A Model-Free Loop-Shaping Method based on Iterative Learning Control [⋆]

Li-Wei Shih [*] **Cheng-Wei Chen** [**]

[*] *Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan (e-mail: b04502031@ntu.edu.tw).*
[**] *Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan (e-mail: cwchenee@ntu.edu.tw).*

Abstract:
Many techniques have been developed for the loop-shaping method in control design. While most loop-shaping methods apply a model of the open-loop controlled plant, the resulting performance depends on the accuracy of the dynamical model. This paper aims to develop a model-free loop-shaping technique. The core idea is to convert the model matching problem to a trajectory tracking problem. To achieve the desired loop gain, we need to determine the control input such that the system output tracks the impulse response of the loop gain function. In this paper, a model-free iterative learning control (ILC) algorithm is applied to solve this tracking problem. Once the ILC converges, the feedback controller that meets the desired loop gain can then be constructed. This method does not require the model of the controlled plant, hence it provides better performance of loop-shaping control design. The proposed method is validated through numerical simulation on a 3-rd order plant.

Keywords: loop-shaping, iterative learning control

1. INTRODUCTION

To achieve closed-loop tracking performance, noise rejection, and robustness, loop-shaping has been applied to modern controller design in many applications [Prempain and Postlethwaite (2005)], [Zhu et al. (2003)], [Fujita et al. (1993)]. Several techniques have been developed, such as PID-controller [Grassi and Tsakalis (1996)] and $\mathcal{H}_\infty$ synthesis loop-shaping method [McFarlane and Glover (1992)]. The well-known $\mathcal{H}_\infty$ synthesis loop-shaping is wildly used to design a controller by assigning several weighting functions indicating the requirement of the desired performance and noise suppression.

The abovementioned work needs a parametric model of the open-loop controlled plant to work on. That is, dynamical modeling or system identification should be properly performed before applying these model-based loop-shaping methods. The resulting performance relies on the quality of the provided model. To eliminate the procedure of model construction, several types of research have been working on model-free loop-shaping approaches, such as a data-driven PID controller [Saeki (2014)]. This method is useful for most low-order plants while it is still challenging to compensate for higher-order dynamics. Another approach is utilizing Linear Matrix Inequality (LMI) to design fixed-order controller with transient response data [Saeki and Sugitani (2008)], [Formentin and Karimi (2012)]. Formulated as a convex optimization problem, LMI ensures a globally optimal solution. However, the computation is time-consuming when the variable dimension is large, especially when it is applied to transient response data.

Iterative Learning Control (ILC), on the other hand, has been shown its excellent capability of tracking [Bristow et al. (2006)]. By iteratively working on the same task repeatedly, updating the current input with respect to the output error in the previous run, the input converges and generates the desired output. The learning filter is a critical design parameter in the ILC algorithm, which exerting influence on the convergence property of ILC. Model-free ILC has been proposed, such as PD-type updating law [Chen and Moore (2002)], reverse-time technique [Ye and Wang (2005)], and model-free inversion-based ILC (MIIC) [Kim and Zou (2012)]. As to increase the convergence rate, [Chen and Tsao (2017)] proposed a convergence rate acceleration algorithm. The algorithm learns a feedforward filter and crossly updates the learning filter from the on-learning feedforward filter during the iterative process. This algorithm can be applied to construct the plant inversion FIR filter in only a few iterations. The constructed plant inversion FIR filter is able to work as a learning filter to other ILC applications.

The main contribution of this paper is the development of a model-free loop-shaping method based on ILC. The conventional loop-shaping problem is converted to a tracking problem. The impulse response of the target loop gain function is assigned as the system's desired reference, and then the ILC algorithm is applied to track the reference at the system's output. Once the iterative learning process converges, the output response of the open-loop plant is the same as the impulse response of the target loop gain. The learned input is then used to construct the closed-

[⋆] This work was financially supported from the Young Scholar Fellowship Program by Ministry of Science and Technology (MOST) in Taiwan, under Grant MOST 108-2636-E-002-007.

loop controller. Therefore, the constructed controller along with the controlled plant yields the desired loop gain. A parametric model is unneeded in this method, resulting in better performance of loop-shaping controller design.

The remainder of this paper is organized as follows: Section 2 gives the problem definition and theoretical background; Section 3 demonstrates the proposed method for loop-shaping; simulation validation is presented in Section 4; the conclusion is given in Section 5.

2. PROBLEM DEFINITION AND PRELIMINARIES

2.1 Loop-shaping

For a typical loop-shaping problem, considering a SISO linear time-invariant(LTI) plant $P(z)$, which is in Z-transform representation as

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

where $p(k)$ is the impulse response of the plant $P(z)$ and $N \to \infty$. In the meanwhile, several closed-loop system behavior specifications are also assigned such as tracking performance, noise rejection, and robustness.

Assumption 1. The Z-transform representation plant $P(z)$ is a proper minimum phase transfer function.

The purpose of loop-shaping is to find a controller $C(z)$ such that the overall loop gain $L(z)$ is equal to the desired loop gain $L_d(z)$ that satisfies the requirements. Note that both the controller $C(z)$ and the loop gain $L(z)$ are in the Z-transform representation, and the loop gain $L(z)$ is the series connection of the controller and the plant, i.e.,

$$L(z) = C(z)P(z) \tag{2}$$

For a desired loop gain $L_d(z)$ that is selected to meet the system specifications, the quality of the controller $C(z)$ designed can be verified by the error between $L_d(z)$ and the loop gain $L(z)$ corresponding to $C(z)$. Therefore, the goal is to find a controller $C(z)$ that minimizes the loop gain error. In other words,

$$\min_{C(z)} \|L_d(z) - C(z)P(z)\|_\infty \tag{3}$$

where $\|\cdot\|_\infty$ denotes the $\mathcal{H}_\infty$-norm. This optimization cost function can also be converted to the time domain with the $\mathcal{H}_\infty$-norm substituting to l^2-norm.

$$\min_{c(k)} |L_d(k) - c(k)P(z)|_2 \tag{4}$$

Note that $|\cdot|_2$ denotes the l^2-norm, $L_d(k)$, and $c(k)$ represent the impulse response of the desired loop gain $L_d(z)$ and the controller $C(z)$, respectively. Thus, the original loop-shaping problem is translated into a tracking problem form.

2.2 Iterative Learning Control (ILC)

For a control input $u(k)$ and the corresponding output $y(k)$ passing through a plant $P(z)$, ILC iteratively learns to track a reference tracking $r(k)$ by updates the control input in the following form:

$$\begin{aligned} u_0(k) &= F(z)r(k) \\ u_{j+1}(k) &= u_j(k) + F(z)[r(k) - y_j(k)] \\ &= u_j(k) + F(z)e_j(k) \end{aligned} \tag{5}$$

where $F(z)$ is the learning filter, and $e(k) := r(k) - y(k)$ is the tracking error. Note that the lower index j denotes the j-th iteration in the ILC process.

With some simplification, the error at j-th iteration can be described as

$$e_{j+1}(k) = [1 - P(z)F(z)]^{j+1}r(k) \tag{6}$$

From Eq. 6, it is observable that the iterative process converges if the following criterion is satisfied [Norrlöf and Gunnarsson (2002)]:

$$\gamma := \|I - P(z)F(z)\|_\infty < 1 \tag{7}$$

Note that γ is the error convergence rate. Eq. 7 shows that the closer the multiplication of the plant $P(z)$ and the learning filter $F(z)$ is to 1, the smaller the error convergence rate γ is, which means faster convergence rate. Ideally, under the circumstance that $F(z) = P^{-1}(z)$, γ is equal to 0, which means the iterative learning process converges immediately at the first iteration. With respect to this idea, a satisfactory way to choose the learning filter is by finding the plant inversion $P^{-1}(z)$ as $F(z)$.

ILC-based Feedforward Filter (ILCFF) To obtain a proper learning filter $F(z)$, the method proposed in [Chen et al. (2020)] can be applied to construct a learning filter through plant inversion. This approach utilized ILC to learn an inversion-based FIR feedforward filtered without constructing the parametric plant model by tracking a filter impulse. The ILC algorithm is used to minimize the following equation:

$$\min_{f(k)} |r(k) - f(k)P(z)|_2 \tag{8}$$

where $r(k)$ is a target reference output and $f(k)$ is the corresponding learned feedforward filter coefficients. The reference output $r(k)$ is typically a filtered impulse filtered by a zero-phase low pass filter, avoiding control saturation for high gain occurring at high frequency. The inversion-based FIR feedforward filter is then constructed by

$$F(z) = \sum_{k=-N/2}^{N/2} f(k)z^{-k} \tag{9}$$

Notice that the constructed inversion-based feedforward filter may be non-causal despite the relative order of the plant $P(z)$. To clarify, if the plant has a delay of order n, then its inversion filter has an advance of order n, resulting in a non-causal inversion filter.

Convergence rate acceleration for ILC In comparison to the slow convergence rate in the conventional ILC methods, which use PD-type or reverse-time technique as a learning filter, an approach significantly improving the convergence rate for ILC was proposed in [Chen and Tsao (2017)]. The algorithm proposed in this approach updates the learning filter $L(z)$ from the learned feedforward filter $F(z)$ every constant iteration. Since the $F(z)$ learns the plant inversion FIR filter, each the learning filter is updated, the product $P(z)F(z)$ is closer to 1, leading to a faster convergence rate γ presents in Eq. 7.

3. ILC-BASED LOOP-SHAPING

For the loop-shaping problem of the model matching form described in Eq. 3, we propose a model-free method to design the controller $C(z)$ utilizing ILC. The problem is converted into a time domain tracking problem states in Eq. 4. As the key concept, the loop gain $L(z)$ constructed in Eq. 2 is equal to the desired loop gain $L_d(z)$ as long as the impulse response $l(k)$ learns the reference output $L_d(k)$. In other words, if there is a control input $c(k)$ to the plant $P(z)$ such that its output $l(k)$ is same as $L_d(k)$, then the control input $c(k)$ can be used as an FIR filter shaping the loop gain. The method proposed in this paper can be separate to following four steps:

Step 1: Design the desired loop gain function $L_d(z)$ according to given specifications.
Step 2: Generate the learning filter using ILCFF with convergence rate acceleration.
Step 3: Apply ILC to learn the impulse response of $L_d(z)$.
Step 4: Convert the learned FIR filter into an IIR filter using balanced model reduction.

3.1 Algorithm

Step 1 (Design of the loop gain function $L_d(z)$)

Assumption 2. The relative order of the desired loop gain $L_d(z)$ is higher than the relative order of the plant $P(z)$.

Once the design specification is provided, the desired loop gain $L_d(z)$ can be designed to meet the requirements. However, the limitation for the relative order relationship between $L_d(z)$ and $P(z)$, described in *Assumption 2*, is necessary. The desired controller of the loop-shaping problem is denoted as $C_d(z)$, which can be evaluated by $C_d(z) = L_d(z)/P(z)$. If $C_d(z)$ is improper, the learned FIR filter $c(k)$ is non-causal. Non-causal FIR filters can only be realized with an additional input delay, which is not allowed in the implementation of feedback control. Hence for causal control input, $C_d(z)$ should be limited to be proper, resulting in the limitation between $L_d(z)$ and $P(z)$. To measure the relative order of the plant $P(z)$, a delta impulse is triggered as an input to the plant and the output response is recorded. Then the relative order of $P(z)$ is evaluated from the delay between the impulse input and the moment the output starting to response.

Step 2 (Design of the ILC learning filter)

Assumption 3. The learning filter $F(z)$ is chosen properly such that the ILC convergence criteria stated in Eq. 7 is satisfied. Thus, the ILC algorithm converges to 0, i.e. for any arbitrary small $\epsilon \in \mathbb{R}$, there exists $n^* \in \mathbb{N}$, such that $|e_n(k)|_2 < \epsilon$ for all $n > n^*$, $n \in \mathbb{N}$.

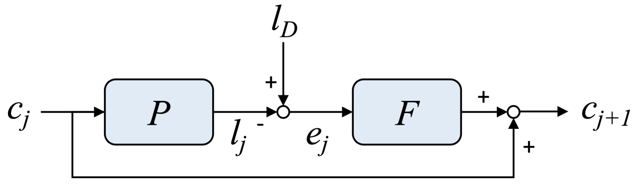

Fig. 1. The block diagram for the propose method.

While all kinds of learning filters that promise the convergence are suitable for the ILC process, while the error convergence rate γ defined in Eq. 7 depends on the quality of the learning filter chosen, however. To achieve a faster convergence rate, the learning filter generated from ILCFF with convergence rate acceleration method is suitable for the proposed ILC-based method. Taking advantage of its exponential error convergence rate acceleration, the method is practical to be implemented on a plant in the real world such that the experimental tasks required are affordable.

Step 3 (ILC for loop shaping design)

With the constructed learning filter, the ILC algorithm is applied to the tracking problem in Eq.4. The block diagram is shown in Fig. 1 and the update law of iterative process is stated as

$$\begin{aligned} c_{j+1}(k) &= c_j(k) + F(z)[L_d(k) - l_j(k)] \\ &= c_j(k) + F(z)e_j(k) \end{aligned} \tag{10}$$

Since the selected learning filter satisfies *Assumption 3*, the ILC algorithm for the tracking problem converges. The corresponding learned FIR filter $C(z)$ is constructed as

$$C(z) = \sum_{k=-N/2}^{N/2} \lim_{j\to\infty} c_j(k) z^{-k} \tag{11}$$

where N is the length of the learned input filter, and $N \to \infty$ ideally. This corresponds to an infinite long FIR filter, which is impractical for implementation. Hence the presence of truncation is necessary, the zero-closed term should be truncated at an appropriate threshold. An instinctive way is to consider the settling property of the impulse response of $P(z)$ and $L_d(z)$, where the acceptable error threshold is a design parameter depends on the implementation, selecting the longer one as the filter length N.

Step 4 (Controller order reduction)

For real-time feedback control, there is a disadvantage of directly using the learned controller $C(z)$ constructed from the learned input $c(k)$. By selecting the threshold, the infinite long impulse response is truncated to a finite length N. Nevertheless, the length is still too long for real-time implementation which required considerable computation for filter convolution. This use of the FIR filter also results in that the closed-loop system needs at least N samples to achieve steady state, causing poor behavior in transient response.

To achieve a better transient response, it is necessary to reduce the order of the learned controller $C(z)$. The controller then becomes an IIR filter denotes as $C_{\text{IIR}}(z)$, which can be obtained by model reduction. The error due to balanced model reduction, $\|C(z) - C_{\text{IIR}}(z)\|_\infty$, is promised through its Hankel singular values [Glover (1984)], it has been shown that

$$\|C(z) - C_{\text{IIR}}(z)\|_\infty \leq 2 \sum_{k \geq r+1}^{N} \sigma_k \tag{12}$$

where σ_k, $k \in \mathbb{N}$, $0 < k \leq N$, is the k-th Hankel singular value of the control input $C(z)$, and r is the order of the reduced IIR filter. Specifies that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq \ldots \geq \sigma_N$.

Once $C(z)$ is truncated by balanced model reduction, the IIR-type loop gain $L_{\text{IIR}}(z)$ is expressed as

$$L_{\text{IIR}}(z) = C_{\text{IIR}}(z)P(z) \tag{13}$$

Since he desired controller $C_d(z)$ can be estimated through directly division $C_d(z) = L_d(z)/P(z)$. The order of the IIR-type controller $C_{\text{IIR}}(z)$ can be evaluated with and no more than the sum of the orders of $L_d(z)$ and $P(z)$.

3.2 Frequency Weight

When the pole of the desired loop gain $L_d(z)$ getting closer to the unit circle, it takes longer for the impulse response of $L_d(z)$ to approaches 0. This causes that the length N of the impulse response should be chosen specific large in order to reduce the error due to truncation. It is even impossible when $L_d(z)$ contains an integrator, which needs an infinity long length and unable to be truncated.

To solve this problem, the desired loop gain $L_d(z)$ can be separated into two parts $H(z)$ and $L'_d(z)$.

$$L_d(z) = H(z)L'_d(z) \tag{14}$$

where $H(z)$ is a frequency weighting function such as integrator or the other slow poles that cannot be truncated under the selected filter length N. $L'_d(z)$ contains all the remaining poles and zero. Therefore, the proposed ILC-based loop-shaping method can be applied to track the impulse response of $L'_d(z)$, denoted by $L'_d(k)$. The controller $c(k)$ can be reconstructed with the learned controller input $c'(k)$ and $H(z)$.

$$c(k) = H(z)c'(k) \tag{15}$$

Note that it is necessary to ensure $L'_d(z)$ is proper, otherwise its impulse response is unstable instead of converging to 0. While $L'_d(z)$ is improper due to the separation of $H(z)$ and $L'_d(z)$, Eq. 14 should be modified to create additional pole-zero pairs. To clarify, $L'_d(z)$ is multiplied by additional poles and hence becomes strictly proper. $H(z)$ is also multiplied by the additional zeros.

$$L_d(z) = (H(z)z^n)(L'_d(z)\frac{1}{z^n}) \tag{16}$$

where n is the number of additional pole-zero pairs. The placement of the additional pole-zero pairs is located infinitely far for simplicity, hence the additional poles are promised not to make the settling property of tracking output slower. In addition, the frequency weight with an additional zero is stable since it is still a proper filter.

4. SIMULATION VALIDATION

4.1 Setup of the Simulation

The performance of the proposed method is validated by numerical simulation. Considering a SISO 3-rd order plant P(z) defined as

$$P(z) = \frac{-0.1(z-0.995)(z-0.99)}{(z-0.4)(z^2-1.998z+0.998)} \tag{17}$$

The frequency response of P(z) is presented in Fig. 2. The plant has a bandwidth of 14.2 Hz. In addition, there is a peak response at 6.51 Hz. The controller sampling rate is assumed to be 10 kHz.

The requirements of the closed-loop controlled system are given as

- Rising time, t_r <5 ms
- Settling time, t_s <10 ms
- Maximum overshoot, $M_p < 1\%$
- Phase margin, $PM > 80°$
- Steady-state error, $e_{ss} < 2\%$

For simplicity in design, assume that no overshoot occurs, i.e. $M_p = 0\%$, hence the open-loop damping ratio ζ is 0. The placement of the dominant pole can be evaluated by the approximation formula $t_r = 1.8/\omega_n$, corresponding to ω_n is 360 rad/s. The closed-loop crossover frequency ω_c is estimated with $\omega_c = \omega_n/(1+\zeta^2)$, since ζ is chosen as 0, the lower bound of ω_c is 360 rad/s. On the other hand, the 2 % steady state error indicates that the position constant K_p of $L_d(z)$ should be higher than 33.8 dB.

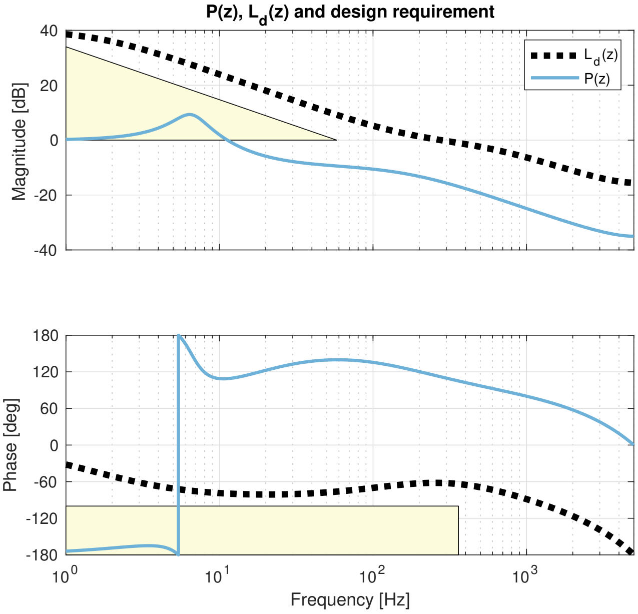

Fig. 2. The Frequency response of the plant $P(z)$ and the desired loop gain $L_d(z)$. Forbidden regions for specifications are shown in the yellow area.

With aforementioned design constraints about ω_c, K_p, the desired loop gain $L_d(z)$ is chosen to be a second-order system composing of a low-frequency pole to reduce the steady-state error and a lead compensator to ensure the phase margin specification.

$$L_d(z) = \frac{0.3(z-0.9)}{(z-0.999)(z-0.7)} \tag{18}$$

The closed-loop poles locate at 0.9444 and 0.4546. The design requirements and the frequency response are presented in Fig. 2.

4.2 ILC-Based Loop-Shaping

To start with the ILC-based loop-shaping method proposed in this paper, the ILCFF with convergence rate acceleration algorithm should be performed to obtain an FIR inverse filter $f(k)$ of the plant $P(z)$. Note that the length of $f(k)$ is chosen as 5000, which is 0.5 s corresponding to 10 kHz sampling rate. Specifically, half of the filter length is for the causal part and the other half for non-causal. The algorithm is run for a total of 100 iterations with the learning filter updating every 10 iterations. The resulting RMS tracking error described in Eq. 4 is around 2×10^{-13}, where $r(k) = \delta(k)$ is a delta impulse function.

With the FIR inverse filter $f(k)$, the proposed loop-shaping method is then applied for learning the control input $c(k)$ using Eq. 10. The length of L_d(k) is truncated to 5000 (0.5 s at 10 kHz sampling rate).

The simulation result of the l^2-norm tracking error convergence curve is shown in Fig. 3. The tracking error converges to around -240 dB at the 5-th iteration, as the error is mostly contributed by the numerical error and hence the error stays around the same level after running 5 iterations. The iterative learning result $l_{\text{FIR}}(k)$ of tracking $L_d(k)$ and the corresponding leaned controller input $c(k)$ are presented in Fig. 4. The maximum tracking error is around 3×10^{-12}, which is also a numerical error.

To perform a real-time feedback control, the learned filter $c(k)$ is converted to an IIR filter $C_{IIR}(z)$ through balanced realization. By investigating the Hankel singular values of the FIR filter, the model can be reduced to a 5-th order controller, which consists of 96 % of the summation of original Hankel singular values. The IIR-type loop gain $L_{IIR}(z)$ is formed by Eq. 13. The error bound of $\|C_{\text{FIR}}(z) - C_{\text{IIR}}(z)\|_\infty$ can be obtained from Eq. 12, where r, m are the reduced model order 5 and the original model order 10 000, respectively. The error bound is promised by Hankel singular value according to Eq. 12, which is 5.52 or 14.84 dB, as the maximum error is around only -38 dB. Finally, Fig. 5 plots the desired loop gain and the resulting loop gain $L_{\text{IIR}}(z)$,indicating that the learned loop gain fits the desired loop gain both in magnitude and phase. The error between the two loop gains is less than -38 dB in all frequency presented in Fig. 6.

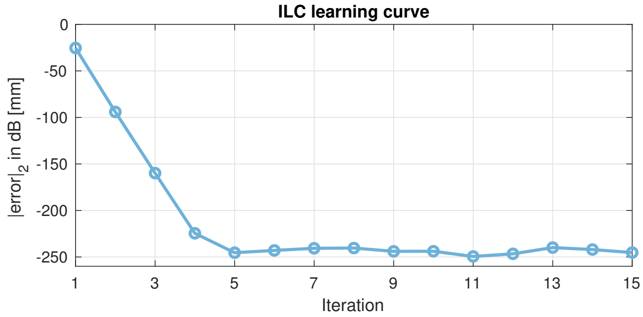

Fig. 3. The ILC learning curve of tracking $l_d(k)$, in terms of the l^2-norm error $|l_d(k) - c(k)P(z)|_2$.

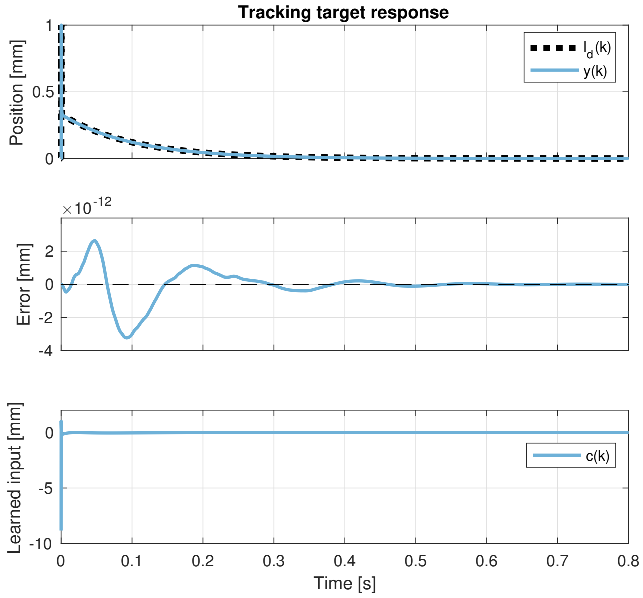

Fig. 4. The converged iterative result of tracking $l_d(k)$ and the corresponding learned input $c(k)$.

	spec.	G_d	G_{IIR}
$t_r(ms)$	5	2.7	2.7
$t_s(ms)$	10	5.7	5.7
$M_p(\%)$	1	0	0
PM(deg)	80	118	118
$e_{ss}(\%)$	2	0.99	0.99

Table 1. The step response simulation results and validation of requirements.

4.3 Closed-Loop Validation

The closed-loop step response is simulated to identify whether the controller meets the design specification. Eq. 19 is used to obtain the closed-loop system for the desired loop gain, learned FIR and IIR loop gain, which are $G_d(z)$ and $G_{\text{IIR}}(z)$, receptively.

$$G(z) = (I + C(z)P(z))^{-1}C(z)P(z) \tag{19}$$

The simulation results of the closed-loop step response are shown in Fig. 7. Specific values of the step response properties are list in Tb. 1. Each specification of $G_d(z)$ and $G_{\text{IIR}}(z)$ are equal up to 4 decimal digits. Therefore the controller learned from the method proposed in this paper meets the provided requirements.

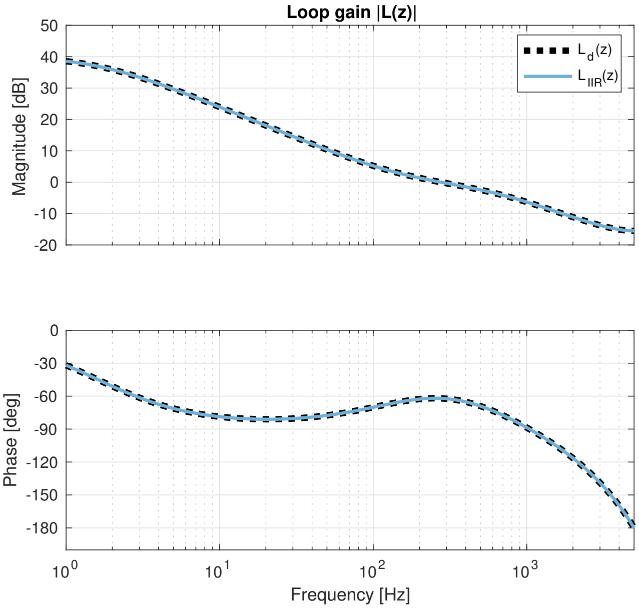

Fig. 5. The frequency response of the desired loop gain and the loop gain learned with ILC.

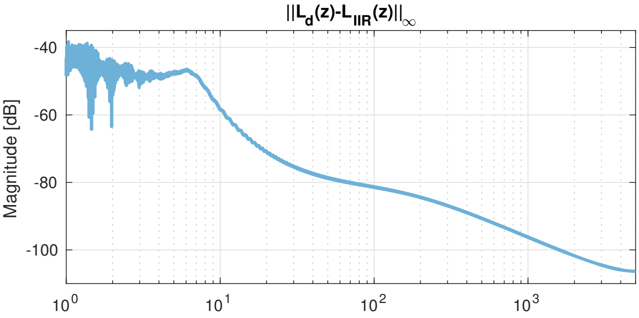

Fig. 6. The frequency response of $\|L_d(z) - L_{\text{IIR}}(z)\|_\infty$.

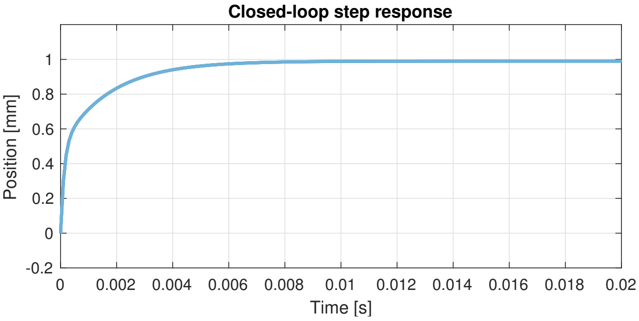

Fig. 7. The step response of the closed-loop system.

5. CONCLUSION

A novel method for model-free loop-shaping based on ILC is proposed in this paper. The conventional loop-shaping problem is converted into a tracking problem, which is solved by the ILC algorithm, taking advantage of its tracking performance and model-free process. While the learning for the impulse response of the desired loop gain is done, the learned control input is used to construct an FIR filter as the closed-loop controller. For real-time feedback implementation, the FIR controller is then further converted into an IIR filter by balanced realization. The proposed is validated on a 3-rd order numerical plant. The simulation results show that the constructed controller meets the desired loop gain, as the learning process converges within only a few iterations.

REFERENCES

Bristow, D.A., Tharayil, M., and Alleyne, A.G. (2006). A survey of iterative learning control. *IEEE control systems magazine*, 26(3), 96–114.

Chen, C.W., Rai, S., and Tsao, T.C. (2020). Iterative learning of dynamic inverse filters for feedforward tracking control. *IEEE/ASME Transactions on Mechatronics*, 25(1), 349–359.

Chen, C.W. and Tsao, T.C. (2017). Data-driven progressive and iterative learning control. *IFAC-PapersOnLine*, 50(1), 4825–4830.

Chen, Y. and Moore, K.L. (2002). An optimal design of pd-type iterative learning control with monotonic convergence. In *Proceedings of the IEEE Internatinal Symposium on Intelligent Control*, 55–60. IEEE.

Formentin, S. and Karimi, A. (2012). A data-driven approach to mixed-sensitivity control with application to an active suspension system. *IEEE Transactions on Industrial Informatics*, 9(4), 2293–2300.

Fujita, M., Hatake, K., and Matsumura, F. (1993). Loop shaping based robust control of a magnetic bearing. *IEEE Control Systems Magazine*, 13(4), 57–65.

Glover, K. (1984). All optimal hankel-norm approximations of linear multivariable systems and their L-infinity-error bounds. *International journal of control*, 39(6), 1115–1193.

Grassi, E. and Tsakalis, K. (1996). PID controller tuning by frequency loop-shaping. In *Proceedings of 35th IEEE Conference on Decision and Control*, volume 4, 4776–4781. IEEE.

Kim, K.S. and Zou, Q. (2012). A modeling-free inversion-based iterative feedforward control for precision output tracking of linear time-invariant systems. *IEEE/ASME Transactions on Mechatronics*, 18(6), 1767–1777.

McFarlane, D. and Glover, K. (1992). A loop-shaping design procedure using H-infinity synthesis. *IEEE transactions on automatic control*, 37(6), 759–769.

Norrlöf, M. and Gunnarsson, S. (2002). Time and frequency domain convergence properties in iterative learning control. *International Journal of Control*, 75(14), 1114–1126.

Prempain, E. and Postlethwaite, I. (2005). Static H-infinity loop shaping control of a fly-by-wire helicopter. *Automatica*, 41(9), 1517–1528.

Saeki, M. (2014). Data-driven loop-shaping design of PID controllers for stable plants. *International Journal of Adaptive Control and Signal Processing*, 28(12), 1325–1340.

Saeki, M. and Sugitani, Y. (2008). Data-driven controller design for loop-shaping using plant transient responses. In *2008 SICE Annual Conference*, 2370–2375. IEEE.

Ye, Y. and Wang, D. (2005). Zero phase learning control using reversed time input runs. *Journal of dynamic systems, measurement, and control*, 127(1), 133–139.

Zhu, C., Khammash, M., Vittal, V., and Qiu, W. (2003). Robust power system stabilizer design using H-infinity loop shaping approach. *IEEE Transactions on Power Systems*, 18(2), 810–818.