Model-Free Repetitive Control and Iterative Learning Control

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

[1] Dept. of Electrical Engineering
National Taiwan University, Taipei 106319, Taiwan
[2] Dept. of Electrical Engineering
Technical University Darmstadt, 64283 Darmstadt, Germany

Keywords: Data-based inverse filter · Model-free RC · Model-free ILC.

1 Introduction

In the final exam, we are asked to design learning-based controllers for the galvanometer to track 400 Hz triangular wave with an amplitude of 1 degree and duration of 2 seconds.

This report is organized as follows. Section 2 introduces the flow of obtaining a data-based inverse filter. Section 3 elaborates on how the performance is when using the data-based inverse filter to do a feedforward control in both simulation and experiment. In Section 4 and Section 5, we apply the data-based inverse filter on repetitive control and iterative learning control. Finally, we draw some conclusions in the last section.

2 FIR Inverse Filter

In Lab02, we have derived the inverse filter $F(z)$ by zero-phase error tracking controller (ZPETC)[1],[3]. 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.

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

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

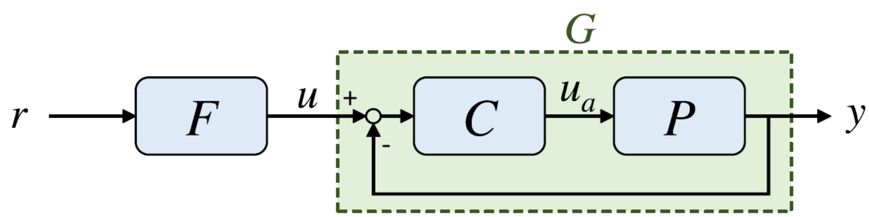


Fig. 1: Block diagram of feedforward control [1]

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

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

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.

Parameter	**Value**
N	47
α	0.5
i	100

Table 1: Parameters

2.2 Data-based inverse filter results

In this section, we show the magnitude of the bodeplot of 4 cases times the estimated plant from Lab01 in Fig.3a

1. $F_{sim} : 2k$: 2000 Hz lowpass filter and the data from the simulation.
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. This results from the system cannot track such high-frequency signals. 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. This is because the area under the impulse response is enlarged to increase the noise-signal ratio. To solve this problem, we use Eqn.7 to regulate our inverse filter. and the result is shown in Fig.3b. We can see that the magnitude of inverse filters from the simulation environment are perfectly equal to 1 and the ones from the experimental data aren't due to the modeling error.

$$F(z) = \frac{F(z)}{\mathrm{Sum}(r)} \tag{7}$$

2.3 Stability analysis

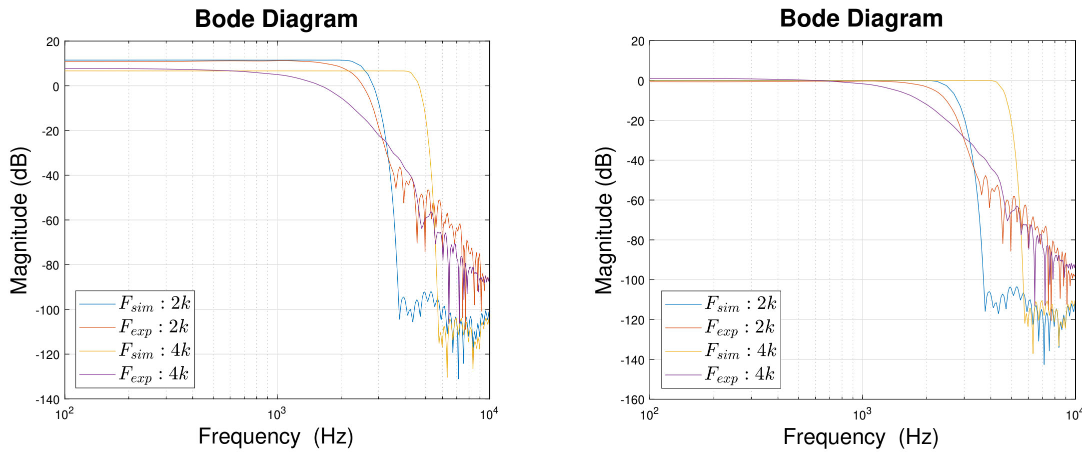
The stability criterion of ILC or RC can be represented as the following equation[1],[2],

$$|Q(z)(1 - F(z)G(z))|_{\infty} < 1 \tag{8}$$

Where $Q(z)$ is a zero-phase lowpass filter and $G(z)$ is a stable closed-loop system. Here we use the estimated Galvaonometer scanner system from Lab01 as $G(z)$ and let $Q(z) = 0.25z^{(1)} + 0.5 + 0.25z^{(-1)}$. Then, we plot out the bodeplot of Eqn.8. The result is shown in Fig.4 It's easy to see that the maximum value in four cases is less than 0dB, which means they all satisfy the stability criterion. In the following section, we only discuss $F_{exp} : 2k$ and $F_{exp} : 4k$.

3 Data-Based Feedforward

The simulation and experiment data-based feedforward tracking are shown in Fig.5. From Fig.3, 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 worse than $F_{exp} : 2k$ at the triangular peak in the experiment. On the other hand, the tracking result from the experiments has a little phase delay compared to the one from the simulation.



(a) 4 cases of inverse filter

(b) 4 cases of inverse filter: normalized

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

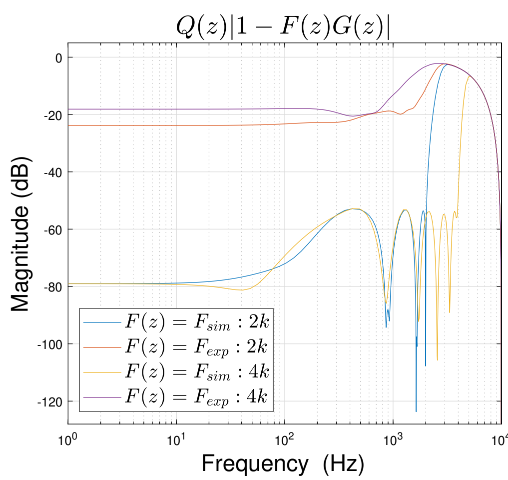
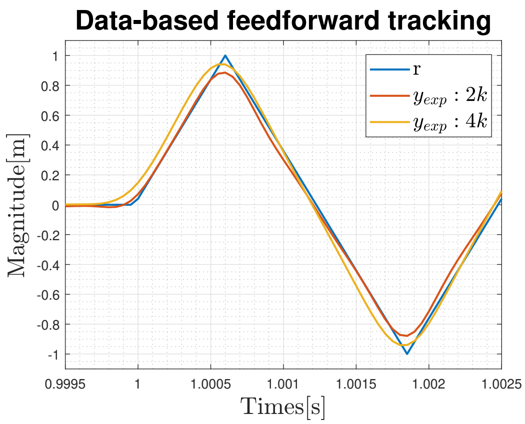


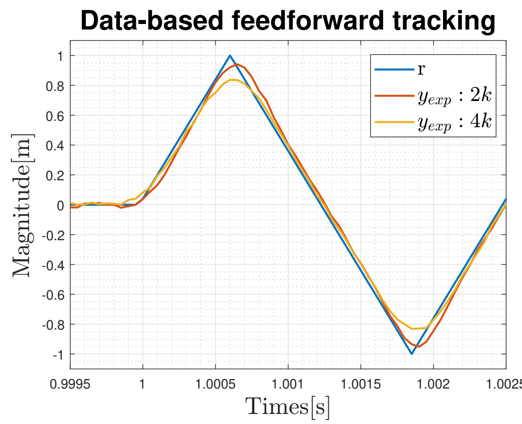
Fig. 4: Bode plot of $Q(z)(1-F(z)G(z))$

4 Model-Free Repetitive Control

The simulation and experiment results of model-free repetitive control are shown in Fig.6a and Fig.6b. Compared with the zero-phase tracking controller(ZPETC)[3] we have used in Lab02 as the inverse filter because we adopt a data-based inverse filter, it's no surprise that we have a better performance at the beginning. After running about 2 periods the output y almost aligns with the reference. Just like the data-based feedforward tracking, $F_{exp}:2k$ outperforms $F_{exp}:4k$ in the peak of the triangular wave. Also, this can be illustrated from Fig.3

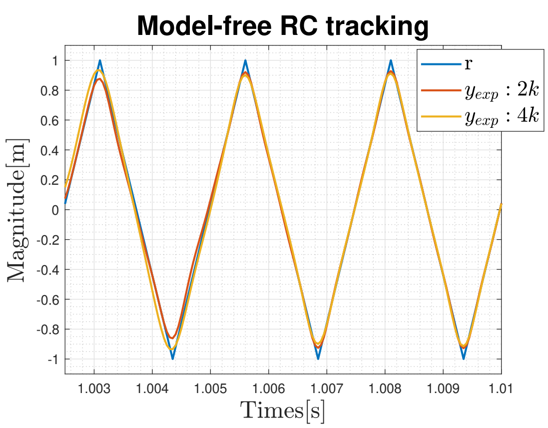


(a) Data-based feedforward tracking: simualtion

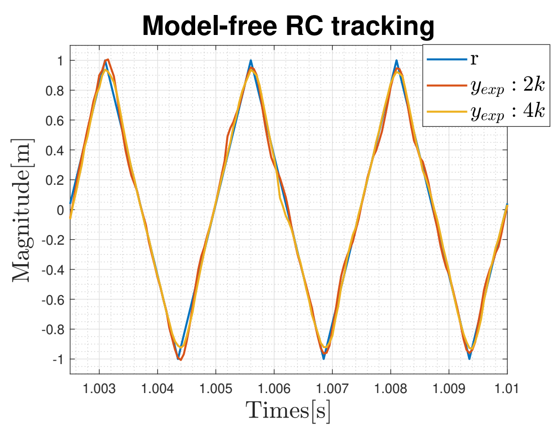


(b) Data-based feedforward tracking: experiment

Fig. 5: Data-based feedforward tracking results



(a) Model-free RC tracking: simualtion



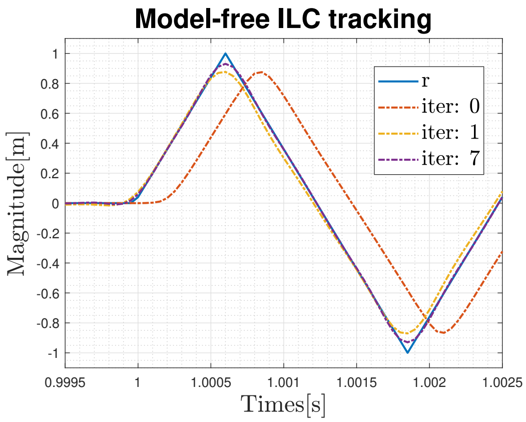
(b) Model-free RC tracking: experiment

Fig. 6: Model-free RC tracking results

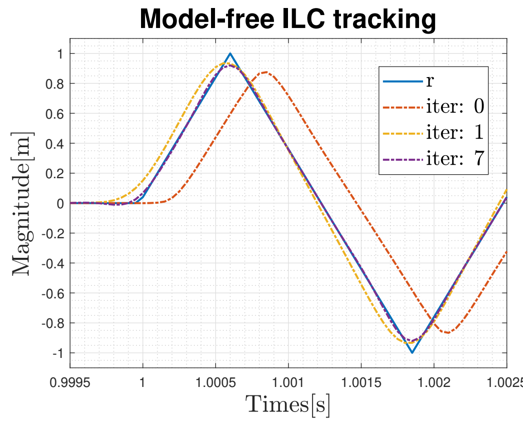
5 Model-Free Iterative Learning Control

We apply the data-based inverse filter F as the learning filter to the ILC and run 10 iterations. The simulation results of model-free ILC tracking are shown in Fig.7a and Fig.7b, and the experimental results are shown in Fig.7c and Fig.7d.

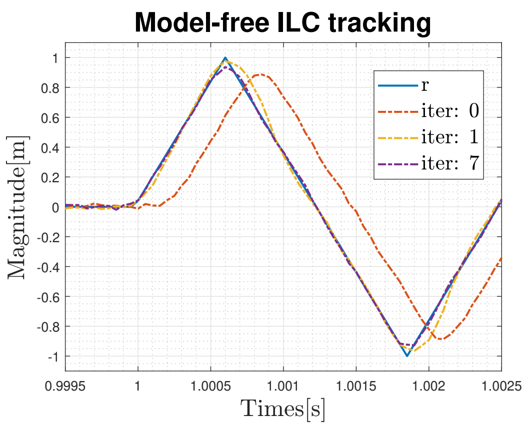
We also consider 2 cases and plot out the result separately after iterations 0, 1, and 7. No matter whether in simulation or experiment, all cases gain huge progress after 1 iteration, and after 7 iterations, the output almost aligns with the reference. Fig.7e shows the tracking result of 2 cases after 7 iterations in the experiment. Also, $F_{exp} : 2k$ outperforms $F_{exp} : 4k$ in the peak of the triangular wave.



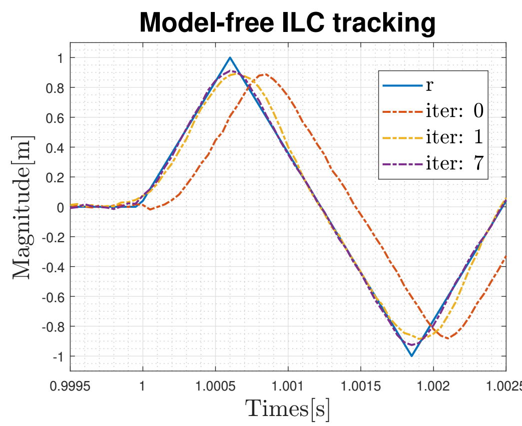
(a) Model-free ILC tracking: F_{exp} : $2k$: simulation



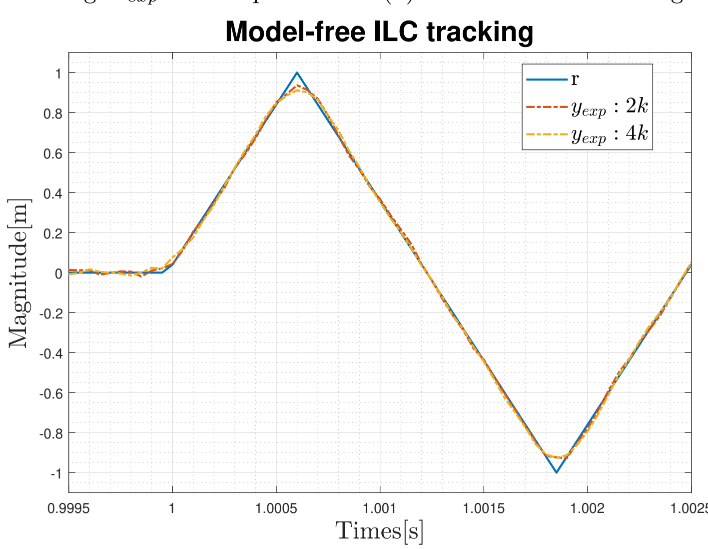
(b) Model-free ILC tracking: F_{exp} : $4k$: simulation



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



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



(e) Model-free ILC tracking: 7 iterations : experiment

Fig. 7: Model-free ILC tracking

6 Conclusion

In the first place, this report elaborates on how to obtain a data-based inverse filter from a mathematical concept to an implementation point of view. Afterward, we considered 2 cases of inverse filter, $F_{exp} : 2k$ and $F_{exp} : 4k$, and proved that both cases work for the simulation environment and real galvanometer system. Furthermore, $F_{exp} : 2k$ outperforms $F_{exp} : 4k$ in the peak of the triangular wave due to the system's inability to track high-frequency signals. As a closing remark, we claim
"We did this exam by ourselves without working with or getting help from any other group."

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.
3. M. Tomizuka, "Zero phase error tracking algorithm for digital control," Journal of Dynamic Systems, Measurement, and Control, 109, pp. 65-68, 1987.