IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 19, 2019, accepted December 31, 2019, date of publication January 13, 2020, date of current version February 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2966300

Enhancing Accuracy and Numerical Stability for Repetitive Time-Varying System Identification: An Iterative Learning Approach

FAZHI SONG[1,2], (Member, IEEE), YANG LIU[1,2], (Member, IEEE), XIANLI WANG[3], WEN JIN[4], AND LI LI[1,2]

[1]Center of Ultra-Precision Optoelectronic Instrument Engineering, Harbin Institute of Technology, Harbin 150080, China
[2]Key Lab of Ultra-Precision Intelligent Instrumentation, Ministry of Industry and Information Technology, Harbin 150080, China
[3]College of Physical and Electronic Information, Luoyang Normal University, Luoyang 471934, China
[4]School of Pharmaceutical Sciences, Harbin Medical University, Harbin 150081, China

Corresponding author: Yang Liu (hitlg@hit.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51907038, in part by the China Postdoctoral Science Foundation under Grant 2019M651278, in part by the Heilongjiang Province Postdoctoral Science Foundation under Grant LBH-Z19066, in part by the Key Scientific Research Program of Higher Education of Henan Province under Grant 18B480002, in part by the National Science and Technology Major Project under Grant 2017ZX02101007-001, and in part by the Scientific Research Foundation for Returned Scholars, Heilongjiang, under Grant LC2018022.

ABSTRACT Time-varying system identification is an appealing but challenging research area. Existing identification algorithms are usually subject to either low estimation accuracy or bad numerical stability. These deficiencies motivate the development of an iterative learning identification algorithm in this paper. Three distinguished features of the proposed method result in the achievement of high estimation accuracy and high numerical stability: i) *recursion along the iteration axis*, ii) *bias compensation*, and iii) *singular value decomposition* (SVD). Firstly, an extra iteration axis associated with the original time axis is introduced in the parameter estimation process. A norm-optimal identification approach with the balance between convergence speed and noise robustness is then proposed along the iteration axis, followed by further analysis on the accuracy and the numerical stability. Secondly, in order to eliminate the estimation bias in the presence of noise and thus to improve the accuracy, a bias compensation algorithm along the iteration axis is proposed. Thirdly, a SVD-based update algorithm for the covariance matrix is developed to avoid the possible numerical instability during iterations. Numerical examples are finally provided to validate the algorithm and confirm its effectiveness.

INDEX TERMS Iterative learning algorithm, time-varying system, parameter estimation, output-error system, bias compensation, singular value decomposition.

I. INTRODUCTION

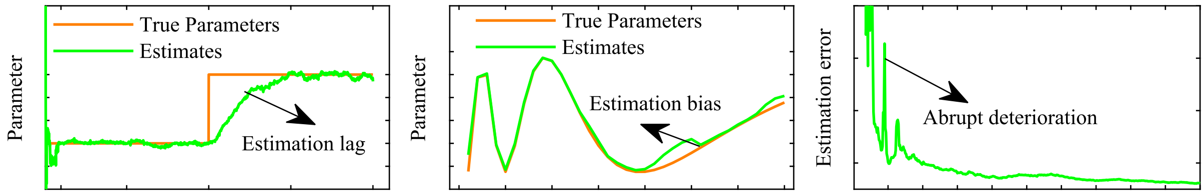
System identification has long been an appealing area in both theory research and practical applications [1]–[3]. Identification methods for systems that can be effectively characterized by time-invariant models have received considerable results [4]. However, there are numerous of practical industrial processes which are inherently time-varying, such as manufacturing processes, aerospace industry and biomedical systems [5]–[7]. In [8], a pick-and-place robot working in an assembly line is investigated. A significant dynamic variation occurs when a mass is picked up or released. Identification for such linear time-varying systems turns out to be a challenging problem and thus attracts wide interests in the community of engineers [9], [10].

In this paper, parameter estimation for the following linear time-varying (LTV) system is investigated.

$$y(k) = \frac{B(k,z)}{A(k,z)}u(k) + v(k), \quad k = 0, 1, 2, \ldots, N \tag{1}$$

where k is the time index, N is the pre-specified time interval, and z^{-1} denotes the backward shift operator with respect to time. Namely, $z^{-1}u(k) = u(k-1)$. $u(k)$ and $y(k)$ are the sampled system input and output respectively, and $v(k)$ is the observation noise acting on the system. $A(k,z)$ and $B(k,z)$ are

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Gaggero.



(a) Estimation lag (b) Estimation bias (c) Numerical instability

FIGURE 1. Deficiencies of existing identification algorithms for time-varying systems.

time-varying polynomials of z^{-1} with the following forms

$$A(k,z) = 1 + a_1(k)z^{-1} + a_2(k)z^{-2} + \ldots + a_n(k)z^{-n}, \quad (2)$$

$$B(k,z) = b_1(k)z^{-1} + b_2(k)z^{-2} + \ldots + b_m(k)z^{-m}, \quad (3)$$

where $a_1(k), a_2(k), \ldots, a_n(k)$ and $b_1(k), b_2(k), \ldots, b_m(k)$ are the time-varying parameters. n and m denote the orders of $A(k,z)$ and $B(k,z)$, respectively.

A. MOTIVATIONS

Generally speaking, the following excellent features are pursued in time-varying system identification.

F1) *High estimation accuracy*. Identification algorithms are expected to achieve estimation of time-varying parameters without time delay. The estimation lag which deteriorates the estimation accuracy at the time when the parameters vary should be avoided. In addition, in consideration of the inevitable observation noise in practice, identification algorithms are desired to eliminate the noise-induced estimation bias.

F2) *High numerical stability*. From a practical industrial point of view, the desired identification algorithms should have good numerical stability, ensuring no abrupt performance deterioration during convergence.

Although important theoretical efforts have been devoted to the time-varying system identification [10]–[12], the following deficiencies of the existing algorithms motivate the further investigation.

1) LIMITED TRACKING SPEED OF EXISTING RECURSIVE ALGORITHMS

Recursive least squares (RLS) algorithm is one of the most popular system identification methods owing to its good convergence property and small mean square error [13]. However, it fails to track time-varying parameters due to the asymptotically vanishing gain [14]. Some efforts have been contributed towards modifying the RLS algorithm for time-varying system identification [15]. In [16], a RLS method is proposed for the permanent magnet synchronous motor by coupling identification with bias compensation. In [17], an identification method based on the expansion of time-varying parameters onto a set of basis functions is proposed. This method is flexible for different applications by selecting different type of basis functions. However, how to characterize time-varying parameters by a fixed basis function set is not a trivial question, and hence the effectiveness of this method remains to be seen.

Introducing forgetting factors (FFs) to RLS is one of the most commonly used approaches for time-varying system identification [18]. In [19], a bias-compensation-based RLS algorithm with a FF (*FFBCRLS*) is proposed to improve the estimation accuracy. In [12], a robust forgetting factor based RLS algorithm is proposed for time-varying disturbances. However, in these RLS algorithms with constant FFs, a large FF close to one reduces the convergence speed while a small FF close to zero leads to large misadjustment [20]. In [20], [21], RLS algorithms with variable FFs have been developed to attain the minimal misadjustment as well as to estimate the optimal FF. Estimation lag, however, is inevitable in these FF-based recursive algorithms since they use previous information to estimate current parameters.

An illuminating example which indicates the slow tracking speed of existing recursive algorithms is shown in Fig. 1(a).

2) LOW ESTIMATION ACCURACY OF EXISTING ITERATIVE ALGORITHMS

Motivated by iterative learning control (ILC) [22], iterative learning identification (ILI) has attracted considerable research attentions in recent studies [23]. ILI is first introduced in [24] where an ILC-based method is proposed for the identification of linear time-invariant (LTI) systems. This method is extended in [25] to achieve robustness against noise through a Kalman filter, and in [26] to estimate parameters in the presence of input disturbance. The first implementation of ILI on a physical system is discussed in [27]. These early investigations mainly focus on identification of time-invariant systems.

In [28], an iterative learning approach based on least squares (*ILLS*) is proposed to identify repetitive systems with time-varying parametric uncertainties. In [29], a similar algorithm to ILLS, called iterative learning recursive least squares (ILRLS) identification algorithm, is proposed for linear LTV systems, and is extended to nonlinear systems. Compared with RLS, ILLS and ILRLS achieve no-lag estimation of time-varying parameters. These methods, however, result in biased estimation for output-error (OE) systems in spite of their unbiased estimation for autoregressive systems with exogenous input (ARX). This is because OE system is

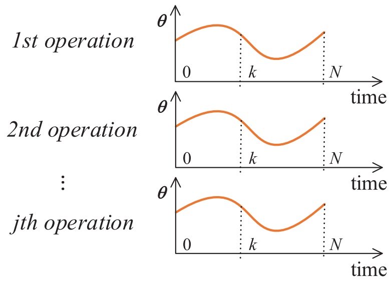


FIGURE 2. Description of repetitive time-varying parameters.

equivalent to equation-error-type system with colored noise while ARX system is white-noise.

The low estimation accuracy of existing iterative algorithms is illuminated in Fig. 1(b).

3) NUMERICAL INSTABILITY OF EXISTING COVARIANCE-MATRIX-BASED ALGORITHMS

There exists inherently numerical instability in RLS-like algorithms [8], [17], [19]–[21], [25], [28]. The covariance matrix of input signals may become poorly conditioned or even singular if input signals are not persistently exciting [30]. In addition, the round-off error yielded by the subtraction during the update of the covariance matrix may make it non-positive. In such situations, the estimation error will increase considerably or even become nonconvergent, as depicted in Fig. 1(c).

B. CONTRIBUTIONS

In this paper, parameter estimation for the LTV system in (1) with repetitive behaviors is studied. The repetitive time-varying parameter is illustrated in Fig. 2 where the parameter vector $\boldsymbol{\theta}(k)$ is defined as

$$\boldsymbol{\theta}(k) \triangleq [a_1(k)\, a_2(k)\, \ldots\, a_n(k)\, b_1(k)\, b_2(k)\, \ldots\, b_m(k)]^T. \quad (4)$$

There are two key features about the considered repetitive LTV system.

1) The time-varying system can operate multiple times over a finite interval $k \in [0\ N]$.
2) Although the parameters are time-varying for each operation, the parameters at the same time are invariant for different operations. More precisely, let $\boldsymbol{\theta}_j(k)$ denotes the time-varying parameters at the time k in the jth operation. It follows that $\boldsymbol{\theta}_1(k) = \boldsymbol{\theta}_2(k) = \ldots = \boldsymbol{\theta}_j(k)$, $k \in [0\ N]$, although $\boldsymbol{\theta}_i(0), \boldsymbol{\theta}_i(1), \ldots, \boldsymbol{\theta}_i(N)$, $i = 1, 2, \ldots, j$, are variant.

A large number of plants in industrial processes have repetitive behaviors such as the robot arm described in [8] where the robot picks up a steel mass at a fixed time and then releases it at other time. The model of the robot has a large variation when the mass is picked up or released.

In view of the limitations of existing identification algorithms, a norm-optimal iterative learning identification algorithm for the repetitive LTV system described in (1) is proposed with the achievement of F1 and F2 simultaneously. The main contributions of this work lie in the following.

C1) Besides the time axis, an extra iteration axis is introduced into the considered LTV system in (1), followed by the development of a norm-optimal identification algorithm that performs recursion along the iteration axis. The inevitable estimation lag in conventional recursive algorithms is eliminated.

C2) A novel bias compensation method in the iteration domain is proposed to mitigate the effect of observation noise on the estimation accuracy.

C3) A singular value decomposition (SVD) based update algorithm for the covariance matrix is developed to solve the numerical instability in conventional RLS-like algorithms.

The rest of this paper is organized as follows. Section II describes the basic regressive equations. Section III derives the norm-optimal ILI algorithm and analyzes its accuracy and numerical stability. In Section IV, a bias compensation scheme is proposed to improve the estimation accuracy, followed by Section V where a SVD based update algorithm for the covariance matrix is derived to improve the numerical stability. Section VI provides examples to show effectiveness of the proposed method.

II. PRELIMINARIES

The system described in (1) can be rewritten in a regressive form as

$$y(k) = \boldsymbol{\varphi}(k)\boldsymbol{\theta}(k) + \boldsymbol{\psi}(k)\boldsymbol{\theta}(k) + v(k) \quad (5)$$

where the information vector $\boldsymbol{\varphi}(k) \in \mathbb{R}^{1\times n_f}$, $n_f = n+m$, and the noise vector $\boldsymbol{\psi}(k) \in \mathbb{R}^{1\times n_f}$ are defined respectively as

$$\boldsymbol{\varphi}(k) \triangleq [-y(k-1)\ \ldots\ -y(k-n)\ u(k-1)\ \ldots\ u(k-m)], \quad (6)$$

$$\boldsymbol{\psi}(k) \triangleq [v(k-1)\ v(k-2)\ \ldots\ v(k-n)\ 0\ \ldots\ 0]. \quad (7)$$

Note that $\boldsymbol{\psi}(k)$ represents the set of previous noise information while $v(k)$ denotes the current noise information.

The following assumptions are made about the system described in (1).

1) For each operation, $v(k)$ is white noise with zero mean. $u(k)$ and $v(k)$ are statistically independent.
2) All the time-varying parameters are bounded, and the orders n and m are fixed and known.

Considering the repetitive nature of the system, an iteration axis as shown in Fig. 3 is introduced into the LTV system in (1). The parameters vary along the time axis for each iteration/operation, but are invariant along the iteration axis for each time. In other words, the time-varying behavior of the plant is iteration-invariant.

The system output in the jth iteration can be written as

$$y_j(k) = \boldsymbol{\varphi}_j(k)\boldsymbol{\theta}(k) + \boldsymbol{\psi}_j(k)\boldsymbol{\theta}(k) + v_j(k) \quad (8)$$

For all iterations, the stacked output vector $\boldsymbol{Y}_j(k)$, the stacked noise vector $\boldsymbol{V}_j(k)$, the stacked information

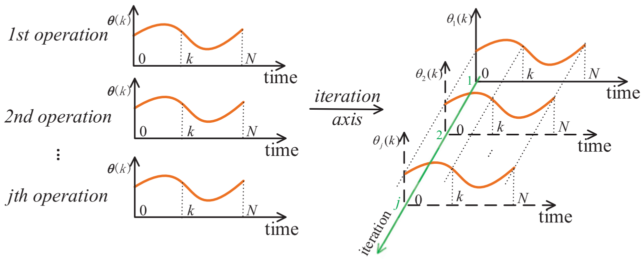


FIGURE 3. Introduction of the iteration axis.

matrix $\Phi_j(k)$ and the stacked noise matrix $\Psi_j(k)$ are defined respectively as

$$\begin{cases} \boldsymbol{Y}_j(k) \triangleq [y_1(k)\ y_2(k)\ \ldots\ y_j(k)]^T \in \mathbb{R}^{j\times 1}, \\ \boldsymbol{V}_j(k) \triangleq [v_1(k)\ v_2(k)\ \ldots\ v_j(k)]^T \in \mathbb{R}^{j\times 1}, \\ \Phi_j(k) \triangleq [\boldsymbol{\varphi}_1(k)\ \boldsymbol{\varphi}_2(k)\ \ldots\ \boldsymbol{\varphi}_j(k))]^T \in \mathbb{R}^{j\times n_f}, \\ \Psi_j(k) \triangleq [\boldsymbol{\psi}_1(k)\ \boldsymbol{\psi}_2(k)\ \ldots\ \boldsymbol{\psi}_j(k))]^T \in \mathbb{R}^{j\times n_f}. \end{cases} \tag{9}$$

Then the following equation can be obtained

$$\boldsymbol{Y}_j(k) = \Phi_j(k)\boldsymbol{\theta}(k) + \Psi_j(k)\boldsymbol{\theta}(k) + \boldsymbol{V}_j(k). \tag{10}$$

III. NORM-OPTIMAL ITERATIVE LEARNING IDENTIFICATION

Conventional recursive identification algorithms are performed along the time axis in the form of $\widehat{\boldsymbol{\theta}}(k) = \widehat{\boldsymbol{\theta}}(k-1) + L(k)\boldsymbol{e}(k)$. In this paper, an identification algorithm that performs recursion along the iteration axis is proposed in the form of

$$\widehat{\boldsymbol{\theta}}_j(k) = \widehat{\boldsymbol{\theta}}_{j-1}(k) + L_j(k)\boldsymbol{E}_j(k) \tag{11}$$

where $\widehat{\boldsymbol{\theta}}_j(k)$ is the estimate vector at the time index k in the jth iteration, $L_j(k)$ is the learning gain and $\boldsymbol{E}_j(k)$ is the estimation error defined as

$$\boldsymbol{E}_j(k) \triangleq \boldsymbol{Y}_j(k) - \boldsymbol{\Phi}_j(k)\widehat{\boldsymbol{\theta}}_j(k). \tag{12}$$

As discussed in Section I, estimation lag is inevitable in conventional recursive identification, even if through forgetting factors. However, due to the iteration-invariant characteristics of the time-varying parameters, the estimation lag can be eliminated in iterative identification algorithms.

A. ITERATIVE LEARNING IDENTIFICATION ALGORITHM

Here a norm-optimal approach is proposed to design the learning law in (11). The quadratic cost function is chosen for the balance between convergence speed and noise robustness, which is given as follows

$$\begin{aligned} J_j(k, \widehat{\boldsymbol{\theta}}_j(k)) &= \boldsymbol{E}_j^T(k)W_1\boldsymbol{E}_j(k) \\ &\quad + [\widehat{\boldsymbol{\theta}}_j(k)-\widehat{\boldsymbol{\theta}}_{j-1}(k)]^T W_2[\widehat{\boldsymbol{\theta}}_j(k)-\widehat{\boldsymbol{\theta}}_{j-1}(k)] \end{aligned} \tag{13}$$

where $W_1 \in \mathbb{R}^{j\times j}$ and $W_2 \in \mathbb{R}^{n_f\times n_f}$ are positive definite weighting matrices on the estimation error and the estimate increment, respectively. In this paper, the weights on the estimation error from the first iteration to the jth iteration are set to be same. So are the weights on the estimate increment. More precisely, we define $W_1 \triangleq \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_1 \end{bmatrix}$ and $W_2 \triangleq \begin{bmatrix} w_2 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_2 \end{bmatrix}$ where w_1 and w_2 are scalar weighting coefficients.

The cost function in (13) is very similar to the norm-optimal ILC [31] except the term for input energy. The inclusion of the penalty on the estimate changing rate improves the robustness against noise although the convergence speed may be decreased.

For the optimization problem in (13), minimizing $J_j(k, \widehat{\boldsymbol{\theta}}_j(k))$ by taking the partial derivative with respect to $\widehat{\boldsymbol{\theta}}_j(k)$ and setting it to zero lead to

$$\widehat{\boldsymbol{\theta}}_j(k) = \widehat{\boldsymbol{\theta}}_{j-1}(k) + W_2^{-1}\Phi_j^T(k)W_1\boldsymbol{E}_j(k). \tag{14}$$

From the comparison with (11), it can be observed that the learning gain is $L_j(k) = W_2^{-1}\Phi_j^T(k)W_1$. However, the parameter update law in (14) is not implementable since $\boldsymbol{E}_j(k)$ is not available before $\widehat{\boldsymbol{\theta}}_j(k)$ is obtained according to (12).

When the iteration index j is large enough, at least n_f, $W_2 + \Phi_j^T(k)W_1\Phi_j(k)$ will be full-rank. Substituting (12) into (14) yields

$$\begin{aligned} \widehat{\boldsymbol{\theta}}_j(k) &= \left[W_2 + \Phi_j^T(k)W_1\Phi_j(k)\right]^{-1} W_2\widehat{\boldsymbol{\theta}}_{j-1}(k) \\ &\quad + \left[W_2 + \Phi_j^T(k)W_1\Phi_j(k)\right]^{-1} \Phi_j^T(k)W_1\boldsymbol{Y}_j(k) \end{aligned} \tag{15}$$

which provides an implementable parameters update law.

Remark 1: The iterative learning law in (15) seems to be similar to the one in [8], but they are fundamentally different in the information data used for iteration. Information matrix $\Phi_j(k)$ in [8] consists of sampled input/output (I/O) data in a time window at the current iteration, i.e. $\Phi_j(k) \triangleq [\boldsymbol{\varphi}_j(k-p)\ \ldots\ \boldsymbol{\varphi}_j(k)\ \ldots\ \boldsymbol{\varphi}_j(k+q)]$ where $[k-p, k+q]$ is the time window around the time k. However, we do not use the concept of *time window* in this paper. $\Phi_j(k)$ here is composed of I/O data from the first to the jth iterations at the current time, i.e. $\Phi_j(k) \triangleq [\boldsymbol{\varphi}_1(k)\ \boldsymbol{\varphi}_2(k)\ \ldots\ \boldsymbol{\varphi}_j(k)]$. In a word, information data in this paper is collected along the iteration axis instead of the time axis.

The iterative update law in (15) uses not only the current I/O data but also the I/O data from all the previous iterations. If it is performed directly, there would be a heavy calculation due to the large size of $\Phi_j(k)$ as well as the requirement of large storage space for all-iterations data. Now, the update law in (15) will be simplified further such that only the sampled I/O data at the current iteration is required.

Define

$$P_j(k) \triangleq \left[W_2 + \Phi_j^T(k)W_1\Phi_j(k)\right]^{-1} \in \mathbb{R}^{n_f\times n_f}. \tag{16}$$

Obviously,

$$P_j^{-1}(k) = P_{j-1}^{-1}(k) + w_1\boldsymbol{\varphi}_j^T(k)\boldsymbol{\varphi}_j(k). \tag{17}$$

Based on the matrix-inversion algorithm, it follows that

$$P_j(k) = P_{j-1}(k) - \frac{w_1 P_{j-1}(k)\boldsymbol{\varphi}_j^T(k)\boldsymbol{\varphi}_j(k)P_{j-1}(k)}{1 + w_1\boldsymbol{\varphi}_j(k)P_{j-1}(k)\boldsymbol{\varphi}_j^T(k)}. \tag{18}$$

Denote $\Delta\widehat{\boldsymbol{\theta}}_j(k) \triangleq \widehat{\boldsymbol{\theta}}_j(k) - \widehat{\boldsymbol{\theta}}_{j-1}(k)$ as the vector of estimate increment. Substituting (16) into (15) leads to the iterative update law for $\Delta\widehat{\boldsymbol{\theta}}_j(k)$ as follows

$$\Delta\widehat{\boldsymbol{\theta}}_j(k) = w_2 P_j(k)\Delta\widehat{\boldsymbol{\theta}}_{j-1}(k) + L_j(k)[y_j(k) - \boldsymbol{\varphi}_j(k)\widehat{\boldsymbol{\theta}}_{j-1}(k)] \tag{19}$$

where

$$L_j(k) = w_1 P_j(k)\boldsymbol{\varphi}_j^T(k) = \frac{w_1 P_{j-1}(k)\boldsymbol{\varphi}_j^T(k)}{1 + w_1\boldsymbol{\varphi}_j(k)P_{j-1}(k)\boldsymbol{\varphi}_j^T(k)}. \tag{20}$$

From (19), it can be observed that, besides the last-iteration estimates and their increments, only sampled I/O data in the current iteration, i.e. $\boldsymbol{\varphi}_j(k)$, is used to update the estimates. No I/O data from previous iterations are required. The proposed iterative identification algorithm can be summarized as follows

$$\begin{cases} \widehat{\boldsymbol{\theta}}_j(k) = \widehat{\boldsymbol{\theta}}_{j-1}(k) + \Delta\widehat{\boldsymbol{\theta}}_j(k) \\ \Delta\widehat{\boldsymbol{\theta}}_j(k) = w_2 P_j(k)\Delta\widehat{\boldsymbol{\theta}}_{j-1}(k) \\ \qquad\qquad + L_j(k)[y_j(k) - \boldsymbol{\varphi}_j(k)\widehat{\boldsymbol{\theta}}_{j-1}(k)], \\ L_j(k) = w_1 P_j(k)\boldsymbol{\varphi}_j^T(k), \\ P_j(k) = P_{j-1}(k) - \dfrac{w_1 P_{j-1}(k)\boldsymbol{\varphi}_j^T(k)\boldsymbol{\varphi}_j(k)P_{j-1}(k)}{1 + w_1\boldsymbol{\varphi}_j(k)P_{j-1}(k)\boldsymbol{\varphi}_j^T(k)}. \end{cases} \tag{21}$$

Remark 2: It is interesting to note that, except its execution along the iteration axis, the proposed algorithm in (21) looks similar to the RLS algorithm. The involvement of the penalty on the estimate changing rate in the cost function, however, makes the proposed algorithm mainly focus on the update law of the estimate increment, i.e. $\Delta\widehat{\boldsymbol{\theta}}_j(k)$, rather than the estimate itself, i.e. $\widehat{\boldsymbol{\theta}}_j(k)$. This is different from the RLS algorithm. The penalty term achieves a tradeoff between the convergence speed and the robustness. What is more, regardless of the iterative execution way, the RLS algorithm can be actually regarded as a special case of the proposed algorithm in (21) by setting w_2 to be zero.

B. PERFORMANCE ANALYSIS

In this section, estimation accuracy and numerical stability are analyzed for the proposed iterative identification algorithm in (21).

1) ESTIMATION ACCURACY

From the assumptions in Section II, it follows that $v_i(k)$ is independent-of-inputs white noise with zero mean and the iterations are mutually independent. According to the law of large numbers, we have in probability as j gets large

$$\begin{cases} \dfrac{1}{j}\sum_{i=1}^{j} v_i^2(k) \to \sigma^2 \\ \dfrac{1}{j}\sum_{i=1}^{j} u_i(k)v_i(q) \to 0 \end{cases} \tag{22}$$

where $q = 0, 1, \ldots, N$. (For simplicity, we will omit in probability late if there is no confusion.) Therefore, it can be obviously obtained that

$$\lim_{j\to\infty}\frac{1}{j}\sum_{i=1}^{j} y_i(k)v_i(\bar{k}) = \begin{cases} \sigma^2, & \bar{k} = k. \\ 0, & \bar{k} \neq k. \end{cases} \tag{23}$$

Based on the above knowledge, the converged estimates are presented in the following theorem.

Theorem 1: Consider the repetitive LTV system in (1) satisfying the assumptions in Section II. Provided that the inputs are persistent-excitation along the iteration axis, and that they are stationary and ergodic, then

$$\lim_{j\to\infty}\widehat{\boldsymbol{\theta}}_j(k) = \boldsymbol{\theta}(k) - w_1\sigma^2[\lim_{j\to\infty} jP_j(k)]\Omega\boldsymbol{\theta}(k) \tag{24}$$

is satisfied where $\Omega \triangleq \begin{bmatrix} I_n & O \\ O & O_m \end{bmatrix}$.

Proof: Substituting (16) into (15), it follows that

$$\widehat{\boldsymbol{\theta}}_j(k) = w_2 P_j(k)\widehat{\boldsymbol{\theta}}_{j-1}(k) + w_1 P_j(k)\Phi_j^T(k)\left[\Phi_j(k)\boldsymbol{\theta}(k) + \Psi_j(k)\boldsymbol{\theta}(k) + \boldsymbol{V}_j(k)\right]. \tag{25}$$

Therefore,

$$w_1^{-1}[P_j^{-1}(k)\widehat{\boldsymbol{\theta}}_j(k) - w_2\widehat{\boldsymbol{\theta}}_{j-1}(k)] = \Phi_j^T(k)\Psi_j(k)\boldsymbol{\theta}(k) + \Phi_j^T(k)\boldsymbol{V}_j(k) + \Phi_j^T(k)\Phi_j(k)\boldsymbol{\theta}(k). \tag{26}$$

For the terms on the right hand of (26), we have

$$\begin{cases} \Phi_j^T(k)\Psi_j(k)\boldsymbol{\theta}(k) = \sum_{i=1}^{j}\boldsymbol{\varphi}_i^T(k)\boldsymbol{\psi}_i(k)\boldsymbol{\theta}(k), \\ \Phi_j^T(k)\boldsymbol{V}_j(k) = \sum_{i=1}^{j}\boldsymbol{\varphi}_i^T(k)v_i(k). \end{cases} \tag{27}$$

Dividing both sides of (27) by j and taking limit, it follows that

$$\begin{cases} \lim_{j\to\infty}\dfrac{1}{j}\Phi_j^T(k)\Psi_j(k)\boldsymbol{\theta}(k) = -\sigma^2\Omega\boldsymbol{\theta}(k), \\ \lim_{j\to\infty}\dfrac{1}{j}\Phi_j^T(k)\boldsymbol{V}_j(k) = \mathbf{0}. \end{cases} \tag{28}$$

If the inputs are persistent-excitation in the iteration domain, the persistent excitation (PE) condition $\frac{1}{j}\Phi_j^T \Phi_j(k) = \frac{1}{j}\sum_{i=1}^{j}\boldsymbol{\varphi}_i^T(k)\boldsymbol{\varphi}_i(k) > 0$ holds. Provided that the inputs are stationary and ergodic, then $\lim_{j\to\infty}\frac{1}{j}\Phi_j^T\Phi_j(k)$ exists and is independent of the iteration index j, i.e. $\lim_{j\to\infty}\frac{1}{j}\Phi_j^T\Phi_j(k)$ is a constant matrix for the fixed k. Therefore, $\lim_{j\to\infty}\frac{1}{j}P_j^{-1}(k) = \lim_{j\to\infty}\frac{1}{j}W_2 + \lim_{j\to\infty}\frac{1}{j}\Phi_j^T W_1\Phi_j(k) = w_1\lim_{j\to\infty}\frac{1}{j}\Phi_j^T\Phi_j(k)$ is also a constant matrix for the fixed k.

Dividing both sides of (26) by j and taking limit, we have

$$\lim_{j\to\infty}\frac{1}{j}w_1^{-1}P_j^{-1}(k)\widehat{\boldsymbol{\theta}}_j(k) = -\sigma^2\Omega\boldsymbol{\theta}(k)+\left[\lim_{j\to\infty}\frac{1}{j}w_1^{-1}P_j^{-1}(k)\right]\boldsymbol{\theta}(k). \quad (29)$$

Then the limit of $\widehat{\boldsymbol{\theta}}_j(k)$ can be obtained as follows

$$\lim_{j\to\infty}\widehat{\boldsymbol{\theta}}_j(k)=\boldsymbol{\theta}(k)-w_1\sigma^2\left[\lim_{j\to\infty}jP_j(k)\right]\Omega\boldsymbol{\theta}(k). \quad (30)$$

This completes the proof. □

Theorem 1 indicates that bias between $\widehat{\boldsymbol{\theta}}_j(k)$ and $\boldsymbol{\theta}(k)$ exists, and it results primarily from the observation noise.

2) NUMERICAL STABILITY

The definition of $P_j(k)$ as shown in (16) indicates that $P_j(k)$ is a symmetric positive definite matrix. However, the iterative update law for $P_j(k)$ in (18) may make it non-positive definite if the input signals are not persistently exciting [30]. On the other hand, the round-off error yielded by the subtraction in (18) may also make $\boldsymbol{P}_j(k)$ non-positive definite. In such situations, the estimates will become unstable as illustrated in Fig. 1(c).

IV. ACCURACY ENHANCEMENT BY BIAS COMPENSATION

In this section, a bais compensation algorithm is proposed to eliminate the noise-induced estimation bias.

A. BIAS COMPENSATION SCHEME

Equation (34) can be rewritten as

$$\boldsymbol{\theta}(k)=\lim_{j\to\infty}\left[\widehat{\boldsymbol{\theta}}_j(k)+jw_1\sigma^2P_j(k)\Omega\boldsymbol{\theta}(k)\right], \quad (31)$$

from which it is indicated that $\widetilde{\boldsymbol{\theta}}_j(k) \triangleq \widehat{\boldsymbol{\theta}}_j(k) + jw_1\sigma^2 P_j(k)\Omega\boldsymbol{\theta}(k)$ provides an unbiased estimation for the true parameter vector $\boldsymbol{\theta}(k)$. Therefore, the compensation term $jw_1\sigma^2 P_j(k)\Omega\boldsymbol{\theta}(k)$ can be introduced into $\widehat{\boldsymbol{\theta}}_j(k)$ to improve the estimation accuracy. Since the noise variance σ^2 and the true parameter vector $\boldsymbol{\theta}(k)$ are both unknown, here the compensation term is designed as $jw_1\widehat{\sigma}^2 P_j(k)\Omega\widetilde{\boldsymbol{\theta}}_{j-1}(k)$ where $\widehat{\sigma}$ is the estimate of σ. The bias compensation algorithm can be consequently given in the following form

$$\widetilde{\boldsymbol{\theta}}_j(k)=\widehat{\boldsymbol{\theta}}_j(k)+jw_1\widehat{\sigma}^2P_j(k)\Omega\widetilde{\boldsymbol{\theta}}_{j-1}(k). \quad (32)$$

Since $\widehat{\boldsymbol{\theta}}_j(k)$ and $P_j(k)$ can be determined by (21) and Ω is a constant known matrix, the key to obtain the unbiased estimate $\widetilde{\boldsymbol{\theta}}_j(k)$ lies in the estimation of the noise variance.

Remark 3: Equation (34) can also be rewritten as $\lim_{j\to\infty}\widehat{\boldsymbol{\theta}}_j(k)=\left\{I-w_1\sigma^2\left[\lim_{j\to\infty}jP_j(k)\right]\Omega\right\}\boldsymbol{\theta}(k)$. If the inverse of $\left[I-jw_1\widehat{\sigma}^2 P_j(k)\Omega\right]$ exists, another bias compensation algorithm can be given as follows

$$\widetilde{\boldsymbol{\theta}}_j(k)=\left[I-jw_1\widehat{\sigma}^2P_j(k)\Omega\right]^{-1}\widehat{\boldsymbol{\theta}}_j(k). \quad (33)$$

B. NOISE VARIANCE ESTIMATION

To estimate the noise variance, we present the following theorem.

Theorem 2: Consider the quadratic cost function $J_j(k)$ shown in (13). The limit of $\frac{1}{j}J_j(k)$ satisfies

$$\lim_{j\to\infty}\frac{1}{j}J_j(k)=w_1\sigma^2[1+\boldsymbol{\theta}^T(k)\Omega\lim_{j\to\infty}\widehat{\boldsymbol{\theta}}_j(k)]. \quad (34)$$

The estimate of the noise variance can be consequently given by

$$\widehat{\sigma}^2=\frac{J_j(k)}{jw_1[1+\widetilde{\boldsymbol{\theta}}_{j-1}^T(k)\Omega\widehat{\boldsymbol{\theta}}_j(k)]}. \quad (35)$$

Proof: From (14), it follows that

$$\boldsymbol{E}_j^T(k)W_1\Phi_j(k)=[\Delta\widehat{\boldsymbol{\theta}}_j(k)]^TW_2. \quad (36)$$

Then the cost function $J_j(k)$ described in (13) can be rewritten as

$$\begin{aligned}J_j(k) &= w_1\boldsymbol{Y}_j^T(k)\Psi_j(k)\boldsymbol{\theta}(k)+w_1\boldsymbol{Y}_j^T\boldsymbol{V}_j(k)\\ &\quad -w_1\widehat{\boldsymbol{\theta}}_j^T(k)\Phi_j^T(k)\Psi_j(k)\boldsymbol{\theta}(k)-w_1\widehat{\boldsymbol{\theta}}_j^T(k)\Phi_j^T(k)\boldsymbol{V}_j(k)\\ &\quad +w_2[\Delta\widehat{\boldsymbol{\theta}}_j(k)]^T[\boldsymbol{\theta}(k)-\widehat{\boldsymbol{\theta}}_{j-1}(k)]. \end{aligned} \quad (37)$$

For the first and second terms on the right hand of (37), we have

$$\begin{cases}\boldsymbol{Y}_j^T(k)\Psi_j(k)=\sum_{i=1}^j y_i(k)\Psi_i(k),\\ \boldsymbol{Y}_j^T(k)\boldsymbol{V}_j(k)=\sum_{i=1}^j y_i(k)v_i(k).\end{cases} \quad (38)$$

Dividing both sides of (38) by j and taking limit lead to

$$\begin{cases}\lim_{j\to\infty}\frac{1}{j}\boldsymbol{Y}_j^T(k)\Psi_j(k)=0,\\ \lim_{j\to\infty}\frac{1}{j}\boldsymbol{Y}_j^T(k)\boldsymbol{V}_j(k)=\sigma^2.\end{cases} \quad (39)$$

According to (28) and (39), dividing both sides of (37) by j and taking limit, it follows that

$$\lim_{j\to\infty}\frac{1}{j}J_j(k)=w_1\sigma^2[1+\boldsymbol{\theta}^T(k)\Omega\lim_{j\to\infty}\widehat{\boldsymbol{\theta}}_j(k)]. \quad (40)$$

Then the noise variance σ^2 can be given as

$$\sigma^2=\frac{\lim_{j\to\infty}\frac{1}{j}J_j(k)}{w_1[1+\boldsymbol{\theta}^T(k)\Omega lim_{j\to\infty}\widehat{\boldsymbol{\theta}}_j(k)]}. \quad (41)$$

Thus the estimate of σ^2 can be obtained as (35) shows. This completes the proof. □

Theorem 2 indicates that the estimation of the noise variance is mainly dependent on the cost function $J_j(k)$. However, from the definition of $J_j(k)$) in (13), $J_j(k)$ uses not only the current-iteration I/O data but also the I/O data from all the previous iterations, which leads it to be an intractable issue to calculate $J_j(k)$ directly from its definition. To tackle this issue, an iterative update law for $J_j(k)$ is necessitated.

C. COST FUNCTION ITERATIVE UPDATE

From (12), the iterative relationship between $\boldsymbol{E}_j(k)$ and $\boldsymbol{E}_{j-1}(k)$ can be expressed as follows

$$\boldsymbol{E}_j(k) = \begin{bmatrix} \boldsymbol{E}_{j-1}(k) - \Phi_{j-1}(k)\Delta\widehat{\boldsymbol{\theta}}_j(k) \\ e_j(k) \end{bmatrix} \tag{42}$$

where $e_j(k) = y_j(k) - \boldsymbol{\varphi}_j(k)\widehat{\boldsymbol{\theta}}_j(k)$.

Then the first term $\boldsymbol{E}_j^T(k)W_1\boldsymbol{E}_j(k)$ in the right hand of (13) can be rewritten as

$$\begin{aligned}\boldsymbol{E}_j^T(k)W_1\boldsymbol{E}_j(k) &= w_1\boldsymbol{E}_{j-1}^T(k)\boldsymbol{E}_{j-1}(k) - 2w_2\Delta\widehat{\boldsymbol{\theta}}_j^T(k)\Delta\widehat{\boldsymbol{\theta}}_{j-1} \\ &\quad + \Delta\widehat{\boldsymbol{\theta}}_j^T(k)\left[P_{j-1}^{-1}(k) - W_2\right]\Delta\widehat{\boldsymbol{\theta}}_j(k) + w_1e_j^2(k).\end{aligned} \tag{43}$$

Substituting (43) into (13), we have

$$\begin{aligned}J_j(k) &= w_1\boldsymbol{E}_{j-1}^T(k)\boldsymbol{E}_{j-1}(k) + w_1e_j^2(k) \\ &\quad - 2w_2\Delta\widehat{\boldsymbol{\theta}}_j^T(k)\Delta\widehat{\boldsymbol{\theta}}_{j-1}(k) + \Delta\widehat{\boldsymbol{\theta}}_j^T(k)P_{j-1}^{-1}(k)\Delta\widehat{\boldsymbol{\theta}}_j(k).\end{aligned} \tag{44}$$

The term $-2w_2\Delta\widehat{\boldsymbol{\theta}}_j^T(k)\Delta\widehat{\boldsymbol{\theta}}_{j-1}(k)+\Delta\widehat{\boldsymbol{\theta}}_j^T(k)P_{j-1}^{-1}(k)\,\Delta\widehat{\boldsymbol{\theta}}_j(k)$ in (44) can be simplified as

$$\begin{aligned}&-2w_2\Delta\widehat{\boldsymbol{\theta}}_j^T(k)\Delta\widehat{\boldsymbol{\theta}}_{j-1}(k) + \Delta\widehat{\boldsymbol{\theta}}_j^T(k)P_{j-1}^{-1}(k)\Delta\widehat{\boldsymbol{\theta}}_j(k) \\ &= \Delta\widehat{\boldsymbol{\theta}}_j^T(k)\Big\{w_1\boldsymbol{\varphi}_j^T(k)[y_j(k) - \boldsymbol{\varphi}_j(k)\widehat{\boldsymbol{\theta}}_j(k)] \\ &\quad - w_2\Delta\widehat{\boldsymbol{\theta}}_{j-1}(k)\Big\}.\end{aligned} \tag{45}$$

Substituting (45) into (44), it follows that

$$\begin{aligned}J_j(k) &= J_{j-1}(k) - w_2\Delta\widehat{\boldsymbol{\theta}}_{j-1}^T(k)[\Delta\widehat{\boldsymbol{\theta}}_j(k) + \Delta\widehat{\boldsymbol{\theta}}_{j-1}(k)] \\ &\quad + w_1[y_j(k) - \boldsymbol{\varphi}_j(k)\widehat{\boldsymbol{\theta}}_j(k)][y_j(k) - \boldsymbol{\varphi}_j(k)\widehat{\boldsymbol{\theta}}_{j-1}(k)].\end{aligned} \tag{46}$$

From (32), (35) and (46), the bias compensation algorithm for *accuracy enhancement* can be summarized as follows

$$\begin{cases}\widetilde{\boldsymbol{\theta}}_j(k) = \widehat{\boldsymbol{\theta}}_j(k) + jw_1\widehat{\sigma}^2P_j(k)\Omega\widetilde{\boldsymbol{\theta}}_{j-1}(k), \\ \widehat{\sigma}^2 = \dfrac{J_j(k)}{jw_1\left[1 + \widetilde{\boldsymbol{\theta}}_{j-1}^T(k)\Omega\widehat{\boldsymbol{\theta}}_j(k)\right]}, \\ J_j(k) = J_{j-1}(k) - w_2\Delta\widehat{\boldsymbol{\theta}}_{j-1}^T(k)[\Delta\widehat{\boldsymbol{\theta}}_j(k) + \Delta\widehat{\boldsymbol{\theta}}_{j-1}(k)] \\ \qquad + w_1[y_j(k) - \boldsymbol{\varphi}_j(k)\widehat{\boldsymbol{\theta}}_j(k)][y_j(k) - \boldsymbol{\varphi}_j(k)\widehat{\boldsymbol{\theta}}_{j-1}(k)].\end{cases} \tag{47}$$

Remark 4: It should be pointed out that the principle of bias compensation has been widely used in time-invariant parameter estimation [16], [19], [32], [33], but is rarely studied in time-varying systems. In [19], a forgetting factor is involved into the bias-compensation-based RLS algorithm proposed in [32] to estimate time-varying parameters. However, a big estimation lag appears when parameters vary, although high estimation accuracy is achieved when parameters are invariant.

Remark 5: The existing bias compensation algorithms such as the one in [32] can be regarded as special cases of the proposed algorithm in (46) by setting the weighting coefficient w_2 as zero.

V. NUMERICAL STABILITY ENHANCEMENT BY SVD

Since $P_j(k)$ is a symmetric positive definite matrix, there exists an orthogonal matrix $Q_j(k) \in \mathbb{R}^{n_f\times n_f}$ by which $P_j(k)$ can be decomposed as follows based on SVD

$$P_j(k) = Q_j(k)\Lambda_j^2(k)Q_j^T(k) \tag{48}$$

where $\Lambda_j(k) = diag(\sigma_{j\_1}\ \sigma_{j\_2}\ \ldots\ \sigma_{j\_n_f})$, and $\sigma_{j\_1} \geq \sigma_{j\_2} \geq \ldots \geq \sigma_{j\_n_f}$ are singular values of $P_j(k)$.

Substituting (48) into (17), it follows that

$$\begin{aligned}&\left[Q_j(k)\Lambda_j^2(k)Q_j^T(k)\right]^{-1} \\ &= \left[Q_{j-1}^T(k)\right]^{-1}\Big[\Lambda_{j-1}^{-2}(k) \\ &\quad + w_1Q_{j-1}^T(k)\boldsymbol{\varphi}_j^T(k)\boldsymbol{\varphi}_j(k)Q_{j-1}(k)\Big]Q_{j-1}^{-1}(k).\end{aligned} \tag{49}$$

Define $\Gamma_j(k) \triangleq \begin{bmatrix}\sqrt{w_1}\boldsymbol{\varphi}_j(k)Q_{j-1}(k) \\ \Lambda_{j-1}^{-1}(k)\end{bmatrix} \in \mathbb{R}^{(n_f+1)\times n_f}$. It can be obtained that

$$\Gamma_j^T(k)\Gamma_j(k) = \Lambda_{j-1}^{-2}(k) + w_1Q_{j-1}^T(k)\boldsymbol{\varphi}_j^T(k)\boldsymbol{\varphi}_j(k)Q_{j-1}(k). \tag{50}$$

Substituting (50) into (49) yields

$$\left[Q_j(k)\Lambda_j^2(k)Q_j^T(k)\right]^{-1} = \left[Q_{j-1}^T(k)\right]^{-1}\Gamma_j^T(k)\Gamma_j(k)Q_{j-1}^{-1}(k). \tag{51}$$

The SVD of $\Gamma_j(k)$ can be given by

$$\Gamma_j(k) = \overline{Q}_j(k)\begin{bmatrix}\overline{\Lambda}_j(k) \\ 0\end{bmatrix}\left[\overline{R}_j(k)\right]^T \tag{52}$$

where $\overline{Q}_j(k) \in \mathbb{R}^{(n_f+1)\times(n_f+1)}$ and $\overline{R}_j(k) \in \mathbb{R}^{n_f\times n_f}$ are orthogonal matrices, and $\overline{\Lambda}_j(k) \in \mathbb{R}^{n_f\times n_f}$ is the singular value matrix of $\Gamma_j(k)$.

Substituting (52) into (51), it follows that

$$\begin{aligned}&\left[Q_j(k)\Lambda_j^2(k)Q_j^T(k)\right]^{-1} \\ &= \left\{Q_{j-1}(k)\overline{R}_j(k)\overline{\Lambda}_j^{-2}(k)\left[Q_{j-1}(k)\overline{R}_j(k)\right]^T\right\}^{-1}.\end{aligned} \tag{53}$$

Obviously,

$$\begin{cases}Q_j(k) = Q_{j-1}(k)\overline{R}_j(k), \\ \Lambda_j(k) = \overline{\Lambda}_j^{-1}(k).\end{cases} \tag{54}$$

Now the SVD based update law for $P_j(k)$ can be summarized as follows

$$\begin{cases}P_j(k) = Q_j(k)\Lambda_j^2(k)Q_j^T(k), \\ Q_j(k) = Q_{j-1}(k)\overline{R}_j(k), \\ \Lambda_j(k) = \overline{\Lambda}_j^{-1}(k), \\ \begin{bmatrix}\sqrt{w_1}\boldsymbol{\varphi}_j(k)Q_{j-1}(k) \\ \Lambda_{j-1}^{-1}(k)\end{bmatrix} = \overline{Q}_j(k)\begin{bmatrix}\overline{\Lambda}_j(k) \\ 0\end{bmatrix}\left[\overline{R}_j(k)\right]^T.\end{cases} \tag{55}$$

By now, based on the above results in (21), (47) and (55), we have established the enhanced norm-optimal iterative learning identification (ENOILI) algorithm for the LTV system described in (1). The steps involved in the ENOILI algorithm are listed as follows.

S1) Initialize the ENOILI algorithm. The iteration index j is set to be zero. Initialize w_1 as 1 and determine w_2 taking into consideration the balance between convergence speed and noise robustness. Set $P_0(k)$ as $[w_2 I + \Xi]^{-1}$ where Ξ is a small positive definite matrix, for example $\Xi = 10^{-6} I_{n+m}$. Decompose $P_0(k)$ based on SVD and then get the initial values of $Q_j(k)$ and $\Lambda_j(k)$, i.e. $Q_0(k)$ and $\Lambda_0(k)$. The cost function $J_0(k)$ and the estimate vectors $\widehat{\boldsymbol{\theta}}_0(k)$, $\widetilde{\boldsymbol{\theta}}_0(k)$ and $\Delta\widehat{\boldsymbol{\theta}}_0(k)$ are all initialized to be zeros, $k = 0, 1, \ldots, N$.

S2) Increase the iteration index j by one. Consider the LTV system described by (1) and collect the sampled I/O data $u_j(k)$ and $y_j(k)$. Then construct the information vector $\boldsymbol{\varphi}_j(k)$.

S3) Update $P_j(k)$ based on (55). Calculate the learning gain $L_j(k)$ by (20) and then update the estimate vector $\widehat{\boldsymbol{\theta}}_j(k)$ by (19).

S4) Calculate $J_j(k)$ by (46) and estimate the noise variance by (35). Then update the estimate vector $\widetilde{\boldsymbol{\theta}}_j(k)$ by (32).

S5) Compare $\widetilde{\boldsymbol{\theta}}_j(k)$ with $\widetilde{\boldsymbol{\theta}}_{j-1}(k)$, if

$$\sum_{k=1}^{N} \| \widetilde{\boldsymbol{\theta}}_j(k) - \widetilde{\boldsymbol{\theta}}_{j-1}(k) \| \leq \varepsilon \tag{56}$$

where ε is a small positive real number determined by user, then terminate the identification procedure. Otherwise go to **S**2 and begin the $(j+1)$th iteration.

VI. SIMULATION RESULTS

To illustrate the proposed algorithm and validate its effectiveness, two numerical simulation examples are provided in this section. Example 1 is provided to verify the merit of the proposed ENOILI algorithm versus conventional recursive algorithms, while example 2 is provided to show the enhanced performance compared with the existing iterative algorithms.

A. EXAMPLE 1: COMPARISON RESULTS WITH CONVENTIONAL RECURSIVE IDENTIFICATION ALGORITHMS

To show the differences between the proposed ENOILI algorithm with conventional recursion algorithms, here we choose one of the latest recursive identification algorithm for LTV systems, namely FFBCRLS algorithm proposed in [19], as a contrast.

The considered LTV system is a linear motor used in a wafer stage system as shown in Fig. 4. It is responsible for the long-stroke motion of the wafer stage which is an important mechatronic unit in photolithography, one of the significant processes in integrated circuits manufacturing. Two same wafter stages are used in photolithography in order to reduce the overhead time created by wafer exchange. While the first stage performs overhead activities such as wafer unload/load, horizontal alignment and measurement of the surface topography, the second wafer exposes the previously measured wafer. When both stages are finished with their tasks, the stages are swapped and a new cycle begins. See [34], [35] for the more descriptions about the photolithography process.

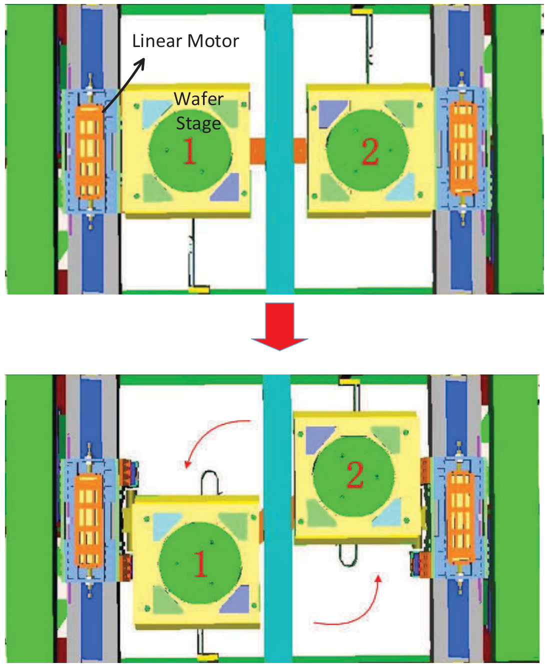


FIGURE 4. Wafer stage system. When the linear motor releases or captures the wafer stage, the load mass has an abrupt variation.

When the swap begins, the linear motor releases the wafer stage, followed by the capture of the other equivalent one. Therefore, a big model variation occurs for the linear motor when the wafer stage is released or captured. Suppose that the period for the wafer stage swap is 4s, the mass of the liner motor is 20kg and the mass of the wafer stage is 30kg. Then based on a rigid body assumption, the true model of the linear motor can be given as follows

$$Y(s) = \frac{1}{Ms^2} U(s) \tag{57}$$

where $U(s)$ and $Y(s)$ are the input and output of the linear motor respectively, and M is the time-varying load given by

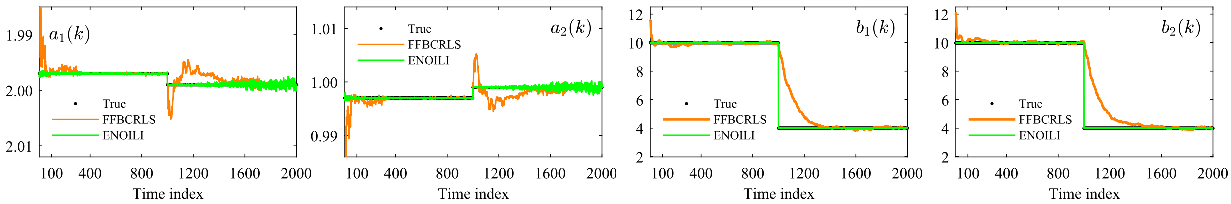
$$M = \begin{cases} 20, & 0 < t \leq 2, \\ 50, & 2 < t \leq 4. \end{cases} \tag{58}$$

The equivalent zero-order-hold discrete-time model with the sampling time $T_s = 2$ms can be given by (1), with the following time-varying polynomials

$$A(k, z) = \begin{cases} 1 - 1.997z^{-1} + 0.997z^{-2}, & 0 \leqslant k < 1000 \\ 1 - 1.999z^{-1} + 0.999z^{-2}, & 1000 < k \leqslant 2000 \end{cases} \tag{59}$$

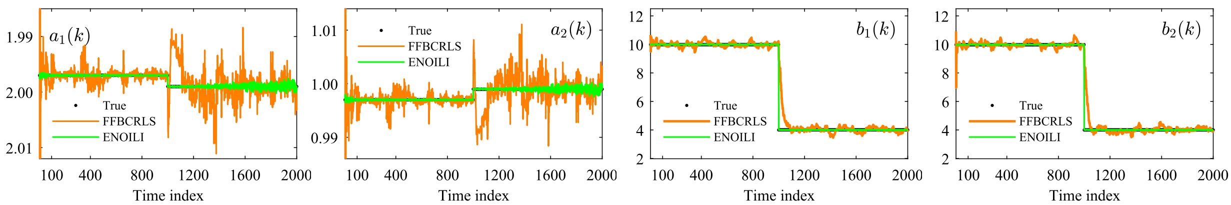
$$B(k, z) = \begin{cases} 9.990z^{-1} + 9.980z^{-2}, & 0 \leqslant k < 1000 \\ 3.998z^{-1} + 3.997z^{-2}, & 1000 < k \leqslant 2000 \end{cases} \tag{60}$$

where $k = 0, 1, 2, \ldots, N$ and $N = 2000$.



(a) Estimates of $a_1(k)$ (b) Estimates of $a_2(k)$ (c) Estimates of $b_1(k)$ (d) Estimates of $b_2(k)$

FIGURE 5. Estimates by FFBCRLS and ENOILI respectively with λ = 0.99.



(a) Estimates of $a_1(k)$ (b) Estimates of $a_2(k)$ (c) Estimates of $b_1(k)$ (d) Estimates of $b_2(k)$

FIGURE 6. Estimates by FFBCRLS and ENOILI respectively with λ = 0.96.

Identification of the parameters in (59) and (60) is performed by the proposed ENOILI algorithm and the FFBCRLS algorithm in [19] respectively. The input signal $u_j(k)$ is taken as i.i.d. persistent-excitation sequence distributed uniformly in $[-1\ 1]$. The output signal $y_j(k)$ is contaminated by the observation noise $v_j(k) \in \mathcal{N}(0, \sigma^2)$ with $\sigma^2 = 0.3^2$.

To describe the estimation accuracy, the parameter estimation error is defined as

$$\delta_j = \frac{1}{N}\sum_{k=0}^{N}\frac{||\widetilde{\boldsymbol{\theta}}_j(k) - \boldsymbol{\theta}(k)||}{||\boldsymbol{\theta}(k)||}. \tag{61}$$

To show the relationship between tracking speed and estimation accuracy, the FFBCRLS algorithm is performed under different forgetting factors $\lambda = 0.99$ and $\lambda = 0.96$. The maximum iteration index for the ENOILI algorithm is set to be 200. Initialize the ENOILI algorithm according to Section V-S1) where the weighting coefficients w_1 and w_2 are both set to be 1, and Ξ is set to be $10^{-6}I$. From the estimates shown in Fig. 5 and Fig. 6, and the estimation error shown in Fig. 7, the following observations can be obtained.

O1) Tracking speed and estimation accuracy conflict with each other in regarding to the choice of the forgetting factor for the FFBCRLS algorithm. As shown in Fig. 5, a large forgetting factor close to one mitigates the estimates fluctuation and improves the estimation accuracy when the parameters are invariant, but leads to a conspicuous estimation lag when the parameters vary drastically. On the other hand, a small forgetting factor decreases the estimation lag and thus improves the tracking speed as shown in Fig. 6, whereas the estimates fluctuate sharply due to noise.

O2) Fast tracking speed and high estimation accuracy are achieved simultaneously by the proposed ENOILI algorithm. On one hand, as shown in Fig. 5 and Fig. 6, the estimates by the ENOILI algorithm track the parameters timely when drastic variation happens. Tracking lag does not exist since the ENOILI algorithm performs identification in the iteration domain rather than the time domain as FFBCRLS does. On the other hand, estimates fluctuation is reduced remarkably by the compensation of the noise-induced estimation error, and hence estimation accuracy is improved significantly. (***F1** is achieved.*)

O3) It can be observed from Fig. 5 that the time span is up to $k = 2000$, which indicates there should be 2000 iterative estimation algorithms running in a parallel manner. Actually, the fast tracking speed and high estimation accuracy achieved by the proposed ENOILI algorithm are at the cost of calculations. Therefore, the proposed algorithm is more suitable for systems where the plant operates over a short interval, or where heavy calculation is allowed.

Remark 6: For a practical industrial system, if the estimation result of the time-varying parameter is not used for real-time control or compensation, we can store the I/O data when the plant runs and then estimate the parameters off line on an industrial computer whose calculating capacity is usually very powerful. Therefore, although thousands of estimation algorithms are required in each iteration, the heavy calculation cannot limit the practical application of the proposed approach due to its off-line feature.

B. EXAMPLE 2: COMPARISON RESULTS WITH EXISTING ITERATIVE IDENTIFICATION ALGORITHMS

To further show the enhanced numerical stability and estimation accuracy, here we make a comparison between the proposed ENOILI algorithm and two existing iterative

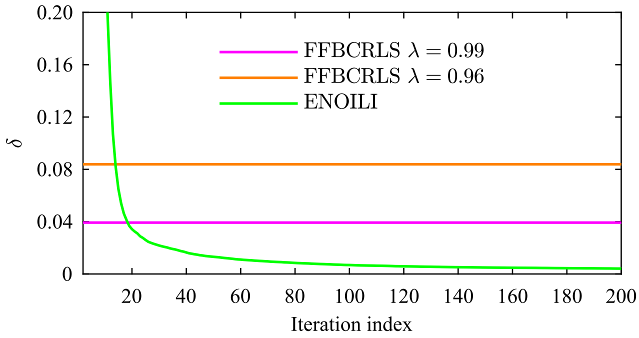
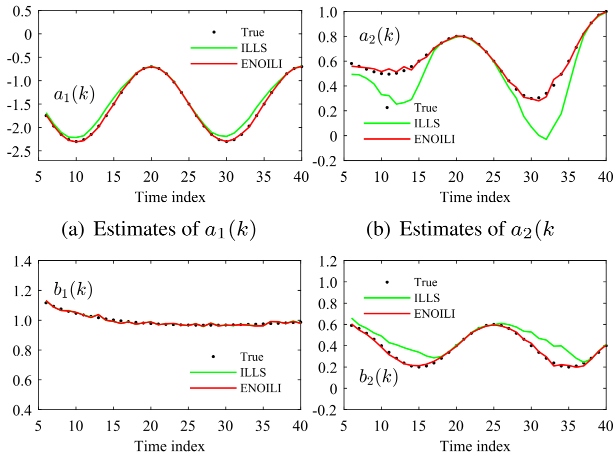


FIGURE 7. Estimation error by FFBCRLS and ENOILI respectively.



(a) Estimates of $a_1(k)$ (b) Estimates of $a_2(k$ (c) Estimates of $b_1(k)$ (d) Estimates of $b_2(k$

FIGURE 8. Estimates by ILLS and ENOILI respectively.

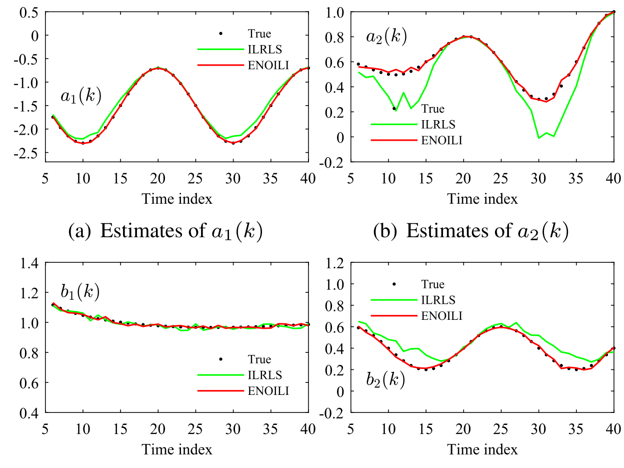
algorithms which are the ILLS algorithm proposed in [28] and the ILRLS algorithm proposed in [29]. ILLS and ILRLS are very similar to ENOILI except the bias compensation term and the SVD-based update law for $P_j(k)$. The LTV system in (1) with the following time-varying parameters is considered

$$\begin{cases} a_1(k) = -1.5 + 0.8\cos(0.1\pi k) \\ a_2(k) = 0.6 + 0.01k\cos(0.1\pi k) \\ b_1(k) = 1 + \dfrac{\cos(2\pi k/61)}{k+1} \\ b_2(k) = 0.4 + 0.2\sin(0.1\pi k) \end{cases} \tag{62}$$

where $k = 0, 1, 2, \ldots, N$ and $N = 40$.

The input signal $u_j(k)$ is taken as same as that in Example 1. The observation noise is taken as $v_j(k) \in \mathcal{N}(0, \sigma^2)$ with $\sigma^2 = 0.04^2$. The maximum iteration index is set to be 500. Initialize ENOILI according to Section V-S1) where the weighting coefficients w_1 and w_2 are both set to be 1, and Ξ is set to be $10^{-6}I$. Initialize $P_j(k)$ in ILLS and ILRLS both to be $10^6 I$. The estimates and estimation error by ILLS, ILRLS and ENOILI are shown in Fig. 8–Fig. 10, from which the following observations are obtained.

O4) Higher estimation accuracy is achieved by the proposed ENOILI algorithm than the existing iterative identification algorithms. Large estimation bias exists for ILLS and ILRLS algorithms at some time indexes, as shown in Fig. 8 and Fig. 9. The bias mainly results from the observation noise. By designing bias compensation term, the proposed ENOILI algorithm obtains more accurate estimates than ILLS and ILRLS algorithms. As shown in Fig. 10, the noise-induced estimation error is compensated significantly as the iteration index increases. (***F1** is achieved.*)



(a) Estimates of $a_1(k)$ (b) Estimates of $a_2(k)$ (c) Estimates of $b_1(k)$ (d) Estimates of $b_2(k)$

FIGURE 9. Estimates by ILRLS and ENOILI respectively.

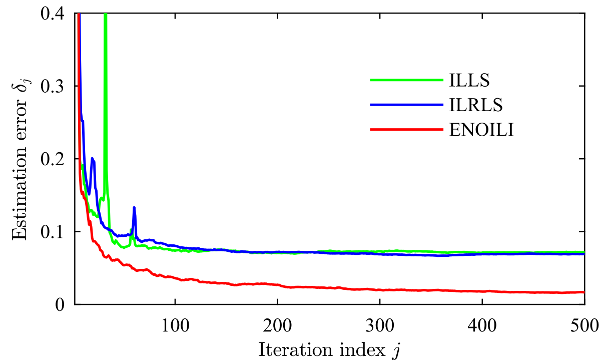


FIGURE 10. Estimation error by ILLS, ILRLS and ENOILI respectively.

O5) Higher numerical stability is achieved by the proposed ENOILI algorithm than the existing iterative identification algorithms. As shown in Fig. 10, the estimation errors by ILLS and ILRLS algorithms increase sharply during some iterations which is undesirable in practice. This numerical instability mainly results from the non-positive definite $P_j(k)$ as discussed in Section III-B. The ENOILI algorithm uses SVD operation instead of the subtraction operation in ILLS and ILRLS algorithms when updating $P_j(k)$, thereby improving numerical stability significantly. (***F2** is achieved.*)

Fig. 11 shows the estimation error in 500th iteration under different noise variance, from which it is indicated that the noise deteriorates the estimation accuracy while the proposed ENOILI algorithm compensates the noise-induced bias

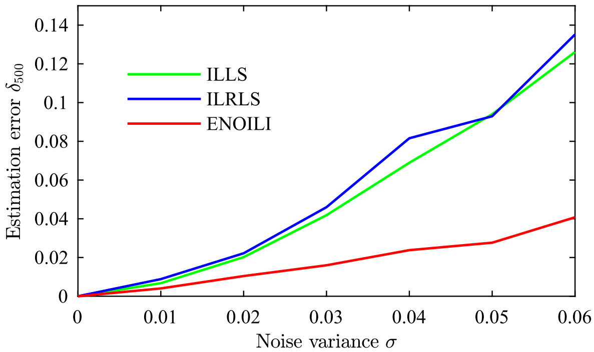


FIGURE 11. **Estimation error under different noise variance.**

and achieves accuracy enhancement compared with existing algorithms. This conclusion is in accordance with the observation **O**4).

VII. CONCLUSION

In this paper, an enhanced normal-optimal iterative identification algorithm for repetitive time-varying systems has been proposed. The main motivations are that 1) the conventional recursive methods have low tracking speed; 2) the presence of observation noise always leads to estimation bias; and 3) there exists inherently numerical instability in the existing RLS-like algorithms. To address these problems, an iteration axis is introduced into the LTV systems. Identification method that performs recursion in the iteration domain is proposed such that the estimation lag is eliminated. To reduce the estimation bias induced by noise, a bias compensation term is developed to improve the estimation accuracy. The convergence and numerical stability of the proposed algorithm is further improved by a SVD-based update law for the covariance matrix. Future work includes the possible application of the proposed method to a real-world identification problem.

ACKNOWLEDGMENT

The authors would like to thank Prof. E. Bai from University of Iowa for his contributions on the improvement of this paper.

REFERENCES

[1] I. Uyanik, U. Saranli, M. M. Ankarali, N. J. Cowan, and O. Morgul, "Frequency-domain subspace identification of linear time-periodic (LTP) systems," *IEEE Trans. Autom. Control.*, vol. 64, no. 6, pp. 2529–2536, Jun. 2019.

[2] M. Babakmehr, F. Harirchi, A. Al-Durra, S. M. Muyeen, and M. G. Simoes, "Compressive system identification for multiple line outage detection in smart grids," *IEEE Trans. Ind. Appl.*, vol. 55, no. 5, pp. 4462–4473, Sep. 2019.

[3] Y. Li, W.-G. Cui, Y.-Z. Guo, T. Huang, X.-F. Yang, and H.-L. Wei, "Time-varying system identification using an ultra-orthogonal forward regression and multiwavelet basis functions with applications to EEG," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 2960–2972, Jul. 2018.

[4] M. Kazemi and M. M. Arefi, "A fast iterative recursive least squares algorithm for Wiener model identification of highly nonlinear systems," *ISA Trans.*, vol. 67, pp. 382–388, Mar. 2017.

[5] Y. Zhao, A. Fatehi, and B. Huang, "A data-driven hybrid ARX and Markov chain modeling approach to process identification with time-varying time delays," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4226–4236, May 2017.

[6] M. Ronde, J. Van Den Bulk, R. Van De Molengraft, and M. Steinbuch, "Feedforward for flexible systems with time-varying performance locations," in *Proc. Amer. Control Conf.*, Jun. 2013, pp. 6033–6038.

[7] R. Zou and K. Chon, "Robust algorithm for estimation of time-varying transfer functions," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 2, pp. 219–228, Feb. 2004.

[8] N. Liu and A. Alleyne, "Iterative learning identification for linear time-varying systems," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 1, pp. 310–317, Jan. 2016.

[9] X. Shan and J. B. Burl, "Continuous wavelet based linear time-varying system identification," *Signal Process.*, vol. 91, no. 6, pp. 1476–1488, Jun. 2011.

[10] H. Rios, D. Efimov, J. A. Moreno, W. Perruquetti, and J. G. Rueda-Escobedo, "Time-varying parameter identification algorithms: Finite and fixed-time convergence," *IEEE Trans. Autom. Control.*, vol. 62, no. 7, pp. 3671–3678, Jul. 2017.

[11] J. Lataire, R. Pintelon, D. Piga, and R. Tóth, "Continuous-time linear time-varying system identification with a frequency-domain kernel-based estimator," *IET Control Theory Appl.*, vol. 11, no. 4, pp. 457–465, Feb. 2017.

[12] T. Pu and J. Bai, "Robust identification of discrete-time linear systems with unknown time-varying disturbance," *Digit. Signal Process.*, vol. 83, pp. 271–279, Dec. 2018.

[13] F. Ding, Y. Shi, and T. Chen, "Performance analysis of estimation algorithms of nonstationary ARMA processes," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 1041–1053, Mar. 2006.

[14] J. Li, Y. Zheng, and Z. Lin, "Recursive identification of time-varying systems: Self-tuning and matrix RLS algorithms," *Syst. Control Lett.*, vol. 66, pp. 104–110, Apr. 2014.

[15] J. M. Bravo, A. Suarez, M. Vasallo, and T. Alamo, "Slide window bounded-error time-varying systems identification," *IEEE Trans. Autom. Control.*, vol. 61, no. 8, pp. 2282–2287, Aug. 2016.

[16] Z. Shi, Y. Wang, and Z. Ji, "Bias compensation based partially coupled recursive least squares identification algorithm with forgetting factors for MIMO systems: Application to PMSMs," *J. Franklin Inst.*, vol. 353, no. 13, pp. 3057–3077, Sep. 2016.

[17] S. R. Seydnejad, "Real-time identification of time-varying ARMAX systems based on recursive update of its parameters," *J. Dyn. Syst., Meas., Control*, vol. 136, no. 3, pp. 031017-1–031017-10, 2014.

[18] K. Nishiyama, "A new approach to introducing a forgetting factor into the normalized least mean squares algorithm," *Signal Process.*, vol. 114, pp. 19–23, Sep. 2015.

[19] A. Wu, S. Chen, and D. Jia, "Bias-compensation-based least-squares estimation with a forgetting factor for output error models with white noise," *Int. J. Syst. Sci.*, vol. 47, no. 7, pp. 1700–1709, May 2016.

[20] S.-H. Leung and C. So, "Gradient-based variable forgetting factor RLS algorithm in time-varying environments," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 3141–3150, Aug. 2005.

[21] B. Kovačević, Z. Banjac, and I. K. Kovačević, "Robust adaptive filtering using recursive weighted least squares with combined scale and variable forgetting factors," *EURASIP J. Adv. Signal Process.*, vol. 2016, no. 1, pp. 1–22, 2016.

[22] T. Meng and W. He, "Iterative learning control of a robotic arm experiment platform with input constraint," *IEEE Trans. Ind. Electron.*, vol. 65, no. 1, pp. 664–672, Jan. 2018.

[23] F. Song, Y. Liu, J.-X. Xu, X. Yang, P. He, and Z. Yang, "Iterative learning identification and compensation of space-periodic disturbance in PMLSM systems with time delay," *IEEE Trans. Ind. Electron.*, vol. 65, no. 9, pp. 7579–7589, Sep. 2018.

[24] T. Sugie and F. Sakai, "Noise tolerant iterative learning control for identification of continuous-time systems," in *Proc. 44th IEEE Conf. Decis. Control*, Oct. 2006, pp. 4251–4256.

[25] M. C. Campi, T. Sugie, and F. Sakai, "An iterative identification method for linear continuous-time systems," *IEEE Trans. Autom. Control.*, vol. 53, no. 7, pp. 1661–1669, Aug. 2008.

[26] T.-H. Kim and T. Sugie, "An iterative learning control based identification for a class of MIMO continuous-time systems in the presence of fixed input disturbances and measurement noises," *Int. J. Syst. Sci.*, vol. 38, no. 9, pp. 737–748, Sep. 2007.

[27] N. Liu and A. G. Alleyne, "Iterative learning identification for an automated off-highway vehicle," in *Proc. Amer. Control Conf.*, Jun. 2011, pp. 4299–4304.

[28] M.-X. Sun and H.-B. Bi, "Learning identification: Least squares algorithms and their repetitive consistency," *Acta Autom. Sinica*, vol. 38, no. 5, pp. 698–706, Dec. 2012.

[29] N. Lin, R. Chi, and B. Huang, "Linear time-varying data model-based iterative learning recursive least squares identifications for repetitive systems," *IEEE Access*, vol. 7, pp. 133304–133313, 2019.
[30] S. C. Chan and Y. J. Chu, "A new state-regularized QRRLS algorithm with a variable forgetting factor," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 3, pp. 183–187, Mar. 2012.
[31] P. Janssens, G. Pipeleers, and J. Swevers, "A data-driven constrained norm-optimal iterative learning control framework for LTI systems," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 2, pp. 546–551, Mar. 2013.
[32] F. Ding, T. Chen, and L. Qiu, "Bias compensation based recursive least-squares identification algorithm for MISO systems," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 5, pp. 349–353, May 2006.
[33] P. Carbone and G. Vandersteen, "Bias compensation when identifying static nonlinear functions using averaged measurements," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 7, pp. 1855–1862, Jul. 2014.
[34] B. G. Sluijk, T. Castenmiller, R. D. C. De Jongh, H. Jasper, T. Modderman, L. Levasier, E. Loopstra, G. Savenije, M. Boonman, and H. Cox, "Performance results of a new generation of 300-mm lithography systems," *Proc. SPIE*, vol. 4346, Sep. 2001, pp. 544–557.
[35] H. Butler, "Position control in lithographic equipment [applications of control]," *IEEE Control Syst. Mag.*, vol. 31, no. 5, pp. 28–47, Oct. 2011.

FAZHI SONG (Member, IEEE) received the B.S. degree in automation from the Henan University of Science and Technology, Luoyang, China, in 2010, the M.S. degree in control science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2013, and the Ph.D. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2019. From March 2017 to March 2018, he was a joint-training Ph.D. Student with the National University of Singapore, Singapore. He is currently a Lecturer with the Harbin Institute of Technology. His research interests include system identification and motion control with applications in high-precision mechanical servo systems.

YANG LIU (Member, IEEE) received the B.S. degree in automation and the M.S. and Ph.D. degrees in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2004, 2006, and 2011, respectively. From 2014 to 2015, he was a Research Fellow with Queen's University Belfast. He is currently an Associate Professor with the Harbin Institute of Technology. His current research interests include system identification and precision motion control.

XIANLI WANG received the B.S. degree in electrical engineering and automation from the Zhengzhou University of Light Industry, Zhengzhou, China, in 2009, and the M.S. degree in agricultural electrification and automation from Northeast Agricultural University, Harbin, China, in 2013. He is currently a Lecturer with Luoyang Normal University. His research interests include modeling and optimization methods and their applications on smart grids.

WEN JIN received the B.S. degree in food science and technology from Northeast Agricultural University, Harbin, China, in 2010, the M.S. degree in food science and the Ph.D. degree in chemical engineering and technology from the Harbin Institute of Technology, Harbin, in 2012 and 2017, respectively. From 2014 to 2015, she was a joint-training Ph.D. Student with the University of California at San Diego, San Diego, California, USA. She is currently a Lecturer with Harbin Medical University. Her research interests include parameter estimation in chemical processes and preparation of nanocomposites.

LI LI received the B.S. degree in automation from the School of Automation, Harbin Engineering University, Harbin, China, in 2013, and the M.S. degree in control science and engineering from the School of Astronautics, Harbin Institute of Technology, in 2015, where he is currently pursuing the Ph.D. degree in instrument science and technology. His current research interests include precision motion control, iterative learning control, sliding mode control, and intelligent control.