

Model-free Iterative Learning Control with Recursive Convergence Acceleration for Tracking Control Application

Gen-Hao Zhang

Advisor: Shao-Yi Chien

Graduate Institute of Electronics Engineering

National Taiwan University

Taipei, Taiwan

May 2020

Abstract

Contents

| | |
|---|------------|
| Abstract | i |
| List of Figures | v |
| List of Tables | vii |
| 1 Introduction | 1 |
| 1.1 Motivation | 1 |
| 1.2 Previous Work and Problem Formulation | 5 |
| 1.3 Proposed Approach and Contributions | 6 |
| 1.4 Thesis Overview | 6 |
| 2 Related Work | 7 |
| 2.1 Repetitive Control and ILC | 7 |
| 2.1.1 Repetitive Control | 7 |
| 2.1.2 Iterative Learning Control | 7 |
| 2.1.3 Inverse Filter Design of RC and ILC | 9 |
| 2.2 Inverse Filter Design Method for Repetitive Control and ILC | 9 |
| 2.2.1 Zero Phase Error Tracking Control (ZPETC) | 9 |
| 2.2.2 Unity Magnitude Error Tracking Control (UMETC) | 9 |
| 2.2.3 Direct Inversion Method | 9 |
| 2.2.4 Iterative Learning of Dynamic Inverse Filters (ILCFF) | 9 |
| 2.3 Model-free ILC | 9 |

| | | |
|----------|--|-----------|
| 2.3.1 | Model-free Adjoint-based ILC | 9 |
| 2.3.2 | Model-free Inversion-based Iterative Control (MFIIIC) . . . | 10 |
| 2.3.3 | Non-Linear Inversion-based Iterative Control (NLIIC) . . . | 11 |
| 2.3.4 | Comparison of existed model-free ILC | 11 |
| 3 | Model-free ILC with Recursive Convergence Acceleration | 13 |
| 3.1 | Algorithm Formulation | 13 |
| 3.2 | Stability Analysis | 18 |
| 3.3 | Robustness Analysis | 19 |
| 3.4 | Simulation Result | 21 |
| 3.4.1 | Simulation Setup | 21 |
| 3.4.2 | Ideal Case Without Output Disturbance | 23 |
| 3.4.3 | Output Tracking with Trial-variant Disturbances | 24 |
| 4 | Case Study 1: Triangular Wave Tracking of Galvo Scanner | 29 |
| 4.1 | System Overview | 29 |
| 4.2 | Repetitive Controller Design | 29 |
| 4.3 | Experiment Setup | 29 |
| 4.4 | Experiment Result | 29 |
| | Reference | 31 |

List of Figures

| | | |
|-----|--|----|
| 1.1 | The structure of feedback control system. | 2 |
| 1.2 | The structure of feedforward control system. $F(z)$ is the feed-forward compensator of the closed-loop system $G(z)$, and the optimal design of $F(z)$ is $F(z) = G^{-1}(z)$, which make the overall system $\frac{Y}{R} := F(z)G(z) = 1$ if the disturbance $d(k)$ is not considered. | 3 |
| 1.3 | The structure of repetitive control system. $F(z)$ is the inversion-based filter of the stable system $G(z)$. The stability condition of RC is $\ I - Q(z)F(z)G(z)\ _{\infty} < 1$, where $Q(z)$ is a zero-phase low pass filter, hence the optimal design of the $F(z)$ is $F(z) = G^{-1}(z)$ | 4 |
| 1.4 | The control architecture of iterative learning control. $L(z)$ is the learning filter of ILC and $G(z)$ is a stable system. The stability condition of ILC is $\ I - L(z)G(z)\ _{\infty} < 1$, hence the optimal design of the $L(z)$ is $L(z) = F(z) = G^{-1}(z)$ | 4 |
| 3.1 | The control architecture of the proposed algorithm. The learning is switched to the acceleration filter $\frac{U_j(z)}{R(z)}$ for every $n \in \mathbb{N}$ iterations; otherwise, the control input is updated by the time-reversal based ILC. | 15 |
| 3.2 | The chirp reference with the amplitude of 100 mm and a maximum frequency component of 100 Hz. | 22 |

| | | |
|-----|--|----|
| 3.3 | The convergence of the ILC tracking error. (a) The case with no output disturbance. (b)The case with the output disturbance $d(k) \in N(0, 1)$ | 26 |
| 3.4 | The convergence of the ILC tracking error when the output disturbance $d(k) \in N(0, 80)$. Both MFIC and NLIIC do not converge. . | 27 |
| 3.5 | The ILC tracking error in steady-state. (a) The case with no output disturbance. (b) The case with the output disturbance $d(k) \in N(0, 1)$ | 28 |
| 4.1 | The schematic diagram of the galvo scanner system. | 30 |

List of Tables

| | |
|--|----|
| 3.1 The ILC tracking error obtained from different methods | 24 |
|--|----|

Chapter 1

Introduction

1.1 Motivation

Tracking control problem, which is to make a physical system to move along a desired time-dependent trajectory, has been a critical issue in control applications for many years [1]. For example, for a CNC machine, each axis has to follow the position command signal produced by the trajectory interpolator [2]. Traditionally, the tracking control problem is solved by closed-loop feedback control structure, as shown in Fig. 1.1, where $P(z)$ is the physical system to be controlled (usually called "plant"), $u(k)$ is the control input signal of $P(z)$, $y(k)$ is the output signal of $P(z)$, and $r(k)$ is the desired reference signal that $P(z)$ to be followed. Note that k is the discrete time index of the signal, and z is the delay operator for a system represented in z-domain transfer function form. For this structure, to let the system output $y(k)$ track along the desired reference signal $r(k)$, the controller $C(z)$ is introduced to minimize the tracking error $e(k) := r(k) - y(k)$. Therefore, the tracking control problem can be stated as the following optimization goal:

$$\min_{C(z)} \|r(k) - y(k)\|_2 \quad (1.1)$$

To implement the controller $C(z)$, which is a causal and LTI digital filter, the computation from $e(k)$ to $u(k)$ is usually done by a real-time program in

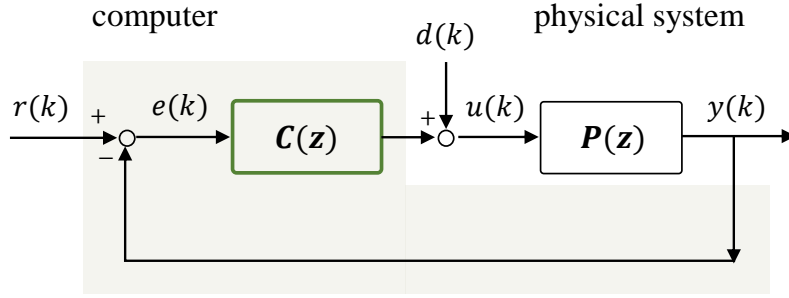


Figure 1.1: The structure of feedback control system.

microcomputer. Specially, proportional-integral-derivative(PID) controller is the most common used ones in industry applications [2]. For PID-based feedback control approach, the pseudo code of the real-time control can be written as:

Algorithm 1 PID-based Feedback Control

Input: The desired reference trajectory $r(k)$ with length N .

Output: The control input u_k

- 1: Initialize the last error $e_{k-1} \leftarrow 0$;
 - 2: Initialize the error summation $e_{\Sigma} \leftarrow 0$;
 - 3: **for** $k = 1$ to N **do**
 - 4: measure the current output signal y_k ;
 - 5: compute the current tracking error: $e_k = r[k] - y_k$;
 - 6: PID control law: $u_k = k_p e_k + k_I e_{\Sigma} + k_D(e_k - e_{k-1})$; $//C(z)$
 - 7: send the control input signal u_k to the system;
 - 8: record the last error $e_{k-1} \leftarrow e_k$;
 - 9: compute the error summation $e_{\Sigma} \leftarrow e_{\Sigma} + e_k$;
 - 10: **end for**
 - 11: Note: The for-loop conducts for 1 sampling period for each loop.
-

The advantages of PID-based feedback control are simple, easy-to-implement and model-free, which only need to tune PID parameters k_P , k_D and k_I without knowing the plant model $P(z)$ in advance, hence it is widely used in industry

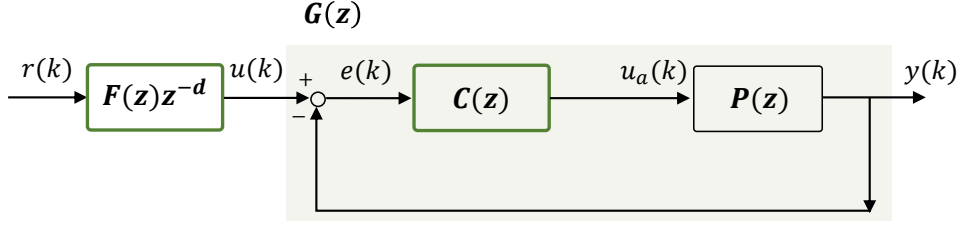


Figure 1.2: The structure of feedforward control system. $F(z)$ is the feedforward compensator of the closed-loop system $G(z)$, and the optimal design of $F(z)$ is $F(z) = G^{-1}(z)$, which make the overall system $\frac{Y}{R} := F(z)G(z) = 1$ if the disturbance $d(k)$ is not considered.

applications. From the pseudo code of the feedback control algorithm, we can observe that once the tracking error occurs, controller $C(z)$ reacts to the error to compute a suitable control input u to remedy the output result. However, this property results the lag in transient tracking, hence the tracking performance of PID-based feedback control algorithm is limited since the feedback controller $C(z)$ only takes action if the error is not zero [3].

To remedy the difficulty that feedback control approach faced, several methods has been proposed, such as feed-forward control[], iterative learning control[] and repetitive control[]. For these approaches, the inversion-based compensator $F(z)$ plays the most important role in the tracking performance, and the optimal design of $F(z)$ is:

$$F(z) = G^{-1}(z) \quad (1.2)$$

$$G(z) = \frac{C(z)P(z)}{1 + C(z)P(z)} \quad (1.3)$$

where $F(z)$ is the inversion-based compensator (also called feedforward compensator, inversion-based filter, and inversion-based learning filter in feedforward

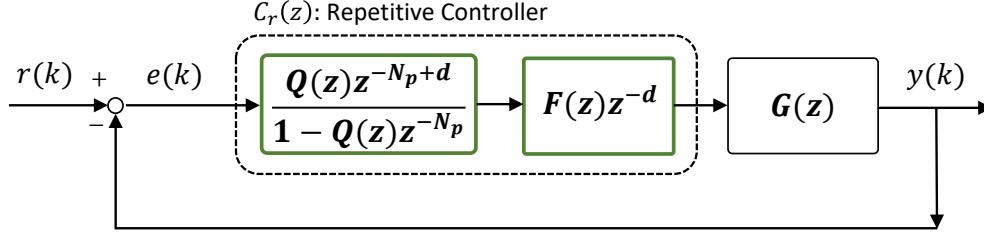


Figure 1.3: The structure of repetitive control system. $F(z)$ is the inversion-based filter of the stable system $G(z)$. The stability condition of RC is $\|I - Q(z)F(z)G(z)\|_\infty < 1$, where $Q(z)$ is a zero-phase low pass filter, hence the optimal design of the $F(z)$ is $F(z) = G^{-1}(z)$.

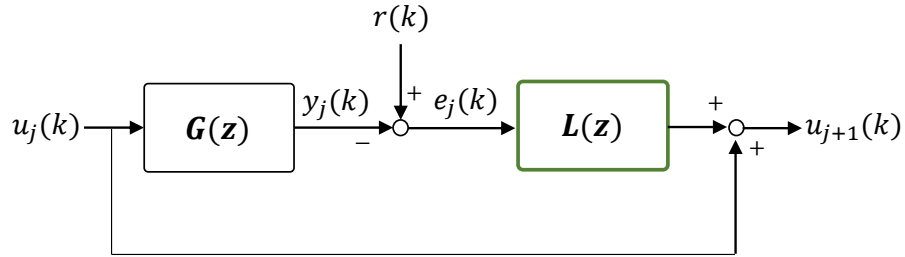


Figure 1.4: The control architecture of iterative learning control. $L(z)$ is the learning filter of ILC and $G(z)$ is a stable system. The stability condition of ILC is $\|I - L(z)G(z)\|_\infty < 1$, hence the optimal design of the $L(z)$ is $L(z) = F(z) = G^{-1}(z)$.

control, repetitive control and iterative learning control), and $G(z)$ is a SISO, LTI and stable system that can be stabilized by the feedback controller $C(z)$.

The motivation of this thesis is to construct a novel algorithm and provide a systematic procedure to improve the performance of each advanced control method mentioned above. Specially, we focus on the development of a model-free iterative learning control algorithm. Our algorithm not only improves the tracking performance of that feedback control, but also

To construct the optimal compensator $F(z)$ for feedforward control, iterative learning control and

1.2 Previous Work and Problem Formulation

To reduce the tracking error iteratively, the ILC updating law has to be well designed. The model-based ILC [[?]; [?]] uses the system inverse model as the learning filter to conduct the learning process. For these approaches, if the system model is identified accurately, the learning process would converge within few iterations [[?]]. However, inversion-based methods require delicate modeling process in advance, which isn't practical for industry applications. Due to the reasons mentioned above, the development of model-free ILC becomes more and more popular in the ILC research community [?].

For the model-free approaches, the PD-type ILC [[?]; [?]] is the simplest way to implement. However, monotonic convergence condition is not always satisfied by tuning the PD-parameters [[?]], and the tuning process may be time-consuming and damage the system. Time-reversal based ILC [[?]] uses the adjoint operator of the system as the learning filter, and the control updating law can be implemented by filtering the reversed error signals using the system itself. However, the learning rate of the time-reversal based ILC is usually slow [[?]].

To achieve a fast error convergence rate, the model-free inversion-based iterative learning control (MFIIC) [[?]] applies the point-by-point division over the

discrete frequency interval to estimate the system inverse dynamics from the input and output signals. Nevertheless, the output disturbances present in the denominator of the computation. Once unpredicted output disturbances are introduced, the magnitude of the denominator may approach to zero. This unpredicted phenomenon leads to a poor transient learning behavior [[?]]. To improve the learning transient performance, NLIIC [[?]] uses an adaptive learning gain to avoid the division by a small numbers. However, since the adaptive learning gain shuts down the learning for the frequency components dominated by the output disturbances, the tracking performance in steady-state is degraded.

1.3 Proposed Approach and Contributions

To overcome the difficulties mentioned above, a model-free learning algorithm is proposed. The proposed algorithm is based on the time-reversal based ILC, and we apply a data-based learning filter to update the control input every n iterations. This approach not only accelerates the convergence rate of time-reversal based ILC. Compared to the MFIIC and NLIIC approaches, our method provides a more flexible updating law that gains the robustness against output disturbances and the tracking performance in steady-state.

1.4 Thesis Overview

The remainder of this paper is organized as follows: Section 2 illustrates the basic iterative learning control algorithm and the existing model-free ILC approaches. Section 3 provides a derivation and analysis of the proposed model-free learning algorithm. Section 4 presents the simulation results to validate the effectiveness of the proposed algorithm. Finally, Section 5 summarizes the main results and contributions of this paper.

Chapter 2

Related Work

2.1 Repetitive Control and ILC

2.1.1 Repetitive Control

2.1.2 Iterative Learning Control

For a SISO, LTI system $G(z)$, which is represented in z-domain and assumed to be asymptotically stable, the ILC learning algorithm can be written in the following form:

$$\begin{aligned} u_{j+1}(k) &= u_j(k) + L(z)[r(k) - y_j(k)] \\ &= u_j(k) + L(z)e_j(k) \end{aligned} \tag{2.1}$$

where k is the time index of the discrete-time signal, the sub-index j represents the j th iteration of learning process, r is the trial-invariant reference to be tracked with time interval N , u represents the control input of $G(z)$, y is the output of $G(z)$, e is the tracking error to be reduced, and $L(z)$ is the learning filter to guarantee the stable learning.

Assumption 1. Since the z-transform and Fourier transformation of a time sequence is computed over an infinite time interval, all the time signals in this

paper are assumed to have infinite length (i.e. $N \rightarrow \infty$) to meet the requirement of frequency-domain analysis [[?]].

Assumption 1 is necessary for ILC analysing in z-domain since the convolution theorem in z-domain is hold iff the time-sequences are all with infinite length.

To analysis the ILC stability condition in frequency domain, the error propagation formula can be obtained from Eq. 2.1:

$$e_{j+1}(k) = [1 - L(z)G(z)]e_j(k) \quad (2.2)$$

By taking Fourier transform of Eq. 2.2, apparently, the tracking error will converge if:

$$\|I - G(z)L(z)\|_{\infty} < 1 \quad (2.3)$$

where $\|\cdot\|_{\infty}$ denotes \mathcal{H}_{∞} -norm. It had been proven when the learning filter $L(z)$ satisfies Eq.2.3, the control input u_{∞} and tracking error e_{∞} will converge and thus the learning is stable [[?]] .

Remark 1. To meet the infinite long time signal assumption, the zero-padding technique is applied to extend the reference signal with sufficient long zeros on the both ends to avoid leakage in frequency-domain computation.

2.1.3 Inverse Filter Design of RC and ILC

2.2 Inverse Filter Design Method for Repetitive Control and ILC

2.2.1 Zero Phase Error Tracking Control (ZPETC)

2.2.2 Unity Magnitude Error Tracking Control (UMETC)

2.2.3 Direct Inversion Method

2.2.4 Iterative Learning of Dynamic Inverse Filters (ILCFF)

2.3 Model-free ILC

To satisfy the stability criteria and the model-free requirement, there are several ways to implement the model-free learning law. Here the time-reversal based ILC and MFIIC approaches are introduced.

2.3.1 Model-free Adjoint-based ILC

Consider the adjoint-based ILC [[?]; [?]], the learning filter $L(z) = \alpha G^*(z)$ is chosen with a sufficiently small learning gain $\alpha > 0$ to guarantee the stability condition. The reason why the adjoint-based learning filter is capable to stabilize the learning is the phase effect of the system could be cancelled, hence the stable learning can be hold if the learning gain is properly selected.

To realize the adjoint-based learning law:

$$u_{j+1}(k) = u_j(k) + \alpha G^*(z) e_j(k) \quad (2.4)$$

the term $\alpha G^*(z) e_j(k)$ can be obtained by using the reversed time filtering technique as shown in [[?]]:

1. Reverse the error signal: $e_{j1}(k) = e_j(N - k)$;
2. Feed the reversed error signal to the system: $e_{j2}(k) = G(z)e_{j1}(k)$;
3. Reverse $e_{j2}(k)$ again, and multiply with α : $e_{j3}(k) = \alpha e_{j2}(N - k)$
4. Conduct the learning law: $u_{j+1}(k) = u_j(k) + e_{j3}(k)$;

For the adjoint-based learning approach, the robustness can be improved compared with the inversion-based approach, especially considering the effect of high frequency model uncertainty [[?]]. However, since the convergence rate is limited by the small learning gain, this disadvantage leads to a critical issue of the time-consuming learning process in real application.

2.3.2 Model-free Inversion-based Iterative Control (MFIIC)

Consider the SISO, LTI and stable system $y_j(k) = G(z)u_j(k)$, the model-free inversion-based learning law can be written as:

$$U_{j+1}(e^{j\omega}) = \begin{cases} U_j(e^{j\omega}) + \rho_j \frac{U_j(e^{j\omega})}{Y_j(e^{j\omega})} E_j(e^{j\omega}), \\ \text{if } Y_j(e^{j\omega}) \neq 0 \text{ and } R(e^{j\omega}) \neq 0; \\ U_j(e^{j\omega}); \text{ otherwise} \end{cases} \quad (2.5)$$

where $U_j(e^{j\omega})$ and $Y_j(e^{j\omega})$ is the frequency-domain representation of $u_j(k)$ and $y_j(k)$ respectively; and the learning gain ρ_j is equal to 1 for MFIIC [[?]] and $\rho_j = f(|Y_j(e^{j\omega})|)$ is a function of $|Y_j(e^{j\omega})|$ for NLIIC [[?]] to improve the robustness.

For this approach, the updating computation in frequency-domain is relative simple and time-efficient compared to the reversed time filtering approach. Moreover, since the estimated learning filter is inversion-based, the learning will converge after few iterations. However, due to the learning filter is constructed by measured data, while the unpredicted disturbances dominate the output signal,

the updated control input may be dramatically amplified in some frequency components. This observation had been indicated in [[?]]. Moreover, since the noisy learning filter is conducted iteration by iteration over the whole learning process, while the adaptive learning gain is introduced [[?]], the tracking performance in steady-state is limited.

2.3.3 Non-Linear Inversion-based Iterative Control (NLIIC)

2.3.4 Comparison of existed model-free ILC

In this paper, we propose a novel method to remedy the difficulties for the model-free ILC mentioned above. Specifically, the proposed method is expected to

1. Improve the learning transient against output disturbances;
2. Achieve lower tracking error in steady-state.

Chapter 3

Model-free ILC with Recursive Convergence Acceleration

The main idea of the proposed algorithm is to formulate the ILC recursive equation (Eq. 2.1) into a closed-form, and a “ recursive convergence rate acceleration equation ” is observed. By using such the equation, the learning rate of any stable learning law can be further accelerated.

Moreover, since the updating computation avoids the division by noisy output signals ,the unpredictable learning transient is reduced naturally, and the tracking performance in steady-state can also be improved by avoiding the use of the noisy data-based learning filter over the whole learning procedure.

The derivation of the proposed algorithm will be discussed in section 3.1, and the stability and robustness of the proposed algorithm will be analysed in section 3.2 and 3.3.

3.1 Algorithm Formulation

Since the ILC learning law is a recurrence relation, given the initial term $u_0 = L(z)r(k)$, Eq. 2.1 can be reformulated as the following form:

$$u_{j-1}(k) = \frac{1 - [1 - L(z)G(z)]^j}{G(z)} r(k) \quad (3.1)$$

$$e_{j-1}(k) = [1 - G(z)L(z)]^j r(k) \quad (3.2)$$

As mentioned in section 2, all the discrete time signals are assumed to be with infinity length, thus the time-domain signal $x(k)$ can be transformed into the z-domain representation:

$$X(z) := \sum_{k=-\infty}^{\infty} x(k)z^{-k} \quad (3.3)$$

As a result, Eq. 3.1 and Eq. 3.2 can be rewritten as:

$$U_{j-1}(z) = \frac{1 - [1 - L(z)G(z)]^j}{G(z)} R(z) \quad (3.4)$$

$$E_{j-1}(z) = [1 - G(z)L(z)]^j R(z) \quad (3.5)$$

where $U_{j-1}(z)$ denotes the z-domain representation of $u_{j-1}(k)$, and $R(z)$ and $E_{j-1}(z)$ are as so on. Inspired by the recursive equation, if the sub-index j is replaced with $2j$, then Eq. 3.4 becomes

$$U_{2j-1}(z) = \frac{1 - [1 - L(z)G(z)]^{2j}}{G(z)} R(z) \quad (3.6)$$

Combine Eq. 3.5 and Eq. 3.6,

$$U_{2j-1}(z) = \frac{1 - [\frac{E_{j-1}(z)}{R(z)}]^2}{G(z)} R(z) \quad (3.7)$$

$$= \frac{1 - [\frac{R(z) - G(z)U_{j-1}(z)}{R(z)}]^2}{G(z)} R(z) \quad (3.8)$$

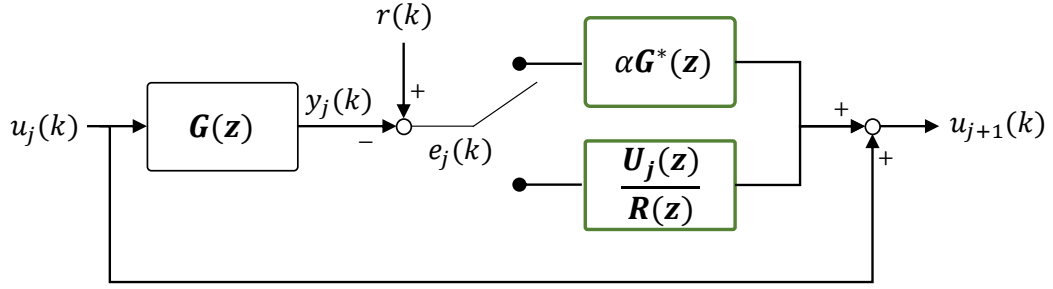


Figure 3.1: The control architecture of the proposed algorithm. The learning is switched to the acceleration filter $\frac{U_j(z)}{R(z)}$ for every $n \in \mathbb{N}$ iterations; otherwise, the control input is updated by the time-reversal based ILC.

$$= \frac{R(z) - \frac{[R(z) - G(z)U_{j-1}(z)]^2}{R(z)}}{G(z)} \quad (3.9)$$

$$= \frac{R(z)^2 - [R(z)^2 - 2G(z)U_{j-1}(z)R(z) + G^2(z)U_{j-1}^2(z)]}{R(z)G(z)} \quad (3.10)$$

$$= \frac{2G(z)U_{j-1}(z)R(z) - G^2(z)U_{j-1}^2(z)}{R(z)G(z)} \quad (3.11)$$

$$= \frac{2U_{j-1}(z)R(z) - G(z)U_{j-1}^2(z)}{R(z)} \quad (3.12)$$

$$= 2U_{j-1}(z) - \frac{U_{j-1}(z)}{R(z)}G(z)U_{j-1}(z) \quad (3.13)$$

$$= 2U_{j-1}(z) - \frac{U_{j-1}(z)}{R(z)}Y_{j-1}(z) \quad (3.14)$$

Since the tracking error is defined as:

$$E_{j-1}(z) := R(z) - Y_{j-1}(z) \quad (3.15)$$

hence

$$Y_{j-1}(z) = R(z) - E_{j-1}(z) \quad (3.16)$$

Combine Eq. 3.14 and Eq. 3.16, then

$$\begin{aligned} U_{2j-1}(z) &= 2U_{j-1}(z) - \frac{U_{j-1}(z)}{R(z)}Y_{j-1}(z) \\ &= 2U_{j-1}(z) - \frac{U_{j-1}(z)}{R(z)}(R(z) - E_{j-1}(z)) \\ &= U_{j-1}(z) + \frac{U_{j-1}(z)}{R(z)}(E_{j-1}(z)) \end{aligned} \quad (3.17)$$

therefore, Eq. 3.14 can be reformulated into the ILC updated form, and the following “recursive convergence rate acceleration equation ” is obtained:

$$U_{2j-1}(z) = U_{j-1}(z) + \frac{U_{j-1}(z)}{R(z)}E_{j-1}(z) \quad (3.18)$$

Eq. 3.18 implies if the learning filter $\frac{U_{j-1}(z)}{R(z)}$ is assigned, the learning rate will be dramatically improved. For example, consider the scenario that $u_{20}(k)$ is the learned control input obtained by using the initial learning filter $L_0(z)$ to conduct the ILC at 20th iteration, while the learning filter is replaced with $\frac{U_{20}(z)}{R(z)}$ at 21th iteration, the updated control input becomes $u_{41}(k)$ learned from $L_0(z)$. Therefore, the learning becomes more time-efficient.

To implement this observation, consider the SISO, LTI and pre-stabilized system $G(z)$, and the initial learning filter $L_0(z)$ is chosen to satisfy the stability criteria (Eq. 2.3). The proposed control input updating approach utilizes Eq. 3.18 to update the control input every $n \in \mathbb{N}$ iterations to accelerate the convergence rate,

and the rest iterations update the control input via $L_0(z)$. The control architecture is as shown in Fig. 3.1, and the procedure of the proposed algorithm is summarized below:

Algorithm 2 Model-free ILC with Recursive Convergence Acceleration

- 1: Initialize $j = 0$, $u_0(k) = r(k)$, $L_0(z) = \alpha G^*(z)$.
- 2: Feed $u_j(k)$ to the system $G(z)$ to obtain the output data $y_j(k)$; Record the tracking error $e_j(k) = r(k) - y_j(k)$; If tracking error is small enough within the tolerance, then stop the learning; otherwise, go to the next step.
- 3: If $j = 0$ or $(j \bmod n) \neq 0$, go to step 4; otherwise, go to step 5;
- 4: Conduct the time-reversal based ILC:

$$u_{j+1}(k) = u_j(k) + \alpha G^*(z) e_j(k) \quad (3.19)$$

Set $j \leftarrow j + 1$, go to step 2;

- 5: Apply DFT to obtain the frequency-domain signal $R(e^{j\omega})$, $U_j(e^{j\omega})$ and $E_j(e^{j\omega})$;
- 6: Conduct the proposed updating law (Eq. 3.18) over the discrete-frequency interval:

$$U_{j+1}(e^{j\omega}) = \begin{cases} U_j(e^{j\omega}) + \frac{U_j(e^{j\omega})}{R(e^{j\omega})} E_j(e^{j\omega}), \\ \text{if } |R(e^{j\omega})| > \text{threshold}; \\ U_j(e^{j\omega}); \text{ otherwise} \end{cases} \quad (3.20)$$

Do inverse DFT to obtain the time-domain sequence $u_{j+1}(k)$; Set $j \leftarrow j + 1$, go to step 2;

Note that the learning threshold of Eq. 3.20 must be set if the output disturbances are considered. How to set the learning threshold to maintain the stable learning will be discussed and analysed in Section 3.3.

3.2 Stability Analysis

To prove the proposed ILC algorithm is stable, some assumptions and definitions are stated below:

Definition 1. (Error Convergence Rate) Consider the stable system $G(z)$ and the learning filter $L(z)$, the error convergence rate of ILC is defined as:

$$\gamma := \|I - G(z)L(z)\|_{\infty} \quad (3.21)$$

The smaller the value of γ is, the faster the tracking error converges.

Definition 2. (Symbol Convention) The learning filter updated at each pn iteration is denoted as $L_{pn}(z)$, where $p \in \mathbb{N}$ is the number of updating times.

Assumption 2. The initial learning filter $L_0(z)$ is stable, i.e.

$$\gamma_0 := \|I - G(z)L_0(z)\|_{\infty} < 1 \quad (3.22)$$

Theorem 1. (Stability of Algorithm 1.)

Assume the learning filter conducted at each pn iteration by algorithm 1 is:

$$L_{pn}(z) = \frac{U_{pn}(z)}{R(z)} \quad (3.23)$$

where $U_{pn}(z)$ is the control input applying algorithm 1 at pn iteration in z -domain representation, $p, n \in \mathbb{N}$, then:

$$\gamma_{pn} := \|I - G(z)L_{pn}(z)\|_{\infty} \leq (\gamma_0)^{2^{p-1}(n+1)} < 1 \quad (3.24)$$

Proof:

According to Eq. 3.4 and Eq. 3.18, the control input conducted by algorithm 1 at each pn iteration can be written as:

$$U_{pn}(z) = \frac{1 - [1 - L_0(z)G(z)]^{(2^p-1)n+2^{p-1}}}{G(z)} R(z) \quad (3.25)$$

Hence the learning filter $L_{pn}(z)$ becomes:

$$L_{pn}(z) = \frac{U_{pn}(z)}{R(z)} = \frac{1 - [1 - L_0(z)G(z)]^{(2^p-1)n+2^{p-1}}}{G(z)} \quad (3.26)$$

Combine Eq. 2.3 and Eq. 3.26, the stability criteria becomes:

$$\gamma_{pn} := \|I - G(z)L_{pn}(z)\|_{\infty} \quad (3.27)$$

$$= \|(I - G(z)L_0(z))^{(2^p-1)n+2^{p-1}}\|_{\infty} \quad (3.28)$$

$$\leq \|I - G(z)L_0(z)\|_{\infty}^{(2^p-1)n+2^{p-1}} \quad (3.29)$$

Since $2^p - 1 \geq 2^{p-1}$ for $p \in \mathbb{N}$, therefore

$$(2^p - 1)n + 2^{p-1} \geq 2^{p-1}n + 2^{p-1} = 2^{p-1}(n + 1) \quad (3.30)$$

From Eq. 3.29 and Eq. 3.30,

$$\gamma_{pn} \leq (\gamma_0)^{(2^p-1)n+2^{p-1}} \leq (\gamma_0)^{2^{p-1}(n+1)} < 1 \quad (3.31)$$

■

Theorem 1 implies if the pre-selected learning filter $L_0(z)$ is stable, then the acceleration learning filter $L_{pn}(z)$ not only satisfies the stability criteria, but also accelerates the learning rate exponentially at each pn iteration, $p, n \in \mathbb{N}$.

3.3 Robustness Analysis

To guarantee the tracking error conducted by the proposed learning algorithm converges monotonically when the output disturbances, such as measurement noise and non-linear dynamics are considered. Assume the system dynamics of a stable $G(z)$ becomes:

$$y_j(k) = G(z)u_j(k) + d(k) \quad (3.32)$$

where $d(k)$ is the output disturbance.

Assumption 3. The output disturbance $d(k)$ is trial-invariant, and can be represented in z-domain and frequency-domain as $D(z)$ and $D(e^{j\omega})$ respectively. $|D(e^{j\omega})|$ is bounded by δ , i.e., $\delta := \max(|D(e^{j\omega})|) \forall \omega \in (-\pi, \pi)$.

Theorem 2. (Learning threshold of Algorithm 1.) Consider the system dynamics of SISO, LTI system $G(z)$ in frequency-domain: $Y_j(e^{j\omega}) = G(e^{j\omega})U_j(e^{j\omega}) + D(e^{j\omega})$, where $|D(e^{j\omega})|$ is bounded by δ , i.e., $\delta := \max(|D(e^{j\omega})|) \forall \omega \in (-\pi, \pi)$. The necessary condition for the monotonic convergence of the learning law Eq. 3.20 is

$$|R(e^{j\omega})| > |E_j(e^{j\omega})| + \delta \quad (3.33)$$

where $R(e^{j\omega})$ is the reference trajectory and $E_j(e^{j\omega}) := R(e^{j\omega}) - Y_j(e^{j\omega})$ is the tracking error.

Proof:

Since the tracking error at j th iteration is:

$$E_j(e^{j\omega}) = R(e^{j\omega}) - G(e^{j\omega})U_j(e^{j\omega}) - D(e^{j\omega}) \quad (3.34)$$

the tracking error at $(j + 1)$ th iteration conducted by the learning law Eq. 3.20 becomes:

$$E_{j+1}(e^{j\omega}) = R(e^{j\omega}) - G(e^{j\omega})U_{j+1}(e^{j\omega}) - D(e^{j\omega}) \quad (3.35)$$

$$= R(e^{j\omega}) - G(e^{j\omega})(U_j(e^{j\omega}) + \frac{U_j(e^{j\omega})}{R(e^{j\omega})}E_j(e^{j\omega})) - D(e^{j\omega}) \quad (3.36)$$

$$= (1 - \frac{GU_j(e^{j\omega})}{R(e^{j\omega})})E_j(e^{j\omega}) \quad (3.37)$$

$$= [\frac{E_j(e^{j\omega}) + D(e^{j\omega})}{R(e^{j\omega})}]E_j(e^{j\omega}) \quad (3.38)$$

Define

$$\kappa_j := \frac{E_j(e^{j\omega}) + D(e^{j\omega})}{R(e^{j\omega})} \quad (3.39)$$

then

$$E_{j+1}(e^{j\omega}) = \kappa_j E_j(e^{j\omega}) \quad (3.40)$$

Therefore,

$$|E_{j+1}(e^{j\omega})| \leq |\kappa_j| |E_j(e^{j\omega})| \quad (3.41)$$

To guarantee the tracking error is monotonically decaying, the following condition must be satisfied:

$$|\kappa_j| := \left| \frac{E_j(e^{j\omega}) + D(e^{j\omega})}{R(e^{j\omega})} \right| < 1 \quad (3.42)$$

Hence the necessary condition for the monotonic convergence of learning law Eq. 3.20 is:

$$|R(e^{j\omega})| > |E_j(e^{j\omega}) + D(e^{j\omega})| \quad (3.43)$$

Since

$$|E_j(e^{j\omega})| + \delta \geq |E_j(e^{j\omega})| + |D(e^{j\omega})| \geq |E_j(e^{j\omega}) + D(e^{j\omega})| \quad (3.44)$$

If the NSR condition Eq. 3.33 is hold, the monotonic convergence of updating law Eq. 3.20 is guaranteed. ■

From *Theorem 2*, the learning threshold to maintain the stable learning at each pn iteration can be derived. *Theorem 2* implies if the learning threshold (Eq. 3.33) is set, the tracking will be shut down when the frequency components do not reach the threshold. However, since the rest iterations use time-reversal based ILC to reduce the tracking error, these frequency components can still be learned. Such a dual-mode learning not only provides a more robust learning algorithm than MFIIC, but also achieves better steady-state tracking performance compared to [[?]].

3.4 Simulation Result

3.4.1 Simulation Setup

To verify the effectiveness of the proposed algorithm, a numerical example of the system $G(z)$ is performed. As mentioned previously, $G(z)$ is pre-stabilized with

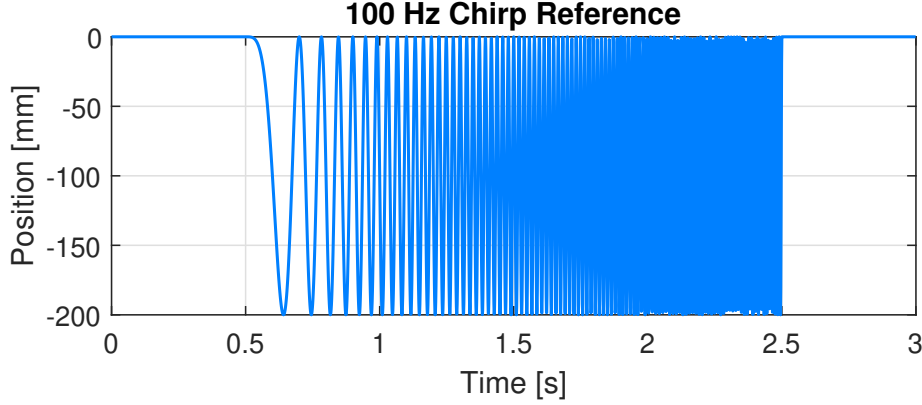


Figure 3.2: The chirp reference with the amplitude of 100 mm and a maximum frequency component of 100 Hz.

the closed-loop feedback control structure. For example, assume the plant $P(z)$ is stabilized with the feedback controller $C(z)$ at sampling frequency 1 kHz, where:

$$P(z) = \frac{0.01(z - 0.9)}{(z - 1)^2}; C(z) = \frac{(20z - 19.9)}{z} \quad (3.45)$$

then the system applied in the simulation $G(z)$ is

$$G(z) = \frac{C(z)P(z)}{1 + C(z)P(z)} \quad (3.46)$$

The chirp signal (linear swept-frequency cosine) is chosen as the reference trajectory in the simulation. Here the chirp trajectory with the amplitude of 100 mm and a maximum frequency component of 100 Hz is applied as the illustrated example in this paper, as shown in Fig. 3.2. Note that to reduce the influence of the frequency leakage as pointed out in Sec. ??, the chirp reference is padded with sufficient long zeros on the both ends.

To compare the performance of each model-free ILC, there are 4 cases are performed, each case is performed the updating law for 1000 iterations, and all the initial control inputs are set as $u_0(k) = r(k)$. The first case is to conduct the time-reversal based ILC, where $L_0(z) = \alpha G^*(z)$, $\alpha = 0.1$ is chosen; the second case is to conduct the MFIIC; the third case is to apply the NLIIC [[?]] to enhance the robustness of MFIIC; and the fourth case is to perform the proposed algorithm,

where the initial learning filter $L_0(z)$ is same as the case 1, and the updating period $n = 5$, i.e., the acceleration updating occurs at 5th, 10th,..., 1000th iterations.

3.4.2 Ideal Case Without Output Disturbance

Consider the ideal case, where the output disturbance is neglected in this section. The only factor impacts the unideal learning for the disturbance-free case in the simulation is the frequency-leakage problem as pointed out in section ??.

Fig. 3.3(a) shows the error convergence of each learning law with no output disturbance. From the simulation result, it can be observed the convergence rate of time-reversal based ILC is the slowest. The convergence rate of the proposed learning law is accelerated every 5 iterations compared with time-reversal based approach, hence the learning rate is further improved. Although the convergence rate of MFIIC is faster than time-reversal based ILC, however, the unpredictable learning transient may appear in some iterations. NLIIC solve the robustness issue of MFIIC, however, the tracking is limited by the adaptive learning gain, hence the tracking performance in steady-state is much worse than MFIIC and proposed algorithm.

On the other hand, although the convergence rate of the proposed learning law is slower than MFIIC and NLIIC, however, the learning transient is improved by avoiding the noisy signal division. Moreover, the proposed algorithm provide better tracking performance in steady-state compared to NLIIC. From Fig. 3.5(a), it can be observed the NLIIC is suffered from the influence of frequency leakage problem. The dual-mode learning of the proposed algorithm further reduces the tracking error caused by the frequency leakage, hence the steady-state performance is improved. The RMS error and maximum error of each learning law at the 1000th iteration with no output disturbance is shown in Table 3.1.

Table 3.1: The ILC tracking error obtained from different methods

| Method | No Disturbance | | With Disturbance | |
|---------------|----------------|----------|------------------|------|
| | RMSE | MaxE | RMSE | MaxE |
| | [mm] | [mm] | [mm] | [mm] |
| Time-reversal | 2.33 | 6.13 | 2.54 | 8.15 |
| MFIIC | 1.64e-06 | 1.75e-06 | 1.24 | 9.61 |
| NLIIC | 0.00410 | 0.0339 | 1.22 | 13.8 |
| Proposed | 6.25e-06 | 1.64e-05 | 1.12 | 4.52 |

3.4.3 Output Tracking with Trial-variant Disturbances

Consider the trial-variant normally distributed random time sequences (the MATLAB function: randn) are applied as the output disturbances, the tracking result of each model-free learning law is shown in this section. Note that to reduce the influence of the output disturbances in error convergence, both the NSR conditions of MFIIC and NLIIC are set as:

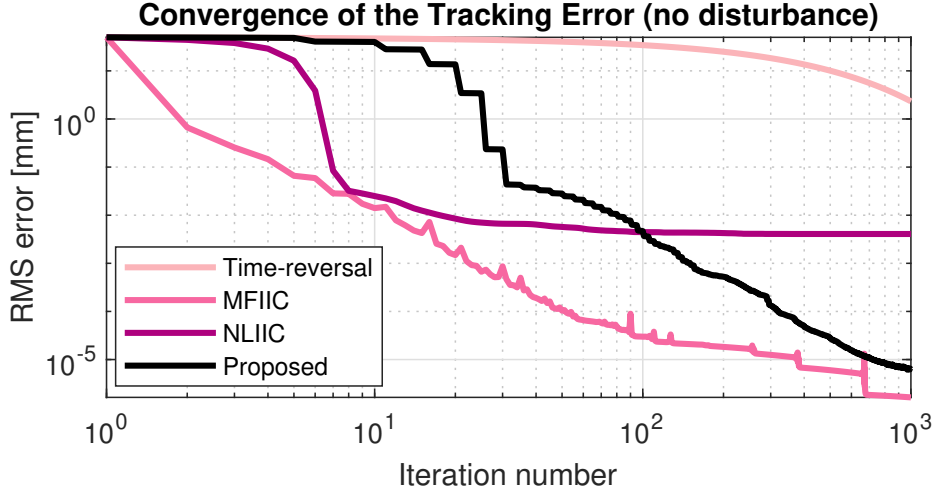
$$U_{j+1}(e^{j\omega}) = \begin{cases} U_j(e^{j\omega}) + \rho_j \frac{U_j(e^{j\omega})}{Y_j(e^{j\omega})} E_j(e^{j\omega}), \\ \text{if } |Y_j(e^{j\omega})| \neq 0 \text{ and } |R(e^{j\omega})| > \delta; \\ U_j(e^{j\omega}); \text{ otherwise} \end{cases} \quad (3.47)$$

Fig. 3.3(b) shows the error convergence of each learning law with the output disturbances, where the standard deviations of the disturbances are 1 mm and the mean values are 0 mm (i.e., $d(k) \in N(0, 1)$). From the simulation results, it can be seen the convergence rate of the proposed algorithm is dramatically improved compared with time-reversal based ILC, and the learning transient is also reduced, hence the robustness is improved. Moreover, the tracking performance at the 1000th iteration of the proposed algorithm is supreme, as shown in the right column of the Table 3.1.

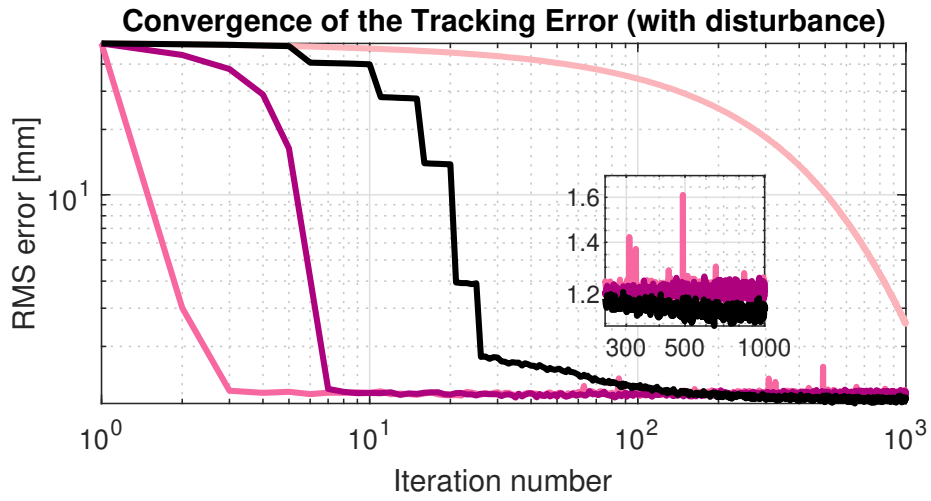
To show the robustness and the steady-state performance of the proposed algorithm outperforms that of the MFIIC and NLIIC, consider the extreme situation that

the output disturbances with the standard deviations of 80 mm and the mean values of 0 mm (i.e., $d(k) \in N(0, 80)$) are applied to track the 100 mm chirp reference, as shown in Fig. 3.4. Although NLIIC provides a smoother learning curve than MFIIC approach, however, the tracking error doesn't converge to a lower level. On the other hand, since the proposed algorithm is based on time-reversal based ILC, the robustness and the limited steady-state tracking performance problems can be remedied.

From Fig. 3.5(b), it can be observed the maximum tracking error of MFIIC and NLIIC occur at around 2.5 sec, i.e., the high-frequency components of the chirp reference. The proposed algorithm not only further reduces the tracking error in high-frequency, but also provides a more robust and flexible updating law compared with MFIIC and NLIIC, hence the tracking performance is improved if the output disturbances are considered.



(a)



(b)

Figure 3.3: The convergence of the ILC tracking error. (a) The case with no output disturbance. (b) The case with the output disturbance $d(k) \in N(0, 1)$.

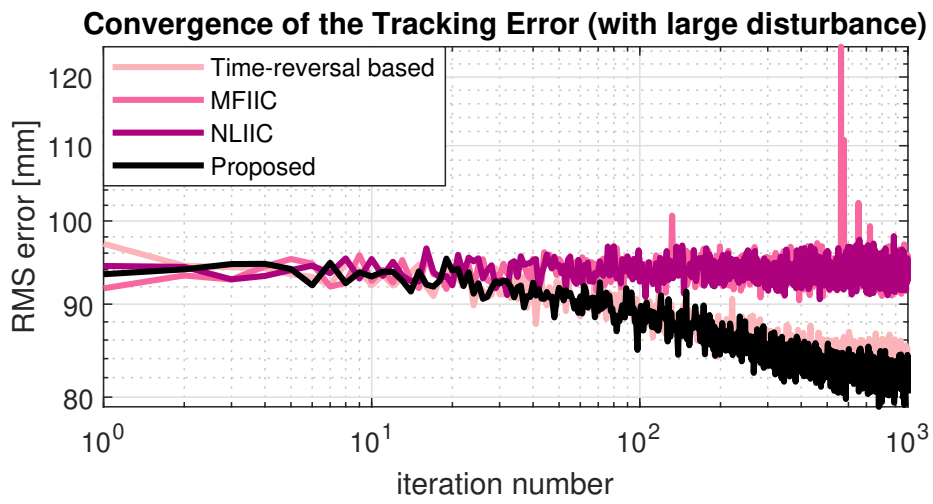
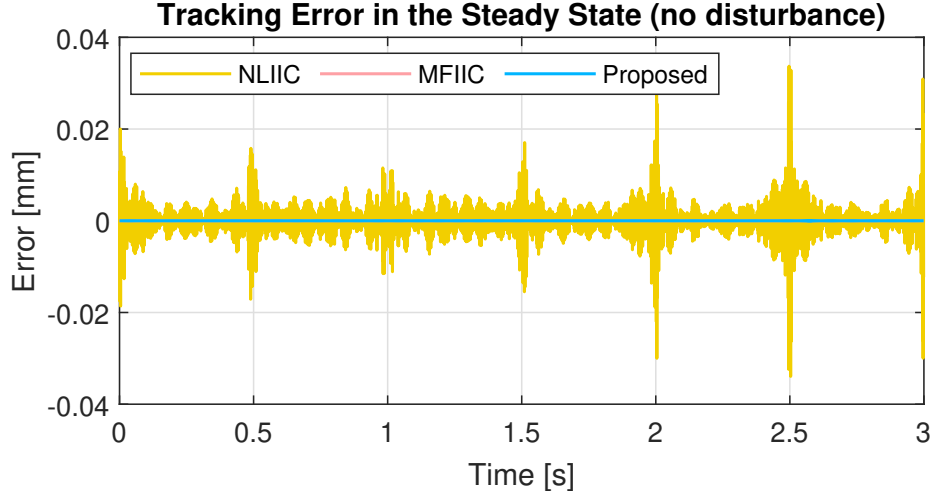


Figure 3.4: The convergence of the ILC tracking error when the output disturbance $d(k) \in N(0, 80)$. Both MFIIC and NLIIC do not converge.



(a)



(b)

Figure 3.5: The ILC tracking error in steady-state. (a) The case with no output disturbance. (b) The case with the output disturbance $d(k) \in N(0, 1)$.

Chapter 4

Case Study 1: Triangular Wave Tracking of Galvo Scanner

4.1 System Overview

4.2 Repetitive Controller Design

4.3 Experiment Setup

4.4 Experiment Result

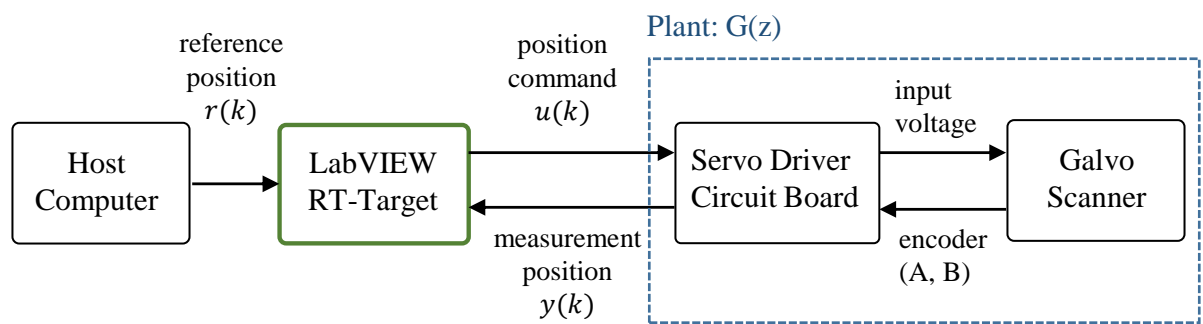


Figure 4.1: The schematic diagram of the galvo scanner system.

Reference

- [1] M. Tomizuka, “Zero phase error tracking algorithm for digital control,” 1987. [1](#)
- [2] R. Ramesh, M. Mannan, and A. Poo, “Tracking and contour error control in cnc servo systems,” *International Journal of Machine Tools and Manufacture*, vol. 45, no. 3, pp. 301–326, 2005. [1](#), [2](#)
- [3] T. Oomen, “Learning for advanced motion control,” *arXiv preprint arXiv:2004.11017*, 2020. [3](#)