Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Model-free inversion-based iterative learning control algorithm with adaptive gain: achieving better robustness and convergence

Zhicheng Kou | Jinggao Sun*

Key Laboratory of Smart Manufacturing in Energy Chemical Process, East China University of Science and Technology, Shanghai, China

Correspondence
Jinggao Sun*, Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China. Email: jgsun@ecust.edu.cn

Present Address
East China University of Science and Technology, No.130 Meilong Road, Xuhui District, Shanghai, China.

Summary

The main objective of this study is to address the challenge of simultaneously ensuring robustness and convergence performance in model-free inversion-based iterative learning control. Initially, this research provides a mathematical analysis of the sources of errors in the iterative process, followed by proposing a design guideline to enhance both convergence speed and the final value error. Based on the design guideline, a gain design method associated with the number of iterations is proposed, resulting in a novel model-free inversion-based iterative learning control algorithm. Subsequently, a robustness analysis of the proposed algorithm is conducted. Finally, a comprehensive simulation and numerical comparison of the proposed algorithm with existing similar algorithms are presented to demonstrate the superior performance of the proposed control algorithm.

KEYWORDS:
iterative learning control, model-free inversion-based algorithm, gain design, robustness

1 | INTRODUCTION

Learning is very important for human beings. When there is no regular summary, people tend to summarize the pattern through historical experience to obtain better results in the next execution. Drawing on this idea in the control domain, Iterative Learning Control (ILC) has been developed[1]. The basic idea of ILC is to optimize the control signal for the next iteration by using data from previous control processes to produce a better control signal. This control method is suitable for repetitive processes, and performance can be gradually improved as the number of iterations increases. ILC has been shown to be effective in a variety of industrial systems, including linear motors[2], cooperative systems[3,4], distributed parameter system[5], general nonlinear systems[6,7,8,9], and a variety of communication systems[10,11,12].

Over time, various ILC algorithms have emerged. Traditional ILC algorithms, such as PID-like ILC, use the input of the current iteration and tracking error to determine the input of the next iteration, without using the system's model information. As a result, these algorithms require a higher number of iterations to converge[13]. Therefore, traditional ILC algorithms are only suitable for specific applications where the number of iterations is not strictly limited.

Suppose the system model is estimated from historical data during the iterative process, and the estimated model information is introduced during the next control signal calculation. In that case, the convergence rate can be significantly increased. With the help of this idea, the model-free inversion-based iterative control (MFIIC) algorithm, which approximates the inverse model of the system through historical data, is proposed[14]. If the control process is a noise-free environment, the estimated inverse model is accurate at this point, so the error is able to converge to very small values after a few iterations[14]. In work [15][15], the

MFIIC and its related algorithms are compared in detail. The MFIIC algorithm has been applied in several scenarios, such as piezoelectric devices [16], in 3D nanopositioning [17], or 2D stick-slip positioners [18].

Disturbances are inevitable in practical applications. For instance, the operation of a particular system may entail random disturbances, and the output signal's measurement may be contaminated by random noise and inaccurate measurements. In the case of MFIIC, the presence of output noise in practical applications often leads to less accurate inversion model identification [19]. As a result, the MFIIC algorithm's performance is seriously degraded, and there are large error fluctuations [20]. In a recent study, [21] explored the relationship between the robustness of the MFIIC algorithm and noise, specifically.

The MFIIC algorithm's performance is known to decrease sharply in the presence of noise in practical applications [19,20]. To address this issue, a model-free inversion-based learning iterative control based on nonlinear function gain (NLIIC) has been proposed [21]. The NLIIC algorithm adjusts the constant gain of the MFIIC algorithm to a nonlinear function and uses nonlinear gain for iteration only when the output is larger than a threshold. This approach allows the gain to vary along the time axis based on the reference signal versus the output, leading to robust performance. However, the NLIIC algorithm's convergence speed is greatly reduced in small or noise-free environments, despite enhancing the system's robustness through the combination of the MFIIC algorithm and the nonlinear gain. Furthermore, the NLIIC algorithm has high requirements for the reference trajectory, as the reference trajectory is required to avoid small values.

Inspired by the work [22] [22], it is worth noting that in addition to designing gains on the time axis, setting different gains on the iteration axis is also a strategy that can significantly improve the algorithm's control performance.

The role of different gain settings in the iteration is distinct. While larger gains can expedite convergence, they do not ensure accurate error control. As the iterative process gradually reduces the error, different gain settings may be necessary at various iteration stages to achieve better control. Continuing to use a large gain when the iterative error converges to a relatively small value for the MFIIC algorithm would over-adjust the input signal for the next iteration, causing the control signal to deteriorate (error divergence). At this stage, the optimal way to improve the control signal further towards the ideal control signal is to reduce the gain. This paper proposes an adaptive gain model-free inversion-based iterative learning control algorithm from this perspective.

This work makes the following contributions:

- C1: The error sources of the MFIIC algorithm are derived to analyze how to design the gain to achieve better convergence with guaranteed robustness. Then the criteria for the adaptive gain setting of the MFIIC algorithm are proposed.
- C2: A specific adaptive gain design is proposed to form a new model-free inversion-based iterative learning control algorithm. This algorithm can achieve smaller final value errors and faster convergence while ensuring robustness. In addition, the proposed algorithm is not strictly limited to the reference trajectory.
- C3: A comprehensive simulation analysis and data comparison of the algorithm under different noise conditions and different reference trajectory tracking difficulties are carried out to illustrate the advantages of this algorithm compared with similar algorithms.

2 | PROBLEM FORMULATION

2.1 | Model-free inversion-based learning control algorithms

For the model-free inversion-based iterative learning control algorithm, the following discrete linear time-invariant control system needs to be considered:

$$y_i(k) = G(z)\, u_i(k), \tag{1}$$

where $G(z)$ is the unknown model while having $G\left(e^{j\omega_k}\right) \neq 0$ through $\omega_k = \frac{2\pi k}{N}$, $\forall k \in \{0, \cdots, N-1\}$. The input signal, output signal, and error signal denoted as $u(k)$, $y(k)$, and $e(k)$, respectively.

Assumption 1. The reference trajectory $r(k)$ is a known time-invariant and bounded signal. The DFT transform of the reference trajectory is $R(k)$.

Assumption 2. In an environment with output noise, the output signal $Y_i(k) = G\left(e^{j\omega_k}\right) U_i(k) + D_i(k)$, where $D_i(k)$ is the unknown interference, the tracking error is $E_i(k) = R(k) - Y_i(k)$.

Assumption 3. The following forms of output noise are considered in this work: $D(k) \in N\left(0, A^2\right)$, A is the standard deviation of the noise signal. The maximum amplitude of the noise is δ, namely $\max(|D(k)|) \leq \delta$.

Under the conditions of $Assumptions$ 1to 3, the control law of this MFIIC algorithm is as follows:
if $Y_i(k) \neq 0$ and $R(k) \neq 0$, then

$$U_{i+1}(k) = U_i(k) + \rho_i \frac{U_i(k)}{Y_i(k)} E_i(k), \tag{2a}$$

if Y_i and R_i do not satisfy the above conditions, then

$$U_{i+1}(k) = U_i(k), \tag{2b}$$

where $\rho_i = 1$.

Remark 1. The MFIIC algorithm, being an inverse model-based approach, exhibits a faster convergence rate with the error converging to a lower level within a few iterations. However, a major drawback of this algorithm is that its accuracy in identifying the inverse model deteriorates significantly in the presence of noise. Consequently, the correctness of the inverse model identification cannot be guaranteed, leading to the possibility of error convergence growing with the number of iterations. Therefore, in the presence of noise, the reliability of this strategy is reduced.

Under the conditions of $Assumptions$ 1to 3, the control law of NLIIC algorithm is the same as MFIIC algorithm except for the different design method of gain. The gain of NLIIC is follows:

$$\rho(|Y|) = \begin{cases} 1, & |Y| > \epsilon \\ \frac{1}{2}\left(1 - \cos\left(\frac{\pi}{\epsilon}|Y|\right)\right), & |Y| \leq \epsilon \end{cases}, \tag{3}$$

where ϵ is a constant value, $\epsilon > 0$.

Remark 2. NLIIC solves the problem of occurrence of non-convergence when noise is present in MFIIC. However, if the noise is very large, the non-convergence still occurs in NLIIC. In addition, due to the introduction of the nonlinear function, the convergence speed is also much slower than MFIIC. The specific disadvantages of the NLIIC algorithm will be further analyzed in Section 4.

The NLIIC algorithm has strict requirements and restrictions on the reference trajectory, which will converge only when the reference trajectory satisfies the following conditions:[21]

$$\begin{aligned} &inf\ \rho(|Y|) \triangleq \rho > \tfrac{\alpha-1}{\alpha}, \\ &sup\ \rho(|Y|) \triangleq \rho \leq 1, \end{aligned} \tag{4a}$$

$$|R| \geq \alpha\delta + \Delta\delta \tfrac{\alpha}{\alpha-1}, \tag{4b}$$

where $\alpha > 1$, ρ is the learning gain of NLIIC, R is the reference trajectory.

However, the reference trajectory often does not satisfy the strict requirements, especially for some high precision engineering applications. This work removes the strict restriction on the reference trajectory.

2.2 | The requirements of model-free inversion-based iterative learning control

In fact, both the MFIIC algorithm and the NLIIC algorithm have their own drawbacks. Therefore, in a noisy environment, a model-free inversion-based iterative learning control algorithm is needed that can simultaneously satisfy the following requirements:

R1: The algorithm maintains the fast convergence speed of the MFIIC algorithm.

R2: The algorithm ensures a high level of robustness.

R3: The algorithm guarantees a very low convergence error.

R4: The algorithm does not have strict requirements on the environment and reference trajectory.

However, the MFIIC algorithm fails to guarantee (R2) and (R3), while the NLIIC algorithm fails to guarantee (R1), (R3), and (R4).

3 | ANALYSIS AND DESIGN OF ALGORITHM

3.1 | Analysis of gain

In the iterative process, the error can be considered to be caused by the difference between the ideal control signal and the actual control signal, which can be analyzed by considering the difference $\Delta U_i(k)$ and the relationship between error and the gain setting to reduce the error from the perspective of reducing $\Delta U_i(k)$.

Specifically, we have

$$\Delta U_i(k) = U_d(k) - U_i(k), \tag{5}$$

considering the noise environment in Assumption 3,

$$\begin{aligned} \Delta U_{i+1}(k) =& U_d(k) - \left(U_i(k) + \rho_i(k) \frac{U_{i-1}(k)}{Y_{i-1}(k)} \left(Y_d(k) - Y_i(k) \right) \right) \\ =& \Delta U_i(k) - \rho_i(k) \widehat{G_i(k)^{-1}} \left(G_i \Delta U_i(k) - D_i(k) \right), \end{aligned} \tag{6}$$

$$\Delta U_{i+1}(k) = \Delta U_i(k) \left(1 - \rho_i(k) \widehat{G_i(k)^{-1}} G \right) + \rho_i(k) \widehat{G_i(k)^{-1}} D_i(k). \tag{7}$$

Taking the vector norm to both sides of Equation (7),

$$\begin{aligned} ||\Delta U_{i+1}(k)||^2 = & \Delta U_i(k)^2 \left(1 - \rho_i(k) \widehat{G_i(k)^{-1}} G \right)^2 \\ & + 2\Delta U_i(k) \left(1 - \rho_i(k) \widehat{G_i(k)^{-1}} G \right) \rho_i(k) \widehat{G_i(k)^{-1}} D_i(k) + \rho_i(k)^2 \widehat{G_i(k)^{-1}}^2 D_i(k)^2. \end{aligned} \tag{8}$$

Taking the conditional expectation of (8) with respect to output noise d_{i-1},

$$\mathbb{E}\left[||\Delta U_{i+1}||^2 | d_{i-1} \right] = \Delta U_i(k)^2 \left(1 - \rho_i(k) \widehat{G_i(k)^{-1}} G \right)^2 + \rho_i(k)^2 \widehat{G_i(k)^{-1}}^2 D_i(k)^2. \tag{9}$$

To make the analysis easier, noting $\mathbb{E}_1 = \Delta U_i(k)^2 \left(1 - \rho_i(k) \widehat{G_i(k)^{-1}} G \right)^2$, $\mathbb{E}_2 = \rho_i(k)^2 \widehat{G_i(k)^{-1}}^2 D_i(k)^2$.

From Equation (9), it can be seen that the factors affecting $\Delta U_i + 1(k)$ can be controlled by $\widehat{G_i(k)^{-1}}$ and $\rho_i(k)$. In the early phase of iteration, the error is mainly caused by $\mathbb{E}_1$ because $\Delta U_i(k)$ is relatively large. Therefore, it is necessary to adjust $\rho_i(k)$ to make it close to 1 and then $\mathbb{E}$ is smaller. At the later stage of the iterative process, when $\Delta U_i(k)$ is small, the main source of error is $\mathbb{E}_2$. Therefore, it is necessary to adjust $\rho_i(k)$ to a smaller value, which can make the error further reduced and make the adjustment amplitude of the control signal smaller to ensure the robustness.

Based on the above analysis, the following guidelines can be derived for setting the gain of the model-free inversion-based iterative learning algorithm, which would allow the algorithm to satisfy both (R1)-(R4):

- The gain should be selected in the range $0 < \rho \leq 1$ throughout the iteration.
- In the early stages of iteration, the gain should be set as close to 1 as possible, which would lead to faster convergence.
- In the later stages of iteration, the gain should be set to a smaller value to ensure the algorithm's final value error and robustness.

For the MFIIC algorithm and NLIIC algorithm, Remark 3 and Remark 4 can be derived from Equation (9.

Remark 3. For the MFIIC algorithm, the variations of $U_i(k)$ will cause a large deviation from the inverse model when the output signal $Y_i(k)$ is very small. This will lead to the occurrence of $1 - \rho_i(k) \widehat{G_i(k)^{-1}} G > 1$ and $\Delta U_{i+1}(k) > \Delta U_i(k)$. As a result, this can make MFIIC's algorithm diverge in noisy environments.

Remark 4. For the NLIIC algorithm, the nonlinear gain Equation 4a can result in excessively small values for $\rho_i(k)$, which can slow down error reduction in the initial iteration stage. Furthermore, some gains remain relatively large in the later stage of iteration, which can cause $\mathbb{E}_2$ to be too large and result in inadequate final value error reduction.

3.2 | Inversion-based model-free iterative learning control with adaptive gain

In order to adapt the gain to the changing iterative process, this work proposes a form of gain in the form of derivatives of the number of iterations according to the guidelines in Section 3.1. This gain form is used in the model-free inversion-based iterative learning control algorithm to obtain model-free inversion-based iterative learning control with adaptive gain, denoted as AG-MFIIC.

The gain of the AG-MFIIC algorithm is set as follows:

$$\rho_i (k) = \frac{1}{i}, \tag{10}$$

where i is the number of iterations, $i \geq 1, \forall k \in \{0, \cdots, N-1\}$. The complete steps of the AG-MFIIC algorithm can be obtained as shown in Algorithm 1.

Algorithm 1 Model-free inversion-based iterative learning control algorithm with adaptive gain

```
1: Select $u_0 (k) = r$;
2: Apply the input $u_i(k)$ to the system and record the tracking error $e_i(k) = r(k) - y_i(k)$.
3: Apply DFT to obtain $U_i (k)$ and $E_i (k)$;
4: Determine $U_{i+1}(k)$ using Equation (2a) and (2b)with $\rho_i (k)$ given by Equation (10).
5: Do IDFT to obtain $u_{i+1} (k)$;
6: Set $i = i + 1$, go to Step 2.
```

3.3 | Robustness analysis

Theorem 1. Consider Algorithm 1 with the Assumption 1 to 3 are satisfied. If the gain ρ is set to Equation (10), the sequence of control signals $\{U_i\}$ generated by Algorithm 1 will result in $\lim_{i\to\infty} E_i \in \mathcal{B}_{re}(0)$, where $r_e = \frac{i\Delta\delta Y_i}{Y_i - D_i}$.

Proof. In environments with output noise, the output signal

$$Y_i (k) = GU_i (k) + D_i (k), \tag{11}$$

$$E_i (k) = R (k) - Y_i (k). \tag{12}$$

Substituting Equation (11) into Equation (12),

$$\begin{aligned}
E_{i+1} &= R - GU_{i+1} - D_{i+1} = R - G\left(U_i + \rho_i \frac{U_i}{Y_i} E_i\right) - D_{i+1} \\
&= \left(1 + \rho_i \frac{GU_i}{Y_i}\right) E_i + D_i - D_{i+1} \\
&= \left(\frac{\left(1-\rho_i\right) Y_i + \rho_i D_i}{Y_i}\right) E_i + D_i - D_{i+1} \\
&= \left(\frac{\left(1-\rho_i\right)\left(R - E_i\right) + \rho_i D_i}{Y_i}\right) E_i + D_i - D_{i+1}.
\end{aligned} \tag{13}$$

Assume that

$$\kappa_i = |\frac{\left(1-\rho_i\right)\left(R - E_i\right) + \rho_i D_i}{R - E_i}|, \tag{14}$$

then

$$|E_{i+1}| \leq \kappa_i |E_i| + |D_i - D_{i+1}| = \kappa_i |E_i| + \Delta\delta, \tag{15}$$

If $|E_{i+1}|$ gradually converges, there is $|E_{i+1}| \leq \kappa_i |E_i| + \Delta\delta \leq |E_i|$. To make $\kappa_i |E_i| + \Delta\delta \leq |E_i|$, i.e,

$$\left(1 - \kappa_i\right) |E_i| \geq \Delta\delta. \tag{16}$$

We can substituting Equation (14) into Equation (16),

$$\left(\rho_i - \frac{\rho_i D_i}{Y_i}\right)|E_i| \geq \Delta\delta, \tag{17}$$

$$|E_i| \geq \frac{i\Delta\delta Y_i}{Y_i - D_i}, \tag{18}$$

Therefore, the boundary value of convergence is $|E_i| = \frac{i\Delta\delta Y_i}{Y_i - D_i}$, i.e., when the convergence error $|E_i| \geq \frac{i\Delta\delta Y_i}{Y_i - D_i}$, then it will continue to converge to $|E_i| \leq \frac{i\Delta\delta Y_i}{Y_i - D_i}$. Therefore, the proof of Theorem 1 is completed.

□

4 | SIMULATION COMPARISON

4.1 | Simulation Settings

Simulations is performed using a mathematical model extracted from a linear motor. By fitting a polynomial to the frequency response of the closed-loop system with a sampling rate of 10 KHz, the following fitted fourth-order model is finally obtained[14]:

$$G(z) = \frac{-0.02\,(z - 1.664)\,(z - 0.648)\,(z + 0.036)}{\left(z^2 - 1.804z + 0.835\right)\left(z^2 - 1.599z + 0.764\right)}. \tag{19}$$

In order to provide a comprehensive comparison of the proposed algorithms, simulations are performed under various noise environments and tracking trajectories of different difficulties. The reference trajectory is set as a chirp signal, and the environments are set with different reference signal frequencies and noise standard deviations. The step size is set to N=2000 and the sampling period is set to T=0.001. The chirp reference trajectory at 100Hz is shown in Figure 1.

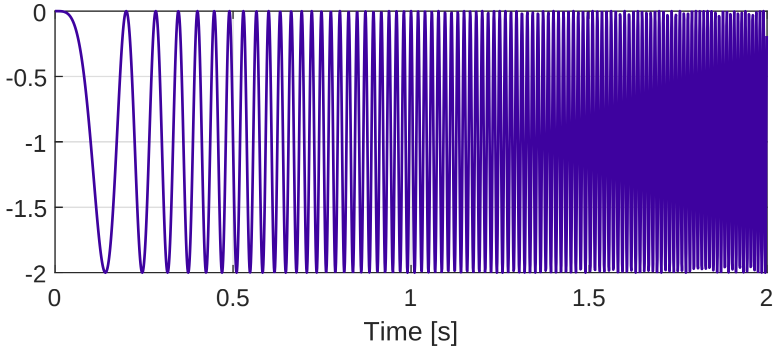

Figure 1 Reference trajectory (chrip signal) with the frequency of 100 Hz and the amplitude of 1.

In the simulations, consider the output noise described in Assumption 3, the standard deviation A of random noise is set to 0.02, 0.1, 0.5, 1 respectively. Meanwhile, the frequencies of the reference trajectory are set to 100, 1000, 10000Hz respectively.

4.2 | Simulation results and comparisons

The root mean square (RMS) error variations of the simulation results are presented in Figure 2 to Figure 4. The simulation results indicate that the AG-MFIIC and NLIIC algorithms exhibit strong robustness and do not experience significant error fluctuations, while the robustness of the MFIIC algorithm decreases as noise increases. Additionally, the simulation results were analyzed based on two aspects: convergence speed and final value error.

Comparison of convergence speed: In all environments, except in Figure 4(a) (10000 Hz, A=0.02), the AG-MFIIC algorithm shows the best convergence speed. The slower convergence in this environment can be attributed to the gradually decreasing

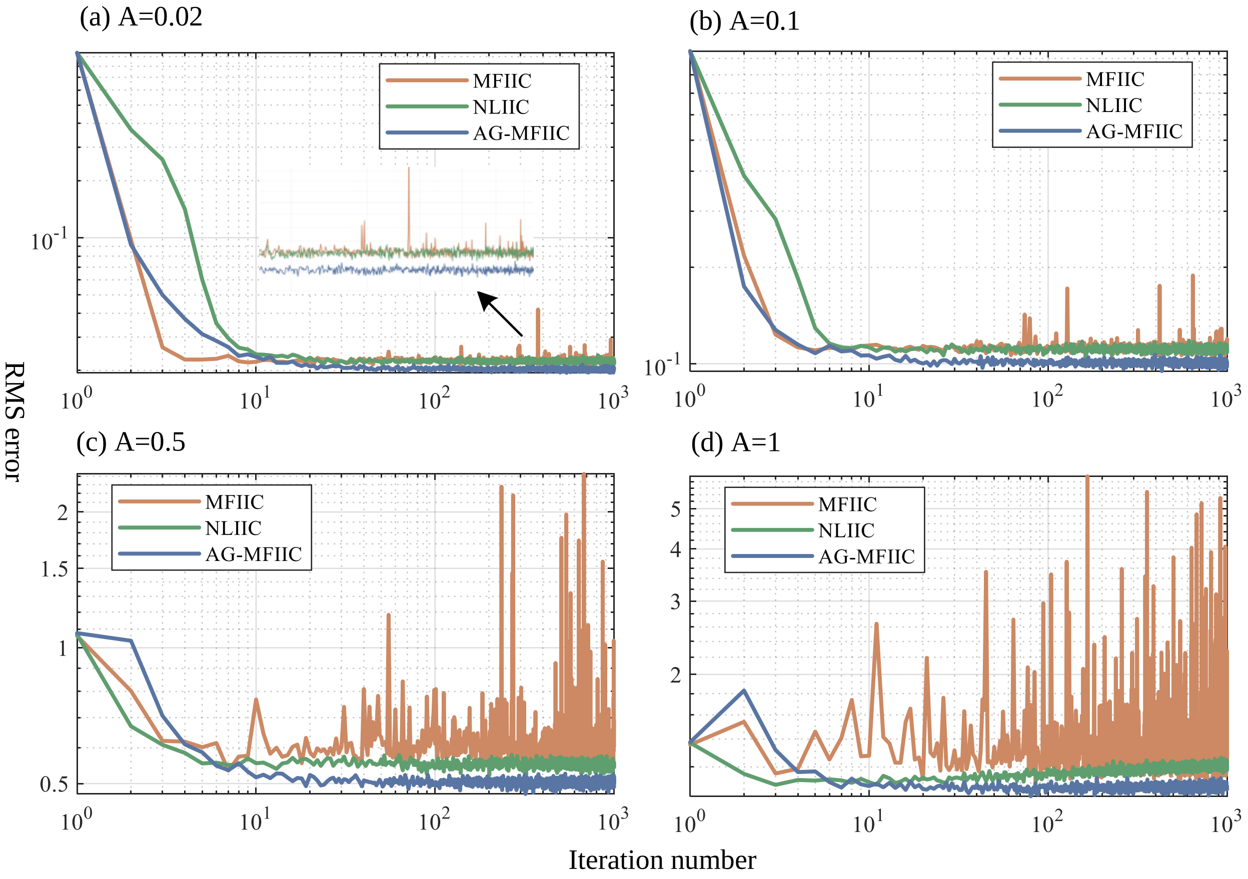

Figure 2 Comparison of the RMS error of different algorithms in the noisy environment with the reference trajectory of 100Hz

gain that slows down the convergence of the error. This situation can be improved by further enhancing the gain settings, which will be addressed in future work.

Comparison of the convergence final value error: The AG-MFIIC algorithm performs the best, followed by the NLIIC algorithm, while the MFIIC algorithm exhibits the worst performance. This can be attributed to the smaller gain setting of the AG-MFIIC algorithm, which allows for further tuning of the control signal, bringing it closer to the ideal control signal than the other algorithms.

In order to more clearly compare and analyze the control performances of different iterative processes, this work uses the mean root mean square (M-RMS) errors at different stages for the comparison of convergence. The M-RMS error is calculated as

$$\mathscr{E}_p = \frac{\sum_{i=n}^{i=n+m} E_{rms}}{m}. \tag{20}$$

The whole iterative process is divided into 3 stages: $i = 1 \rightarrow i = 20$, $i = 21 \rightarrow i = 100$, and $i = 101 \rightarrow i = 1000$. The M-RMS error of these stages are recorded as $\mathscr{E}_1$, $\mathscr{E}_2$, $\mathscr{E}_3$, where

$$\mathscr{E}_1 = \frac{\sum_{i=0}^{i=20} E_{rms}}{20}, \mathscr{E}_2 = \frac{\sum_{i=21}^{i=100} E_{rms}}{80}, \mathscr{E}_3 = \frac{\sum_{i=101}^{i=1000} E_{rms}}{900}.$$

It should be noted that $\mathscr{E}_3$ is more important. The smaller the $\mathscr{E}_3$, the better the robustness and convergence final value error of the corresponding algorithm.

The data from the three iterative stages were extracted and calculated, and Tables 1 to 3 were created based on the corresponding data from Figures 2 to 4.

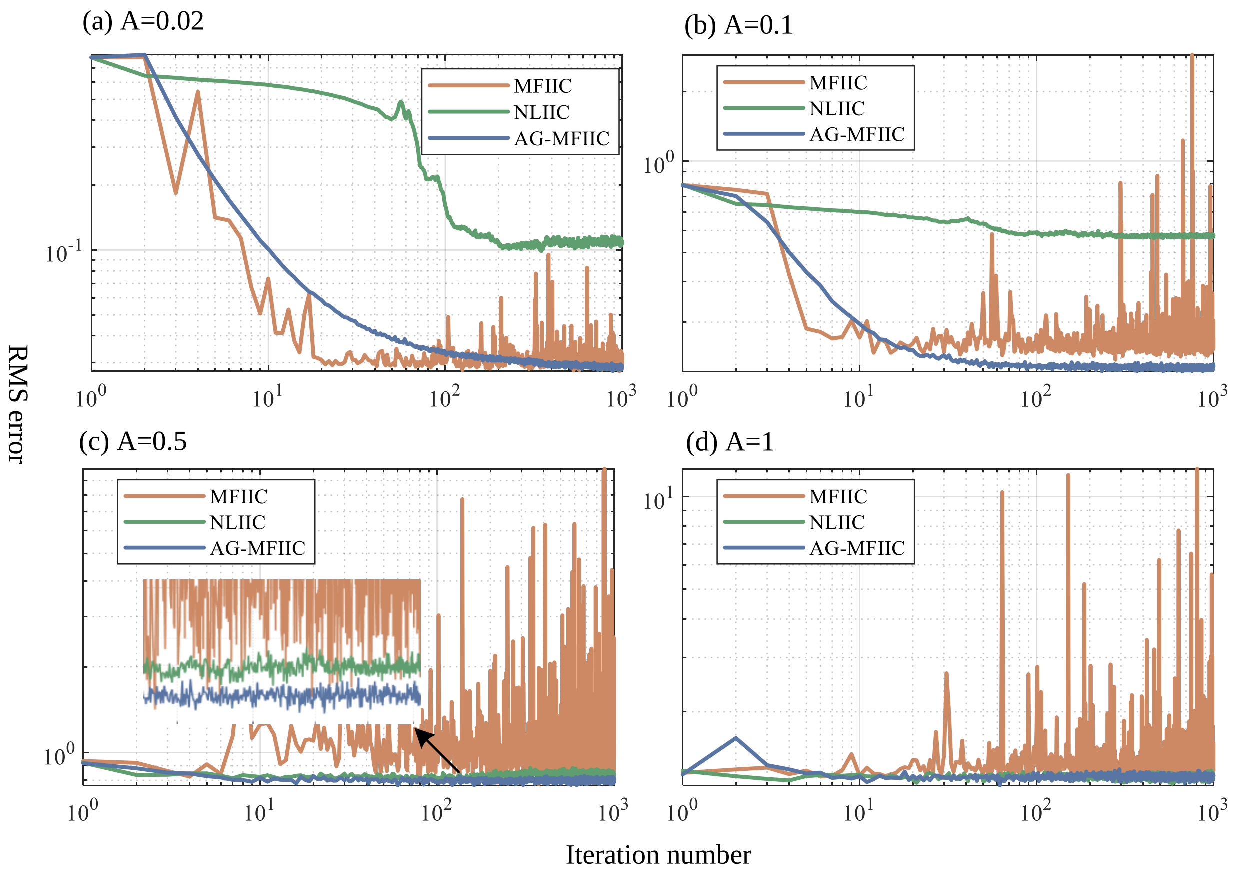

Figure 3 Comparison of the RMS error of different algorithms in the noisy environment with the reference trajectory of 1000Hz

- Table 1 presents the simulation results data for the reference trajectory of 100 Hz. The table reveals that the optimal ratio of AG-MFIIC for this reference trajectory is $\frac{10}{12}$.
- Table 2 displays the simulation results data for the reference trajectory of 1000 Hz. The table indicates that the optimal ratio of AG-MFIIC for this reference trajectory is $\frac{8}{12}$.
- Table 3 exhibits the simulation results data for the reference trajectory of 10000 Hz. The table shows that the optimal ratio of AG-MFIIC for this reference trajectory is $\frac{7}{12}$.

Combining Table 1 to Table 3, the M-RMS error of the three stages in 12 different noise environments were analyzed. The optimal ratio of AG-MFIIC algorithm for the M-RMS error is $\frac{25}{36}$, the optimal ratio of the MFIIC algorithm is $\frac{6}{36}$, and the optimal ratio of the NLIIC algorithm is $\frac{5}{36}$. For $\mathscr{E}_3$, which is the most important index among $\mathscr{E}_1$, $\mathscr{E}_2$, $\mathscr{E}_3$, the optimal ratio of AG-MFIIC algorithm is $\frac{11}{12}$. In summary, the AG-MFIIC algorithm is superior to the other two algorithms.

4.3 | Comparison and analysis of gain in simulation

In order to provide a more detailed explanation of the different roles played by different gains during the iterative process, this subsection analyzes the gains during the simulation to aid in illustrating the effectiveness of the proposed algorithms.

As depicted in Figure 3, the gain comparison of three algorithms for a complete control process at the 100th iteration under a specific environment (A=1, 1000Hz) is presented. At this point, the MFIIC algorithm has a gain of $\rho_1 = 1$, the AG-MFIIC algorithm has a gain of $\rho_2 = 0.01$, and the NLIIC algorithm varies within a range due to its nonlinear nature.

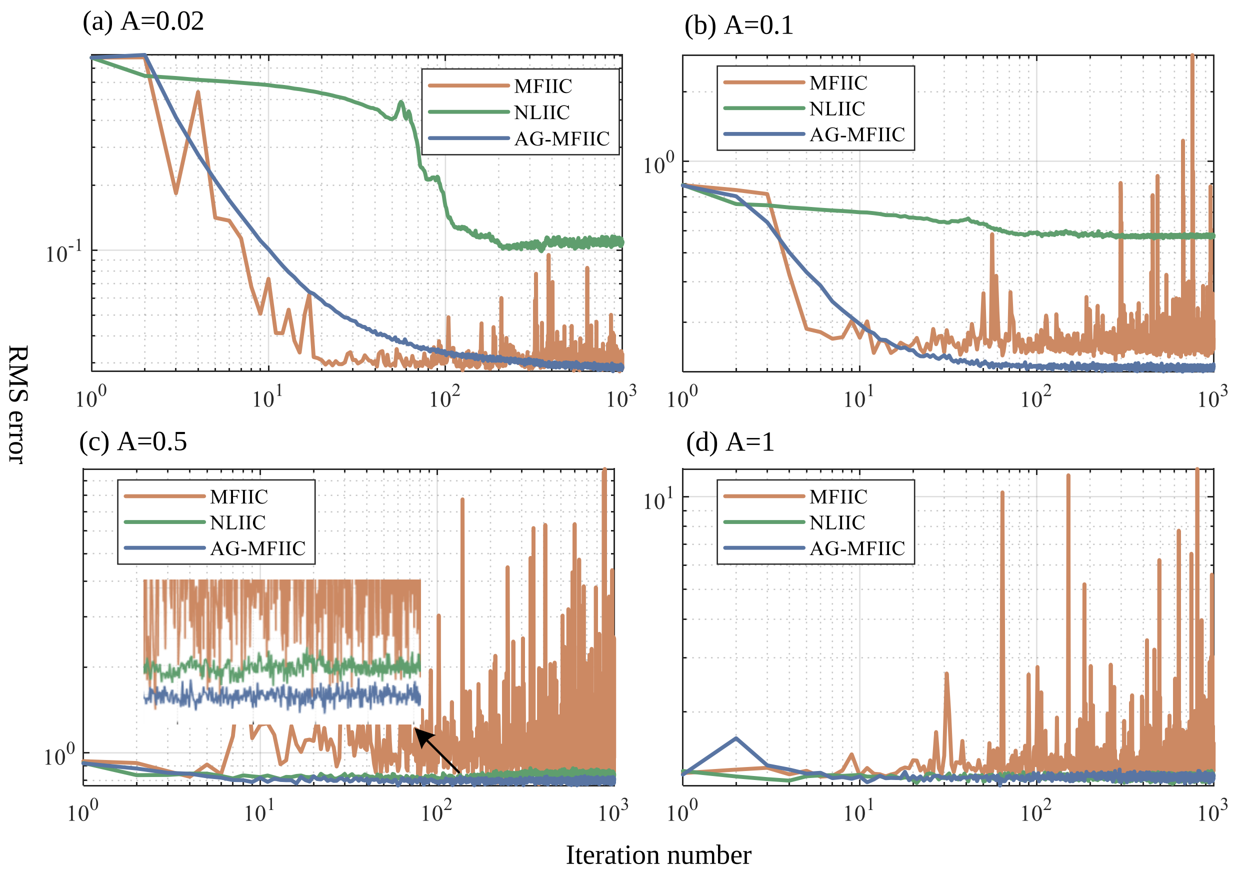

Figure 4 Comparison of the RMS error of different algorithms in the noisy environment with the reference trajectory of 10000Hz

Table 1 Comparison of the M-RMS error of different algorithms in the noisy environment with the reference trajectory of 100Hz.

	A=0.02			A=0.1			A=0.5			A=1		
	AG	MFIIC	NLIIC	AG	MFIIC	NLIIC	AG	MFIIC	NLIIC	AG	MFIIC	NLIIC
$\mathscr{E}_1$	0.1279	0.1226	0.1913	0.2019	0.2073	0.2499	0.6702	0.6774	0.6267	1.2331	1.3647	1.1405
$\mathscr{E}_2$	0.0207	0.0226	0.0226	0.1014	0.1126	0.1108	0.5061	0.6262	0.5543	1.0688	1.4151	1.1370
$\mathscr{E}_3$	0.0203	0.0226	0.0224	0.1006	0.1121	0.1112	0.5027	0.6225	0.5515	1.0741	1.4108	1.1948

†AG is the abbreviation for AG-MFIIC.

Based on the analysis in Section 3.1, it is evident that at the 100th iteration, the error is already small. Therefore, the primary goal should be to minimize the error caused by noise and prevent error fluctuations due to over-adjusting the control signal. However, as demonstrated in Figure 3, the gain of the NLIIC algorithm remains mostly around 1, which can lead to less precise control of the signal. Moreover, for the MFIIC algorithm, the gain is consistently set at 1, which may not be appropriate in situations where the reference trajectory is small, leading to significant deviations in the inverse model calculation. Therefore, it is more suitable to set the gain at a small value, as shown in the proposed AG-MFIIC algorithm.

Table 2 Comparison of the M-RMS error of different algorithms in the noisy environment with the reference trajectory of 1000Hz.

	A=0.02			A=0.1			A=0.5			A=1		
	AG	MFIIC	NLIIC	AG	MFIIC	NLIIC	AG	MFIIC	NLIIC	AG	MFIIC	NLIIC
$\mathscr{E}_1$	0.2306	0.2253	0.2941	0.3930	0.3664	0.6379	0.8354	1.1252	0.8413	1.2969	1.2897	1.2397
$\mathscr{E}_2$	0.0207	0.0226	0.0226	0.1361	0.1726	0.5223	0.7998	1.1138	0.8214	1.2218	1.4831	1.2311
$\mathscr{E}_3$	0.0203	0.0226	0.0224	0.1273	0.1830	0.4749	0.7984	1.1831	0.8422	1.2307	1.4513	1.2289

†AG is the abbreviation for AG-MFIIC.

Table 3 Comparison of the M-RMS error of different algorithms in the noisy environment with the reference trajectory of 10000Hz.

	A=0.02			A=0.1			A=0.5			A=1		
	AG	MFIIC	NLIIC	AG	MFIIC	NLIIC	AG	MFIIC	NLIIC	AG	MFIIC	NLIIC
$\mathscr{E}_1$	0.3351	0.1506	0.6355	0.2893	0.2561	0.6450	0.7054	0.6695	0.8186	1.2025	1.1992	1.1970
$\mathscr{E}_2$	0.0330	0.0216	0.2815	0.1033	0.1059	0.3596	0.5306	0.5438	0.7606	1.1026	1.1534	1.1780
$\mathscr{E}_3$	0.0207	0.0216	0.0216	0.1013	0.1057	0.1473	0.5264	0.5453	0.7538	1.0957	1.2674	1.1709

†AG is the abbreviation for AG-MFIIC.

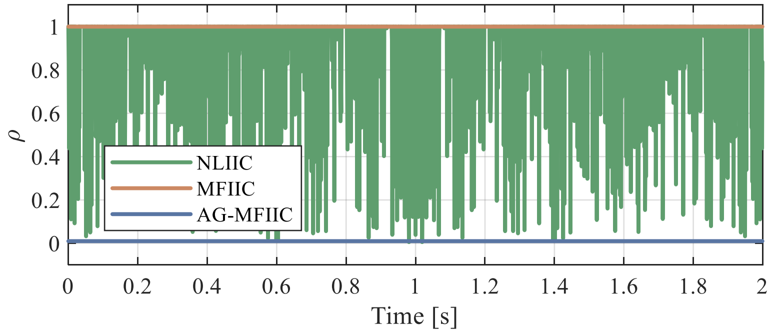

Figure 5 Gain comparison of different algorithms at the 100th iteration in a specific environment (A=1, 1000Hz).

5 | CONCLUSION

An important problem in applying the model-free inversion-based iterative learning control algorithm is that the robustness of the algorithm and the convergence effect cannot be guaranteed simultaneously. In this work, the error sources of the iterative process are derived and analyzed, and the design guideline of the gain is proposed, followed by a specific gain-setting method. Subsequently, the robustness analysis of the proposed algorithm is performed. Finally, the proposed algorithm is compared with existing similar algorithms in a detailed simulation and data comparison, thus illustrating the effectiveness of the proposed algorithm.

Designing the gain along the iterative axis is an effective way to improve the control effect more in line with the characteristics of iterative learning iterations. In future work, we will consider designing better gains along the iteration axis to make each iteration optimal. Also, it is an attractive direction to combine the gain design of the time axis with the iterative axis to carry out the work.

References

1. Bristow DA, Tharayil M, Alleyne AG. A survey of iterative learning control. *IEEE control systems magazine* 2006; 26(3): 96–114.

2. Tien S, Zou Q, Devasia S. Iterative control of dynamics-coupling-caused errors in piezoscanners during high-speed AFM operation. *IEEE Trans. Control Syst. Technol.* 2005; 13(6): 921–931.

3. Chi R, Hui Y, Huang B, Hou Z. Adjacent-agent dynamic linearization-based iterative learning formation control. *IEEE Trans. Cybern.* 2019; 50(10): 4358–4369.

4. Liu G, Hou Z. Cooperative adaptive iterative learning fault-tolerant control scheme for multiple subway trains. *IEEE Trans. Cybern.* 2020; 52(2): 1098–1111.

5. Liu Y, Lai J, Jin Z, Ren Z, Xie S, Wang Y. Iterative learning control for linear parabolic distributed parameter systems with multiple point sensors. *Optim. Control. Appl. Methods* 2021.

6. Dong J. Robust data-driven iterative learning control for linear-time-invariant and Hammerstein-Wiener systems. *IEEE Trans. Cybern.* 2021.

7. Jin X. Iterative learning control for MIMO nonlinear systems with iteration-varying trial lengths using modified composite energy function analysis. *IEEE Trans. Cybern.* 2020; 51(12): 6080–6090.

8. Jin X. Fault tolerant nonrepetitive trajectory tracking for MIMO output constrained nonlinear systems using iterative learning control. *IEEE Trans. Cybern.* 2018; 49(8): 3180–3190.

9. Yu X, Hou Z, Polycarpou MM. A data-driven ILC framework for a class of nonlinear discrete-time systems. *IEEE Trans. Cybern.* 2021; 52(7): 6143–6157.

10. Shen D, Liu C, Wang L, Yu X. Iterative learning tracking for multisensor systems: A weighted optimization approach. *IEEE Trans. Cybern.* 2019; 51(3): 1286–1299.

11. Bu X, Yu W, Yu Q, Hou Z, Yang J. Event-triggered model-free adaptive iterative learning control for a class of nonlinear systems over fading channels. *IEEE Trans. Cybern.* 2021; 52(9): 9597–9608.

12. Shen D, Zhang C. Zero-error tracking control under unified quantized iterative learning framework via encoding–decoding method. *IEEE Trans. Cybern.* 2020; 52(4): 1979–1991.

13. Owens DH, Hätönen J. Iterative learning control—An optimization paradigm. *Annual reviews in control.* 2005; 29(1): 57–70.

14. Kim KS, Zou Q. A modeling-free inversion-based iterative feedforward control for precision output tracking of linear time-invariant systems. *IEEE/ASME Trans. Mechatron.* 2012; 18(6): 1767–1777.

15. Teng KT, Tsao TC. A comparison of inversion based iterative learning control algorithms. In: IEEE. ; 2015: 3564–3569.

16. Wu Y, Zou Q. Robust inversion-based 2-DOF control design for output tracking: piezoelectric-actuator example. *IEEE Trans. Control Syst. Technol.* 2009; 17(5): 1069–1082.

17. Yan Y, Wang H, Zou Q. A decoupled inversion-based iterative control approach to multi-axis precision positioning: 3D nanopositioning example. *Automatica.* 2012; 48(1): 167–176.

18. Tian Y, Huo Z, Wang F, Shi B. Precision tracking of a 2-DOF stick-slip positioner using modeling-free inversion-based iterative control and modified inverse hysteresis compensator. *Sens. Actuators Phys.* 2021; 331: 112959.

19. Pintelon R, Schoukens J. *System identification: a frequency domain approach.* John Wiley & Sons . 2012.

20. Hou ZS, Wang Z. From model-based control to data-driven control: Survey, classification and perspective. *Inform. Sci.* 2013; 235: 3–35.

21. Rozario dR, Oomen T. Data-driven iterative inversion-based control: Achieving robustness through nonlinear learning. *Automatica.* 2019; 107: 342–352.

22. Cheng X, Jiang H, Shen D, Yu X. A Novel Adaptive Gain Strategy for Stochastic Learning Control. *IEEE Trans. Cybern.* 2022: 1-12. doi: 10.1109/TCYB.2022.3192031

□