbf{P}(\lambda)$ in (3) as well as the associating matrix $M$ in (14) will have large size. This will inevitably lead to high computation complexity making the optimization in Theorem 6 much less tractable. To circumvent that, we resort to the frequency-domain approach by assuming infinite trial length. ①

①This is a standard assumption in ILC when frequency domain analysis methods are exploited. See [18] for more details.
In this section, we consider the uncertain plant represented in $z$-domain as

$$P(\lambda, z) = \frac{b_m(\lambda)z^m + b_{m-1}(\lambda)z^{m-1} + \cdots + b_0(\lambda)}{z^n + a_{n-1}(\lambda)z^{n-1} + \cdots + a_0(\lambda)}$$  \quad (17)$$

where $\lambda$ is the uncertainty vector. To make the frequency domain representation well defined, we assume that $P(\lambda, z)$ are stable for all $\lambda$ constrained in some set. This can be verified by various methods, see [19], [20] for instance. The system (1) is described by

$$Y_j(z) = P(z, \lambda)U_j(z) + D(z)$$  \quad (18)$$

and the learning algorithm (5) is transformed to

$$U_{j+1}(z) = Q(z) [U_j(z) + zL(z)E_j(z)]$$  \quad (19)$$

where $E_j(z) = Y_d(z) - Y_j(z)$. Note that the combined effects of the initial condition of the plant and the disturbance can be reflected in $D(z)$.

Without loss of generality, we consider non-causal Q-filter $Q(z)$ and learning function $L(z)$ with finite impulse responses

$$L(z) = l_{-k_1}z^{k_1} + \cdots + l_{-1}z + l_0 + l_1z^{-1} + \cdots l_{-k_2}z^{-k_2}$$ \quad (20)

$$Q(z) = q_{-k_2}z^{k_3} + \cdots + q_{-1}z + q_0 + q_1z^{-1} + \cdots q_{-k_4}z^{-k_4}.$$ \quad (21)

As it will be shown later, the subsequent results are applicable to arbitrarily chosen $k_1, k_2, k_3, k_4 \in \mathbb{N}$. This enables the designers to decide the complexity of the learning algorithm based on the memory and computational capability allowed by the computing platform.

As in the previous section, we aim at achieving the robust monotonic convergence. To this end, observe that the error dynamics can be obtained as

$$E_{\infty}(z) - E_{j+1}(z) = Q(z) [1 - zL(z)P(z, \lambda)] (E_{\infty}(z) - E_j(z)).$$

Analogous to Definition 2, the uncertain system (18) with the ILC algorithm (19) is said to be robustly monotonically convergent if there exists $0 \leq \gamma < 1$ such that

$$\|E_{\infty}(z) - E_{j+1}(z)\|_2 < \gamma \|E_{\infty}(z) - E_j(z)\|_2, \forall j \in \mathbb{N}, \forall \lambda \in \Lambda.$$  \quad (22)$$

This is equivalent to

$$\gamma \triangleq \|Q(z) [1 - zL(z)P(z, \lambda)]\|_\infty < 1, \forall \lambda \in \Lambda.$$  \quad (23)$$

The problem addressed in this section is as follows.

**Problem 8.** Given a fixed $Q(z)$, find the learning function $L(z)$ such that the convergent rate $\gamma$ defined in (22) is minimized, i.e.,

$$\min_{L} \sup_{\lambda \in \Lambda} \|Q(z) [1 - zL(z)P(z, \lambda)]\|_\infty.$$  \quad (24)$$

**A. Nominal case**

Let us start by considering the nominal case with the plant given by a rational transfer function

$$P(z) = \frac{b_mz^m + b_{m-1}z^{m-1} + \cdots + b_0}{z^n + a_{n-1}z^{n-1} + \cdots + a_0}.$$  \quad (25)$$

The optimization problem in (23) is now simplified as

$$\min_{\gamma, L} \gamma$$

$$\gamma^2 - \frac{Q(z) [1 - zL(z)P(z)]Q(z) [1 - zL(z)P(z)]}{\forall z \in \mathbb{C}, |z| = 1 > 0}$$ \quad (25)$$
where \( l \) is the vector containing \( l_{-k_1}, \ldots, l_0, \ldots, l_{k_2} \) to be optimized. Since \( z \) is constrained into a compact set, one has that the constraint in (25) holds if and only if there exists a scalar \( \epsilon > 0 \) satisfying
\[
\gamma^2 - Q(z) [1 - zL(z)P(z)]Q(z) [1 - zL(z)P(z)] - \epsilon \geq 0, \forall z \in \mathbb{C}, |z| = 1. \tag{26}
\]

Let \( x \in \mathbb{R} \) be a new variable and define the function \( \phi : \mathbb{R} \to \mathbb{C} \) as \( \phi(x) = \frac{1-x^2+j2x}{1+x^2} \). It can be observed that the complex unit circle \( |z| = 1 \) can be parameterized by the real variable \( x \in \mathbb{R} \). As a result, the optimization problem (25) can be equivalently reformulated as
\[
\min_{\gamma, \eta, \epsilon > 0, l} \gamma^2 - \epsilon - Q(\phi(x)) [1 - \phi(x)L(\phi(x))P(\phi(x))] \cdot Q(\phi(x)) [1 - \phi(x)L(\phi(x))P(\phi(x))] \geq 0, \forall x \in \mathbb{R}. \tag{27}
\]

Next, let us express
\[
Q(\phi(x)) [1 - \phi(x)L(\phi(x))P(\phi(x))] = \left( \sum_{i=-k_3}^{k_4} q_i \left( \frac{1-x^2+j2x}{1+x^2} \right)^{-i} \right) \left[ 1 - \frac{1-x^2+j2x}{1+x^2} \left( \sum_{i=-k_1}^{k_2} l_i \right) \right] \left( \frac{1-x^2+j2x}{1+x^2} \right)^{-i} \left( \frac{1-x^2+j2x}{1+x^2} \right)^{-i} \left( \sum_{i=1}^{n-1} a_i \left( \frac{1-x^2+j2x}{1+x^2} \right)^i \right) \] \tag{28}
\]

where \( \tau_1 \) and \( \tau_2 \) are polynomial in \( x \) with their coefficients depending linearly on \( l \).

**Lemma 9.** The optimal solution \((\gamma^*, l^*)\) to (25) can be obtained by solving the following SOS program
\[
\min_{\eta > 0, \epsilon > 0, l} \eta \begin{pmatrix} (\eta - \epsilon) \tau_3(x)^2 & \tau_1(l, x) & \tau_2(l, x) \\ \tau_1(l, x) & 1 & 0 \\ \tau_2(l, x) & 0 & 1 \end{pmatrix} \text{ is SOS} \tag{29}
\]

where \( \gamma^* = \sqrt{\eta^*} \) and \( l^* \) is given by the optimal solution.

**Proof:** By (28), the constraint in (27) can be rewritten into
\[
(\gamma^2 - \epsilon)^2 \tau_3(x)^2 - \tau_1(l, x)^2 - \tau_2(l, x)^2 \geq 0,
\]
which is equivalent to
\[
\begin{pmatrix} (\gamma^2 - \epsilon) \tau_3(x)^2 & \tau_1(l, x) & \tau_2(l, x) \\ \tau_1(l, x) & 1 & 0 \\ \tau_2(l, x) & 0 & 1 \end{pmatrix} \geq 0
\]
according to the Schur complement lemma.

From Theorem 4 in [16], one has that a necessary and sufficient condition for a univariate matrix polynomial being positive semidefinite is that it is an SOS matrix. Consequently, the above constraint can be equivalently expressed as
\[
\begin{pmatrix} (\eta - \epsilon) \tau_3(x)^2 & \tau_1(l, x) & \tau_2(l, x) \\ \tau_1(l, x) & 1 & 0 \\ \tau_2(l, x) & 0 & 1 \end{pmatrix} \text{ is SOS.}
\]

Therefore, addressing (29) equates to solving (25) with \( \gamma = \eta^2 \). \qed
B. Uncertain case

In this rest of this section, we proceed to the case whereas the plant is affected by uncertainty.

Consider the uncertain plant (1) with \(a_n, a_{n-1}, \ldots, a_0\) and \(b_m, \ldots, b_0\) allowed to be any rational functions in the uncertainty vector \(\lambda\). We assume that the uncertainty vector \(\lambda \in \mathbb{R}^n\) is constrained into a simplex \(\Lambda\).

The optimization problem (23) is now given as

\[
\min_{\gamma, d} \gamma \quad \gamma^2 - Q(z) \left[ 1 - z L(z) P(z, \lambda) \right] - Q(z) \left[ 1 - z L(z) P(z, \lambda) \right] > 0
\]

\[
\forall z \in \mathbb{C}, |z| = 1, \forall \lambda \in \Lambda.
\]

Let \(x_1, x_2 \in \mathbb{R}\) be new variables. By substituting \(z\) with \(\varphi(x_1, x_2) = x_1 + jx_2\), one can observe that \(|z| = 1\) can be parameterized by \(x_1, x_2 \in \mathbb{R}\) subject to \(x_1^2 + x_2^2 = 1\). Then (30) can be transformed to

\[
\gamma^2 - Q(\varphi(x_1, x_2)) \left[ 1 - \varphi(x_1, x_2) L(\varphi(x_1, x_2)) P(\varphi(x_1, x_2), \lambda) \right] - Q(\varphi(x_1, x_2)) \left[ 1 - \varphi(x_1, x_2) L(\varphi(x_1, x_2)) P(\varphi(x_1, x_2), \lambda) \right] > 0
\]

\[
\forall x_1, x_2 \in \mathbb{R}, x_1^2 + x_2^2 = 1, \forall \lambda \in \Lambda.
\]

Similar to (28), we can express

\[
Q(\varphi(x_1, x_2)) \left[ 1 - \varphi(x_1, x_2) L(\varphi(x_1, x_2)) P(\varphi(x_1, x_2), \lambda) \right] = \frac{Q_1(l, x_1, x_2, \lambda) + jQ_2(l, x_1, x_2, \lambda)}{\nu_3(l, x_1, x_2, \lambda)}
\]

where \(\nu_1\) and \(\nu_2\) are polynomials in \((x_1, x_2, \lambda)\) with their coefficients depending linearly on \(l\).

Define

\[
T(l, \gamma^2, x_1, x_2, \lambda) = \begin{pmatrix}
\gamma^2 \nu_3(\lambda, x_1, x_2)^2 & \nu_1(l, \lambda, x_1, x_2) & \nu_2(l, \lambda, x_1, x_2) \\
\nu_1(l, \lambda, x_1, x_2) & 1 & 0 \\
\nu_2(l, \lambda, x_1, x_2) & 0 & 1
\end{pmatrix}
\]

and let

\[
\tilde{T}(l, \gamma^2, x_1, x_2, \lambda) = \text{hom} \left( T(l, \gamma^2, x_1, x_2, \lambda), \lambda \right)
\]

where the operator \(\text{hom}\) is defined in (13). Denote the degree of \(T\) defined above in \((x_1, x_2)\) as \(\text{deg} (T, x)\), and the degree of \(T\) in \(\lambda\) as \(\text{deg}(T, \lambda)\).

By substituting \(x_1\) and \(x_2\) with \(x_1 = \frac{1-x^2}{1+x^2}\) and \(x_2 = \frac{2x}{1+x^2}\) respectively, let us further define

\[
\tilde{T}(l, \gamma^2, x, \lambda) = \tilde{T}(l, \gamma^2, \frac{1-x^2}{1+x^2}, \frac{2x}{1+x^2}, \lambda),
\]

and observe that the variables \(x_1, x_2 \in \mathbb{R}\) subject to \(x_1^2 + x_2^2 = 1\) is now parameterized by \(x \in \mathbb{R}\). It is straightforward to verify that \((1+x^2)^{\text{deg}(T,x)} \tilde{T}(l, \gamma^2, x, \lambda^2)\) is a matrix polynomial in \((x, \lambda)\), and is a matrix homogeneous polynomial in \(\lambda\).

**Lemma 10.** The following condition holds

\[
\gamma^2 - Q(z) \left[ 1 - z L(z) P(z, \lambda) \right] - Q(z) \left[ 1 - z L(z) P(z, \lambda) \right] > 0
\]

\[
\forall z \in \mathbb{C}, |z| = 1, \forall \lambda \in \Lambda
\]

if and only if there exists a scalar \(\epsilon > 0\) and an integer \(k \in \mathbb{N}\) satisfying that

\[
(1+x^2)^{\text{deg}(T,x)} \left( \tilde{T}(l, \gamma^2, x, \lambda^2) - \epsilon \|\lambda\|^2_{\text{deg}(T,\lambda)} I \right) \|\lambda\|^{2k}_{\text{deg}(T,\lambda)}\text{ is SOS}.
\]
Proof: “⇒” Assume that there exists a scalar $\epsilon > 0$ an integer $k \in \mathbb{N}$ such that the condition (37) is satisfied. Then, we can derive that

$$\hat{T}(l, \gamma^2, x, \lambda^2) > 0, \forall x \in \mathbb{R}, \forall \lambda \in \mathbb{R}_0^+.$$ 

Next, let us recall that $\hat{T}$ is a matrix homogeneous polynomial in $\lambda$ according to (34). Hence, it can be inferred from Theorem 11 in [16] that the above condition holds if and only if

$$\hat{T}(l, \gamma^2, x, \lambda) > 0, \forall x \in \mathbb{R}, \forall \lambda \in \Lambda.$$ 

It follows from (35) that the above condition is equivalent to

$$\hat{T}(l, \gamma^2, x_1, x_2, \lambda) > 0, \forall x_1^2 + x_2^2 = 1, \forall \lambda \in \Lambda,$$

which yields that $T(l, \gamma^2, x_1, x_2, \lambda) > 0, \forall x_1^2 + x_2^2 = 1, \forall \lambda \in \Lambda.$ Based on (31) and (32), it can verified that the condition (36) is satisfied.

“⇒” Suppose the condition in (36) holds. It follows from the variable substitution $z = x_1 + jx_2$ that (31) holds. Therefore, it holds that

$$T(l, \gamma^2, x_1, x_2, \lambda) > 0, \forall x_1^2 + x_2^2 = 1, \forall \lambda \in \Lambda.$$ 

Due to the compactness of $\Lambda$ and the set of unit circle of $(x_1, x_2)$, there exists a small enough scalar $\epsilon > 0$ such that

$$T(l, \gamma^2, x_1, x_2, \lambda) - \epsilon I > 0, \forall x_1^2 + x_2^2 = 1, \forall \lambda \in \Lambda.$$ 

Since $\sum_{i=1}^n \lambda_i = 1$, one has that

$$\hat{T}(l, \gamma^2, x_1, x_2, \lambda) - \epsilon (\sum_{i=1}^n \lambda_i)^{\deg(T, \lambda)} I > 0, \forall x_1^2 + x_2^2 = 1, \forall \lambda \in \Lambda.$$ 

Next, let us apply the result on positivity of matrix homogeneous polynomials over the simplex in Theorem 3 in [17]: for each fixed pair of $(x_1, x_2)$, $\hat{T}(l, \gamma^2, x_1, x_2, \lambda) - \epsilon (\sum_{i=1}^n \lambda_i)^{\deg(T, \lambda)} I$ is a matrix homogeneous polynomial in $\lambda$ with $\lambda$ constrained in the simplex $\Lambda$, which follows that there exists $k(x_1, x_2) \in \mathbb{N}$ such that all the matrix coefficients of $(\hat{T}(l, \gamma^2, x_1, x_2, \lambda) - \epsilon (\sum_{i=1}^n \lambda_i)^{\deg(T, \lambda)} I) (\sum_{i=1}^n \lambda_i)^{k(x_1, x_2)}$ with respect to $\lambda$ are positive definite. As a consequence, the matrix coefficients of

$$(\hat{T}(l, \gamma^2, x_1, x_2, \lambda) - \epsilon (\sum_{i=1}^n \lambda_i)^{\deg(T, \lambda)} I) (\sum_{i=1}^n \lambda_i)^{\hat{k}}$$

with any integer $\hat{k} \geq k(x_1, x_2)$ can be shown to be positive definite. Hence, it can be inferred that there exists $k \in \mathbb{N}$ such that the matrix coefficients of

$$(\hat{T}(l, \gamma^2, x_1, x_2, \lambda) - \epsilon (\sum_{i=1}^n \lambda_i)^{\deg(T, \lambda)} I) (\sum_{i=1}^n \lambda_i)^{k}$$

with respect to $\lambda$ are positive definite for all $x_1, x_2 \in \mathbb{R}, x_1^2 + x_2^2 = 1$. Let us denote $G_d(x_1, x_2)$ with $d \in \mathcal{D} \triangleq \{d \in \mathbb{N}^n : \sum_{i=1}^n d_i = \deg(T, \lambda) + k\}$ by the matrix coefficients, i.e.,

$$G_d(x_1, x_2) = \sum_{d \in \mathcal{D}} G_d(x_1, x_2) \lambda_1^{d_1} \cdots \lambda_n^{d_n}$$

(38)

wherein $G_d(x_1, x_2) > 0$ for all $x_1, x_2 \in \mathbb{R}, x_1^2 + x_2^2 = 1$. 

By the variable substitution $x_1 = \frac{1-x^2}{1+x^2}$ and $x_2 = \frac{2x}{1+x^2}$, we can rewrite (38) equivalently into

$$\left(\hat{T}(l, \gamma^2, x, \lambda) - \epsilon (\sum_{i=1}^n \lambda_i)^{\deg(T, \lambda)} I\right) (\sum_{i=1}^n \lambda_i)^{k} = \sum_{d \in \mathcal{D}} G_d \left(\frac{1-x^2}{1+x^2}, \frac{2x}{1+x^2}\right) \lambda_1^{d_1} \cdots \lambda_n^{d_n}.$$
Thus, \( G_d(\frac{1-x^2}{1+x^2}, \frac{2x}{1+x^2}) > 0 \) for all \( x \in \mathbb{R} \). Multiplying both sides of the above equation with \((1 + x^2)^{\text{deg}(T,x)}\), we obtain that
\[
(1 + x^2)^{\text{deg}(T,x)} \left( \hat{T}(l, \gamma^2, x, \lambda^2) - \epsilon \left( \sum_{i=1}^{n} \lambda_i \right)^{\text{deg}(T,\lambda)} I \right) \left( \sum_{i=1}^{n} \lambda_i \right)^k
= \sum_{d \in \mathbb{D}} (1 + x^2)^{\text{deg}(T,x)} G_d(\frac{1-x^2}{1+x^2}, \frac{2x}{1+x^2}) \lambda_1^{d_1} \cdots \lambda_n^{d_n},
\]
and it can be inferred that \((1 + x^2)^{\text{deg}(T,x)} G_d(\frac{1-x^2}{1+x^2}, \frac{2x}{1+x^2})\) is a matrix polynomial in \( x \).

Since \((1 + x^2)^{\text{deg}(T,x)} G_d(\frac{1-x^2}{1+x^2}, \frac{2x}{1+x^2})\) is positive definite for all \( x \in \mathbb{R} \), it can be obtained from Theorem 4 in [16] that it is an SOS matrix polynomial in \( x \), i.e., it can be rewritten as
\[
(1 + x^2)^{\text{deg}(T,x)} G_d(\frac{1-x^2}{1+x^2}, \frac{2x}{1+x^2}) = \sum_i G_{di}(x)^T G_{di}(x)
\]
for some matrix polynomials \( G_{di} \) in \( x \).

It can now be derived that
\[
(1 + x^2)^{\text{deg}(T,x)} \left( \hat{T}(l, \gamma^2, x, \lambda^2) - \epsilon \left( \sum_{i=1}^{n} \lambda_i \right)^{\text{deg}(T,\lambda)} I \right) \left( \sum_{i=1}^{n} \lambda_i \right)^2 k
= \sum_{d \in \mathbb{D}} (1 + x^2)^{\text{deg}(T,x)} G_d(\frac{1-x^2}{1+x^2}, \frac{2x}{1+x^2}) \lambda_1^{2d_1} \cdots \lambda_n^{2d_n}
= \sum_{d \in \mathbb{D}} \sum_i G_{di}(x)^T G_{di}(x) \lambda_1^{d_1} \cdots \lambda_n^{d_n}
= \sum_{d \in \mathbb{D}} \sum_i (\lambda_1^{d_1} \cdots \lambda_n^{d_n} G_{di}(x))^T (\lambda_1^{d_1} \cdots \lambda_n^{d_n} G_{di}(x)),
\]
and hence we can conclude that \((1 + x^2)^{\text{deg}(T,x)} \left( \hat{T}(l, \gamma^2, x, \lambda^2) - \epsilon \left\| \lambda \right\|^2_{\text{deg}(T,\lambda)} I \right) \left\| \lambda \right\|^2_k\) is SOS. \(\blacksquare\)

Remark 11. We note that the result in Lemma [10] cannot be extended trivially from Lemma [4] due to the existence of an extra independent variable \( z \) in (36) apart from the uncertainty \( \lambda \). In fact, Lemma [7] proposes a necessary and sufficient condition that can be verified by solving convex optimization in the form of SDP programs for characterizing positivity of a matrix rational function on both the standard simplex and the unit circle simultaneously.

At this point, we are ready to present the following theorem.

**Theorem 12.** The optimal solution \((\gamma^*, l^*)\) to (39) can be obtained by solving the following SOS program
\[
\min_{\eta, \epsilon \geq 0, k \in \mathbb{N}, l} \eta
\text{s.t.} \quad (1 + x^2)^{\text{deg}(T,x)} \left( \hat{T}(l, \gamma^2, x, \lambda^2) - \epsilon \left\| \lambda \right\|^2_{\text{deg}(T,\lambda)} I \right) \left\| \lambda \right\|^2_k\text{ is SOS}
\]
where \( \gamma^* = \sqrt{\eta} \) and \( l^* \) is given by the optimal solution of (39).

**Proof:** It can be proved directly from Lemma [10] \(\blacksquare\)

Theorem [12] provides an approach to solve Problem [8] by solving a set of optimization problems following the same procedures described in Remark [7].

It is worth mentioning that when \( Q(q) \) and \( L(q) \) are both designed to be causal, the optimal \( \gamma^* \) and \( l^* \) obtained from (39) will satisfy \( \left\| e_{\infty} - e_{j+1} \right\| < \gamma \left\| e_{\infty} - e_j \right\| \) for \( j \in \{1, 2, \ldots \} \) for the ILC system (18) and (19) with any finite trial length \( N \) (See [3] for more details).

**Remark 13.** The methods proposed in this work can be applied to solve Problem [3] or [8] with variable Q-filter \( Q(q) \) by slight modifications. This can be done as follows. First, choose a proper initial \( Q(q) \) and solve the SOS program in (16) or (39). Next, fix \( L(q) \) as the obtained optimal solution, and solve the SOS program again with variable \( Q(q) \). If there is any constraint on \( Q(q) \), it can be imposed as an additional constraint in the SOS program. Then, repeat the previous steps by iterating between \( L(q) \) and \( Q(q) \) to obtain a decreasing \( \gamma \) as \( T \) in (16) or \( \hat{T} \) in (39) is bilinear in \( L(q) \) and \( Q(q) \).
IV. Numerical Example

In this section, we present a numerical example to illustrate the proposed results. The computations are done by Matlab with the toolbox SeDuMi [21] and SOSTOOLS [22]. Consider a linear uncertain plant given by

\[
A = \begin{pmatrix}
\theta & -0.5 \\
-2\theta - 0.1 & 0.2
\end{pmatrix}, \quad B = \begin{pmatrix}1 \\ 1\end{pmatrix}, \quad C = (1 \ 1)
\]

where \(\theta\) is an uncertainty variable that can take any value in \([-0.7, -0.5]\). Suppose the desired output trajectory is described by \(y_d = \sin(k2\pi/N)\) with \(N = 100\), and fix \(Q = I\) for simplicity. We aim at finding the optimal learning function with the structure (7) or (20) that minimizes the robust convergent rate \(\gamma\). Given \(N = 100\), one can predict high computational complexity if the approach in Theorem 16 is adopted. Hence, we resort to the frequency-domain method to solve Problem 8.

First of all, by exploiting the Jury stability criterion, it can be verified that the plant is internally robust stable for all \(\theta \in [-0.7, -0.5]\).

Next, let us express the uncertain plant in \(z\)-domain as

\[
P(z, \theta) = \frac{-40z + 60\theta + 16}{20z^2 + (4 + 20\theta)z + 16\theta + 1}, \quad \theta \in [-0.7, -0.5].
\]

By utilizing the variable transform described in Remark 1, the plant can be rewritten into

\[
P(z, \lambda) = \frac{40z + 30\lambda_1 + 42\lambda_2 - 16}{20z^2 + (10\lambda_1 + 14\lambda_2 - 4)z + 8\lambda_1 + 11.2\lambda_2 - 1}, \quad \lambda \in \Lambda.
\]

Then, we solve the SOS program proposed in Theorem 12. To avoid numerical error, we set \(\epsilon = 0.001\). With the structure of \(L(z) = l_0, \ L(z) = l_0 + l_1z^{-1}\) and \(L(z) = l_0 + l_1z^{-1} + l_2z^{-2}\), we obtain the optimal solution of \(\gamma^*, \ l^*\) as shown in the following table.

| \(k\) | \(\gamma^*\) | \(l_0\) | \(\gamma^*\) | \((l_0^*, l_1^*)\) | \(\gamma^*\) | \((l_0^*, l_1^*, l_2^*)\) |
|---|---|---|---|---|---|---|
| 0 | 0.812 | 0.303 | 0.683 | (0.326, -0.132) | 0.463 | (0.491, 0.0252, 0.310) |
| 1 | 0.812 | 0.303 | 0.683 | (0.326, -0.132) | 0.463 | (0.492, 0.0268, 0.311) |
| 2 | 0.812 | 0.303 | 0.682 | (0.326, -0.132) | 0.463 | (0.492, 0.0272, 0.311) |
| 3 | 0.809 | 0.303 | 0.677 | (0.325, -0.127) | 0.463 | (0.493, 0.0273, 0.311) |
As it can be seen from the table, the additional variable in $L(z)$ give a boost to performance. In this example, the simulation results also show that adding the non-causal term in $L(z)$ does not improve the performance. This can be implied from the relative degree of $1 - z L(z) P(z, \lambda)$.

When the learning function of the form $L(z) = l_0 + l_1 z^{-1} + l_2 z^{-2} + l_3 z^{-3}$ is considered, by solving the SOS program in Theorem 12, we obtain that $\gamma^* = 0.319$ and the optimal leaning function is

$$(l_0^*, l_1^*, l_2^*, l_3^*) = (0.508, -0.0716, 0.189, -0.197).$$

With randomly values of $\theta \in [-0.7, -0.5]$, and randomly generated disturbance $d$, the trajectories for the $l_2$ norm of the error $e_j$ are depicted in Figure 1 (since $Q = I$, it follows from [23] that $e_{\infty} = 0$). Since $\gamma^* < 1$, we can conclude that with the learning function $L^*(q)$, the ILC system is robustly monotonically converges with converging rate $\gamma^*$.

V. Conclusion

This work has considered robust monotonic convergent ILC for uncertain linear time-invariant systems. Both time domain and frequency domain are discussed. To establish whether the ILC system is monotonically convergent, a necessary and sufficient condition in the form of SOS program has been provided. This condition provides us with an approach to optimize the convergence speed by solving a set of convex optimization problems. Inspired by [13], [24] An interesting future direction is to further develop non-causal ILC algorithms in frequency domain while removing the assumption on finite trial interval.

REFERENCES

[1] 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.
[2] S. Hara, Y. Yamamoto, T. Omata, and M. Nakano, “Repetitive control system: A new type servo system for periodic exogenous signals,” IEEE Transactions on automatic control, vol. 33, no. 7, pp. 659–668, 1988.
[3] M. Norrlöf and S. Gunnarsson, “Time and frequency domain convergence properties in iterative learning control,” International Journal of Control, vol. 75, no. 14, pp. 1114–1126, 2002.
[4] D. A. Bristow, M. Tharayil, and A. G. Alleyne, “A survey of iterative learning control,” IEEE control systems magazine, vol. 26, no. 3, pp. 96–114, 2006.
[5] K. L. Moore, Y. Chen, and V. Bahl, “Monotonically convergent iterative learning control for linear discrete-time systems,” Automatica, vol. 41, no. 9, pp. 1529–1537, 2005.
[6] T. Seel, T. Schauer, and J. Raisch, “Monotonic convergence of iterative learning control systems with variable pass length,” International Journal of Control, vol. 90, no. 3, pp. 393–406, 2017.
[7] T. Harte, J. Hätönen, and D. Owens*, “Discrete-time inverse model-based iterative learning control: stability, monotonicity and robustness,” International Journal of Control, vol. 78, no. 8, pp. 577–586, 2005.
[8] H. S. Ahn, K. L. Moore, and Y. 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.
[9] D. Owens, J. Hatonen, and S. Daley, “Robust monotone gradient-based discrete-time iterative learning control,” International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal, vol. 19, no. 6, pp. 634–661, 2009.
[10] T. D. Son, G. Pipeleers, and J. Swevers, “Robust monotonic convergent iterative learning control,” IEEE Transactions on Automatic Control, vol. 61, no. 4, pp. 1063–1068, 2015.
[11] X. Ge, J. L. Stein, and T. Ersal, “Frequency-domain analysis of robust monotonic convergence of norm-optimal iterative learning control,” IEEE Transactions on Control Systems Technology, vol. 26, no. 2, pp. 637–651, 2017.
[12] D. de Roover, “Synthesis of a robust iterative learning controller using an $H_{\infty}$ approach,” in Proceedings of 35th IEEE Conference on Decision and Control, vol. 3, IEEE, 1996, pp. 3044–3049.
[13] J. Van de Wijdeven, T. Donkers, and O. Bosgra, “Iterative learning control for uncertain systems: Robust monotonic convergence analysis,” Automatica, vol. 45, no. 10, pp. 2383–2391, 2009.
[14] K. L. Moore, H.-S. Ahn, and Y. Q. Chen, “Iteration domain $H_{\infty}$-optimal iterative learning controller design,” International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal, vol. 18, no. 10, pp. 1001–1017, 2008.
[15] D. A. Bristow and A. G. Alleyne, “Monotonic convergence of iterative learning control for uncertain systems using a time-varying filter,” IEEE Transactions on Automatic Control, vol. 53, no. 2, p. 583, 2008.
[16] G. Chesi, “LMI techniques for optimization over polynomials in control: a survey,” IEEE Transactions on Automatic Control, vol. 55, no. 11, pp. 2500–2510, 2010.
[17] C. W. Scherer and C. W. Hol, “Matrix sum-of-squares relaxations for robust semi-definite programs,” Mathematical programming, vol. 107, no. 1-2, pp. 189–211, 2006.
[18] N. Amann, D. H. Owens, E. Rogers, and A. Wahl, “An $H_{\infty}$ approach to linear iterative learning control design,” *International Journal of Adaptive Control and Signal Processing*, vol. 10, no. 6, pp. 767–781, 1996.

[19] L. Su and G. Chesi, “On the design of output feedback controllers for LTI systems over fading channels,” *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1503–1508, 2017.

[20] ——, “Robust stability analysis and synthesis for uncertain discrete-time networked control systems over fading channels,” *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1966–1971, 2016.

[21] J. F. Sturm, “Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones,” *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.

[22] S. Prajna, A. Papachristodoulou, and P. A. Parrilo, “Introducing SOSTOOLS: A general purpose sum of squares programming solver,” in *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002., vol. 1. IEEE, 2002, pp. 741–746.

[23] M. Q. Phan, R. W. Longman, and K. L. Morre, “Unified formulation of linear iterative learning control,” *Spaceflight mechanics 2000*, pp. 93–111, 2000.

[24] M. M. G. Ardakani, S. Z. Khong, and B. Bernhardsson, “On the convergence of iterative learning control,” *Automatica*, vol. 78, pp. 266–273, 2017.