Online Wavelet Denoising via a Moving Window

XIA Rui[1] MENG Ke[1] QIAN Feng[1] WANG Zhen-Lei[1]

Abstract In this paper, shortcoming of traditional wavelet denoising in real-time signal processing is discussed, requirements of online denoising are considered, and a moving window is introduced into traditional wavelet transform. Using the moving window, an online wavelet denoising approach is proposed. Some problems of online denoising, such as border distortion and pseudo-Gibbs phenomena, are discussed. To solve these problems, window extension and window cycle spinning are also proposed. Different approaches are tested by the signal widely used in denoising domain. Both the visual results and the quantitative measures are presented to highlight the availability of the new approach.

Key words Discrete wavelet transform, online denoising, moving window, window extension, window circle spinning

1 Introduction

Because of some key advantages over Fourier analysis, wavelet analysis has become a widely used tool in signal estimation, classification, and compression. Wavelet transform tends to concentrate the signal energy into a relatively small number of large coefficients. On this basis, a method called wavelet shrinkage to use thresholding in wavelet domain was proposed, and it was shown to be asymptotically near optimal for a wide range of signals corrupted by additive Gaussian noise[1,2].

Denoising with traditional wavelet transforms always exhibit visual artifacts because of translation-variant, such as pseudo-Gibbs phenomena in the neighborhood of discontinuities. Methods of cycle spinning and translation-invariant denoising produced reconstructions with significantly weaker artifacts than with traditional orthogonal wavelet transform[3].

However, in most cases, wavelet denoising for real-time signal is actualized *via* offline processing, which limits the real-time applications. In this paper, an online wavelet denoising method using a moving window is proposed. Here, some problems that may occur in real-time wavelet denoising, such as border distortion and pseudo-Gibbs phenomena, are discussed. To solve these problems, methods of using window extension and window circle spinning, are developed. Quantitative measures describing the denoising effect and computation complexity of different approaches are presented.

2 Online wavelet denoising approach

2.1 Review of offline wavelet denoising

The energy concentration property makes wavelet transform a powerful method for data compression and noise elimination.

In denoising, the pure signal from the noise-corrupted data should be reconstructed

$$\boldsymbol{x} = \boldsymbol{y} + \boldsymbol{z} \tag{1}$$

where $\boldsymbol{y}$ is the desired signal and $\boldsymbol{z}$ is the additive noise.

After wavelet transform in wavelet domain, we have

$$\begin{cases} \boldsymbol{X} = T\boldsymbol{x}, \boldsymbol{Y} = T\boldsymbol{y}, \boldsymbol{Z} = T\boldsymbol{z} \\ \boldsymbol{X} = \boldsymbol{Y} + \boldsymbol{Z} \end{cases} \tag{2}$$

where T denotes the wavelet coefficient matrix.

A reasonable method for wavelet-based signal denoising is to shrink the small entries of X while retaining the large entries that contain the desired information. The nonlinear wavelet shrinkage can be viewed as a diagonal filtering operation in wavelet domain. If filter is denoted by H, then after inverse wavelet transform, we get the denoised data

$$\hat{\boldsymbol{x}} = T^{-1}HT\boldsymbol{x} \tag{3}$$

One method of reducing visual artifacts in wavelet denoising is to first perform cycle spinning for the signal, and then to do thresholding. Another method, based on circle spinning, is to spin the whole circle shifts of signal, termed fully translation-invariant (TI) denoising. The latter method can suppress the artifacts more effectively. For details regarding about wavelet shrinkage, see [4, 5].

2.2 Moving window based online denoising

Although normal wavelet shrinkage plays an important role in denoising, it can hardly satisfy the requirements of real-time application. We may use a moving window to solve this problem. A similar idea also can be seen in [6].

At the first stage of online denoising, when sampled data are not long enough for a wavelet transform, we keep the data as such. As soon as the minimum length is reached, the first window initiates and online denoising begins. Subsequently, the window moves ahead step by step with the width fixed.

During the interval of sampling time t_s between $x(i)$ and $x(i+1)$, we obtain the window

$$W_i = \begin{cases} \text{none}, & i < lx \\ \{x(i-lx+1), x(i-lx+2), \cdots, x(i)\}, & i \geq lx \end{cases} \tag{4}$$

where $x(i)$ is the ith value of real-time signal, lx is the length of the signal cut by current window. If there is no signal extension, the width of window lw is equal to lx.

We define the scaling function $\varphi_{j,k}(t)$ and the wavelet function $\psi_{j,k}(t)$ as

$$\begin{cases} \varphi_{j,k}(t) = 2^{-\frac{j}{2}}\varphi(2^{-j}t-k) \\ \psi_{j,k}(t) = 2^{-\frac{j}{2}}\psi(2^{-j}t-k) \end{cases} \tag{5}$$

where j and k are integers scaling and shifting the mother function to generate families of functions.

Based on multi-resolution analysis, we can get the double-scaling equations as

$$\begin{cases} \varphi(2^{-j}t) = \sqrt{2}\sum\limits_k h_k\varphi(2^{-(j-1)}t-k) \\ \psi(2^{-j}t) = \sqrt{2}\sum\limits_k g_k\varphi(2^{-(j-1)}t-k) \end{cases} \tag{6}$$

Received June 30, 2006; in revised form December 14, 2006

Supported by National Science Fund for Distinguished Young Scholars (60625302), National Key Fundamental Research Project of China (2002CB3122000), and National High Technology Research and Development Program of China (863 Program) (20060104Z1081)

1. State-Key Laboratory of Chemical Engineering, East China University of Science and Technology, Shanghai 200237, P. R. China

DOI: 10.1360/aas-007-0897

We let $c_{i,0,k}$ be equal to $W_i(k)$, do discrete wavelet transform (DWT) inside the current window W_i, compute the scaling coefficients $c_{i,j,k}$ and the wavelet coefficients $d_{i,j,k}$ as the following functions.

$$\begin{cases} c_{i,j,k} = \sum\limits_n h_{n-2k} c_{i,j-1,n} \\ d_{i,j,k} = \sum\limits_n g_{n-2k} c_{i,j-1,n} \end{cases}, \quad j \geq 1 \tag{7}$$

where the coefficients h_{n-2k} and g_{n-2k} can be deduced from (6) as

$$\begin{cases} h_{n-2k} = \langle \varphi_{j,k}, \varphi_{j-1,n} \rangle \\ g_{n-2k} = \langle \psi_{j,k}, \varphi_{j-1,n} \rangle \end{cases} \tag{8}$$

After threshold shrinkage, we obtain the shrunk wavelet coefficients $\hat{d}_{i,j,k}$. The signal reconstruction can be computed by

$$\hat{c}_{i,j-1,n} = \sum_k h_{n-2k} \hat{c}_{i,j,k} + \sum_k g_{n-2k} \hat{d}_{i,j,k}, \quad j \geq 1 \tag{9}$$

In digital signal processing, the wavelet decomposition is equivalent to the analysis part of a two-channel filter bank, and the inverse transform corresponds to the synthesis part. The fast DWT algorithm based on this idea is also called Mallat algorithm. For more about that, see [4,7].

Notice that at the coarsest level, we only need to compute the lxth denoised value

$$\hat{c}_{i,0,lx} = \sum_k h_{lx-2k} \hat{c}_{i,1,k} + \sum_k g_{lx-2k} \hat{d}_{i,1,k} \tag{10}$$

$\hat{W}_i(lx)$ is the updated data, which is equal to $\hat{c}_{i,0,lx}$. $\hat{W}_i(lx)$ comprises the denoised value $\hat{\boldsymbol{x}}$.

$$\hat{x}(i) = \hat{W}_i(lx) = \hat{c}_{i,0,lx} \tag{11}$$

As soon as $x(i+1)$ is sampled, the above steps are repeated to get $\hat{x}(i+1)$.

2.3 Problem of distortion

We choose two from the four demosignals constructed and analyzed for wavelet denoising in [8]. Fig. 1 shows the two original signals (Blocks and HeaviSine) and their noisy versions.

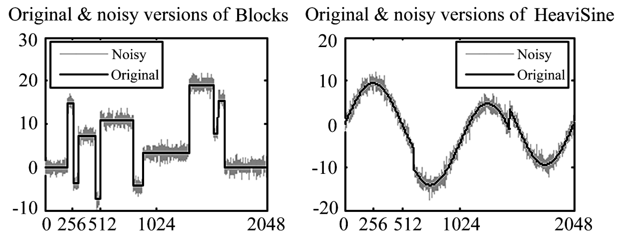


Fig. 1 Original and noisy version of the demosignals

For comparison of different approaches, in this section we use the same wavelet basis (sym4), decompose at the same depth ($J = 5$) and apply soft shrinkage with the universal threshold $\tau = \sigma\sqrt{2\ln(N)}$. Fig. 2 presents the results of simple online wavelet denoising, where $lw = lx = 512$. After $x(512)$ is sampled, the window initiates and moves. To each window, simple wavelet shrinkage is applied. Wavelet shrinkage of the latest window is also shown in Fig. 2.

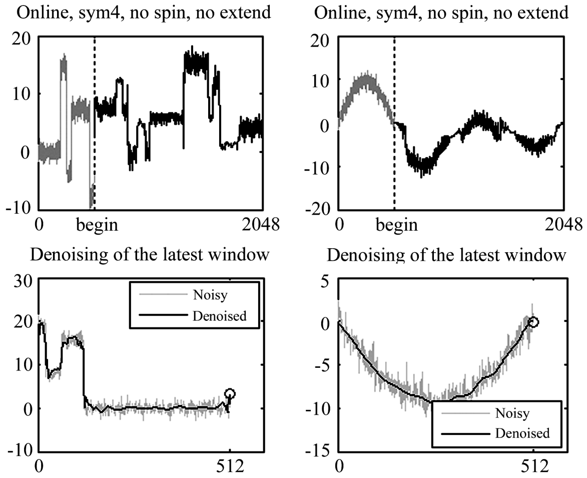


Fig. 2 Simple online wavelet denoising

The oscillations we see from the latest window denoising are pronounced, especially on the right border $W_i(lw)$, which is marked with a circle in the figure. Here, each updated data $\hat{W}_i(lx)$ is located at the border, so the whole denoised signal is awful.

Because circle spinning and TI denoising are effective weapons against pseudo-Gibbs phenomena and border distortion, we may bring the idea into real-time applications. Fig. 3 shows the effect when TI is applied to online denoising. Although TI denoising makes denoised signal in a single window smoother than that in Fig. 2, the whole online denoised signal still has a serious distortion compared to the original version. Averaging does not affect online applications because only one denoised value goes into effect in each window, furthermore, it is on the border.

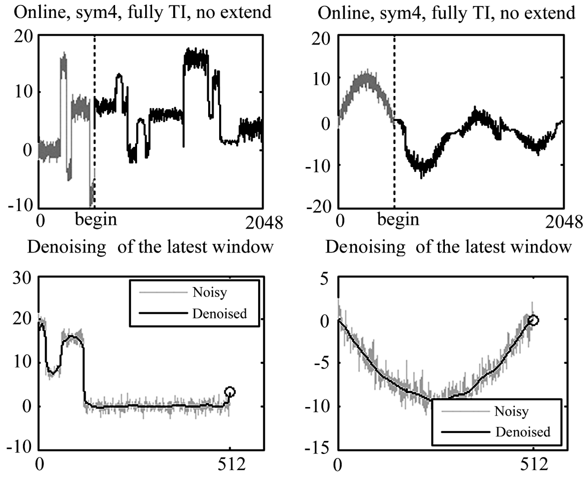


Fig. 3 Fully TI based online wavelet denoising

2.4 Window extension

We know that in offline wavelet denoising, signal extension can be used to reduce the border distortion. Familiar extensions include zero extension, smooth extension, symmetric extension, periodized extension, and so on[9].

In real-time wavelet denoising, one key point is that the updated data must be located beyond discontinuities. Zero extension and periodized extension always add great discontinuity to the border, although the distortions of the low frequency component are canceled by reconstruction when using periodized extension. If we truncate some of the coefficients corresponding to the high frequency signal as we do denoising, the border effect may be slightly noticeable and annoying in the reconstructed signal. Here, we suggest some extension methods that keep the continuity in the area of updated data, such as smooth extension and symmetric extension.

Fig. 4 shows the results of online denoising, where smooth window extension is used. Here, $lx = 256$, and $lw = 2lx$.

$$W_i = \begin{cases} \text{none}, & i < lx \\ \{x(i-lx+1), \cdots, x(i), \overbrace{x(i), \cdots, x(i)}^{\text{smooth}}\}, & i \geq lx \end{cases} \tag{12}$$

After wavelet shrinkage, $x(i)$ located at the center of $W_i(W_i(256))$ updates, as marked by a circle in the figure of the latest window.

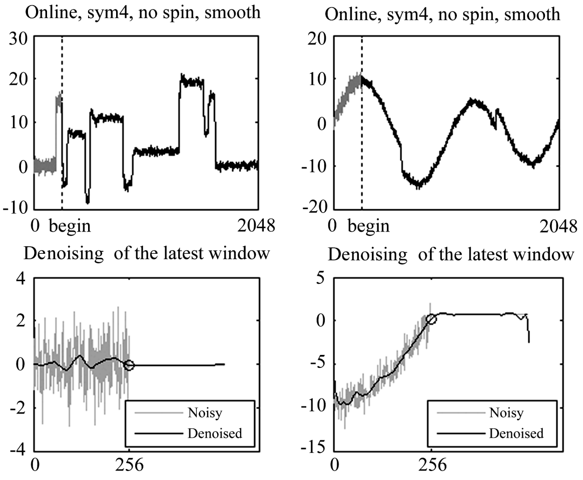


Fig. 4 Online denoising with smooth window extension

Fig. 5 shows the result when symmetric window extension is applied. Here,

$$W_i = \begin{cases} \text{none}, & i < lx \\ \{x(i-lx+1), \cdots, x(i), \overbrace{x(i), \cdots, x(i-lx+1)}^{\text{symmetric}}\}, & i \geq lx \end{cases} \tag{13}$$

As is evident in Figs. 4 and 5, artifacts are reduced efficiently compared with approaches without extensions. Symmetric extension even gets a higher score than smooth extension.

To reduce the load of computation, a short extension method is suggested. The length for extension is denoted by lt ($lt < lx$). The length of extended window is then given as $lw = lx + lt$. The short-extended window can be expressed as

$$W_i = \begin{cases} \text{none}, & i < lx \\ \{x(i-lx+1), \cdots, x(i), \overbrace{x(i), \cdots, x(i-lt+1)}^{\text{short-symmetric}}\}, & i \geq lx \end{cases} \tag{14}$$

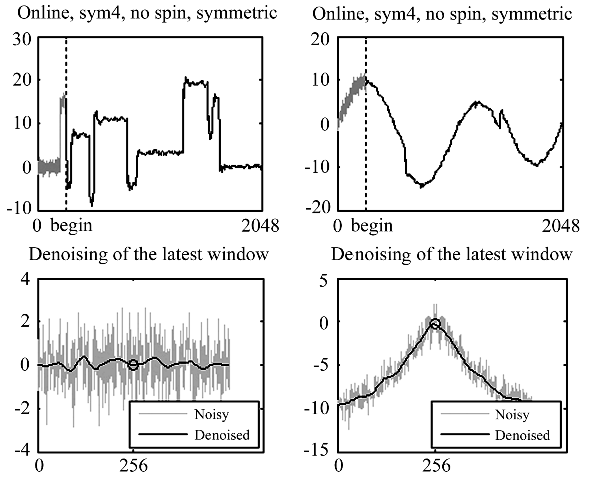


Fig. 5 Online denoising with symmetric extension

Fig. 6 gives the results of this approach with $lx = 240$ and $lt = 16$. More results of different window extension approaches are presented in quantitative measures below.

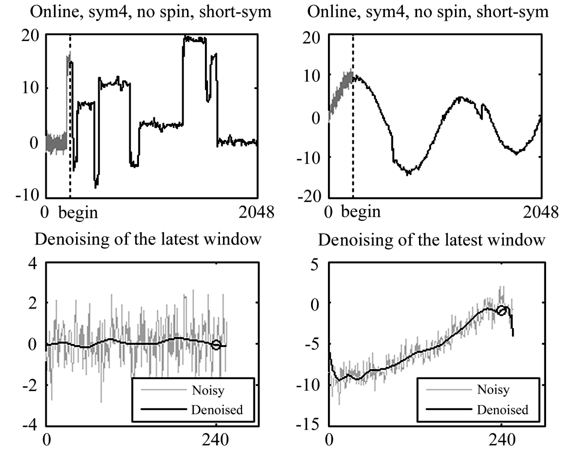


Fig. 6 Online denoising with short-symmetric window extension

2.5 Window circle spinning

Approach of circle spinning is successful in reducing oscillation in [3]. Here, some cases of applying it to online denoising are presented. Fig. 7 gives the results using symmetric window extension and averaging over a range of 16 circle spinning shifts.

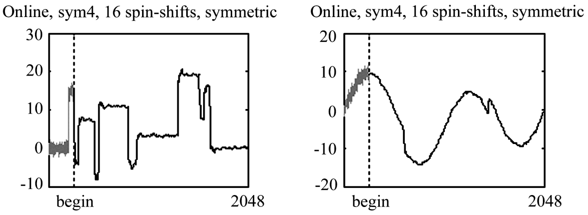


Fig. 7 Online denoising with symmetric extension and circle spinning

Fig. 8 presents the results of online denoising with symmetric window extension and fully TI.

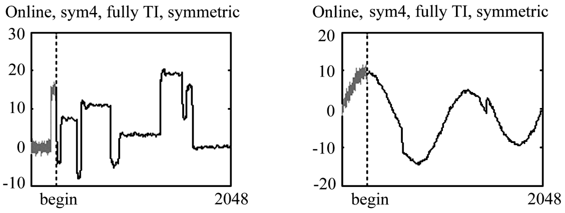


Fig. 8 Online denoising with symmetric extension and full TI

Although the oscillations are reduced to some extent, averaging of all shifts only benefits one window. In the whole denoised signal, the averaging effect is reduced. On the other hand, one key element of real-time application is speediness. In this case, few shifts may be available, fully TI denoising that costs a large mount of computation is not desirable.

3 Quantitative measures

The original signal without noise is denoted by $x(i)$ and the denoised one is denoted by $\hat{x}(i)$, and define root of mean square error (*RMSE*) as

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x(i)-\hat{x}(i))^2} \tag{15}$$

Table 1 summarizes the numerical performances of different online denoising approaches shown in the above figures. Performances of offline denoising are also summarized by comparison.

Table 2 summarizes the results where more parameters are adjusted to meet different real-time requirements.

Table 1 *RMSE* of the above wavelet denoising approaches

Parameters	Blocks	HeaviSine
Offline, sym4, no spin	0.8190	0.2621
Offline, sym4, 16 spin-shifts	0.7465	0.2574
Offline, sym4, fully TI	0.7244	0.2607
Online, sym4, no spin, no extend (512+0)	3.2810	3.3809
Online, sym4, fully TI, no extend (512+0)	3.1197	3.1934
Online, sym4, no spin, smooth (256+256)	0.8060	0.4999
Online, sym4, no spin, symmetric (256+256)	0.7598	0.4156
Online, sym4, no spin, short-sym (240+16)	0.7475	0.6752
Online, sym4, 16 spin-shifts, symmetric (256+256)	0.7407	0.4725
Online, sym4, fully TI, symmetric (256+256)	0.7299	0.4278

Table 2 *RMSE* for more choices of parameters

Parameters	Blocks	HeaviSine
Online, sym8, no spin, symmetric (256+256)	0.8104	0.4285
Online, sym8, 16 spin-shifts, symmetric (256+256)	0.7633	0.4513
Online, db4, no spin, symmetric (256+256)	0.8398	0.4897
Online, db8, no spin, symmetric (256+256)	0.8823	0.4613
Online, haar, no spin, symmetric (256+256)	0.7484	0.6736
Online, sym4, no spin, short-sym (384+128)	0.7187	0.4250
Online, sym4, no spin, short-sym (500+12)	0.7221	0.5247

Table 3 Mean denoising time of online wavelet denoising

Parameters	Blocks(ms)	HeaviSine(ms)
Online, sym4, J=5, no spin, smooth (256+256)	3.8	3.7
Online, sym4, J=5, no spin, symmetric (256+256)	3.6	3.4
Online, sym4, J=5, 16 spin-shifts, symmetric (256+256)	53.9	54.9
Online, sym4, J=5, fully TI, symmetric (256+256)	1619.1	1634.6
Online, sym4, J=5, no spin, short-sym(240+16)	3.2	3.1
Online, sym4, J=7, no spin, symmetric (256+256)	4.6	4.5
Online, sym4, J=3, no spin, symmetric (256+256)	2.7	2.8
Online, sym4, J=3, no spin, short-sym (240+16)	2.3	2.2
Online, sym4, J=3, no spin, short-sym (56+8)	1.9	1.9
Online, sym4, J=3, 16 spin-shifts, short-sym (56+8)	27.8	27.0

Although computation complexity is not the main issue in this paper, online denoising must take into account not only denoised extent but also speediness. We record denoising time $T(i)$ of $x(i)$ and define the mean time of online denoising MDT as

$$MDT = \sqrt{\frac{1}{N}\sum_{i=1}^{N} T(i)} \tag{16}$$

As a reference, Table 3 summarizes performances of different approaches mostly in correspondence to Tables 1 and 2. These results are all obtained at the same simulation environment.

4 Conclusions

In this paper, we emphasize on the availability of online wavelet denoising. It is visually and quantitatively demonstrated that the online wavelet denoising approach has the following properties:

1) Adaptivity to real-time applications.
2) Noise almost suppressed .
3) Weak visual artifacts.
4) Adjustable parameters.

To meet different real-time conditions, some parameters are made adjustable in this approach, such as wavelet basis, decomposing depth, window width, window extension methods, and spinning shifts.

According to the quantitative measures, both the accuracy and the speediness of online wavelet denoising are indispensable. The choice of different parameters should cater for real-time requirement. In principle, if it is a severe real-time application, we suggest an approach with a short window width and short extension, but without circle spinning; else methods using available shifts of circle spinning are preferred.

References

1 Dohono D L, Johnstone I M. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, 1994, **81**(9): 425∼455

2 Donoho D L. Denoising by soft-thresholding. *IEEE Transactions on Information Theory*, 1995, **41**(3): 613∼627

3 Coifman R R, Dohono D L. *Translation-invariant Denoising Wavelet and Statistics*. New York: Springer-Verlag, 1995. 125∼150

4 Cheng Li-Zhi, Wang Hong-Xia, Luo Yong. *Wavelet Theory and Application*. Beijing: Science Publishing House, 2004. 271∼282 (in Chinese)

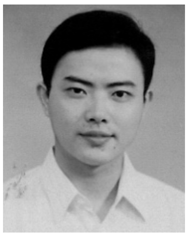
5 Fecit Technology. *Wavelet Analysis and Applications with Matlab*. Beijing: Publishing House of Electronics Industry, 2005. 49∼80 (in Chinese)

6 Jiang Dong-Fang, Chen Ming. A real-time wavelet denoising algorithm. *Instrument and Control*, 2004, **25**(6): 781∼783 (in Chinese)

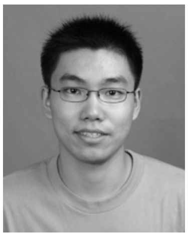
7 Mallat S. Theory for multi-resolution signal decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1989, **11**(7): 674∼693

8 Donoho D L, Johnstone I M, Kerkyacharian G, Picard D. Wavelet shrinkage: asymptopia. *Statist*, 1995, **57**(2): 301∼369

9 Tang Yuan-Yan, Wang Ling. *Wavelet Analysis and Text Recognition*. Beijing: Science Publishing House, 2005. 44∼94 (in Chinese)

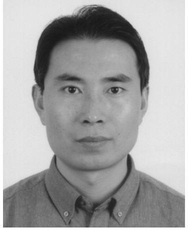


XIA Rui Ph. D. candidate at Institute of Automation, Chinese Academy of Sciences. He received his bachelor degree from Southeast University in 2004 and master degree from East China University of Science and Technology in 2007. His research interest covers signal processing and pattern recognition.
Website: http://www.librawill.com
E-mail: librawill@gmail.com



MENG Ke Ph. D. candidate at School of Information Technology and Electrical Engineering, University of Queensland. He received his bachelor degree from Nanjing University of Technology in 2004 and master degree from East China University of Science and Technology in 2007. His research interest covers power system control and analysis.
E-mail: funnyjason@gmail.com

QIAN Feng Professor at Institute of Automation, East China University of Science and Technology. He received his Ph. D. degree from East China University of Science and Technology in 1995. His research interest covers intelligent control and system engineering. Corresponding author of this paper.
E-mail: fqian@ecust.edu.cn



WANG Zhen-Lei Associate professor at Institute of Automation, East China University of Science and Technology. He received his Ph. D. degree from Northeast University in 2002. His research interest covers intelligent algorithm and control.
E-mail: wangzhen_l@ecust.edu.cn