

A New Iterative Learning Control Algorithm for Final Error Reduction ^{*}

Zhu Chen ^{*} Xiao Liang ^{**} Minghui Zheng ^{***}

^{*} *Mechanical and Aerospace Engineering, University at Buffalo, Buffalo, NY 14260 USA (e-mail: zhuchen@buffalo.edu).*

^{**} *Civil, Structural and Environmental Engineering, University at Buffalo, Buffalo, NY 14260 USA (e-mail: liangx@buffalo.edu).*

^{***} *Mechanical and Aerospace Engineering, University at Buffalo, Buffalo, NY 14260 USA (e-mail: mhzheng@buffalo.edu).*

Abstract: Iterative learning control (ILC) is a powerful technique for tracking performance improvement by adding the iteratively updated learning signal to the feedforward loop of a repetitively operated system. To increase the robustness to uncertainties and noises, usually a low-pass filter is needed in the learning law which can result in a final steady-state error as a trade-off. This study proposes a new ILC algorithm to further reduce the final error associated with the traditional ILC design. This new algorithm can be easily applied to most ILC systems. In the proposed learning algorithm, when the learning converges and a final error is obtained, the learning law is updated by adding a compensating term which consists of a compensatory filter and the obtained final error. The to-be-designed compensatory filter introduces no more design trade-offs and uncertainties. By updating the learning law, the reached learning convergence can be broken and the final error is further reduced. The compensatory filter design guideline along with four different designs are provided and analyzed. Numerical studies with the four designs have been conducted, and the results are compared to that with the traditional ILC design. It shows that the proposed learning algorithm is able to further reduce the final error effectively.

Copyright © 2022 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: ILC, convergence, learning law, final error reduction, low-pass filter.

1. INTRODUCTION

Iterative learning control (ILC) is a control technique to improve the tracking performance of a repetitively operated system. ILC was first introduced by Arimoto et al. (1984), and the main idea of ILC is that, for each iteration, a learning signal is generated by utilizing data from previous iterations to update the control or reference input. ILC has been applied to many applications including hard disk drives, see Xu et al. (2002), industrial manipulators, see Yeon et al. (2009), wafer scanner systems, see Mishra and Tomizuka (2009), and drone systems, see Purwin and D'Andrea (2009). A review of the ILC applications can be found in Bien and Xu (2012).

Control systems are inevitably subject to disturbance, noises, and uncertainties. ILC is effective to suppress repetitive disturbances, see Zheng et al. (2017, 2018); Wang et al. (2018) as the error associated with the repetitive disturbance can be learned and removed over iterations. However, traditional ILC is weak to completely cancel non-repetitive disturbances and thus non-repetitive errors will accumulate and eventually diverge. To handle this, high-order ILCs, see Chen and Moore (2002), and Q filter-based ILCs, see Sun et al. (2014); Lee et al. (2000) have been designed. The Q filter is usually designed to be a low-pass filter which aims to filtrate the non-repetitive influence of high frequency signals. In Lin et al. (2015), a time-varying filter is designed to first identify the principal components of the non-repetitive disturbance and then separate it from the repetitive

disturbance. In Merry et al. (2008), a wavelet filter is designed for ILC, and by comparing the two error realizations for each iteration, the non-repetitive part can be identified and removed.

One important parameter in ILCs is the learning filter; it is usually designed using the model inverse-based method and norm optimization-based one, see Harte et al. (2005); Ge et al. (2017); Meng (2018); Adlakha and Zheng (2021). As the learning filters are usually designed based on the nominal model, to handle the adverse effects resulting from modelling uncertainties, robustness of the ILC designs have been investigated. For example, adaptive ILCs, see Yin et al. (2010), and iterative learning identification-integrated ILCs, see Liu and Alleyne (2016), have been designed to deal with parametric uncertainties. Robust ILCs which leverage the H_∞ optimal control theory, see Chen et al. (2022b) and μ -synthesis, see Zheng et al. (2017), have been proposed to handle the modeling uncertainties and un-modeled dynamics. There are some ILC variants proposed for heterogeneous systems, see Chen et al. (2020, 2022a) that also encounter the balancing design between the learning performance and the robustness.

It is worth mentioning that increasing the robustness to the disturbances, noises and uncertainties is at the cost of the system performance. The final error usually involves the Q filter term which can not be eliminated, see Wang et al. (2014). This can be understood that the Q filter has a cutoff frequency, and this will filtrate partial benign learning signals which is outside the designed bandwidth region. To alleviate the trade-off, Ye and Wang (2006) have designed a P-type ILC which uses a negative learning gain to learn more frequency components and im-

^{*} This work is in part supported by the National Science Foundation CAREER Award under Grant No. 2046481. Corresponding author: Minghui Zheng.

proves the final error level. Li et al. (2021) have designed a dual-loop ILC to reduce the final error, in which the inner loop uses the traditional ILC and the outer loop generates a feedforward signal to the ILC when the error converges, and no additional dynamic parameter design efforts are needed in their study. ILC combining other advanced feedback control techniques has been proposed. For example, Xie and Ren (2018) have designed the iterative learning-based model predictive control (MPC) to achieve high precision level, in which the MPC controller limits the tracking error for all iterations, while the ILC is applied to ensure high precision tracking. Nevertheless, the above-mentioned methods are usually designed for specific ILC algorithms. In this paper, we will introduce a new final error reduction law that is applicable to most of the ILCs.

To further reduce the final error associated with traditional ILC, this paper designs a new ILC algorithm that is applicable to most of ILC systems with easy implementation. When the error converges, the learning law is updated by added an extra term, and then the iteration resumes with the updated learning law. This extra term mainly consists a compensatory filter which is designed based on the known dynamics. The compensatory filter will not introduce further uncertainties and inaccuracy. This updated learning law aims to break the current convergence by learning more components to further reduce the error.

The main contributions of this paper can be highlighted as follows. (a) The compensatory filter design is introduced systematically with theoretically-proved performance enhancement and will not affect the stability of the system. (b) It can be easily applied to most of the existing ILC systems by just adding an extra term and does not introduce additional design trade-offs. (c) An analytical design guideline along with four selected different designs for compensatory filters are provided.

The remainder is organized as follows. Section 2 illustrates the traditional ILC basics. Section 3 presents the new ILC, where the updating mechanism, compensatory filter design, and convergence analysis are demonstrated. Section 4 shows the numerical studies with four compensatory filter designs along with the traditional ILC, and Section 5 concludes the paper.

2. TRADITIONAL ILC BASICS

This section first reviews the traditional ILC learning law. Then the final error associated with it is derived and discussed.

2.1 Learning law

The ILC framework is given in Fig. 1, where $\mathbf{r}$ denotes a repetitive trajectory reference; P denotes a stable closed-loop system with a baseline controller included; j as the iteration index; $\mathbf{s}_j$, $\mathbf{y}_j$, $\mathbf{e}_j$ as the learning signal, tracking trajectory, and tracking error at iteration j , respectively. E contains the converged errors which is elaborated in Section 3.

In this study, we focus on linear time invariant (LTI) systems, and the theoretical derivations are presented in the frequency domain. Define $\mathbf{e}_0$ as the tracking error of the initial iteration when there is no learning signal added to the control loop. From Fig. 1, we have $\mathbf{e}_0 = \mathbf{r} - \mathbf{y}_0 = \mathbf{r} - P\{\mathbf{r}\}$ and

$$\mathbf{e}_j = \mathbf{r} - \mathbf{y}_j = \mathbf{r} - P\{\mathbf{r} + \mathbf{s}_j\} = \mathbf{e}_0 - P\{\mathbf{s}_j\}. \quad (1)$$

The traditional ILC iteratively updates the learning signal as

$$\mathbf{s}_j = Q\{\mathbf{s}_{j-1} + L\{\mathbf{e}_{j-1}\}\} \quad (2)$$

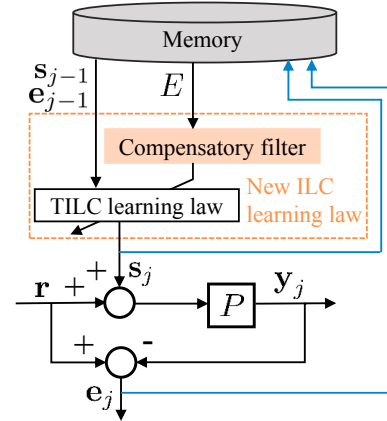


Fig. 1. ILC framework. The memory contains the tracking information from previous iterations. TILC is short for traditional ILC. The proposed new ILC adds a compensatory filter to the TILC, and the new ILC learning law along with the compensatory filter design is given in Section 3.

where L is a learning filter, and Q is usually a low-pass filter which is close to 1 over a wide range of frequencies. With (1) and (2), the error and learning signal can be represented in the iteration domain as

$$\mathbf{e}_j = Q(1 - PL)\{\mathbf{e}_{j-1}\} + (1 - Q)\{\mathbf{e}_0\} \quad (3)$$

$$\mathbf{s}_j = Q(1 - PL)\{\mathbf{s}_{j-1}\} + QL\{\mathbf{e}_0\}. \quad (4)$$

The main design task in traditional ILC is to compose two suitable parameters Q and L to make $\mathbf{e}_j$ decrease monotonously, that is, to make $\|\mathbf{e}_j\| < \|\mathbf{e}_{j-1}\|$. With (3), a sufficient condition to guarantee the stability of $\mathbf{e}_j$ is that Q is close to 1, and

$$\|\Lambda\|_\infty < 1 \quad (5)$$

where $\Lambda = Q(1 - PL)$.

2.2 Final error

The error usually converges after a few learning iterations. Define the convergence criteria as

$$\begin{aligned} \|\mathbf{e}_j\|_2 &\leq \|\mathbf{e}_{j-1}\|_2 \\ \text{and} \\ (|\|\mathbf{e}_j\|_2 - \|\mathbf{e}_{j-1}\|_2|) / (\|\mathbf{e}_{j-1}\|_2) &< \sigma \end{aligned} \quad (6)$$

where σ is a user-defined threshold. That is, (6) judges the error converges if $\mathbf{e}_j$ is no longer reduced compared to $\mathbf{e}_{j-1}$ in the sense of 2-norm, and the absolute value of the difference between $\mathbf{e}_j$ and $\mathbf{e}_{j-1}$ is within a defined small range. Then it is rational to treat $\mathbf{e}_j = \mathbf{e}_{j-1}$.

Assume that at iteration a where a is a number index, the error converges. By setting $\mathbf{e}_j = \mathbf{e}_{j-1} = \mathbf{e}_a$ in (3), and we have $\mathbf{e}_a = Q(1 - PL)\{\mathbf{e}_a\} + (1 - Q)\{\mathbf{e}_0\}$, and

$$\mathbf{e}_a = \Gamma\{\mathbf{e}_0\} \quad (7)$$

where

$$\Gamma = [1 - Q(1 - PL)]^{-1}(1 - Q). \quad (8)$$

As mentioned above, the design task of ILC is to make $\mathbf{e}_j$ in (3) decrease monotonously. Then it is rational to assume $\|\mathbf{e}_a\| < \|\mathbf{e}_0\|$, and in the frequency range of the interest (e.g., the low-frequency range or the frequency of the input signal to Γ), we have

$$\|\Gamma\| < 1. \quad (9)$$

In practice, Q is designed as a dynamic parameter and $Q \neq 1$. Therefore, $\Gamma \neq 0$ and the final error $\mathbf{e}_a \neq 0$ based on (7) and (8).

Though ILC can reduce the error from $\mathbf{e}_0$ through multiple learning iterations, (7) and (8) show that the final error of traditional ILC $\mathbf{e}_a$ will not reach zero partially due to the Q term. In practice, the final error can also be caused by noises, disturbances, uncertainties. On the other hand, the low-pass filter Q plays an important role of filtrating high-frequency noises and disturbances, and a non-zero final error $\mathbf{e}_a$ is inevitable, and its value is determined by P , L , Q and $\mathbf{e}_0$ as (7) indicates. In other words, for a given reference $\mathbf{r}$, plant P , the designed Q and L , the final error can be determined. However, our proposed method is able to break the convergence and further reduce the final error $\mathbf{e}_a$ by updating the learning law and continue the learning iterations, as presented in the following section.

3. NEW ILC ALGORITHM

This section presents the new ILC algorithm including the learning law, compensatory filter design, and convergence analysis.

3.1 Overview of the new ILC

As given in Fig. 1, the proposed ILC updates the learning law when a learning convergence is reached. To elaborate the updating algorithm, we first illustrate the process of updating the learning law of traditional ILC when the final error $\mathbf{e}_a$ is reached. Then we generalize the algorithm.

When the learning convergence is reached and $\mathbf{e}_a$ is obtained, the learning law in (2) is updated as

$$\mathbf{s}_j = Q\{\mathbf{s}_{j-1} + L\{\mathbf{e}_{j-1}\}\} + \Delta\{\mathbf{e}_a\} \quad (10)$$

where Δ is the to-be-designed compensatory filter. Compare it to the learning law in (2), the new learning law in (10) only adds a fixed extra term $\Delta\{\mathbf{e}_a\}$ which aims to break the convergence. Then with the new learning law in (10), the iterations continue, and the error can be further reduced. Assume that with a few more iterations, the error converges with the new learning law, and the new final error is labelled as $\mathbf{e}_b$. Then the learning law in (10) can be updated as

$$\mathbf{s}_j = Q\{\mathbf{s}_{j-1} + L\{\mathbf{e}_{j-1}\}\} + \Delta\{\mathbf{e}_a\} + \Delta\{\mathbf{e}_b\} \quad (11)$$

to further reduce $\mathbf{e}_b$ with further iterations. Assume that by using (11) with a few iterations, the error converges and a new final error $\mathbf{e}_c$ is obtained.

During the iterating process, we label the final errors as $\mathbf{e}_a$, $\mathbf{e}_b$, $\mathbf{e}_c$, ... in the chronological order. Define the set E contains those final errors and $E = \{\mathbf{e}_a, \mathbf{e}_b, \mathbf{e}_c, \dots, \mathbf{e}_{last}\}$, where $\mathbf{e}_{last}$ is the last element of E . When a new final error is obtained, it will be added to set E as its last element. The extra term $\Delta\{\mathbf{e}_{last}\}$ will be added to update the learning law. The proposed learning law updating algorithm is given in algorithm 1.

3.2 Compensatory filter design

This section presents the the compensatory filter, and its design guidelines. The compensatory filter is designed as

$$\Delta = QL + (1 - Q)\left(-\sum_{i=1}^n \binom{n}{i} [P^{k \times i - 1} (-L^k Q^m)^i]\right) \quad (12)$$

where k , m , n are user-defined integers and $k \geq 1$, $m \geq 0$, $n \geq 0$. The design guideline for Δ is that it will not introduce any

Algorithm 1 Proposed learning law updating algorithm

Inputs:

dynamic parameters P , Q , L

reference $\mathbf{r}$

Initialization:

set iteration index as $j = 0$

set (2) as the current learning law

set $E = \{\}$

while *satisfactory $\mathbf{e}_j$ is not obtained* **do**

if *$\mathbf{e}_j$ is not converged* **then**

 1. continue the iterations with current learning law

 2. record $\mathbf{e}_j$

else

 1. add the new final error $\mathbf{e}_j$ to set E as the last element

 2. select a compensatory filter Δ

 3. Update the current learning law by adding the extra term $\Delta\{\mathbf{e}_{last}\}$

 4. set $j = j + 1$; set the updated learning law as the current learning law

end

end

design trade-offs and compromises which is adverse to the system performance. For example, an exact model inverse may be impossible to obtain if the model is non-minimum phase system, and instead, an approximation of the model inverse is used. This approximation is considered as a design trade-off between design feasibility and accuracy. In our case, the compensatory filter designed in (12) meets the following two conditions: 1) it does not introduce new dynamic parameters but just includes P , Q , L terms; 2) it does not include model inverse terms such as P^{-1} , Q^{-1} , L^{-1} .

Define a new variable

$$\Omega = (1 - P^k L^k Q^m)^n \quad (13)$$

and based on the binomial theorem, see Weisstein (2002), Ω can be expanded as

$$\Omega = \sum_{i=0}^n \binom{n}{i} (-P^k L^k Q^m)^i \quad (14)$$

where i is a dummy variable and

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \quad (15)$$

With (12) and (13), we have

$$P\Delta = PQL + (1 - Q)[1 - \Omega]. \quad (16)$$

Different k , m , n integers can be selected to generate different compensatory filters and this can lead to different tracking performance and error convergence, as observed in the numerical studies in Section 4. The design guideline for Ω is to achieve

$$\|\Omega\| \leq 1 \quad (17)$$

at the frequency range of the interest, and this will be further explained in later sections. Four example designs are given in Table 1.

3.3 Convergence analysis

With the new ILC algorithm, the initial learning law in (2) is used, and then the learning law is updated as (10). We first analyze the convergence by comparing $\mathbf{e}_a$ and $\mathbf{e}_b$.

Plug (10) into (1) to have

Table 1. Four compensatory filters

Design #	Integers	Ω	Δ
1	$n = 0$	1	QL
2	$n = 1, m = 0, k = 1$	$1 - PL$	L
3	$n = 1, m = 1, k = 1$	$1 - PLQ$	$QL(2 - Q)$
4	$n = 2, m = 1, k = 1$	$(1 - PLQ)^2$	$QL + (1 - Q)(2QL - PL^2Q^2)$

$$\begin{aligned}
\mathbf{e}_{j+1} &= \mathbf{e}_0 - P\{\mathbf{s}_{j+1}\} \\
&= \mathbf{e}_0 - P[Q(\mathbf{s}_j + L\mathbf{e}_j) + \Delta\{\mathbf{e}_a\}] \\
&= \mathbf{e}_0 - Q(P\{\mathbf{s}_j\}) - PQL\{\mathbf{e}_j\} - P\Delta\{\mathbf{e}_a\} \\
&= \mathbf{e}_0 - Q(\mathbf{e}_0 - \mathbf{e}_j) - PQL\{\mathbf{e}_j\} - P\Delta\{\mathbf{e}_a\} \\
&= Q(1 - PL)\{\mathbf{e}_j\} + (1 - Q)\mathbf{e}_0 - P\Delta\{\mathbf{e}_a\}.
\end{aligned} \quad (18)$$

Similarly, to obtain the final error $\mathbf{e}_b$ which is associated with the learning law in (10), we treat $\mathbf{e}_b = \mathbf{e}_j = \mathbf{e}_{j+1}$ and plug it into (18), and with (7), (8), (16), we have

$$\begin{aligned}
\mathbf{e}_b &= [1 - Q(1 - PL)]^{-1}(1 - Q)\{\mathbf{e}_0\} \\
&\quad - [1 - Q(1 - PL)]^{-1}P\Delta\{\mathbf{e}_a\} \\
&= \mathbf{e}_a - [1 - Q(1 - PL)]^{-1}P\Delta\{\mathbf{e}_a\} \\
&= (1 - [1 - Q(1 - PL)]^{-1}[PQL + (1 - Q)(1 - \Omega)])\{\mathbf{e}_a\} \\
&= \Omega\Gamma\{\mathbf{e}_a\}.
\end{aligned} \quad (19)$$

If (9) and (17) hold, then (19) indicates that $\|\mathbf{e}_b\| < \|\mathbf{e}_a\|$.

If the learning law is updated from (10) to (11), and the final error $\mathbf{e}_c$ which is associated with the learning law in (11) can be related to $\mathbf{e}_b$ as $\mathbf{e}_c = \Omega\Gamma\{\mathbf{e}_b\}$ by following the same derivation process in (18) and (19). Therefore, $\|\mathbf{e}_c\| < \|\mathbf{e}_b\|$ can be observed. That is, with this algorithm, the final errors in set E decrease monotonously in the chronological order which guarantees the learning convergence. The convergence rate within set E (reduction between two adjacent final errors) depends on the parameter Ω and Γ , as (19) indicates.

Eq. (3) and (18) indicates that the new ILC and the traditional ILC share the same convergence rate from iteration to iteration since $\mathbf{e}_j$ and $\mathbf{e}_{j-1}$ are related with the same Λ . Therefore, the stability condition in (5) is also applied to (18) with respect to $\mathbf{e}_j$. This is straightforward to observe, similar to the traditional ILC, in this control framework the new ILC generates learning signals only to modify the reference which will not affect the system stability.

4. NUMERICAL STUDIES

In this section, a drone model is used to simulate the trajectory tracking in its heading direction. The new ILC algorithm which uses the four different Δ in Table 1 are simulated and their results are compared to the learning with traditional ILC. The drone's closed-loop dynamics for position tracking is identified as an LTI system $P(z)$,

$$P(z) = \frac{0.01241z^{-1}}{0.9829 - 1.775z^{-1} + 0.8041z^{-2}} \quad (20)$$

where z is the discrete-time operator. The sampling time equals 0.1 seconds, and this is the suitable sampling time to implement with a quadrotor drone in practice. The stable learning filter L is designed to approximate the plant inverse P^{-1} ; and Q is a low-pass filter with a cutoff frequency of around 0.6 Hz. The low-pass filter is designed with a small bandwidth to provide more robustness to high-frequency noises. The bode plots of P , Q , L are given in Fig. 2. The two to-be-tracked references are given

in Fig. 3, where reference #1 contains several sharp corners which is challenging for the drone to follow, and reference #2 is a sinusoidal signal. Three tracking scenarios are simulated. In scenario 1 and scenario 2, the system is tracking reference #1, and in scenario 3, the system is tracking reference #2.

4.1 Scenario 1

In this scenario, an ideal case is simulated without introducing modelling uncertainties and noises. 14 repetitive iterations are carried out for the drone tracking reference #1, and the learning law with each of the 4 compensatory filter in Table 1 is implemented. The 2-norm of the tracking error for each iteration is recorded. To compare results, the traditional ILC is also implemented. The abbreviation ‘TILC’ in the figures and table stands for traditional ILC. In this study, we set $\sigma = 0.00001$. The 2-norm of the tracking error is given in Fig. 4, where it shows that at iteration 5, the error with traditional ILC converges, and $\mathbf{e}_a$ is obtained; The proposed algorithm then updates the learning law, and each of the 4 new learning laws can further decrease the error; at iteration 9 with design #1, and iteration 10 with design #2~4, the error converges once more and $\mathbf{e}_b$ is obtained.

The trajectory tracking with design #1 is given in Fig. 5 and Fig. 6, and it shows that with iterations, the tracking is getting closer to the reference. Results of partial iterations are hidden for clear view purposes since they can be overlapped with the current curves in the figure. Note that this reference contains sharp corners which can be dynamically infeasible, and therefore the steady-state error can be persistent.

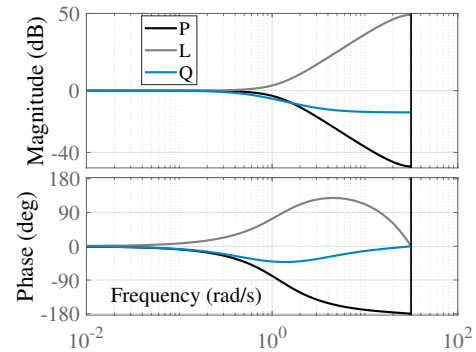


Fig. 2. Bode plots of P , Q , L

The bode plots of the dynamic parameters Λ , Γ , and the three Ω designs in Table 1 are given in Fig. 7, where it shows that at the low frequency range (< 1 Hz), the magnitude of these parameters are all smaller than 1, and this meets the design guidelines in (5), (9), (17). Not that in this study, we only investigated the frequency range of our interest, which is at the low-frequency range. However, this method has the potential to be effective for all frequency range if parameters can be designed such that conditions in (5), (9), (17) are guaranteed

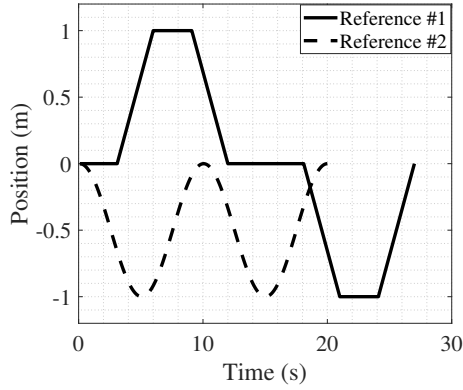


Fig. 3. Trajectory reference

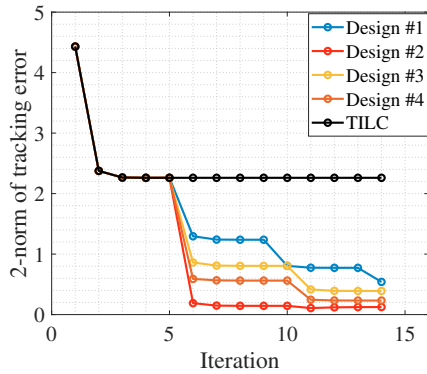


Fig. 4. Scenario 1: tracking errors (2-norm) over iterations

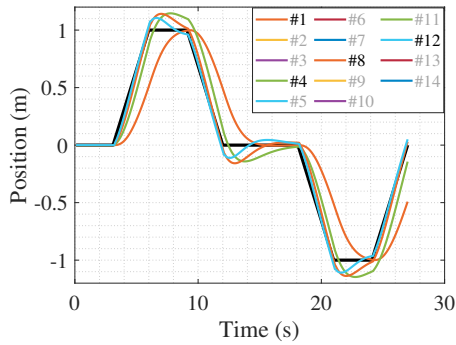


Fig. 5. Scenario 1: trajectory tracking with design #1.

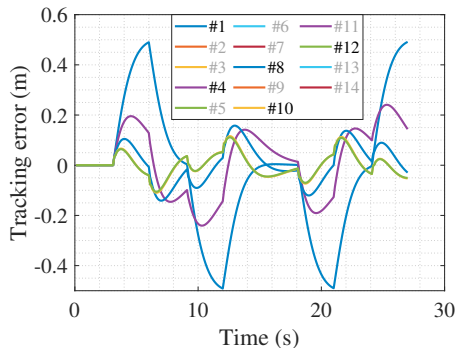
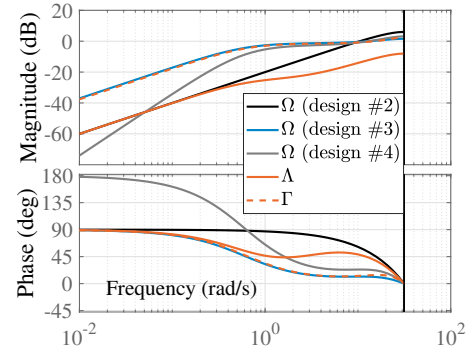


Fig. 6. Scenario 1: tracking errors with design #1

for the whole frequency range. And this will be explored in the future work.

Fig. 7. Bode plots of Ω , Λ , Γ

4.2 Scenario 2

In this scenario, while keeping the same settings used in scenario 1, some model mismatch is introduced in the simulation. A nominal model $\hat{P}(z)$ is generated as

$$\hat{P}(z) = \frac{0.01241z^{-1}}{0.7175 - 1.292z^{-1} + 0.587z^{-2}}. \quad (21)$$

This study does not explicitly consider the robustness to model mismatch, however, we reasonably assume the model mismatch is bounded and small for derivation purposes. The parameters L and Δ are designed based on $\hat{P}(z)$, while the learning signal is injected to the actual model $P(z)$.

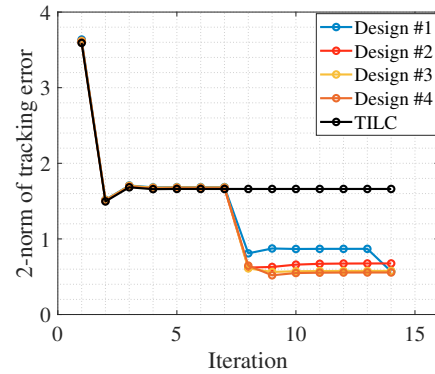


Fig. 8. Scenario 2: tracking errors (2-norm) over iterations

A random measurement noise is generated and added to the output iteratively. With the same reference, parameter designs, and simulation settings as in scenario 1, the drone's trajectory tracking is simulated, and the 2-norm of the tracking error is given in Fig. 8, where it shows that with the traditional ILC, the error converges at iteration 7, and remains unchanged afterwards. With the new ILC algorithm, after iteration 7, the learning law is updated, and all the 4 learning laws are able to further reduce the error. With design #1, the error converges at iteration 13, and with design #2~#4, the error converges at iteration 14. Note that the error might not be decrease monotonously, such as the error at iteration 3 is larger than that at iteration 2, and this is caused by the model mismatch. Also, this phenomenon will not affect the convergence. As the first

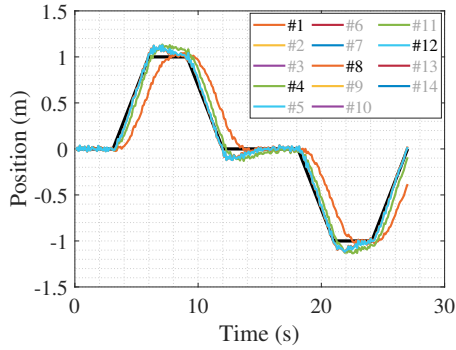


Fig. 9. Scenario 2: trajectory tracking with design #1

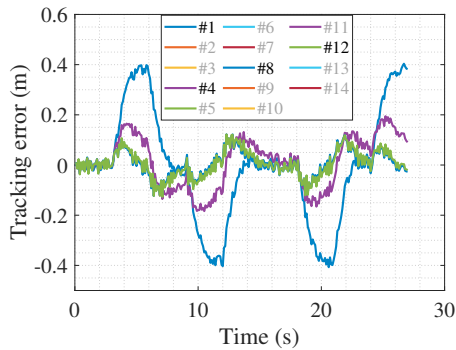


Fig. 10. Scenario 2: trajectory tracking error with design #1

condition in (6) is not met, further iterations will be conducted until the error converges. The tracking with design #1 is given in Fig. 9 and Fig. 10, and it indicates that overall, the error is reduced iteratively.

Table 2. Selected 2-norm of the tracking errors

Iteration #	Design #1	Design #2	Design #3	Design #4	TILC
Scenario 1					
5	2.26	2.26	2.26	2.26	2.26
6	1.29	0.19	0.86	0.59	2.26
10	0.81	0.14	0.81	0.56	2.26
11	0.77	0.11	0.42	0.25	2.26
Scenario 2					
7	1.67	1.67	1.67	1.67	1.67
8	0.65	0.62	0.61	0.65	1.67
14	0.60	0.68	0.58	0.56	1.67
Scenario 3					
7	1.60	1.60	1.60	1.60	1.60
8	0.75	0.54	0.49	0.53	1.60
14	0.49	0.60	0.47	0.43	1.60

4.3 Scenario 3

In scenario 3, the drone is tracking reference #2 while other settings (model mismatch and measurement noise) are kept the same as that in scenario 2. The 2-norm of the tracking error is given in Fig. 11; the tracking is given in Fig. 12 and Fig. 13. The results from scenario 3 also indicate that the proposed ILC outperforms the traditional ILC, which proves that the proposed ILC is applicable to different references.

To provide some insights statistically, partial 2-norm values recorded from the simulation are given in Table 2 and it indicates that in all the three scenarios, the 4 new designs can

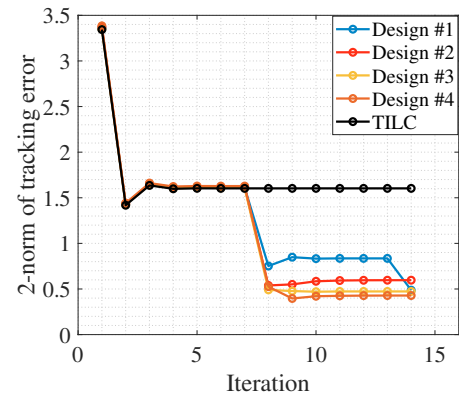


Fig. 11. Scenario 3: tracking errors (2-norm) over iterations

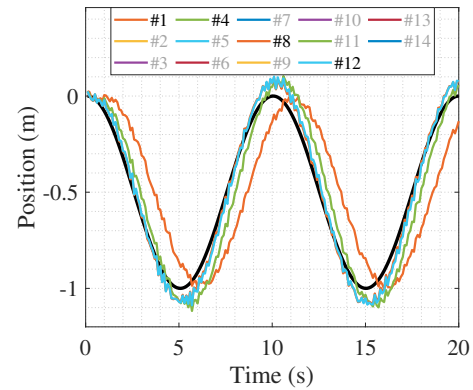


Fig. 12. Scenario 3: trajectory tracking with design #1

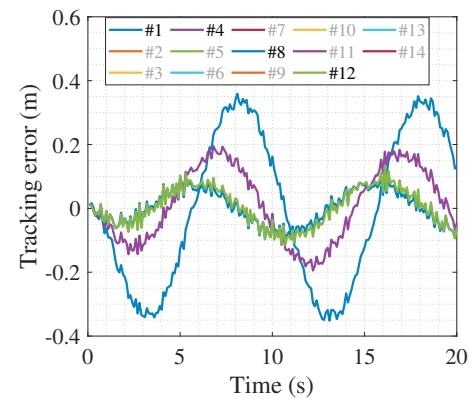


Fig. 13. Scenario 3: trajectory tracking error with design #1

further reduce the error to a large extent compared to that with traditional ILC. Within the performed 14 iterations, the results from scenario 1 indicate that the learning law with design #2 has the best performance, and in scenario 2 and scenario 3, the learning law with design #4 has the best performance regarding the tracking accuracy in the sense of 2-norm. Note that the noise and model mismatch exist in scenario 2, and this can lead to different initial tracking performance with that in scenario 1. Also, when compare the 2-norm values between different scenarios, the factor that the number of tracking points is different in scenario 2 and scenario 3 should also be considered. The results verify that the proposed algorithm is effective to

break the convergence with traditional ILC and have final error reduction.

5. CONCLUSIONS

This paper presents a new ILC algorithm with theoretically-proved performance enhancement compared to the traditional ILC. As explained in the paper, in traditional ILCs, a low-pass filter is usually needed to obtain good robustness to modeling uncertainties and noises, which results in a final steady-state error as a trade-off. The proposed new ILC can break the convergence of traditional ILC and further reduce the final error by updating the learning law. The learning law is updated by adding a compensating term which consists of a compensatory filter and the obtained final error. Extensive studies on four selected designs have been conducted, and the results are compared to that of the traditional ILC design. It shows that the proposed learning algorithm is able to further reduce the final error effectively.

REFERENCES

- Adlakha, R. and Zheng, M. (2021). A Two-Step Optimization-Based Iterative Learning Control for Quadrotor Unmanned Aerial Vehicles. *Journal of Dynamic Systems, Measurement, and Control*, 143(7). doi:10.1115/1.4049566. URL <https://doi.org/10.1115/1.4049566>. 071006.
- Arimoto, S., Kawamura, S., and Miyazaki, F. (1984). Bettering operation of robots by learning. *Journal of Robotic systems*, 1(2), 123–140.
- Bien, Z. and Xu, J.X. (2012). *Iterative learning control: analysis, design, integration and applications*. Springer Science & Business Media.
- Chen, Y. and Moore, K.L. (2002). Harnessing the nonrepetitiveness in iterative learning control. In *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002., volume 3, 3350–3355. IEEE.
- Chen, Z., Hajidavalloo, M., Li, Z., and Zheng, M. (2022a). A Cascaded Learning Framework for Road Profile Estimation Using Multiple Heterogeneous Vehicles. *Journal of Dynamic Systems, Measurement, and Control*. doi:10.1115/1.4055041. URL <https://doi.org/10.1115/1.4055041>.
- Chen, Z., Liang, X., and Zheng, M. (2020). Knowledge transfer between different uavs for trajectory tracking. *IEEE Robotics and Automation Letters*, 5(3), 4939–4946. doi: 10.1109/LRA.2020.3004776.
- Chen, Z., Liang, X., and Zheng, M. (2022b). Iterative learning for heterogeneous systems. *IEEE/ASME Transactions on Mechatronics*, 27(3), 1510–1521. doi: 10.1109/TMECH.2021.3085211.
- Ge, X., Stein, J.L., and Ersal, T. (2017). Frequency-domain analysis of robust monotonic convergence of norm-optimal iterative learning control. *IEEE Transactions on Control Systems Technology*, 26(2), 637–651.
- Harte, T., Hästönen, J., and Owens*, D. (2005). Discrete-time inverse model-based iterative learning control: stability, monotonicity and robustness. *International Journal of Control*, 78(8), 577–586.
- Lee, J.H., Lee, K.S., and Kim, W.C. (2000). Model-based iterative learning control with a quadratic criterion for time-varying linear systems. *Automatica*, 36(5), 641–657.
- Li, M., Chen, T., Cheng, R., Yang, K., Zhu, Y., and Mao, C. (2021). Dual-loop iterative learning control with application to an ultraprecision wafer stage. *IEEE Transactions on Industrial Electronics*.
- Lin, C.Y., Sun, L., and Tomizuka, M. (2015). Robust principal component analysis for iterative learning control of precision motion systems with non-repetitive disturbances. In *2015 American Control Conference (ACC)*, 2819–2824. IEEE.
- Liu, N. and Alleyne, A. (2016). Iterative learning identification/iterative learning control for linear time-varying systems. *Journal of Dynamic Systems, Measurement, and Control*, 138(10), 101005.
- Meng, D. (2018). Convergence conditions for solving robust iterative learning control problems under nonrepetitive model uncertainties. *IEEE transactions on neural networks and learning systems*, 30(6), 1908–1919.
- Merry, R., van de Molengraft, R., and Steinbuch, M. (2008). Iterative learning control with wavelet filtering. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 18(10), 1052–1071.
- Mishra, S. and Tomizuka, M. (2009). Projection-based iterative learning control for wafer scanner systems. *IEEE/ASME Transactions on Mechatronics*, 14(3), 388–393.
- Purwin, O. and D’Andrea, R. (2009). Performing aggressive maneuvers using iterative learning control. In *2009 IEEE international conference on robotics and automation*, 1731–1736. IEEE.
- Sun, L., Chen, X., and Tomizuka, M. (2014). Selective iterative learning control with non-repetitive disturbance rejection. In *Proceedings of 2014 International Symposium on Flexible Automation*.
- Wang, C., Zheng, M., Wang, Z., Peng, C., and Tomizuka, M. (2018). Robust iterative learning control for vibration suppression of industrial robot manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 140(1), 011003.
- Wang, D., Ye, Y., and Zhang, B. (2014). *Practical iterative learning control with frequency domain design and sampled data implementation*. Springer.
- Weisstein, E.W. (2002). Binomial theorem. <https://mathworld.wolfram.com/>.
- Xie, S. and Ren, J. (2018). Note: Precision control of nanopositioning stage: An iterative learning-based model predictive control approach. *Review of Scientific Instruments*, 89(7), 076103.
- Xu, J.X., Lee, T.H., and Zhang, H.W. (2002). On the ilc design and analysis for a hdd servo system. *IFAC Proceedings Volumes*, 35(1), 265–270.
- Ye, Y. and Wang, D. (2006). Learning more frequency components using p-type ilc with negative learning gain. *IEEE Transactions on Industrial Electronics*, 53(2), 712–716.
- Yeon, J.S., Park, J.H., Son, S.W., and Lee, S.H. (2009). Model-based iterative learning control for industrial robot manipulators. In *2009 IEEE International Conference on Automation and Logistics*, 24–28. IEEE.
- Yin, C., Xu, J.X., and Hou, Z. (2010). A high-order internal model based iterative learning control scheme for nonlinear systems with time-iteration-varying parameters. *IEEE Transactions on Automatic Control*, 55(11), 2665–2670.
- Zheng, M., Wang, C., Sun, L., and Tomizuka, M. (2017). Design of arbitrary-order robust iterative learning control based on robust control theory. *Mechatronics*, 47, 67–76.
- Zheng, M., Zhang, F., and Liang, X. (2018). A systematic design framework for iterative learning control with current feedback. *IFAC Journal of Systems and Control*, 5, 1–10.