*ICIC Express Letters*
Volume **16**, Number **8**, August **2022**
ICIC International ©2022 ISSN 1881-803X
pp. 859–867

# RECURSIVE IDENTIFICATION OF TIME-VARYING SYSTEMS WITH RAPID CHANGING

Lianming Sun[1,*], Xinyu Liu[2] and Junbiao Zeng[2]

[1]Department of Information Systems Engineering
Faculty of Environmental Engineering
[2]Graduate School of Environmental Engineering
The University of Kitakyushu
1-1 Hibikino, Wakamatsu-ku, Kitakyushu 808-0135, Japan
{ z8dcb001; c1mca401 }@eng.kitakyu-u.ac.jp
[*]Corresponding author: sun@kitakyu-u.ac.jp

Received December 2021; accepted February 2022

Abstract. *When the system characteristics vary with time, the time-varying models are required to be promptly identified from the observation data. In the recursive algorithms based on the parameter approximation of cosine series (RCS), the time-varying parameters are approximated by cosine series (CS), and the model parameter estimation becomes estimation of CS coefficients. However, Gibbs effect, which occurs at the discontinuous points, causes parameter fluctuation in CS approximation, and deteriorates the estimation accuracy of RCS. In order to improve the identification performance, a novel approach to reducing the influence of Gibbs effect is investigated in this paper. Detection of abrupt variation points through a soft threshold implemented by neural network is introduced into the recursive identification algorithm, the CS approximation is compensated at the detected abrupt variation points to reduce the estimate fluctuation, and then the compensated approximation leads to better identification performance for the time-varying systems with rapid changing. The implementation and effectiveness of the proposed algorithm are demonstrated by numerical simulation examples.*
**Keywords:** Cosine series, Gibbs effect, Time-varying system, Recursive identification

## 1. Introduction.

Due to aging of components, variation of environment, system fault or operation malfunction, the dynamic characteristics of a physical system often vary with time. Such examples can be found in industrial processes whose manipulation depends on the operating region [1, 2], the rapidly fading communication channels in remote sensing or mobile communication systems [3], the robotic manipulator and car steering in automatic driving [4], network with time-varying delays [5], system monitoring [6, 7], fault detection and diagnosis [8, 9], etc. When the influence of dynamics variation on the operating performance cannot be ignored, the models of dynamic characteristics must be time-varying and should be identified in real time through adaptive algorithms.

Several categories of identification methods for time-varying systems have been developed. 1) The linear parameter varying (LPV) model detects the variation information from some special measurable variables that determine the process characteristics [10]. 2) The segmentation methods separate a time-varying model into several local models by segmenting the observation data with respect to the large variation [11, 12]. 3) Some adaptive algorithms such as the recursive least squares (RLS) with a forgetting factor [13], the least mean square (LMS) or the normalized least mean square (NLMS) algorithms [14, 15], the affine projection (AP) algorithm and block orthogonal projection (BOP) [16], Kalman filter, wavelets are used to track the varying dynamics [17]. 4) The explicit approximation of the parameter variation is used for identification through orthogonal basis

DOI: 10.24507/icicel.16.08.859

series such as the trigonometric or Legendre functions [18, 19, 20]. It helps to approximate the dynamics at an arbitrary rate if the series have sufficiently high degree; consequently, the approximation commonly has better tracking performance than the conventional adaptive algorithms or the segmentation approaches for the rapid time-varying systems [21].

In practical applications, too high degree cannot be chosen due to the influence of noise and computational complexity. On the other hand, in the low degree of approximation, Gibbs effects occur at the discontinuous or abrupt variation points. They yield parameter fluctuation in the series based approximation and then deteriorate the identification performance. Cosine series (CS) based approximation is considered in the recursive identification (RCS) to remove the discontinuity at the data window edges [21] and a forgetting weight is imposed on the system output in order to mitigate the variation in the past virtual parameters [22]. However, at the discontinuous points or abrupt changing points inside the data window, fluctuation caused by Gibbs effect still exists in parameter approximation of rapid varying system; therefore, it should be detected and compensated in order to guarantee the identification performance for time-varying systems.

Thresholds are often used to distinguish the abrupt variation. Due to the complicated variation situations in the practical systems, however, the preset hard thresholds cannot deal with abrupt variation well, especially several varying parameters mutually influence each other. Consequently, a soft threshold determined by neural network is investigated and its compensation is applied in the new algorithm. By contrast with the existing methods, the proposed algorithm detects the abrupt variation under various variations, and mitigates the fluctuation in CS parameter approximation, so it has better track performance and is more effective for rapid time-varying systems.

The rest of the paper is organized as follows. In the next section, the main preliminaries for parameter approximation based on CS and Gibbs effect are reviewed. In Section 3, the recursive algorithm of RCS is discussed. Then the neural network based approach to detecting the abrupt variation, and the compensation approach to reducing the Gibbs effect are investigated in Section 4. Section 5 illustrates the numerical example. Finally, the conclusion and the future research work are given in Section 6.

## 2. Preliminaries.

### 2.1. Problem statement of time-varying identification.

Consider a linear time-varying system that can be described by the following model

$$y(k) = h_0^k u(k) + h_1^k u(k-1) + \cdots + h_n^k u(k-n) + e(k), \tag{1}$$

where $n$ is the model order, while $u(k)$, $y(k)$ and $e(k)$ are the input, output and noise at a discrete instant $k$, respectively. The superscript and subscript of the parameter $h_i^k$ indicate the parameter with lag $i$ at instant $k$. It indicates that the model parameters $h_i^k$ vary with time $k$, and should be estimated promptly from the data of $u(k)$, $y(k)$.

### 2.2. CS based approximation.

Assume that the current instant is $k$, the sliding data window is defined as $[k-K, k]$ with window length $K$. If the parameters $h_i^k$ within the data window are virtually expanded into an even periodic function with respect to $[k-2K, k-K]$ and $[k-K, k]$, then the parameters of $h_i^{k-K}, \ldots, h_i^k$ can be approximated by the cosine series with period $2K$ as follows.

$$h_i^{k_0+k_1} \approx \sum_{m=0}^{M} c_{i,m}^k \cos(m\omega k_1), \ \omega = \frac{\pi}{K}, \tag{2}$$

where $k_0 = k-K$, $0 \le k_1 \le K$, $M$ is the degree of the series, and the virtual parameters for $-K \le k_1 < 0$ can also be calculated from (2). In the data window $[k-K, k]$, the coefficients $c_{i,m}^k$ can be treated as constants. Correspondingly, the identification problem of $h_i^k$, $i = 0, 1, \ldots, M$, $k = 0, 1, \ldots$ in (1) becomes estimation of the constant coefficients within the sliding data window. Furthermore, in the virtual expansion of even periodic functions, the virtual parameter $h_i^{k_0+k_1}$ for $k_1 = -2K$ equals $h_i^k$ for $k - k_0 + 2K$ in CS approximation, and the Gibbs effect can be reduced at the window edges.

In order to reduce the series degree $M$, a forgetting factor $\lambda$, $0 < \lambda < 1$ is introduced into the system output $y(k-K), \ldots, y(k)$ [22], and then the weighted output at $k-k_0+k_1$

$$\begin{aligned}\lambda^{K-k_1} y(k_0+k_1) &= \left(\lambda^{K-k_1} h_0^{k_0+k_1}\right) u(k_0+k_1) + \left(\lambda^{K-k_1} h_1^{k_0+k_1}\right) u(k_0+k_1-1) + \cdots \\ &\quad + \left(\lambda^{K-k_1} h_n^{k_0+k_1}\right) u(k_0+k_1-n) + \lambda^{K-k_1} e(k_0+k_1) \\ &= \overline{h}_0^{k_0+k_1} u(k_0+k_1) + \overline{h}_1^{k_0+k_1} u(k_0+k_1-1) + \cdots \\ &\quad + \overline{h}_n^{k_0+k_1} u(k_0+k_1-n) + \overline{e}(k_0+k_1) \end{aligned} \tag{3}$$

holds for $0 \le k_1 \le K$. (3) can be regarded as imposing the forgetting factor on the model parameters, hence the rapid variation at past instants is mitigated virtually, and then $\overline{h}_i^{k_0+k_1}$ is approximated better than the original parameters $h_i^{k_0+k_1}$ with low series degree $M$.

2.3. **Gibbs effect in CS approximation.** The trigonometric basis based approximation suffers from Gibbs effect at abrupt variation or discontinuous points if the series degree $M$ is not high enough. An example is shown in Figure 1, where the time-varying parameter $h_i^k$ has an abrupt jump at $k = 700$. Though the CS approximation fits the true parameter well in the smooth region, it has fluctuation at the discontinuous point, and yields about 10% approximation error even for $M = 50$. The approximation error leads to much larger prediction errors and then results in large bias in parameter estimation. In order to improve the identification performance, it is important to promptly detect the rapid changing points and further to compensate the fluctuation in CS approximation.

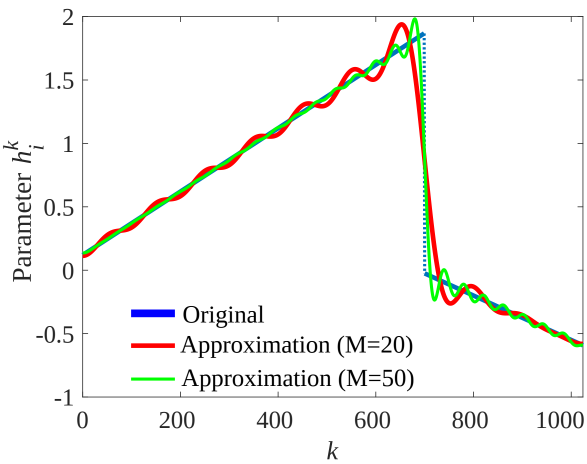



Figure 1. Illustration of Gibbs effect at discontinuous point

A threshold is often applied to detecting the abrupt variations. Due to the various types of unpredictable variations, the preset hard threshold based on the amplitude of variation only has misjudgments. To be adaptive to the various changing characteristics, a soft threshold function using the prediction errors and estimated parameters is considered.

3. **Recursive Identification.** Commonly it is expected to identify the time-varying models recursively. The data vectors and matrices are defined first in the algorithm.

3.1. **Definition of data matrices and vectors.** Let the regression vectors and trigonometric function matrices be defined as follows:

$$\begin{aligned}
&\boldsymbol{\phi}_c^k(k_1) = \begin{bmatrix} \boldsymbol{\phi}_{c,0}^k(k_1) \\ \boldsymbol{\phi}_{c,1}^k(k_1) \\ \vdots \\ \boldsymbol{\phi}_{c,M}^k(k_1) \end{bmatrix}, \ \boldsymbol{\phi}_s^k(k_1) = \begin{bmatrix} \boldsymbol{\phi}_{s,1}^k(k_1) \\ \vdots \\ \boldsymbol{\phi}_{s,M}^k(k_1) \end{bmatrix}, \\
&\boldsymbol{\phi}_{c,0}^k(k_1) = [u(k_0+k_1), \ \ldots, \ u(k_0+k_1-n)]^T, \\
&\boldsymbol{\phi}_{c,m}^k(k_1) = \boldsymbol{W}_{c,m}^{k_1}\boldsymbol{\phi}_{c,0}^k(k_1), \ \boldsymbol{\phi}_{s,m}^k(k_1) = \boldsymbol{W}_{s,m}^{k_1}\boldsymbol{\phi}_{c,0}^k(k_1), \\
&\boldsymbol{W}_{c,m}^{k_1} = \cos(m\omega k_1)\boldsymbol{I}, \ \boldsymbol{W}_{s,m}^{k_1} = \sin(m\omega k_1)\boldsymbol{I},
\end{aligned} \tag{4}$$

where $\boldsymbol{I}$ is an identity matrix with the appropriate dimension, $\boldsymbol{W}_{c,m}^{k_1}$ and $\boldsymbol{W}_{s,m}^{k_1}$ are the diagonal matrices of $\cos(m\omega k_1)$, $\sin(m\omega k_1)$, respectively, while the parameter vectors are

$$\boldsymbol{\theta}_{c,m}^k = \left[c_{0,m}^k, \ c_{1,m}^k, \ \ldots, \ c_{n,m}^k\right]^T, \ \ \boldsymbol{\theta}_c^k = \left[\left(\boldsymbol{\theta}_{c,0}^k\right)^T, \ \left(\boldsymbol{\theta}_{c,1}^k\right)^T, \ \ldots, \ \left(\boldsymbol{\theta}_{c,M}^k\right)^T\right]^T, \tag{5}$$

and then the model in (3) can be approximated as a compact formula

$$\lambda^{K-k_1}y(k_0+k_1) = \left(\boldsymbol{\phi}_c^k(k_1)\right)^T\boldsymbol{\theta}_c^k + \bar{e}(k_0+k_1). \tag{6}$$

Furthermore, define the correlation vector, correlation matrix and its inverse as

$$\boldsymbol{\phi}_{cy}^k = \sum_{k_1=0}^{K}\lambda^{K-k_1}\boldsymbol{\phi}_c^k(k_1)y^k(k_1), \ \ \boldsymbol{\Phi}_{cc}^k = \sum_{k_1=0}^{K}\boldsymbol{\phi}_c^k(k_1)\left(\boldsymbol{\phi}_c^k(k_1)\right)^T, \ \ \boldsymbol{P}^k = \left(\boldsymbol{\Phi}_{cc}^k\right)^{-1}.$$

Then the estimation of the coefficient vector $\boldsymbol{\theta}_c^k$ of the CS approximation can be given by

$$\hat{\boldsymbol{\theta}}_c^k = \left(\boldsymbol{\Phi}_{cc}^k\right)^{-1}\boldsymbol{\phi}_{cy}^k = \boldsymbol{P}^k\boldsymbol{\phi}_{cy}^k. \tag{7}$$

3.2. **Update of data matrices and vectors.** (7) is implemented recursively in RCS algorithm. When the sliding data window shifts from $[k-K,k]$ forward to $[k+1-K,k+1]$ at the next instant $k+1$, the updated correlation matrix $\boldsymbol{\Phi}_{cc}^{k+1}$ in the new data window is

$$\begin{aligned}
\boldsymbol{\Phi}_{cc}^{k+1} &= \sum_{k_1=0}^{K}\boldsymbol{\phi}_c^{k+1}(k_1)\left(\boldsymbol{\phi}_c^{k+1}(k_1)\right)^T \\
&= \begin{bmatrix} \boldsymbol{W}_c & \boldsymbol{W}_s \end{bmatrix}\left(\begin{bmatrix} \boldsymbol{W}_c^{-1}\boldsymbol{\phi}_c^k(K+1) \\ \mathbf{0} \end{bmatrix}\begin{bmatrix} \left(\boldsymbol{\phi}_c^k(K+1)\right)^T\boldsymbol{W}_c^{-1} & \mathbf{0}^T \end{bmatrix}\right. \\
&\quad \left. - \begin{bmatrix} \boldsymbol{\phi}_c^k(0) \\ \mathbf{0} \end{bmatrix}\begin{bmatrix} \left(\boldsymbol{\phi}_c^k(0)\right)^T & \mathbf{0}^T \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Phi}_{cc}^k & \boldsymbol{\Phi}_{cs}^k \\ \boldsymbol{\Phi}_{sc}^k & \boldsymbol{\Phi}_{ss}^k \end{bmatrix}\right)\begin{bmatrix} \boldsymbol{W}_c \\ \boldsymbol{W}_s \end{bmatrix},
\end{aligned} \tag{8}$$

where $\boldsymbol{W}_c$ and $\boldsymbol{W}_s$ are the diagonal matrices with the diagonal blocks $\boldsymbol{W}_{c,m}^1$, $\boldsymbol{W}_{s,m}^1$ for $m = 0, 1, \ldots, M$, respectively.

From (8), it is seen that it is difficult to update the inverse of $\boldsymbol{\Phi}_{cc}^{k+1}$ due to extra terms such as $\boldsymbol{\Phi}_{cs}^k$, $\boldsymbol{\Phi}_{sc}^k$, $\boldsymbol{\Phi}_{ss}^k$ and $\boldsymbol{\phi}_c^k(0)$. Then, the updating should be simplified in the recursion to guarantee the recursive computability. Let the matrices $\boldsymbol{\Phi}_1$, $\boldsymbol{\Phi}_2$ be denoted as

$$\begin{aligned}
&\boldsymbol{\Phi}_1 = \boldsymbol{W}_c^{-1}\boldsymbol{\phi}_c^k(K+1)\left(\boldsymbol{\phi}_c^k(K+1)\right)^T\boldsymbol{W}_c^{-1} - \boldsymbol{\phi}_c^k(0)\left(\boldsymbol{\phi}_c^k(0)\right)^T + \boldsymbol{\Phi}_{cc}^k = \boldsymbol{\Psi}^{k+1} + \boldsymbol{\Phi}_{cc}^k, \\
&\boldsymbol{\Phi}_2 = \boldsymbol{W}_{cs}^{-1}\boldsymbol{\Phi}_{sc}^k + \boldsymbol{\Phi}_{cs}^k\boldsymbol{W}_{cs}^{-1} + \boldsymbol{W}_{cs}^{-1}\boldsymbol{\Phi}_{ss}^k\boldsymbol{W}_{cs}^{-1}, \\
&\boldsymbol{\psi}^{k+1} = \begin{bmatrix} \boldsymbol{W}_c^{-1}\boldsymbol{\phi}_c^k(K+1), & \boldsymbol{\phi}_c^k(0) \end{bmatrix}, \ \bar{\boldsymbol{\psi}}^{k+1} = \begin{bmatrix} \boldsymbol{W}_c^{-1}\boldsymbol{\phi}_c^k(K+1), & -\boldsymbol{\phi}_c^k(0) \end{bmatrix}, \\
&\bar{\boldsymbol{y}}^{k+1} = \begin{bmatrix} y^k(K+1), & -\lambda^{K+1}y^k(0) \end{bmatrix}^T,
\end{aligned} \tag{9}$$

where $\boldsymbol{W}_{cs}^{-1} = \boldsymbol{W}_c^{-1}\boldsymbol{W}_s$, then $\boldsymbol{\Phi}_{cc}^{k+1}$ in (8) can be expressed by $\boldsymbol{\Phi}_{cc}^{k+1} = \boldsymbol{W}_c\big(\boldsymbol{\Phi}_1 + \boldsymbol{\Phi}_2\big)\boldsymbol{W}_c$. It implies that the extra matrices in $\boldsymbol{\Phi}_2$ make the inverse of $\boldsymbol{\Phi}_{cc}^{k+1}$ be very complicated. On the other hand, generally $\|\boldsymbol{\Phi}_1\| \gg \|\boldsymbol{\Phi}_2\|$ holds when the data window length $K$ is much larger than the series degree $M$, and the entries of $\boldsymbol{\Phi}_1^{-1}\boldsymbol{\Phi}_2$ are much smaller than 1. Consequently, the following inversion can be approximated by

$$\left(\boldsymbol{I}+\boldsymbol{\Phi}_1^{-1}\boldsymbol{\Phi}_2\right)^{-1}=\boldsymbol{I}-\boldsymbol{\Phi}_1^{-1}\boldsymbol{\Phi}_2+\left(\boldsymbol{\Phi}_1^{-1}\boldsymbol{\Phi}_2\right)^2-\cdots\approx\boldsymbol{I}-\boldsymbol{\Phi}_1^{-1}\boldsymbol{\Phi}_2, \tag{10}$$

and it yields the approximation of inverse of $\boldsymbol{P}^{k+1}$, i.e., the inverse of $\boldsymbol{\Phi}_{cc}^{k+1}$ as follows

$$\boldsymbol{P}^{k+1}=\left(\boldsymbol{\Phi}_{cc}^{k+1}\right)^{-1}=\boldsymbol{W}_c^{-1}(\boldsymbol{\Phi}_1+\boldsymbol{\Phi}_2)^{-1}\boldsymbol{W}_c^{-1}\approx\boldsymbol{W}_c^{-1}\left(\boldsymbol{I}-\boldsymbol{\Phi}_1^{-1}\boldsymbol{\Phi}_2\right)\boldsymbol{\Phi}_1^{-1}\boldsymbol{W}_c^{-1}, \tag{11}$$

where $\boldsymbol{\Phi}_1^{-1}$ can be updated following matrix inversion lemma [13]

$$\boldsymbol{\Phi}_1^{-1}=\left(\boldsymbol{I}-\boldsymbol{g}^{k+1}\left(\bar{\boldsymbol{\psi}}^{k+1}\right)^T\right)\boldsymbol{P}^k, \tag{12}$$

whereas $\boldsymbol{g}^{k+1}$ is a gain vector given by

$$\boldsymbol{g}^{k+1}=\boldsymbol{P}^k\boldsymbol{\psi}^{k+1}\left(\boldsymbol{I}_2+\left(\bar{\boldsymbol{\psi}}^{k+1}\right)^T\boldsymbol{P}^k\boldsymbol{\psi}^{k+1}\right)^{-1}. \tag{13}$$

Notice that $\boldsymbol{I}_2+\left(\bar{\boldsymbol{\psi}}^{k+1}\right)^T\boldsymbol{P}^k\boldsymbol{\psi}^{k+1}$ is a $2\times 2$ matrix, so it is easy to calculate $\boldsymbol{g}^{k+1}$.

Similarly as $\boldsymbol{\Phi}_{cc}^{k+1}$, the correlation vectors $\boldsymbol{\phi}_{cy}^{k+1}$ and $\boldsymbol{\phi}_{sy}^{k+1}$ can be updated by

$$\begin{aligned}\boldsymbol{\phi}_{cy}^{k+1}&=\boldsymbol{W}_c\left(\boldsymbol{\psi}^{k+1}\bar{\boldsymbol{y}}^{k+1}+\lambda\boldsymbol{\phi}_{cy}^k+\boldsymbol{W}_{cs}^{-1}\lambda\boldsymbol{\phi}_{sy}^k\right),\\ \boldsymbol{\phi}_{sy}^{k+1}&=-\boldsymbol{W}_s\left(\boldsymbol{\psi}^{k+1}\bar{\boldsymbol{y}}^{k+1}+\lambda\boldsymbol{\phi}_{cy}^k\right)+\boldsymbol{W}_c\lambda\boldsymbol{\phi}_{sy}^k.\end{aligned} \tag{14}$$

From (14), the extra term $\boldsymbol{\phi}_{sy}^k$ can be expressed by the past data

$$\boldsymbol{\phi}_{sy}^k=-\boldsymbol{W}_s\left(\boldsymbol{\psi}^k\bar{\boldsymbol{y}}^k+\lambda\boldsymbol{\phi}_{cy}^{k-1}\right)+\boldsymbol{W}_c\lambda\boldsymbol{\phi}_{sy}^{k-1}. \tag{15}$$

Since the matrices $\boldsymbol{W}_c$, $\boldsymbol{W}_s$ and $\boldsymbol{W}_{cs}^{-1}$ are diagonal, $\boldsymbol{\phi}_{sy}^k$ can be rewritten as

$$\boldsymbol{\phi}_{sy}^k=\boldsymbol{W}_c\left(\boldsymbol{I}+\boldsymbol{W}_{cs}^{-2}\right)\lambda\boldsymbol{\phi}_{sy}^{k-1}-\boldsymbol{W}_{cs}^{-1}\lambda\boldsymbol{\phi}_{cy}^k. \tag{16}$$

3.3. **Update of parameter estimation.** Now substitute the approximated formulae of $\boldsymbol{P}^{k+1}$ and $\boldsymbol{\phi}_{cy}^{k+1}$ to deduce the recursive estimation $\hat{\boldsymbol{\theta}}_c^{k+1}=\boldsymbol{P}^{k+1}\boldsymbol{\phi}_c^{k+1}$ in the new window $[k+1-K,k+1]$, where $\boldsymbol{\phi}_{cy}^{k+1}$ in (14) is split into two parts: $\boldsymbol{W}_c\left(\boldsymbol{\psi}^{k+1}\bar{\boldsymbol{y}}^{k+1}+\lambda\boldsymbol{\phi}_{cy}^k\right)$ and $\boldsymbol{W}_s\lambda\boldsymbol{\phi}_{sy}^k$. Multiplying $\boldsymbol{\Phi}_1^{-1}\boldsymbol{W}_c^{-1}$ by the first part of $\boldsymbol{\phi}_{cy}^{k+1}$ yields that

$$\boldsymbol{\Phi}_1^{-1}\boldsymbol{W}_c^{-1}\boldsymbol{W}_c\left(\boldsymbol{\psi}^{k+1}\bar{\boldsymbol{y}}^{k+1}+\lambda\boldsymbol{\phi}_{cy}^k\right)=\left(\boldsymbol{\psi}^{k+1}\left(\bar{\boldsymbol{\psi}}^{k+1}\right)^T+\boldsymbol{\Phi}_{cc}^k\right)^{-1}\left(\boldsymbol{\psi}^{k+1}\bar{\boldsymbol{y}}^{k+1}+\lambda\boldsymbol{\phi}_{cy}^k\right). \tag{17}$$

Similarly as the standard recursive formula in [13], (17) can be compactly rewritten as

$$\lambda\boldsymbol{P}^k\boldsymbol{\phi}_{cy}^k+\boldsymbol{g}^{k+1}\varepsilon^{k+1}=\lambda\hat{\boldsymbol{\theta}}_c^k+\boldsymbol{g}^{k+1}\varepsilon^{k+1}, \tag{18}$$

where the prediction error $\varepsilon^{k+1}$ is defined by

$$\varepsilon^{k+1}=\bar{\boldsymbol{y}}^{k+1}-\left(\bar{\boldsymbol{\psi}}^{k+1}\right)^T\lambda\hat{\boldsymbol{\theta}}_c^k. \tag{19}$$

Let $\boldsymbol{\theta}_s^k=\boldsymbol{P}^k\boldsymbol{W}_s\boldsymbol{\phi}_{sy}^{k-1}$. For the rest part of $\boldsymbol{\phi}_{cy}^{k+1}$ in (8), substituting (15) into the multiplication of $\boldsymbol{\Phi}_1^{-1}\boldsymbol{W}_c^{-1}$ and $\boldsymbol{W}_s\boldsymbol{\phi}_{sy}^k$ yields that

$$\boldsymbol{\Phi}_1^{-1}\boldsymbol{W}_c^{-1}\boldsymbol{W}_s\boldsymbol{\phi}_{sy}^k\approx\left(\boldsymbol{I}-\boldsymbol{g}^{k+1}\left(\bar{\boldsymbol{\psi}}^{k+1}\right)^T\right)\left(\left(\boldsymbol{I}+\boldsymbol{W}_{cs}^{-2}\right)\hat{\boldsymbol{\theta}}_s^k-\boldsymbol{W}_{cs}^{-2}\hat{\boldsymbol{\theta}}_c^k\right). \tag{20}$$

Furthermore, define the gain matrices to simplify the recursive formulae of $\boldsymbol{P}^{k+1}$ and $\hat{\boldsymbol{\theta}}_c^{k+1}$

$$\begin{aligned}\boldsymbol{G}^{k+1}&=\boldsymbol{W}_c^{-1}\left(\boldsymbol{I}-\left(\boldsymbol{I}-\boldsymbol{g}^{k+1}\left(\bar{\boldsymbol{\psi}}^{k+1}\right)^T\right)\boldsymbol{\Omega}^k\right),\\ \boldsymbol{G}_s^{k+1}&=\boldsymbol{G}^{k+1}\left(\boldsymbol{I}-\boldsymbol{g}^{k+1}\left(\bar{\boldsymbol{\psi}}^{k+1}\right)^T\right),\end{aligned} \tag{21}$$

where $\boldsymbol{\Omega}^k = \boldsymbol{P}^k \boldsymbol{\Phi}_2$. Then, by combining (18) with (20), the new parameter vector and the inverse of the correlation matrix can be concluded as follows:

$$\begin{aligned} \hat{\boldsymbol{\theta}}_s^{k+1} &= \lambda \boldsymbol{G}_s^{k+1} \left( \left( \boldsymbol{I} + \boldsymbol{W}_{cs}^{-2} \right) \boldsymbol{\theta}_s^k - \boldsymbol{W}_{cs}^{-2} \boldsymbol{\theta}_c^k \right), \\ \hat{\boldsymbol{\theta}}_c^{k+1} &= \boldsymbol{P}^{k+1} \boldsymbol{\phi}_{cy}^{k+1} = \boldsymbol{G}^{k+1} \left( \lambda \hat{\boldsymbol{\theta}}_c^k + \boldsymbol{g}^{k+1} \varepsilon^{k+1} \right) + \hat{\boldsymbol{\theta}}_s^{k+1}, \\ \boldsymbol{P}^{k+1} &= \boldsymbol{G}^{k+1} \boldsymbol{\Phi}_1^{-1} \boldsymbol{W}_c^{-1} = \boldsymbol{G}^{k+1} \left( \boldsymbol{I} - \boldsymbol{g}^{k+1} \left( \bar{\boldsymbol{\psi}}^k \right)^T \right) \boldsymbol{P}^k \boldsymbol{W}_c^{-1}. \end{aligned} \tag{22}$$

The estimate in (22) is composed of two parts: the first part projects the term $\lambda \hat{\boldsymbol{\theta}}_c^k + \boldsymbol{g}^{k+1} \varepsilon^{k+1}$ onto the cosine basis in the new window, while the second part $\hat{\boldsymbol{\theta}}_s^{k+1}$ corresponds to transition effect on the extra terms $\boldsymbol{\phi}_s^k$, $\boldsymbol{\phi}_{sy}^k$ appearing with the data window sliding.

Moreover, the correlation matrix and vector are updated as follows:

$$\begin{bmatrix} \boldsymbol{\Phi}_{cc}^{k+1} & \boldsymbol{\Phi}_{cs}^{k+1} \\ \boldsymbol{\Phi}_{sc}^{k+1} & \boldsymbol{\Phi}_{ss}^{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}_c & \boldsymbol{W}_s \\ -\boldsymbol{W}_s & \boldsymbol{W}_c \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_{cc}^k + \boldsymbol{\Psi}^{k+1} & \boldsymbol{\Phi}_{cs}^k \\ \boldsymbol{\Phi}_{sc}^k & \boldsymbol{\Phi}_{ss}^k \end{bmatrix} \begin{bmatrix} \boldsymbol{W}_c & -\boldsymbol{W}_s \\ \boldsymbol{W}_s & \boldsymbol{W}_c \end{bmatrix}. \tag{23}$$

3.4. **Some computational techniques.** In (22) the matrix dimension $(M+1)(n+1)$ makes the computation of matrix multiplication complicated for the high degree $M$. Following the orthogonality of cosine basis, if $u(k)$ is assumed as a pseudo-stationary signal,

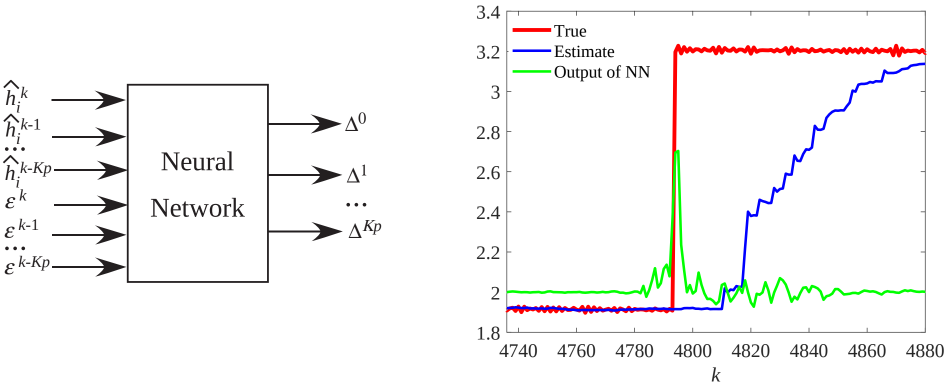
$$\begin{aligned} &\frac{1}{K} \left[ \sum_{k_1=0}^{K} \boldsymbol{\phi}_{c,m_1}^k(k_1) \left( \boldsymbol{\phi}_{c,m_2}^k(k_1) \right)^T \right] \\ &\approx E\left\{ \boldsymbol{\phi}_{c,0}^k(k_1) \left( \boldsymbol{\phi}_{c,0}^k(k_1) \right)^T \right\} \frac{1}{K} \sum_{k_1=0}^{K} \cos(m_1 \omega k_1) \cos(m_2 \omega k_1) \to \begin{cases} \neq 0 \ (m_1 = m_2) \\ = 0 \ (m_1 \neq m_2) \end{cases} \end{aligned} \tag{24}$$

holds for the data vectors of $\boldsymbol{\phi}_{c,m_1}^k(k_1)$ and $\boldsymbol{\phi}_{c,m_2}^k(k_1)$ within the data window. Therefore, corresponding to the structure of $\boldsymbol{\phi}_c^k(k_1)$ and $\boldsymbol{\phi}_{c,m}^k(k_1)$ defined in (4), it is seen that $\boldsymbol{\Phi}_{cc}^k$ can be divided into $(M+1)$ sub-blocks with respect to $m$. Consequently, the parameter update can also be implemented separately for $m = 0, 1, \ldots, M$, where the parameters corresponding to $m = 0$ are similar as the recursive least squares (RLS).

Moreover, the parameters within the data window at $k, k-1, k-2, \ldots, k-K$ are obtained from the CS approximation. Then the parameters at $k-1, k-2, \ldots$ can smooth the past estimates to reduce the influence of noise; however, the influence of Gibbs effect caused by the discontinuity still remains in the estimates.

If there exist measurable process variables that indicate the process characteristics variation, the variation information can be obtained through monitoring the process variables; otherwise the variation points have to be detected through appropriate thresholds. In the next section, a soft threshold is investigated for RCS algorithm.

4. **Detection of Rapid Changing Points.** In practical applications, some abrupt parameter variations occur randomly with unpredictable variation amplitudes, whereas the variation of one parameter influences the estimation of the other parameters through the prediction errors $\varepsilon^{k+1}$; correspondingly, in order to be adaptive to various variations, a threshold should involve much information of identification results. Such threshold is commonly a complicated nonlinear one that is difficult to give a theoretical formula. A preset hard threshold given by variation of the estimated parameters only often misjudges under various situations, hence a nonlinear soft threshold is approximated by using a series of information obtained in identification. To avoid the theoretical deduction, the nonlinear threshold is implemented by neural network, as illustrated in Figure 2(a). The input data to the neural network are the estimated parameters $\hat{h}_i^k, \ldots, \hat{h}_i^{k-K_p}$ and the prediction




(a) Illustration of neural network (b) An example of network output

Figure 2. Detection of abrupt variation point

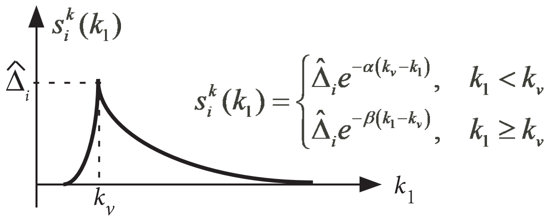


Figure 3. Illustration of compensation function

errors $\varepsilon^k, \ldots, \varepsilon^{k-K_p}$ within a sliding parameter window $[k-K_p,\ k]$, the network outputs are the approximated soft thresholds $\Delta^0, \ldots, \Delta^{k-K_p}$ at the instants $k, k-1, \ldots, k-K_p$.

The coefficients of neural network are determined by optimization algorithms, while the training samples are the simulation data of the estimated and true parameters generated by various time-varying models. When it detects the rapid changing point $k_v$ as shown in Figure 3, a compensation function $s_i^k(k_1)$ that jumps to $\hat{\Delta}_i$ and slowly attenuates to 0 is used to compensate $\bar{h}_i^{k_0+k_1}$ by $\bar{h}_i^{k_0+k_1} - s_i^k(k_1)$, where $\alpha$ and $\beta$ are constants of $\alpha > \beta > 0$. It mitigates the variation of parameters, while the estimates $\hat{h}_i^{k_0+k_1}$ are calculated by the summation of compensated CS approximation and $s_i^k(k_1)$.

5. **Numerical Example.** A simple 3rd order linear model is considered in the numerical example, where the model parameters have abrupt variations with large variation magnitudes. To make the approximation error be no more than 10% for the large abrupt variations, the degree of CS approximation is chosen as $M = 50$. The estimated parameter of $h_2^k$ around a discontinuous point $k = 4893$ is shown in the left of Figure 4. It is seen

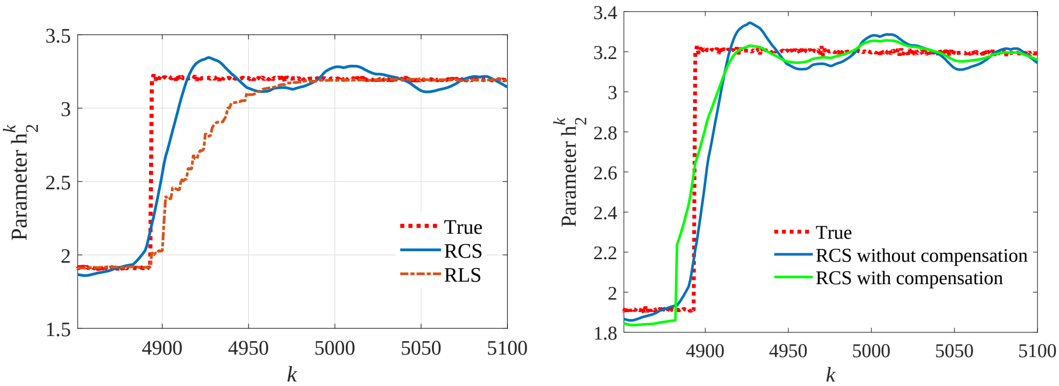



Figure 4. Example of identification result for $h_2^k$

that the estimate obtained by RCS algorithm tracks the true time-varying parameters faster than RLS; however, there still is fluctuation around the discontinuous point.

The trained neural network detects the abrupt variation point at $k_v = 4894$ using the estimated parameters $\hat{h}_i^k, \ldots, \hat{h}_i^{k-30}$ and prediction errors $\varepsilon^k, \ldots, \varepsilon^{k-30}$, where $K_p = 30$, and then the parameter estimation is compensated by $s_i^k(k_1)$, where $\alpha = 0.05$, $\beta = 0.0005$. As shown in the right of Figure 4, the large variation is compensated by $s_i^k(k_1)$, so the fluctuation is largely mitigated, and the tracking becomes slightly faster at the variation point. The situation where several large variations occur in a short period has not been investigated yet in the example. If the detection is further optimized around $k_v$ to reduce the detection error, the detection and compensation performance can be improved greatly.

6. **Conclusions.** The recursive identification algorithm is presented for rapid time-varying systems by using cosine series approximation with the forgetting factor on the system output, and detecting rapid changing is introduced in the proposed algorithm. It has been illustrated that the forgetting factor can reduce the degree of cosine series, and the compensation of the parameter approximation at the rapid changing point can mitigate the fluctuation in the parameter approximation; therefore, the identification performance can be improved for the rapid changing systems. Some meaningful issues such as the accuracy improvement of changing points detection, implementation of the parameter compensation into recursion of parameter update will be considered in the future work.

## REFERENCES

[1] W. Hakvoort, R. Aarts, J. Dijk and J. Jonker, A computationally efficient algorithm of iterative learning control for discrete-time linear time-varying systems, *Automatica*, vol.45, no.12, pp.2925-2929, 2009.

[2] H. Wang, L. Bai, J. Xu and W. Fei, EEG recognition through time-varying vector autoregressive model, *The 12th Int. Conf. on Fuzzy Systems and Knowledge Discovery*, pp.292-296, 2015.

[3] M. Tsatsanis and G. Giannakis, Modelling and equalization of rapidly fading channels, *International Journal of Adaptive Control and Signal Processing*, vol.10, nos.2-3, pp.159-176, 2010.

[4] G. Rödönyi, P. Körös, P. Dániel, A. Soumelidis, Z. Szabo and J. Bokor, Structure selection and LPV model identification of a car steering dynamics, *The 18th IFAC Symposium on System Identification*, pp.1086-1091, 2018.

[5] S. Ruangsang, K. Jirasereeamornkul and W. Assawinchaichote, Control of time-varying delay systems with uncertain parameters via fuzzy-modeled prescribed performance control approach, *International Journal of Innovative Computing, Information and Control*, vol.16, no.2, pp.457-479, 2020.

[6] K. Smerpitak, W. Pinthong and A. Julsereewong, Real-time remote monitoring and alerting system using low-cost IoT platform for odor control scrubbers, *ICIC Express Letters, Part B: Applications*, vol.10, no.6, pp.523-531, 2019.

[7] F. E. Gunawan, A new damage indicator in time and frequency domains for structural health monitoring: The case of beam with a breathing crack, *International Journal of Innovative Computing, Information and Control*, vol.16, no.5, pp.1579-1591, 2020.

[8] S. Simani and R. Patton, Fault diagnosis of an industrial gas turbine prototype using a system identification approach, *Control Engineering Practice*, vol.16, no.7, pp.769-786, 2008.

[9] M. Gao, S. Jiang and F. Pan, Fault detection for time-delayed networked control systems with sensor saturation and randomly occurring faults, *International Journal of Innovative Computing, Information and Control*, vol.14, no.4, pp.1511-1525, 2018.

[10] Q. Zhang and L. Ljung, From structurally independent local LTI models to LPV model, *Automatica*, vol.84, pp.232-235, 2017.

[11] L. Blackmore, S. Rajamanoharan and B. C. Williams, Active estimation for jump Markov linear systems, *IEEE Trans. Automat. Contr.*, vol.53, no.10, pp.2223-2236, 2008.

[12] M. Basseville and I. Nikiforov, *Detecting of Abrupt Changes – Theory and Applications*, Prentice Hall, Englewood Cliffs, NJ, 1993.

[13] L. Ljung, *System Identification – Theory for the User*, Prentice Hall, Englewood Cliffs, NJ, 1999.

[14] Z. Zhang and D. Zhang, New variable step size LMS adaptive filtering algorithm and its performance analysis, *Systems Engineering & Electronics*, vol.31, no.9, pp.2238-2241, 2009.

[15] N. Li, Y. Zhang, Y. Hao and J. Chambers, A new variable step-size NLMS algorithm designed for applications with exponential decay impulse responses, *Signal Processing*, vol.88, no.9, pp.2346-2349, 2008.
[16] K. Ikeda, Convergence analysis of block orthogonal projection and affine projection algorithms, *Signal Processing*, vol.82, pp.491-496, 2002.
[17] S. Broughton and K. Bryan, Discrete Fourier analysis and wavelets, *Journal of Applied Statistics*, vol.37, 2008.
[18] M. Spiridonakos and S. Fassois, Non-stationary random vibration modelling and analysis via functional series time dependent ARMA (FS-TARMA) models – A critical survey, *Mechanical Systems and Signal Processing*, vol.47, nos.1-2, pp.175-224, 2014.
[19] M. Niedzwiecki, *Identification of Time-Varying Processes*, John Wiley & Sons, Ltd., 2000.
[20] M. Niedzwiecki, Recursive functional series modeling approach to identification of time-varying plants, *IEEE Trans. Automat. Contr.*, vol.35, no.5, pp.610-616, 1990.
[21] X. Liu and L. Sun, Recursive identification algorithm based on cosine basis for rapid time-varying systems, *International Journal of Innovative Computing, Information and Control*, vol.15, no.2, pp.617-628, 2019.
[22] X. Liu, L. Sun and J. Liu, Trigonometric basis functions based time-varying identification algorithm with output weight factor, *Proc. of the 12th Asian Control Conference*, Kitakyushu, Japan, pp.1072-1077, 2019.