IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

PD-type Parameter Optimization Iterative Learning Control Algorithm Based on Inverse Model

To cite this article: Zhanlei Xiong *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **569** 052001

IOP Conf. Series: Materials Science and Engineering **569** (2019) 052001 doi:10.1088/1757-899X/569/5/052001

PD-type Parameter Optimization Iterative Learning Control Algorithm Based on Inverse Model

Zhanlei Xiong, Qingshan Zeng, Mingjun Yin

School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China

xiongzlzzu@163.com

Abstract. In this paper, a proportion-differentiation parameter optimization iterative learning control (POILC) algorithm based on inverse model (IM) is proposed for the tracking control problem of a class of single input and single output discrete linear time-invariant (LTI) systems. The algorithm establishes the parameter optimal performance index function and adds the learning gain matrix to proportion and differential terms of the control law, which enable the algorithm to be applied to non-positive definite systems and to converge monotonously and rapidly. The purpose of above method is to reduce the influence of model accuracy on tracking performance. Compared with previous algorithms, the proposed algorithm has been improved to a certain extent in tracking accuracy, convergence speed and robustness.

1. Introduction

With the development of iterative learning control (ILC) technology, the theoretical research content of ILC focuses on optimizing the ILC control law to improve its tracking performance.

The idea of norm optimization iterative learning control (NOILC) was first proposed by Amann. Later, many scholars continued to improve and innovate on this basis. A PID-based fast POILC algorithm based on norm performance index was proposed for discrete systems, which improves the learning efficiency of the algorithm[1]. A quasi-optimal ILC algorithm was proposed to deal with the uncertainties in the system[2]. A NOILC algorithm based on interpolation points was designed for continuous linear time invariant systems, which verified the geometric convergence of error norm[3]. Aimed at the terminal tracking control of discrete systems, an optimal ILC algorithm based on data-driven was proposed[4].

Compared with the norm optimization theory, the idea of parameter optimization is easier to implement in engineering. Owens et al. proposed the parameter optimization theory on the basis of ILC norm optimization theory[5]. Since then, many scholars have continuously improved the tracking performance of algorithm to deal with various practical problems. In [6], a POILC algorithm was proposed for discrete LTI systems, which improves the tracking performance. Later, the high-order POILC algorithm was proposed to further improve the tracking performance[7-8]. The relationship between the positive definiteness of the system and the POILC algorithm was explored[9]. The robustness of parameter optimization iterative learning control algorithm was studied[10-12].

However, the limitation of traditional POILC algorithm makes it only applicable to positive definite systems. In order to make the parameter optimization algorithm applicable to non-positive systems, the learning gain matrix is added to the proportional term of control law, which greatly improves tracking performance of the algorithm[13]. However, the learning gain matrix is derived from singular value decomposition (SVD) of the system model. When there exist model errors, the tracking accuracy and convergence speed of the algorithm are easily affected.

In order to make the POILC algorithm applied to non-definite system and improve its tracking performance, this paper makes the following improvements based on literature[13]: (1) The learning gain matrix is added to both the proportional and the differential terms of control law. (2) The learning gain matrix is obtained by inverting the system model matrix, which ensures that the product of the system model matrix and the learning gain matrix is a unit vector. As a result, $\boldsymbol{e_k}$ does not contain any model information during each iteration, which reduces the influence of model accuracy on system tracking performance. (3) The selection of learning gain matrix can also reduce the influence of model accuracy on parameter selection.

Compared with the POILC algorithm proposed in [13], this paper presents a POILC algorithm based on inverse model. It reduces the influence of system model accuracy on tracking performance and can better deal with the uncertainties in practical process.

2. Problem Formulation

Consider the following single input and single output discrete linear time invariant system:

$$\begin{cases} \boldsymbol{x}_k(t+1) = \boldsymbol{A}\boldsymbol{x}_k(t) + \boldsymbol{B}\boldsymbol{u}_k(t) \\ \boldsymbol{y}_k(t) = \boldsymbol{C}\boldsymbol{x}_k(t) \end{cases} . \tag{1}$$

$t \in [0, T]$ denotes the sampling time interval. k indicates repetition number. $\boldsymbol{x}_k(t) \in \boldsymbol{R}^m$, $\boldsymbol{u}_k(t) \in \boldsymbol{R}^m$, $\boldsymbol{y}_k(t) \in \boldsymbol{R}^m$ represent the states, inputs and outputs at time t of the k-th trial, respectively. A, B, C are the system matrices with appropriate dimensions. $\boldsymbol{CB} \neq \boldsymbol{0}$. $\boldsymbol{x}_k(0) = \boldsymbol{x}_0$ is the identical initial condition during each trial. The optimal input is $\boldsymbol{u}_d = \begin{bmatrix} u_d(0) & u_d(1) & \cdots & u_d(N-1) \end{bmatrix}^{\mathrm{T}}$ .The expected output trajectory of system (1) is $\boldsymbol{y}_d(t) \in \boldsymbol{R}$. $\boldsymbol{e}_k(t) = \boldsymbol{y}_d(t) - \boldsymbol{y}_k(t)$. $\boldsymbol{I}$ is the n-order unit matrix.

The input and output responses of the system can be written as follows:

$$\boldsymbol{y}_k = \boldsymbol{G}\boldsymbol{u}_k + \boldsymbol{d} . \tag{2}$$

Then, $\boldsymbol{e}_k = \boldsymbol{y}_d - \boldsymbol{y}_k = \boldsymbol{y}_d - \boldsymbol{G}\boldsymbol{u}_k - \boldsymbol{d}$.

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{CB} & 0 & 0 & \cdots & 0 \\ \boldsymbol{CAB} & \boldsymbol{CB} & 0 & \cdots & 0 \\ \boldsymbol{CA}^2\boldsymbol{B} & \boldsymbol{CAB} & \boldsymbol{CB} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{CA}^{N-1}\boldsymbol{B} & \cdots & \cdots & \boldsymbol{CAB} & \boldsymbol{CB} \end{bmatrix} . \ \boldsymbol{d} = \begin{bmatrix} \boldsymbol{CA} \\ \boldsymbol{CA}^2 \\ \boldsymbol{CA}^3 \\ \vdots \\ \boldsymbol{CA}^{N-1} \end{bmatrix} x_0 .$$

Consider applying the following algorithm to non-positive definite systems.

$$\boldsymbol{u}_{k+1}(t) = \boldsymbol{u}_k(t) + \boldsymbol{\Gamma}(\alpha_{k+1}\boldsymbol{e}_k(t+1) + \beta_{k+1}\Delta\boldsymbol{e}_k(t)) . \tag{3}$$

$\Delta\boldsymbol{e}_k(t) = \boldsymbol{e}_k(t+1) - \boldsymbol{e}_k(t)$. α_{k+1} and β_{k+1} represent the parameters corresponding to the error proportional term and derivative term, respectively. $\boldsymbol{\Gamma} \in \boldsymbol{R}^{N\times N}$ represents the learning gain matrix.

It is easy to get that $\boldsymbol{G}$ is a n-order real non singular matrix. Then, there exists $\boldsymbol{\Gamma}$ so that $\boldsymbol{G\Gamma} = \boldsymbol{I}$. The latter analysis will prove that the algorithm (3) can still guarantee the tracking error monotonously converge to zero when it is applied to non positive definite system.

In the algorithm (3), the parameters α_{k+1} and β_{k+1} are selected to be the solution of the quadratic objective function J_{k+1} as follows.

$$\begin{bmatrix} \alpha_{k+1} & \beta_{k+1} \end{bmatrix} = \arg\min\left\{ J_{k+1}(\alpha_{k+1}, \beta_{k+1}) \right\} . \tag{4}$$

$$J_{k+1} = \left\| e_{k+1} \right\|^2 + w_1\alpha_{k+1}^2 + w_2\beta_{k+1}^2, w_1 > 0, w_2 > 0 . \tag{5}$$

According to formula (3) and $\boldsymbol{G\Gamma} = \boldsymbol{I}$, we can get the following formula (6).

$$\begin{aligned} \left\|\boldsymbol{e}_{k+1}\right\|^2 &= \left\|\boldsymbol{y}_d - \boldsymbol{y}_{k+1}\right\|^2 = \left\|\boldsymbol{y}_d - \boldsymbol{y}_k + \boldsymbol{y}_k - \boldsymbol{y}_{k+1}\right\|^2 = \left\|\boldsymbol{e}_k - \boldsymbol{G}(\boldsymbol{u}_{k+1} - \boldsymbol{u}_k)\right\|^2 = \left\|\boldsymbol{e}_k - \boldsymbol{G}(\alpha_{k+1}\boldsymbol{\Gamma}\boldsymbol{e}_k + \beta_{k+1}\boldsymbol{\Gamma}\Delta\boldsymbol{e}_k)\right\|^2 \\ &= \left\|\boldsymbol{e}_k\right\|^2 + \alpha_{k+1}^2\left\|\boldsymbol{G\Gamma e}_k\right\|^2 + \beta_{k+1}^2\left\|\boldsymbol{G\Gamma}\Delta\boldsymbol{e}_k\right\|^2 - 2\alpha_{k+1}\left\langle \boldsymbol{e}_k, \boldsymbol{G\Gamma e}_k \right\rangle - 2\beta_{k+1}\left\langle \boldsymbol{e}_k, \boldsymbol{G\Gamma}\Delta\boldsymbol{e}_k \right\rangle + 2\alpha_{k+1}\beta_{k+1}\left\langle \boldsymbol{G\Gamma e}_k, \boldsymbol{G\Gamma}\Delta\boldsymbol{e}_k \right\rangle . \\ &= \left\|\boldsymbol{e}_k\right\|^2 + \alpha_{k+1}^2\left\|\boldsymbol{e}_k\right\|^2 + \beta_{k+1}^2\left\|\Delta\boldsymbol{e}_k\right\|^2 - 2\alpha_{k+1}\left\langle \boldsymbol{e}_k, \boldsymbol{e}_k \right\rangle - 2\beta_{k+1}\left\langle \boldsymbol{e}_k, \Delta\boldsymbol{e}_k \right\rangle + 2\alpha_{k+1}\beta_{k+1}\left\langle \boldsymbol{e}_k, \Delta\boldsymbol{e}_k \right\rangle \end{aligned} \tag{6}$$

According to (5)-(6), let $\partial J / \partial\alpha = 0, \partial J / \partial\beta = 0$ and we can get the optimal solution of the parameters.

$$\left(\begin{bmatrix} \|\boldsymbol{e}_k\|^2 & \langle \boldsymbol{e}_k, \Delta\boldsymbol{e}_k\rangle \\ \langle \boldsymbol{e}_k, \Delta\boldsymbol{e}_k\rangle & \|\Delta\boldsymbol{e}_k\|^2 \end{bmatrix} + \begin{bmatrix} w_1 & 0 \\ 0 & w_2 \end{bmatrix}\right) \cdot \begin{pmatrix} \alpha_{k+1}^* \\ \beta_{k+1}^* \end{pmatrix} = \begin{pmatrix} \langle \boldsymbol{e}_k, \boldsymbol{e}_k\rangle \\ \langle \boldsymbol{e}_k, \Delta\boldsymbol{e}_k\rangle \end{pmatrix}. \tag{7}$$

Let $\boldsymbol{A}_k = \begin{bmatrix} \|\boldsymbol{e}_k\|^2 & \langle \boldsymbol{e}_k, \Delta\boldsymbol{e}_k\rangle \\ \langle \boldsymbol{e}_k, \Delta\boldsymbol{e}_k\rangle & \|\Delta\boldsymbol{e}_k\|^2 \end{bmatrix} + \begin{bmatrix} w_1 & 0 \\ 0 & w_2 \end{bmatrix}$ and $\boldsymbol{B}_k = \begin{pmatrix} \langle \boldsymbol{e}_k, \boldsymbol{e}_k\rangle \\ \langle \boldsymbol{e}_k, \Delta\boldsymbol{e}_k\rangle \end{pmatrix}$. As we can see, $\boldsymbol{A}_k$ is non singular and positive definite. According to formula (7), the following formula holds.

$$[\alpha_{k+1}^*, \beta_{k+1}^*] = \boldsymbol{A}_k^{-1}\boldsymbol{B}_k \tag{8}$$

It can be seen that the system parameters α_{k+1}^* and β_{k+1}^* depend entirely on the datas obtained during each iteration. The model accuracy has little influence on the selection of parameters α_{k+1}^* and β_{k+1}^*.

Theorem 1. If the algorithm (3) is applied to the system (1), for any $k > 0$, there is $\|\boldsymbol{e}_{k+1}\| < \|\boldsymbol{e}_k\|$ and there are $\lim_{k\to\infty}\alpha_{k+1}^* = 0$, $\lim_{k\to\infty}\beta_{k+1}^* = 0$, $\lim_{k\to\infty}\|\boldsymbol{e}_k\| = 0$.

Proof. Introduce the non optimal parameters $\alpha_{k+1}=0$ and $\beta_{k+1}=0$ into equation (5).

$$\|e_{k+1}\|^2 \le J_{k+1}(\alpha_{k+1}^*, \beta_{k+1}^*) = \|e_{k+1}\|^2 + w_1(\alpha_{k+1}^*)^2 + w_2(\beta_{k+1}^*)^2 \le J_{k+1}(0,0) = \|e_k\|^2.$$

$$\sum_{i=0}^{k}(w_1\cdot(\alpha_{i+1}^*)^2 + w_2\cdot(\beta_{i+1}^*)^2) \le \|e_0\|^2 - \|e_{k+1}\|^2.$$

So when k tends to infinity, $\sum_{i=0}^{\infty}(w_1(\alpha_{k+1}^*)^2 + w_2(\beta_{k+1}^*)^2)$ approaches $\|e_0\|^2$. So $\{\alpha_{k+1}^*\}$ and $\{\beta_{k+1}^*\}$ are convergent sequences, then $\lim_{k\to\infty}\alpha_{k+1}^* = \lim_{k\to\infty}\beta_{k+1}^* = 0$. According to formula (8) and **Theorem 1**, $\lim_{k\to\infty} B_k = 0$, then $\lim_{k\to\infty}\|\boldsymbol{e}_k\| = 0$.

3. Robustness Analysis

In practical applications, the algorithm is required to be robust to parameter, structure and output uncertainties. The robustness problem of ILC generally considers the convergence and stability of the system with uncertainties. This paper mainly focuses on the influence of model errors and external disturbances on the tracking performance of the algorithm.

According to formula (2)-(3), it can be obtained that $\boldsymbol{e}_{k+1} = \boldsymbol{e}_k - \boldsymbol{G\Gamma}(\alpha_{k+1}\boldsymbol{e}_k + \beta_{k+1}\Delta\boldsymbol{e}_k) = (1-\alpha_{k+1})\boldsymbol{e}_k - \beta_{k+1}\Delta\boldsymbol{e}_k$. It can be seen that the algorithm (3) can ensure that $\boldsymbol{e}_{k+1}$ does not contain any model information during each iteration. Therefore, it has stronger robustness when there exist model errors.

Let $e_k(N+1) = e_k(1)$, it is easy to get the following formula.

$$\begin{bmatrix} e_{k+1}(1) \\ e_{k+1}(2) \\ e_{k+1}(3) \\ \vdots \\ e_{k+1}(N) \end{bmatrix} = (1-\alpha_{k+1}+\beta_{k+1})\begin{bmatrix} e_k(1) \\ e_k(2) \\ e_k(3) \\ \vdots \\ e_k(N) \end{bmatrix} - \beta_{k+1}\begin{bmatrix} e_k(2) \\ e_k(3) \\ \vdots \\ e_k(N) \\ e_k(N+1) \end{bmatrix} = \left\{(1-\alpha_{k+1}+\beta_{k+1})\boldsymbol{I} - \beta_{k+1}\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & 1 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}\right\}\cdot\begin{bmatrix} e_k(1) \\ e_k(2) \\ e_k(3) \\ \vdots \\ e_k(N) \end{bmatrix} = \boldsymbol{\Psi}\begin{bmatrix} e_k(1) \\ e_k(2) \\ e_k(3) \\ \vdots \\ e_k(N) \end{bmatrix}.$$

When $|1-\alpha_{k+1}+\beta_{k+1}|+|\beta_{k+1}|<1$, there are $\|\boldsymbol{\Psi}\|<1$ and $\|\boldsymbol{e}_{k+1}\| < \|\boldsymbol{e}_k\|$. It is worth mentioning that the tracking error of algorithm (3) can converge to zero after two iterations in theoretical analysis. However, due to the uncertainties of model and external disturbances, it is difficult to achieve completely tracking after two iterations. To achieve better tracking performance, we use the formula (8) to optimize the parameters.

Consider the following three situations:

1) When model errors $\Delta\boldsymbol{G}$ exist, then the established model is $\boldsymbol{G}+\Delta\boldsymbol{G}$. Since $\boldsymbol{\Gamma}$ is obtained by inverting the system model matrix, it still satisfies $(\boldsymbol{G}+\Delta\boldsymbol{G})\cdot\boldsymbol{\Gamma} = \boldsymbol{I}$. It can be concluded from the previous analysis that the convergence and tracking accuracy of the proposed algorithm are only related to the selection of parameters, and $\boldsymbol{e}_{k+1}$ does not contain any model information during each iteration. This is the advantage compared with other algorithms.

Parameter optimization guarantees the realization of $|1-\alpha_{k+1}+\beta_{k+1}|+|\beta_{k+1}|<1$, and then $\|\boldsymbol{e}_{k+1}\|<\|\boldsymbol{e}_k\|$. And, performance index $J_{k+1}(\cdot)$ guarantees the optimal solution of parameters during each iteration, thus the system (1) can be tracked quickly and efficiently.

2) When there are external disturbances γ_k, suppose $\|\gamma_k\|\le\lambda$:

$$\boldsymbol{e}_{k+1}=\boldsymbol{e}_k-\boldsymbol{G}(\alpha_{k+1}\boldsymbol{\Gamma}\boldsymbol{e}_k+\beta_{k+1}\boldsymbol{\Gamma}\Delta\boldsymbol{e}_k)+\gamma_k-\gamma_{k+1}=(1-\alpha_{k+1})\boldsymbol{e}_k-\beta_{k+1}\Delta\boldsymbol{e}_k-(\gamma_{k+1}-\gamma_k)$$

$$\|\boldsymbol{e}_{k+1}\|\le\|\boldsymbol{\Psi}\|\cdot\|\boldsymbol{e}_k\|+\|\gamma_{k+1}-\gamma_k\|\le\|\boldsymbol{\Psi}\|(\|\boldsymbol{\Psi}\|\cdot\|\boldsymbol{e}_{k-1}\|+\|\gamma_k-\gamma_{k-1}\|)+\|\gamma_{k+1}-\gamma_k\|\le\cdots\le\|\boldsymbol{\Psi}\|^{k+1}\cdot\|\boldsymbol{e}_0\|+2\lambda\cdot\frac{1-\|\boldsymbol{\Psi}\|^{k+1}}{1-\|\boldsymbol{\Psi}\|}$$

$$\limsup_{k\to\infty}\|\boldsymbol{e}_{k+1}\|\le\lim_{k\to\infty}\|\boldsymbol{\Psi}\|^{k+1}\cdot\|\boldsymbol{e}_0\|+2\lambda\cdot\lim_{k\to\infty}\frac{1-\|\boldsymbol{\Psi}\|^{k+1}}{1-\|\boldsymbol{\Psi}\|}\le\frac{2\lambda}{1-\|\boldsymbol{\Psi}\|}$$

3) When $\Delta\boldsymbol{G}$ and γ_k both exist, it still satisfies $(\boldsymbol{G}+\Delta\boldsymbol{G})\cdot\boldsymbol{\Gamma}=\boldsymbol{I}$.

$$\|\boldsymbol{e}_{k+1}\|\le\|\boldsymbol{e}_k-(\boldsymbol{G}+\Delta\boldsymbol{G})\cdot(\alpha_{k+1}\boldsymbol{\Gamma}\boldsymbol{e}_k+\beta_{k+1}\boldsymbol{\Gamma}\Delta\boldsymbol{e}_k)+\gamma_k-\gamma_{k+1}\|\le\|(1-\alpha_{k+1})\boldsymbol{e}_k-\beta_{k+1}\Delta\boldsymbol{e}_k-(\gamma_{k+1}-\gamma_k)\|\le\cdots\le\|\boldsymbol{\Psi}\|^{k+1}\cdot\|\boldsymbol{e}_0\|+2\lambda\cdot\frac{1-\|\boldsymbol{\Psi}\|^{k+1}}{1-\|\boldsymbol{\Psi}\|}$$

$$\limsup_{k\to\infty}\|\boldsymbol{e}_{k+1}\|\le\lim_{k\to\infty}\|\boldsymbol{\Psi}\|^{k+1}\cdot\|\boldsymbol{e}_0\|+2\lambda\cdot\lim_{k\to\infty}\frac{1-\|\boldsymbol{\Psi}\|^{k+1}}{1-\|\boldsymbol{\Psi}\|}\le\frac{2\lambda}{1-\|\boldsymbol{\Psi}\|}$$

It can be seen from above analysis that the proposed algorithm can reduces the influence of model uncertainties on tracking accuracy and convergence in practical application. Parameter optimization makes the tracking control fast and effectively. It is worth mentioning that if the external disturbances are constant during each iteration, then $\lim_{k\to\infty}\|\boldsymbol{e}_k\|=0$.

4. Simulation Analysis

In order to verify the validity of the algorithm (3), it is compared with the algorithm in [13]. Consider the following discrete linear time-invariant system.

$$\begin{cases}\boldsymbol{x}_k(t+1)=\begin{bmatrix}0.8454 & -0.0928\\ 0.0464 & 0.9976\end{bmatrix}\boldsymbol{x}_k(t)+\begin{bmatrix}0.0464\\ 0.0012\end{bmatrix}\boldsymbol{u}_k(t)\\ \boldsymbol{y}_k(t)=[1\quad 6]\boldsymbol{x}_k(t), t=0,1,2\cdots N\end{cases}. \tag{9}$$

Expected output trajectory $y_d(t)=\sin(2\pi t/N)$, sampling period $N=20$. $\boldsymbol{x}_k(0)=0$. The range of eigenvalues of $\boldsymbol{G}+\boldsymbol{G}^{\mathrm{T}}$ is in $[-0.023\quad 1.424]$. Therefore, the system is non-positive, so the traditional POILC algorithm can not ensure that the tracking error monotonically converges to zero.

Use the proposed algorithm (3) to track the system (9).

$$\boldsymbol{u}_{k+1}(t)=\boldsymbol{u}_k(t)+\boldsymbol{\Gamma}(\alpha_{k+1}\cdot\boldsymbol{e}_k(t+1)+\beta_{k+1}\cdot\Delta\boldsymbol{e}_k(t))$$

$\boldsymbol{\Gamma}=\boldsymbol{G}^{-1}$. α_{k+1} and β_{k+1} are solved by the equation(9). For better tracking performance, the weight parameters w_1, w_2 are as small as possible.

Fig.1 shows the convergence of the system tracking error norm in the iterative domain. Although the system (9) is non-positive, it can also guarantee monotonic convergence of tracking error. Because the learning gain and the control law are designed uniquely, $\boldsymbol{e}_{k+1}$ is only related to $\boldsymbol{e}_k$, $\Delta\boldsymbol{e}_k$, α_{k+1}, β_{k+1}. The model errors do not directly affect the tracking performance. After parameter optimization, the norm of the tracking error of the algorithm at the second iteration has reached 9.87×10^{-7}.

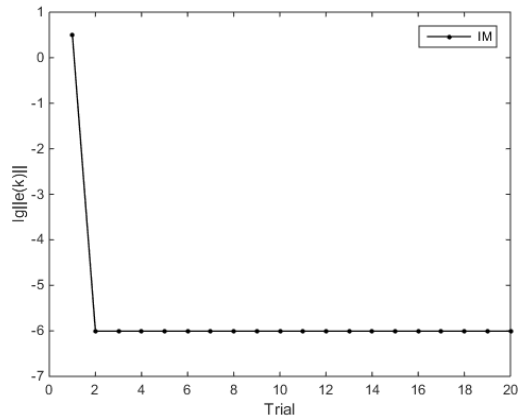

Fig.1 The norm of the tracking error of the algorithm (3)

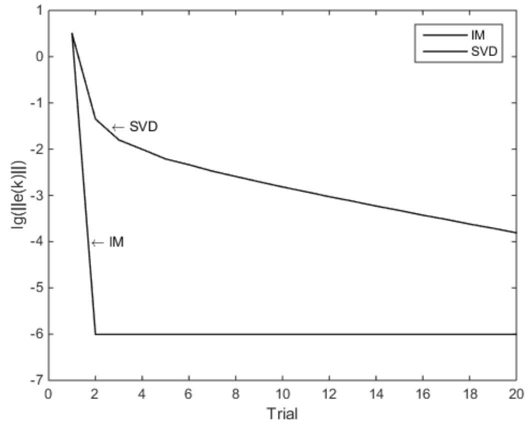

Fig.2 The Comparison between the IM and the SVD algorithm in [13]

Fig.2 shows the convergence of the system tracking error norm of the two algorithms in the iterative domain. It can be seen from the comparison that the algorithm proposed in this paper has some improvements in convergence speed and tracking accuracy.

In order to compare the robustness of the two algorithms, consider comparing the tracking performance in the presence of model errors and external disturbances. When there are model errors and external disturbances in the system (9), it may be assumed that:

$$\Delta \boldsymbol{A} = \begin{bmatrix} 0.2 & 0.2 \\ 0.2 & 0.2 \end{bmatrix}. \Delta \boldsymbol{B} = \begin{bmatrix} 0.1 \\ 0.01 \end{bmatrix}. \Delta \boldsymbol{C} = \begin{bmatrix} 1 & 1 \end{bmatrix}.\ r = rand(1) \cdot \boldsymbol{y}_d = rand(1) \cdot \sin(2\pi t / N),\ rand(1) \in [0,1].$$

Fig.3 and Fig.4 show the convergence of the tracking error norm of the two algorithms in the presence of model errors $\Delta \boldsymbol{G}$ and external disturbances r, respectively.

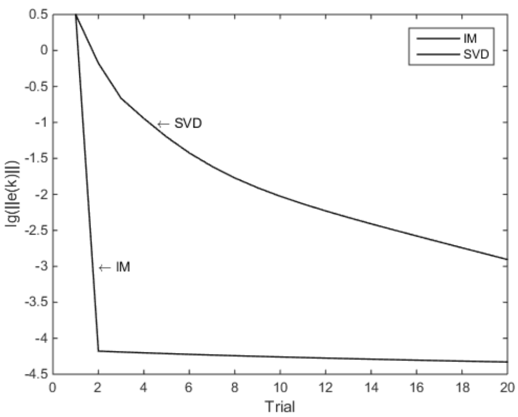

Fig.3 The Comparison between the IM and the SVD algorithm in [13] when $\Delta \boldsymbol{G}$ exist.

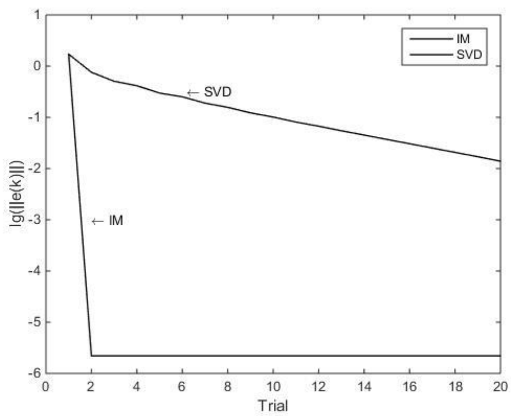

Fig.4 The Comparison between the IM and the SVD algorithm in [13] when r exist.

Fig.5 compares the convergence of the tracking error norm of the two algorithms in the presence of model errors $\Delta \boldsymbol{G}$ and external disturbances r. By comparison, we can conclude that the proposed algorithm has stronger robustness and better tracking performance, which can better deal with various uncertainties in practical systems. The IM-POILC algorithm is hardly affected by the accuracy of modelling. For some practical systems that are difficult to model or have model errors, this algorithm will have excellent performance because of its stronger robustness.

In the presence of model errors $\Delta \boldsymbol{G}$ and external disturbances r, Fig.6 shows the expected output and actual output based on the IM algorithm at the 1st and 2nd iterations. Combining Fig.5 and Fig.6, we can conclude that at the second iteration the actual output is extremely close to the expected output, which verifies the effectiveness of the algorithm.

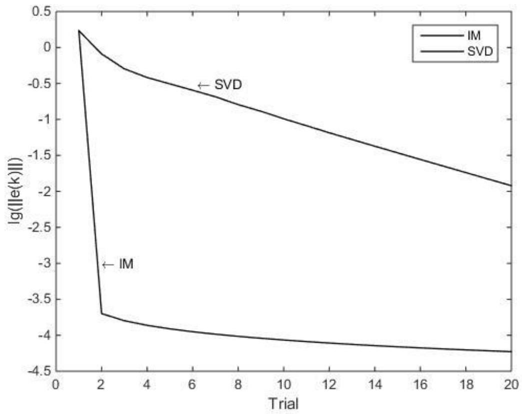

Fig.5 The Comparison between the IM and the SVD algorithm when $\Delta\boldsymbol{G}$ and r exist.

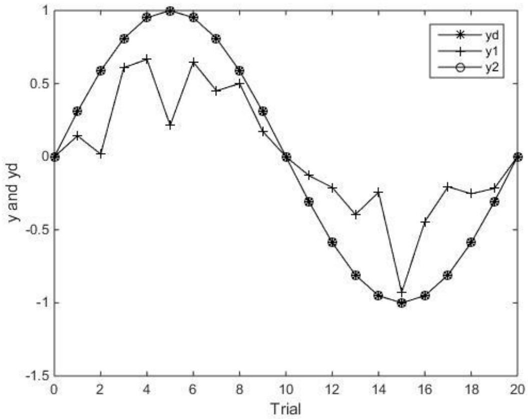

Fig.6 The IM algorithm tracks the desired output at the 1st and 2nd iterations when $\Delta\boldsymbol{G}$ and r exist.

5. Conclusion

For the tracking control problems of a class of single input and single output discrete linear time invariant systems (1), an IM-based PD-type POILC algorithm is proposed. This algorithm has a wide range of applications and can show the advantages of faster convergence, higher tracking accuracy and stronger robustness even if there are external disturbances and model errors. At each iteration, $\boldsymbol{e}_{k+1}$ does not contain any model information, its performance is hardly affected by model accuracy. According to the simulation results and theoretical analysis, compared with the algorithm in [13], the proposed algorithm has some improvements in tracking accuracy, convergence speed and robustness.

Compared with previous research results, this algorithm has the following two advantages: (1) Because the learning gain and the control law are designed uniquely, the algorithm can be applied to non positive definite systems and the model accuracy has little effect on tracking performance of the algorithm. (2) In parameter optimization, the product of system model matrix and learning gain matrix is a unit vector, which makes that α_{k+1} and β_{k+1} are only related to $\boldsymbol{e}_k$, $\Delta\boldsymbol{e}_k$ and not affected by accuracy of the model. Based on the above two points, the proposed algorithm shows high tracking performance and robustness, which can deal with various uncertainties in practical systems.

Acknowledgments

This work was supported by the Innovation Research Team of Science & Technology of Henan Province (No. 17IRTSTHN013).

References

[1] Li Hengjie, Hao Xiaohong. A fast PID type parameter optimal iterative learning control algorithm for non-positive plants[J]. 29th Chinese Control Conference, pp.2091-2096, 2010.
[2] Yan Qiuzhen, Sun Mingxuan. Suboptimal learning control for nonlinear systems with both parametric and nonparametric uncertainties[J]. Acta Automatica Sinica, 2015, 41(9):1659-1668(in Chinese).
[3] D. H. Owens, C. T. Freeman, Bing Chu. Generalized norm optimal iterative learning control with intermediate point and sub-interval tracking[J]. International Journal of Automation and Computing, 2015, 12(3):243-253.
[4] Chi Ronghu, Wang Danwei, Hou Zhongsheng. Data-driven optimal terminal iterative learning control[J]. Journal of Process Control, 2012, 22(10):2026-2037.
[5] D. H. Owens, J. Hatonen. Parameter optimal iterative learning control: a unifying formulation and analysis[J]. International Workshop on Multidimensional System, 2005, 1(1):129-135,.
[6] D. H. Owens, K. Feng. Parameter optimisation in iterative learning control[J]. 2003 European

Control Conference(ECC), pp.228-233, 2003.
[7] Li Hengjie, Hao Xiaohong, Xu Weitao. A fast parameter optimal iterative learning control algorithm[C]. Embedded Software and Systems Symposia 2008, pp.375-379, 2008.
[8] Pang Bo, Shao Cheng. High-order parameter optimization iterative learning control algorithm[J]. Control Theory and Application, 2015, 32(4):561-567(in Chinese).
[9] J. Hatonen, K. Feng, D. H. Owens. New connections between positivity and parameter-optimal iterative learning control[J]. Proceedings of the 2003 IEEE International Symposium on Intelligent Control, pp.69-74, 2003.
[10] Mutita Songjun. Parameter optimal iterative learning control with application to a robot arm[J]. Universal Journal of Control and Automation, 2015, 3(2):28-32,.
[11] Bing Chu, D. H. Owens, C. T. Freeman. Iterative learning control with predictive trial information: convergence, robustness, experimental verification[J]. IEEE Transactions on Control Systems Technology, 2016, 24(3):1101-1108.
[12] Pang Bo, Shao Cheng. A robust iterative learning control with parameter-optimization for discrete nonlinear systems [J]. Control and Decision, 2014, 29(3):449-454(in Chinese).
[13] Pang Bo, Shao Cheng. PID-type parameter optimal iterative learning control algorithm based on singular value decomposition[J]. Information and Control, 2014, 43(4): 483-489(in Chinese).