Model-Free Repetitive Control and Iterative Learning Control

Ching Hsiang, Wu[1] and Lucas Herfurth[2]

[1] Dept. of Electrical Engineering
[2]
National Taiwan University, Taipei 106319, Taiwan

Abstract. **Keywords:** First keyword · Second keyword · Another keyword.

1 Introduction

2 FIR Inverse Filter

In Lab02, we have derived the inverse filter $F(z)$ by zero-phase error tracking controller (ZPETC)[1]. However, ZPETC is model-based so the tracking result will be enormously affected by the model accuracy. Here, we apply time-reversal-based ILC to obtain a purely data-based inverse filter.

2.1 Data-based Inverse filter Introdution

The basic idea of the data-based inverse filter by ILC can borrow the concept of the feedforward control. Fig.1 shows the block diagram of feedforward control.

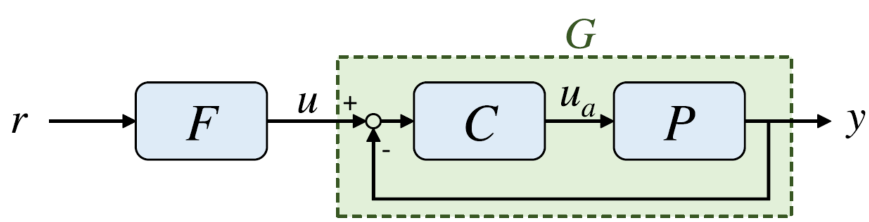


Fig. 1: Block diagram of feedforward control [1]

Where, r is reference, u is control input, y is output, F is a feedforward controller and G is the closed-loop stable galvanometer scanner system. Assuming that F is a perfect inversion plant of G and r is an impulse function $\delta(t)$, then y should also be $\delta(t)$. Intuitively, we can find u to track an impulse function $\delta(t)$ perfectly by designing an ILC algorithm. Once we obtain the control input u, which is also the impulse response of the inversion plant F, we can easily construct the inverse filter F by Eqn.1

$$F(z) = \sum_{k=-N/2}^{N/2} u(k) z^{-k} \tag{1}$$

Note that the inversion plant is a high pass filter for the galvanometer scanner system, which will induce the high-gain input and cause input saturation. To avoid the input saturation, we apply a lowpass filter to the reference impulse signal[1]

$$r_M(k) = M(z) \cdot \delta(k) \tag{2}$$

A generic ILC algorithm can be written as

$$u_{j+1}(k) = u_j(k) + L(z) e_j(k) \tag{3}$$

Define $e_j(k) = r(k) - y_j(k)$. To construct a learning filter without using the model of the system, we adopt time-reversal-based ILC. We let the learning filter $L(z)$ equal

$$L(z) = \alpha \cdot G^*(z) \tag{4}$$

Where $G^*(z)$ is conjugate of $G(z)$. Next, we substitute Eqn.4 into Eqn.3

$$\begin{aligned} u_{j+1}(k) &= u_j(k) + L(z) e_j(k) && (5) \\ &= u_j(k) + \alpha \cdot G^*(z) e_j(k) && (6) \end{aligned}$$

In the time domain, we can represent $G^*(z)e_j(k)$ as the following diagram Where ”ïs a reverse

Fig. 2: Time domain of $G^*(z)e_j(k)$

operator. So far, we have no problem obtaining the inverse filter $F(z)$.

In this report, we first use *”ref_imp.m”* provided by Prof. to generate the reference impulse response and then consider 2 ways to obtain the converged control input $u(k)$. The first one is simply injecting the reference impulse into *”GS_TR.vi”* and collecting the input data. The second one is we construct the time-reversal-based ILC in the simulation environment and the estimated system is from Lab01. For comparison purposes, the cutoff frequency of the lowpass filter is adopted as 2000Hz and 4000Hz.

The other design parameters are listed in Table.1. We define i as the learning iterations.

2.2 Data-based Inverse filter Results

In this section, we show the magnitude of the bodeplot of 4 cases in Fig.3a

1. F_{sim} : $2k$: 2000 Hz lowpass filter and the data from the simulation.

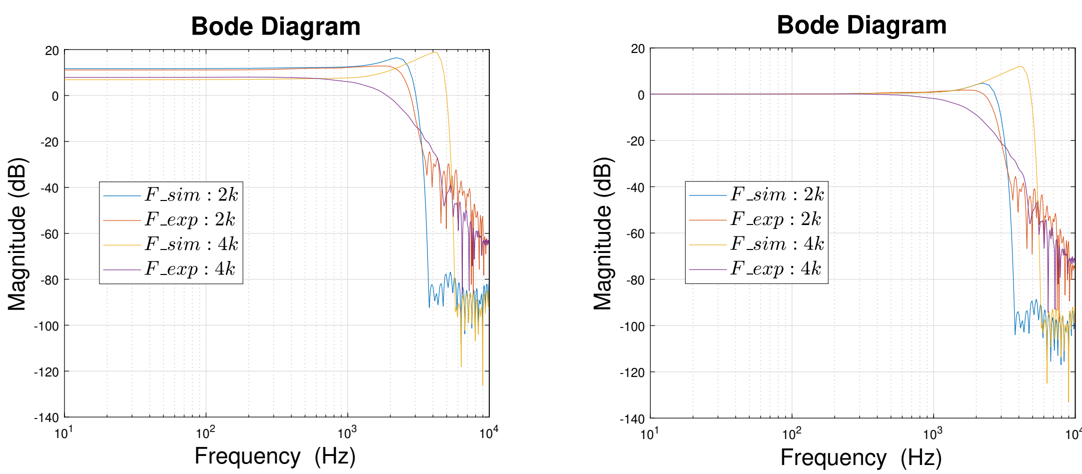
Parameter	Value
N	47
α	0.5
i	100

Table 1: Parameters

2. $F_{exp}: 2k$: 2000 Hz lowpass filter and the data from the experiment.
3. $F_{sim}: 4k$: 4000 Hz lowpass filter and the data from the simulation.
4. $F_{exp}: 4k$: 4000 Hz lowpass filter and the data from the experiment.

We can find out two things from this bodeplot. The first is the magnitude of $F_{exp}: 4k$ doesn't drop at around 4K Hz. The second is the DC gain of these inverse filters is not equal to 1, which will amplify the signal in the low-frequency region. To solve this problem we use Eqn.7 to normalize the DC gain to 1 and the result is shown in Fig.3b.

$$F = \frac{F}{\mathrm{dcgain}(F)} \tag{7}$$



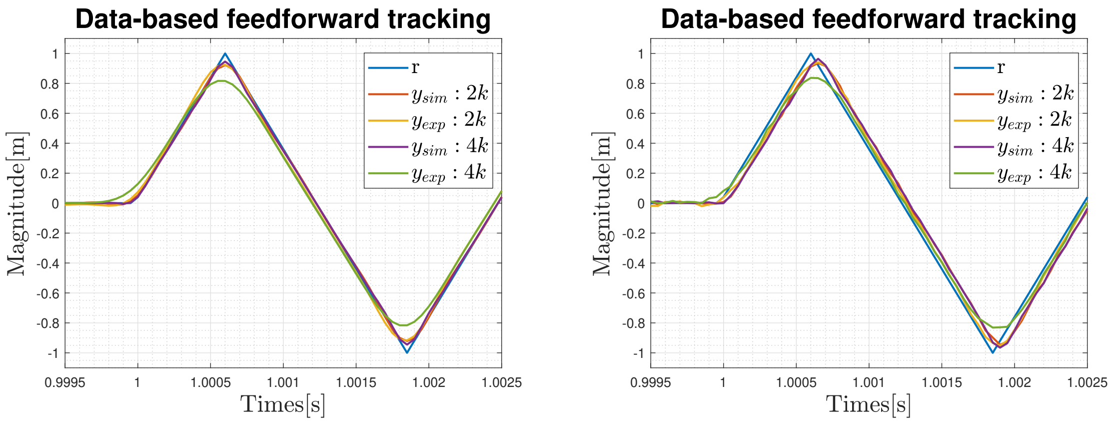
(a) 4 cases of inverse filter

(b) 4 cases of inverse filter: normalized

Fig. 3: Bode plot of data-based inverse filter

3 Data-Based Feedforward

The simulation and experiment data-based feedforward tracking are shown in Fig.4. From Fig.3,



(a) Data-based feedforward tracking: simualtion (b) Data-based feedforward tracking: experiment

Fig. 4: Data-based feedforward tracking results

we know that the magnitude of $F_{exp}:4k$ has an unexpected early drop before 4k Hz. That's why the tracking performance of using $F_{exp}:4k$ is the worst at the triangular peak. In comparison with $F_{exp}:4k$, $F_{sim}:4k$ has the best performance at the peak of the triangular wave, and it makes sense from observing the bodeplot of $F_{sim}:4k$ because it has the widest bandwidth. On the other hand, the tracking result from the experiments has a phase delay compared to the one from the simulation.

4 Model-Free Iterative Learning Control

We apply the data-based inverse filter F as the learning filter to the ILC and run 10 iterations. The experimental model-free ILC tracking results are shown in Fig.5 We also consider four cases and plot out the result separately after iterations 0, 1, and 7. All cases gain huge progress after 1 iteration and after 7 iterations, the output almost aligns with the reference. Fig.5e shows the tracking result of 4 cases after 7 iterations. Just like the data-based feedforward tracking, $F_{sim}:4k$ outperforms other cases in the peak of the triangular wave and $F_{exp}:4k$ has the worst performance. Also, this can be illustrated from Fig.3

5 Model-Free Repetitive Control

6 Discussion

time domain data check

Control saturatio
4k too large

1. (FIR invserse filter)quick drop of $F_exp:4k$, not unity gain at low frequency? Fig.3
2. (Data-based FF)phase shift problem of data-based feedforward. Fig.7
3. (Model-free RC) simulation failed, but the experiment works Fig.6 ,Fig.8

obtain

phd

truncation problem

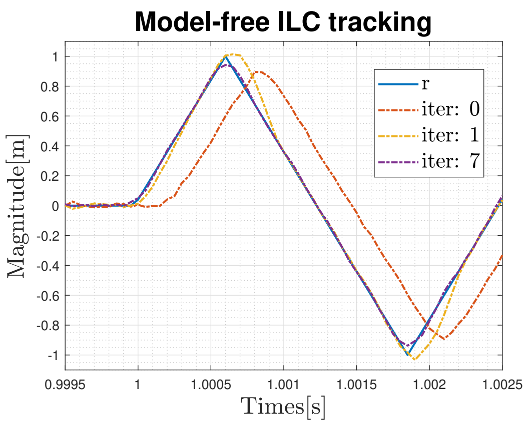
signal noise ratio

programing problem

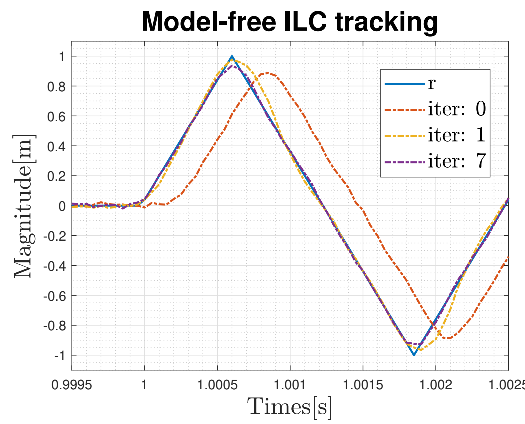
feedforward tracking
delay count
impulse

References

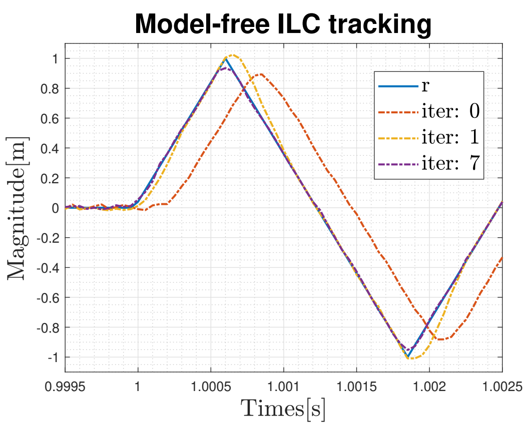
1. C. W. Chen, S. Rai, and T. C. Tsao, "Iterative Learning of Dynamic Inverse Filters for Feedforward Tracking Control," IEEE/ASME Transactions on Mechatronics, 25(1),pp. 349-359, 2020.
2. L. W. Shih and C. W. Chen, "Model-Free Repetitive Control Design and Implementation for Dynamical Galvanometer-Based Raster Scanning," Control Engineering Practice, 122, p. 105124, 2022.



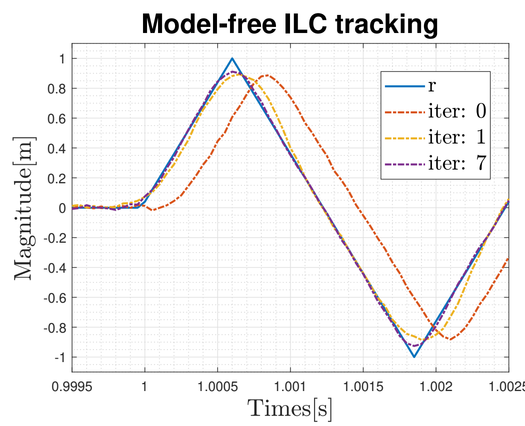
(a) Model-free ILC tracking: $F_{sim}:2k$



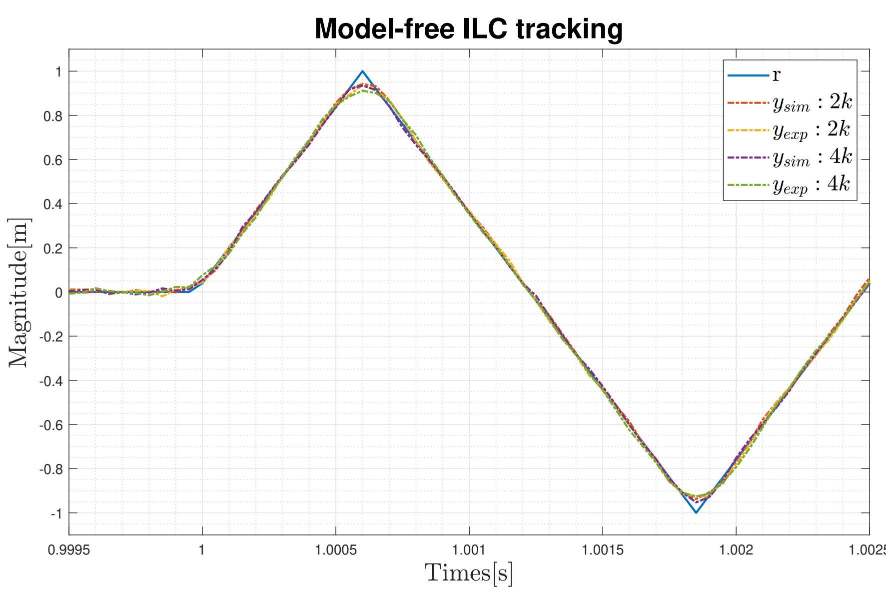
(b) Model-free ILC tracking: $F_{exp}:2k$



(c) Model-free ILC tracking: $F_{sim}:4k$

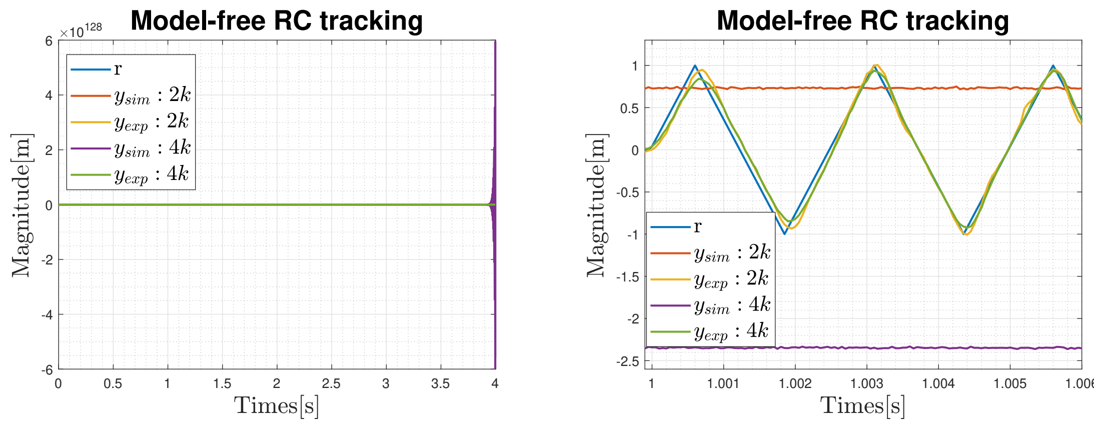


(d) Model-free ILC tracking: $F_{exp}:4k$



(e) Model-free ILC tracking: 7 iterations

Fig. 5: Model-free ILC tracking



(a) Model-free RC tracking: simualtion (b) Model-free RC tracking: experiment

Fig. 6: Model-free RC tracking results

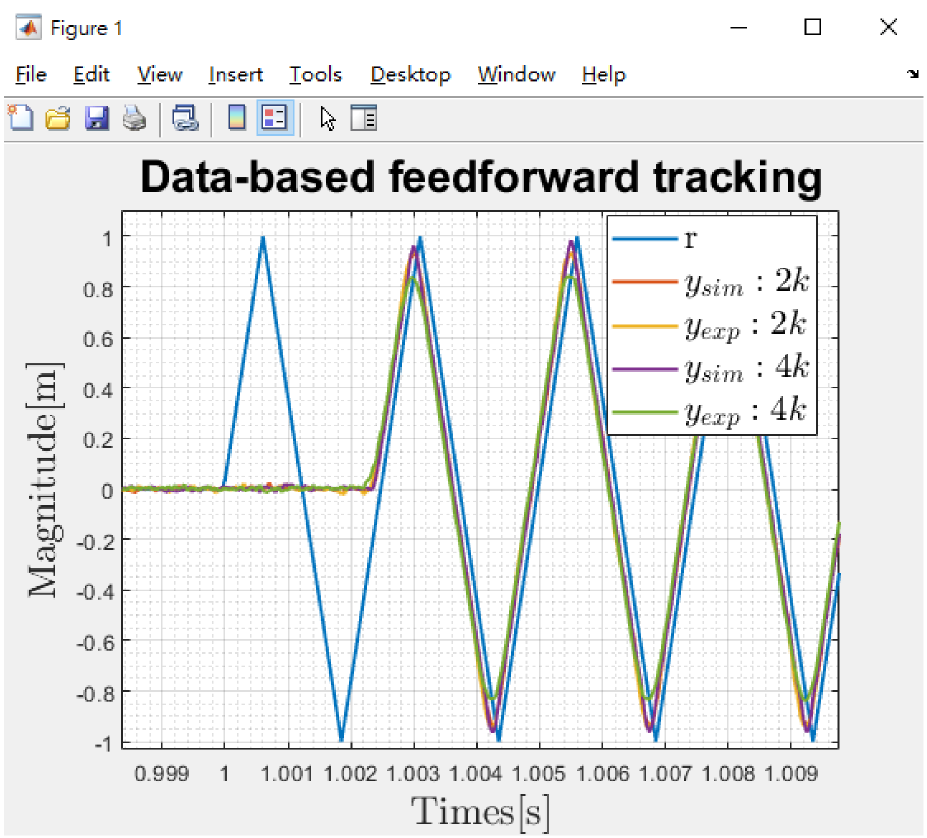


Fig. 7: phase shift of DBFF

high order may cause problem

check data.

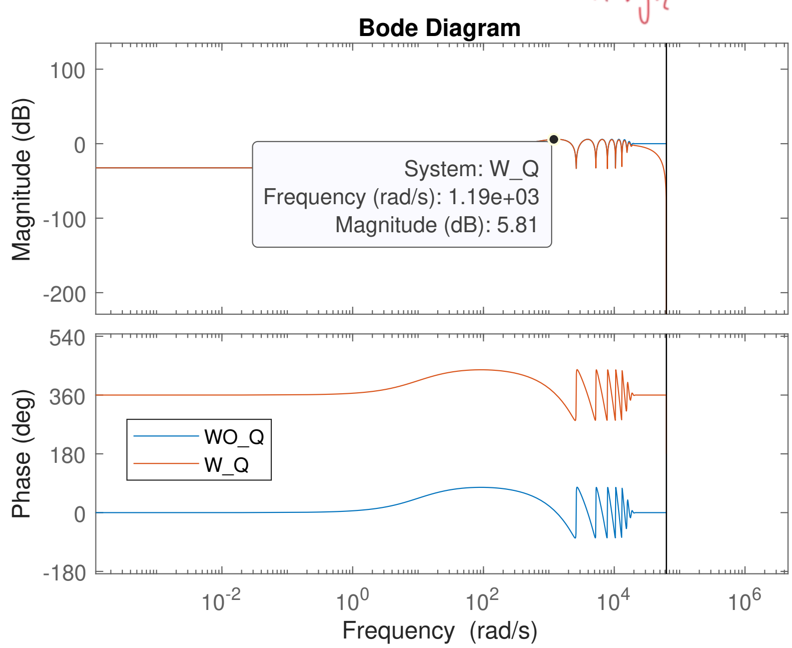


Fig. 8: Bodeplot of stability criterion