An Improved Real-time Denoising Method Based on Lifting Wavelet Transform

Zhaohua Liu [1], Yang Mi[2], Yuliang Mao[2]

[1] Tianjin Key Laboratory of High Speed Cutting and Precision Machining, Tianjin University of Technology and Education, Tianjin, 300222, China, e-mail: liuzhaohuamail@163.com
[2] School of Automation, Beijing Institute of Technology, Beijing, 100081, China

Signal denoising can not only enhance the signal to noise ratio (SNR) but also reduce the effect of noise. In order to satisfy the requirements of real-time signal denoising, an improved semisoft shrinkage real-time denoising method based on lifting wavelet transform was proposed. The moving data window technology realizes the real-time wavelet denoising, which employs wavelet transform based on lifting scheme to reduce computational complexity. Also hyperbolic threshold function and recursive threshold computing can ensure the dynamic characteristics of the system, in addition, it can improve the real-time calculating efficiency as well. The simulation results show that the semisoft shrinkage real-time denoising method has quite a good performance in comparison to the traditional methods, namely soft-thresholding and hard-thresholding. Therefore, this method can solve more practical engineering problems.

Keywords: Real-time denoising, semisoft shrinkage, lifting wavelet transform, moving data window.

1. INTRODUCTION

NOISE is random and unpredictable, so noise reduction processing is necessary to get an accurate measurement result. Wavelet denoising is one of the important applications in engineering practice. Since 1989, when S. Mallat presented wavelet transform fast algorithm [1], people have proposed a series of wavelet denoising methods by using the wavelet transform. So far, there are three main types of the wavelet denoising method which are in essence the same: spatially selective noise filtration(SSNF) [2], modulus maxima denoising [3] and wavelet thresholding denoising [4]. First, we obtain the wavelet coefficients by the wavelet transform; second, process wavelet coefficients by using a certain algorithm; at last, reconstruct the signal by previous processed coefficients to denoise. Wavelet threshold denoising method can achieve optimum in the sense of minimum mean square error approximation, and it has better visual quality, so it is widely used in the signal denoising domain.

In 1996, W. Sweldens presented a method to construct the wavelet function based on lifting scheme [5], and subsequently presented the concept of second generation wavelet transform [6]. Literature [7] proved that any wavelet transform which has finite impulse response filter (FIR) can be decomposed by the finite number of steps lifting scheme. The conclusion establishes the relationship between the traditional discrete wavelet transform and the second generation wavelet transform, and also provides the foundation for constructing a richer wavelet basis function. Comparing with traditional wavelet transform based on the Mallat algorithm, lifting scheme does not rely on the concept of translations and expansions, and it also avoids the Fourier transform absolutely. Lifting scheme can be calculated in-place without auxiliary memory, thus it has fast computational speed. In the condition of same data length, the computation speed of lifting scheme wavelet transform can be increased by a factor of two in comparison to the traditional wavelet transform. Second generation wavelet transform has so many merits, thus it becomes a hotspot of discrete wavelet transform.

Liu [8] applied lifting wavelet algorithm to the signal denoising of inertial instruments and SINS alignment, experimental results show that lifting wavelet algorithm can improve the rate and precision of strap-down inertial navigation system (SINS) alignment. M. Abid [9] proposed an image denoising method by using the adaptive lifting scheme, this method estimates the noise energies in the subband and uses lifting wavelet to perform image denoising. Such scheme can adapt itself well to the analyzed signal, which allows keeping important information for de-noising applications. Although Literature [8] and [9] made a good noise reduction effect by using lifting wavelet transform, this wavelet denoising method is usually employed in offline mode, thus its application scope is limited. In order to satisfy the requirements of real-time signal denoising, Xia [10] proposed an online wavelet denoising approach. Xia introduced a moving data window into traditional wavelet transform to carry out online wavelet denoising, and discussed some problems of online denoising, such as border distortion and pseudo-Gibbs phenomena, but the computation load of this denoising algorithm is large because it uses traditional wavelet transform.

In this paper, in order to satisfy the requirements of real-time wavelet signal denoising and enhance the effect of denoising, an improved semisoft shrinkage real-time denoising method based on lifting wavelet transform is proposed. Theoretical analysis and MATLAB simulation experimental results prove the effectiveness of this method.

2. THE LIFTING WAVELET TRANSFORM
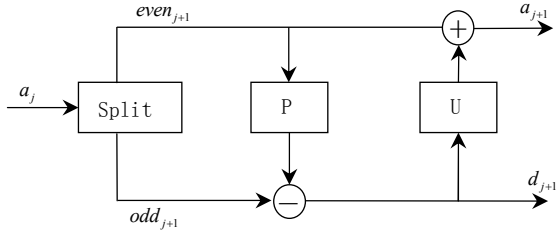
A. Principle of lifting scheme.

Fig.1. Forward wavelet lifting transform.

Due to the high efficiency, little memory and low computational complexity, the lifting wavelet transform is widely used in signal processing. The forward wavelet lifting transform is described in Fig.1. A typical lifting scheme consists of three steps: split, predict and update. Let a_j be original data, lifting step is described as follows:

1) Split step: the sample data are divided into two subsets: the odd sample set odd_{j+1} and the even sample set $even_{j+1}$. This signal split processing is called lazy wavelet transform. Other methods can be used in data split processing as well.

$$\begin{cases} odd_{j+1}(k) = a_j(2k+1), k \in Z \\ even_{j+1}(k) = a_j(2k), k \in Z \end{cases} \quad (1)$$

2) Predict step: in this step, the odd sample set odd_{j+1} is predicted from the neighboring even sample set $even_{j+1}$. The high-pass coefficient d_{j+1} is calculated as the error in predicting the odd sample set odd_{j+1} from the even one $even_{j+1}$ by using a prediction factor P, such that:

$$d_{j+1} = odd_{j+1} - P(even_{j+1}) \quad (2)$$

Predict step is also called dual lifting. d_{j+1} is called wavelet coefficient or detail coefficient.

3) Update step: to produce the low-pass coefficient a_{j+1}, the even sample set $even_{j+1}$ is updated from the wavelet coefficient d_{j+1} by using the updating factor U

$$a_{j+1} = even_{j+1} + U(d_{j+1}) \quad (3)$$

Update step is also termed as primal lifting. a_{j+1} is called scale coefficient or approximation coefficient.

The structure of inverse transform process is symmetrical with forward transformation process of lifting scheme. Thereby, it ensures not only the integrity of reversible transform but also the transform accuracy of reconstruction.

The inverse wavelet lifting transform is described in Fig.2. Inverse transform process of lifting scheme also consists of three steps.

1) Update step:

$$even_{j+1} = a_{j+1} - U(d_{j+1}) \quad (4)$$

2) Forecast step:

$$odd_{j+1} = d_{j+1} + P(even_{j+1}) \quad (5)$$

3) Merge step:

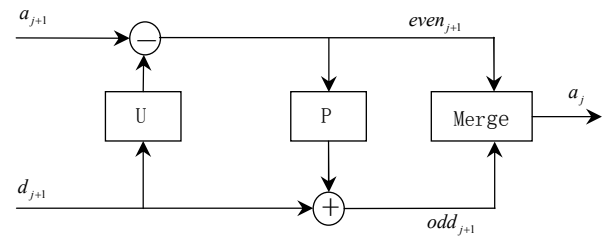$$a_j = Merge(even_{j+1}, odd_{j+1}) \quad (6)$$

Fig.2. Inverse wavelet lifting transform.

B. Lifting implementation of traditional wavelet transform.

The construction of traditional wavelet which is based on lifting scheme does not create the new wavelet. The construction of wavelet can be completed entirely in the spatial domain. At the same time, the computation amount of the first generation traditional wavelet transform based on lifting algorithm is reduced by half in comparison with the classical Mallat algorithm, and lifting algorithm can save a lot of storage units, so it is more suitable for real-time noise reduction processing.

Lifting implementation of wavelet transform consists of split, predict and update. The lifting implementation of forward wavelet transform is also comprised of the above three steps. First, employing lazy wavelet to split the original input data $x(z)$ and obtaining the odd sample set $o(z)$ and the even sample set $e(z)$, as depicted in Fig.3.
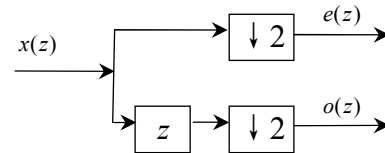

Fig.3. z-transform of lazy wavelet.

And then, employing odd sample set $o(z)$ to predict even sample set $e(z)$ via factor $u_1(z)$, employing prediction error to update odd sample set $o(z)$ via operator $p_1(z)$, and repeating lifting and dual lifting steps. At last, wavelet

approximation coefficients $a(z)$ and detail coefficients $d(z)$ can be obtained by correction of odd sample $o(z)$ and even sample $e(z)$ via scaling $1/K$ and K, respectively. The flow chart of lifting implementation of forward wavelet transform is shown in Fig.4.

In Fig.4., $\downarrow 2$ denotes down sampling, suppose signal data is $x = (..., x_{-2}, x_{-1}, x_0, x_1, x_2, ...)$, the down sampling data is $B_x = (..., x_{-2}, x_0, x_2, ...)$; $\uparrow 2$ denotes up sampling, suppose signal data is $x = (..., x_{-2}, x_{-1}, x_0, x_1, x_2, ...)$, the up sampling data is $E_x = (..., x_{-2}, 0, x_{-1}, 0, x_0, 0, x_1, 0, x_2, ...)$.

The inverse wavelet transform using lifting is described in Fig.5. as follows: First a scaling, then alternating dual lifting and lifting steps, and finally the inverse Lazy transform. The inverse transform can immediately be derived from the forward by running the scheme backwards.

Suppose $x = \{x_1, x_2, \cdots, x_N\}$ is input data whose data length is N, e_l and o_l denote even sample set and odd sample set, respectively. The lifting implementation algorithm of wavelet transform is provided in the Appendix.
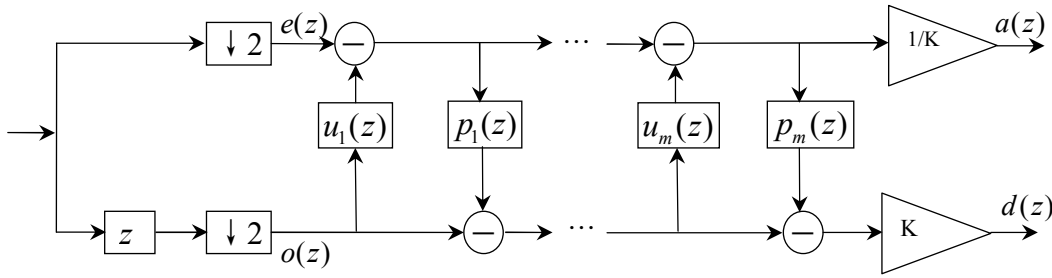

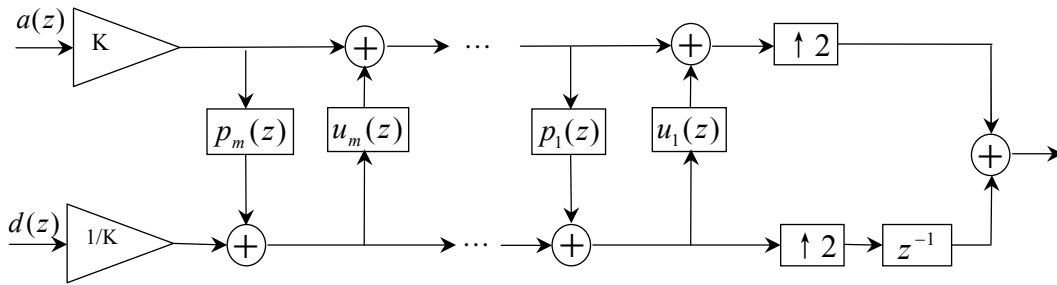Fig.4. The lifting implementation of forward wavelet transform.

Fig.5. The lifting implementation of inverse wavelet transform.

3. IMPROVED SEMISOFT SHRINKAGE REAL-TIME DENOISING METHOD

A. Overall design.

Precision and computational efficiency of wavelet denoising are two important performance criteria for real-time application. In order to satisfy the requirements of the above performance criteria, an improved semisoft shrinkage real-time wavelet denoising method based on lifting scheme was proposed. The method, which adopts moving data window technology, makes real-time wavelet denoising become possible, employs wavelet transform based on lifting scheme to reduce computational complexity significantly, uses hyperbolic thresholding function and recursive threshold computing method to guarantee the dynamic characteristics of the system and further improve the real-time calculating efficiency, and adopts symmetric extension for handling finite length signal boundary distortion, the improved scheme is shown in Fig.6.
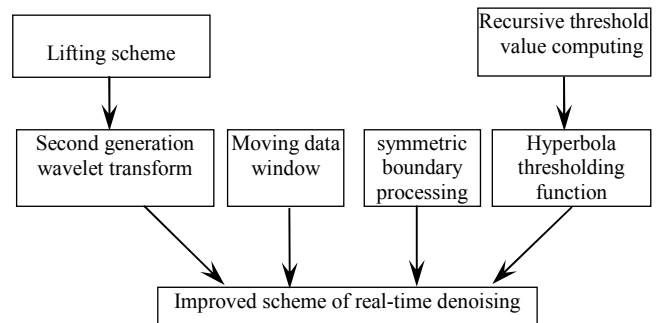

Fig.6. The improved scheme of real-time denoising.

B. Moving data window.

It turns out that every FIR wavelet can be decomposed into lifting steps. As discrete wavelet transform has characteristics of shift variant and non-causality, it is difficult to get the recursive calculations about wavelet transform and wavelet denoising. Consequently, it affects the online real time application of wavelet denoising. For

realizing real-time processing of signal denoising, the moving data window technology [10] was adopted. The basic concept of moving data window is to get the latest piece of real-time data every time, and then by using wavelet denoising method to process it, the end data of the piece of denoised data is output as final denoising result.

Define $W = 2^k, k \in Z, k > 1$ is the length of moving data window, W is also the required minimum number of sample data for the multi-resolution wavelet decomposition, x_i is the ith value of real-time sample signal. At the first stage of online denoising, when sample 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 fixed width. The concrete steps of moving data window are described as follows:

- When $i < 2^k$, due to the length of sample data sequence $(x_1, x_2, \cdots, x_i)$ is too short, it does not meet the requirement of minimum length, so wavelet denoising method does not perform, the sample data are directly output without any processing.

- When $i = 2^k$, the length of sample data sequence $(x_1, x_2, \cdots, x_{2^k})$ meets the requirement of minimum length of moving data window, then the first wavelet denoising performs. The data sequence after denoising is denoted as $(\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_{2^k})$. Finally, take signal $\hat{x}_{2^k}$ as the denoising result of signal x_{2^k} and output it.

- When $i = 2^k + 1$, using wavelet denoising in the sample data sequence $(x_2, x_3, \cdots, x_{2^k+1})$. The data sequence after denoising is denoted as $(\hat{x}_2, \hat{x}_3, \cdots, \hat{x}_{2^k+1})$. Finally, take signal $\hat{x}_{2^k+1}$ as the denoising result of signal x_{2^k+1} and output it.

- The data window moves ahead step by step, and the above steps are repeated until all the sample data is processed.

C. Recursive threshold value computing method.

The selection of threshold value is the key issue in the method of wavelet threshold denoising. Currently, literature [4] proposed the widely used threshold selection method named as universal threshold method. The universal threshold is defined as

$$\lambda = \sigma \sqrt{2 \log N}$$
$$\sigma = \frac{median(| d_{1,j} |)}{0.6745}$$
(7)

Where, N is the signal length, σ denotes the standard deviation of the noise, σ can be estimated from the median of the detail coefficients $d_{1,j}$ at the first level of signal decomposition, $median(\cdot)$ is median function.

The universal threshold may be unwarrantedly large due to its dependence on the number of samples. It will yield an overly smoothed estimate. Considering that the wavelet coefficients within subbands are, in fact, locally stationary and have dependency both in scale and across scales, here a level-dependent threshold, which is more adaptive to the noise and signal characteristics, is presented by the formula:

$$\begin{cases} \lambda_1 = \dfrac{median(| d_{1,j} |)}{0.6745} \sqrt{2 \log N_1} \\ \lambda_i = \dfrac{median(| d_{i,j} |)}{0.6745} \sqrt{2 \log N_i}, \quad i > 1 \end{cases}$$
(8)

Where, N_i is the signal length, if the width of moving data window is 2^k, then $N_1 = 2^k$, $N_2 = 2^{k-1}$, $N_3 = 2^{k-2}, \cdots$; λ_i is the ith level threshold value; $d_{i,j}$ is the detail coefficient at the ith level of signal decomposition.

From (8), level-dependent threshold method considers the propagation characteristics of the noise wavelet coefficients in the wavelet domain. Noise wavelet coefficient has different threshold in different scale. However, this method has shortcoming of threshold value depending on the signal length. At the same time, median function $median(\cdot)$ needs to be calculated when calculating each level noise standard deviation, thus, the computational complexity is large and the real-time performance is poor.

This paper presents a more efficient recursive threshold computing method [11], and its expression is

$$\begin{cases} \lambda_1 = \beta \cdot \dfrac{median(| d_{1,j} |)}{0.6745} \sqrt{2 \log N_1} \\ \lambda_i = \lambda_{i-1} \cdot \dfrac{i-1}{i+\alpha-1}, \quad i > 1 \end{cases}$$
(9)

Where α and β are adjustable parameters and greater than zero, other parameters are the same as (8).

From (9), this approach is simply to calculate the threshold of the first level, and then employ the recursive method to calculate the other level wavelet threshold. α, β can be obtained by offline experimental analysis or experience. Such improved threshold method not only considers the propagation characteristics of the noise wavelet coefficients in the wavelet domain but also makes threshold value determination independent of the signal length. Due to calculating the median function only one time, recursive way further enhances the calculation speed.

D. Hyperbola thresholding function.

Soft-thresholding and hard-thresholding function are usually used in the wavelet threshold denoising. Hard-thresholding denoising method retains the high frequency detail coefficients whose absolute value is greater than the threshold, and set the detail coefficients whose absolute value is less than the threshold to be zero. The hard-thresholding function is given as follows

$$\tilde{d}_{i,j} = \begin{cases} d_{i,j} & |d_{i,j}| \geq \lambda_i \\ 0 & |d_{i,j}| < \lambda_i \end{cases} \qquad (10)$$

Where, $d_{i,j} = \{d_{i,1}, d_{i,2}, d_{i,3}, \cdots, d_{i,n}\}$ is the *ith* level detail coefficient, n is the data length of the *ith* level detail coefficient; λ_i is the *ith* level threshold value; $\tilde{d}_{i,j}$ is the *ith* level detail coefficient after threshold processing.

The soft-thresholding function is given as follows:

$$\tilde{d}_{i,j} = \begin{cases} d_{i,j} - \lambda_i & d_{i,j} \geq \lambda_i \\ 0 & |d_{i,j}| < \lambda_i \\ d_{i,j} + \lambda_i & d_{i,j} \leq -\lambda_i \end{cases} \qquad (11)$$

The hard-thresholding method can better retain local characteristics of the signal edge and is suitable for denoising of sudden change noise signal. But the hard-thresholding method presents oscillations in the vicinity of the function's discontinuities at $\pm\lambda$, such oscillations are called the pseudo-Gibbs phenomena, which will affect the visual quality of the denosing result.

Signal processed by the soft-thresholding method is relatively smooth, and to some extent reduces the pseudo-Gibbs phenomenon. However, there exists a permanent bias between $\tilde{d}_{i,j}$ and $d_{i,j}$, which will certainly affect the precision of signal reconstruction and give rise to the phenomenon of edge blur.

Consequently, this paper adopts a hyperbolic semisoft shrinkage function [12] which is given by

$$\tilde{d}_{i,j} = \begin{cases} sign(d_{i,j})\sqrt{(d_{i,j})^2 - \lambda_i^2} & |d_{i,j}| \geq \lambda_i \\ 0 & |d_{i,j}| < \lambda_i \end{cases} \qquad (12)$$

Where, $sign(\cdot)$ is Signum function, λ_i is the *ith* level threshold value obtained from (9). Hyperbolic semisoft shrinkage function overcomes the shortcomings of the hard-thresholding function and the soft-thresholding function. Three thresholding function curves are shown in Fig.7.
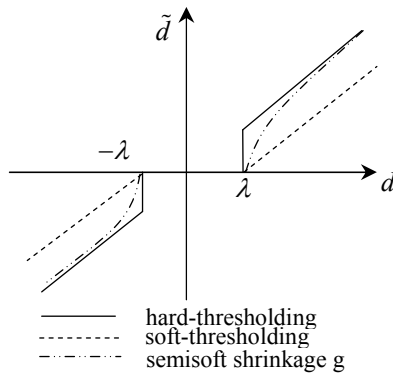

Fig.7. Thresholding function curve.

E. Boundary processing.

Wavelet transform algorithm is derived from the assumption that the signal length is infinite, but in practical applications signal is of finite length, so boundary point must be processed in the application of wavelet transform avoiding boundary distortion. One of the possible ways to eliminate boundary distortion is to extend the signal beyond the boundary. Traditionally, the finite length signal can be zero padded, but this expansive effect is undesirable in many applications. The application of periodic extension in multi-scale filter banks introduces artificial discontinuity in the extended signal. Currently, symmetric extension is a popular technique for handling finite length signal boundary extension in a non-expansive manner [13].

In this paper, symmetric extension is adopted. Suppose sampling signal sequence is $x(t), t = 1, 2, \cdots n$; $a(i)$ is the symmetric extension of $x(t)$; when $t \geq L$,

$$a(i) = \begin{cases} x(t - L + i), i = 1, 2, \cdots, L \\ x(t + L - i), i = L+1, L+2, \cdots, 2L \end{cases} \qquad (13)$$

Where, L is the width of moving data window. $\hat{a}(i)$ is the wavelet denoising result of $a(i)$, take signal $\hat{x}(t)$ as the denoising result at time t by (14).

$$\hat{x}(t) = \frac{\hat{a}(L) + \hat{a}(L+1)}{2} \qquad (14)$$

4. SIMULATION AND ANALYSIS

In order to facilitate the comparison of noise reduction effect, the following performance indices are introduced as the evaluation criteria: Root Mean Square Error (RMSE), Signal to Noise Ratio (SNR).

$$SNR = 10\lg\left\{\frac{\sum_{i=0}^{N-1} x_i^2}{\sum_{i=0}^{N-1}(x_i - \hat{x}_i)^2}\right\} \qquad (15)$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}(x_i - \hat{x}_i)^2} \qquad (16)$$

Where x_i is the original signal, $\hat{x}_i$ is the denoised signal.

In order to verify the validity of the improved semisoft shrinkage real-time denoising method based on lifting wavelet transform, improved semisoft shrinkage method, soft-thresholding method and hard-thresholding method are adopted in the simulation. In the simulation, we used three types of signal: Doppler, HeaviSine and Bumps, the definition of these signals is shown in Appendix B.

The system simulation parameters are set as follows:
1) Add Gaussian white noise whose mean is zero and variance is 0.01 to Doppler signal;

2) Add Gaussian white noise whose mean is zero and variance is 0.4 to HeaviSine signal;

3) Add Gaussian white noise whose mean is zero and variance is 0.2 to Bumps signal;

4) Sum of sample is 2048;

5) Width of moving data window is $n = 2^8 = 256$;

6) Wavelet decomposition scale is 4;

7) The db4 wavelet is selected as lifting base of the second-generation wavelet to implement db4 lifting wavelet transform;
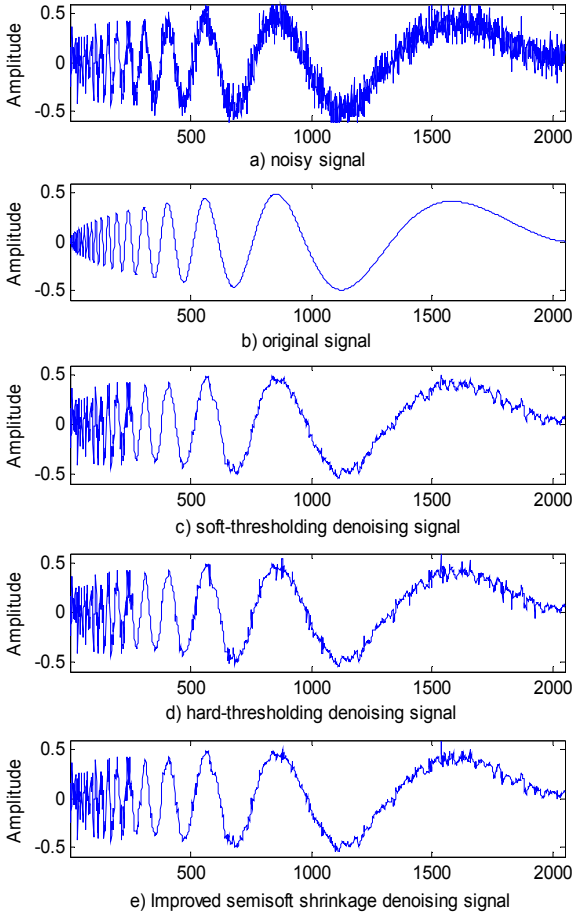
8) Parameters of (9) are $\alpha = 0.3$, $\beta = 1.2$.

Fig.8. Simulation denoising results for the Doppler signal.

Table 1. Comparison of different de-noising method in Doppler.

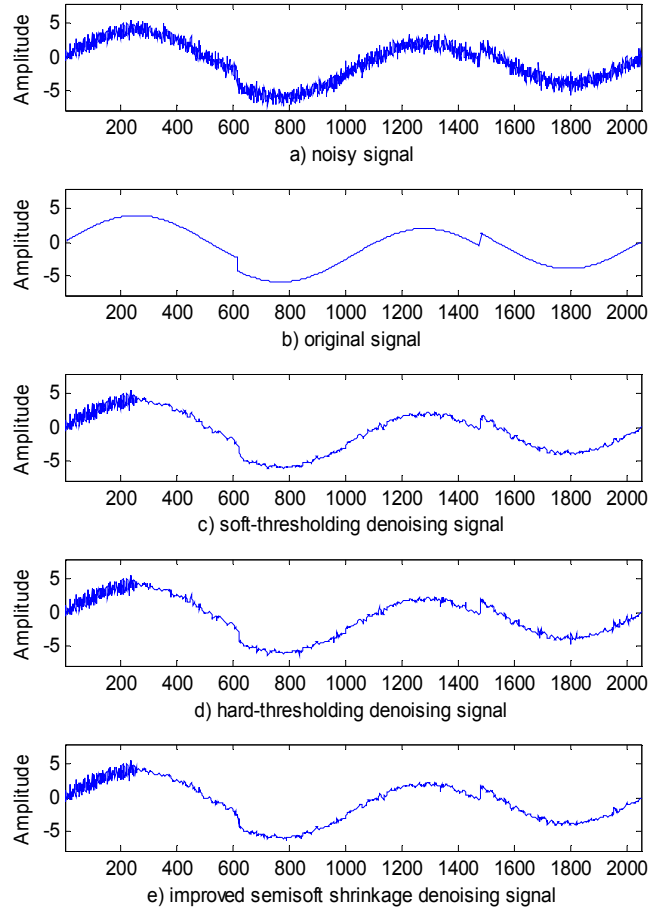|	Doppler signal	
	RMSE	SNR(dB)
No denoising	0.0992	9.7884
Soft thresholding	0.0387	17.9689
Hard thresholding	0.0434	16.9789
Improved semisoft shrinkage	0.0386	17.9997

Fig.9. Simulation denoising results for the Doppler signal.

Table 2. Comparison of different denoising method in HeaviSine

	HeaviSine signal	
	RMSE	SNR(dB)
No denoising	0.6506	13.6204
Soft thresholding	0.2397	22.2950
Hard thresholding	0.2568	21.6957
Improved semisoft shrinkage	0.2369	22.3955

Denoising simulation results for the Doppler signal are shown in Fig.8. We can see by the wavelet real-time denoising that noise has been significantly reduced. From the visual quality, the noise reduction effect of the soft-thresholding method is the best, followed by the improved semisoft shrinkage method, the hard-thresholding method is the worst. Real-time denoising performance of Doppler signal is shown in Table 1. From Table 1., we can see that RMSE, SNR of semisoft shrinkage method are optimal, followed by the soft-thresholding method whose performance criteria are almost the same as the semisoft shrinkage method, the hard-thresholding method is the worst.

Denoising simulation results for the HeaviSine signal are shown in Fig.9. From the visual quality, the noise reduction effect of the soft-thresholding method is the best, followed by improved semisoft shrinkage method, the hard-thresholding method is the worst. Real-time denoising performance of HeaviSine signal is shown in Table 2. From Table 2., we can see that RMSE, SNR of semisoft shrinkage method are optimal, followed by the soft-thresholding method, the hard-thresholding method is the worst.
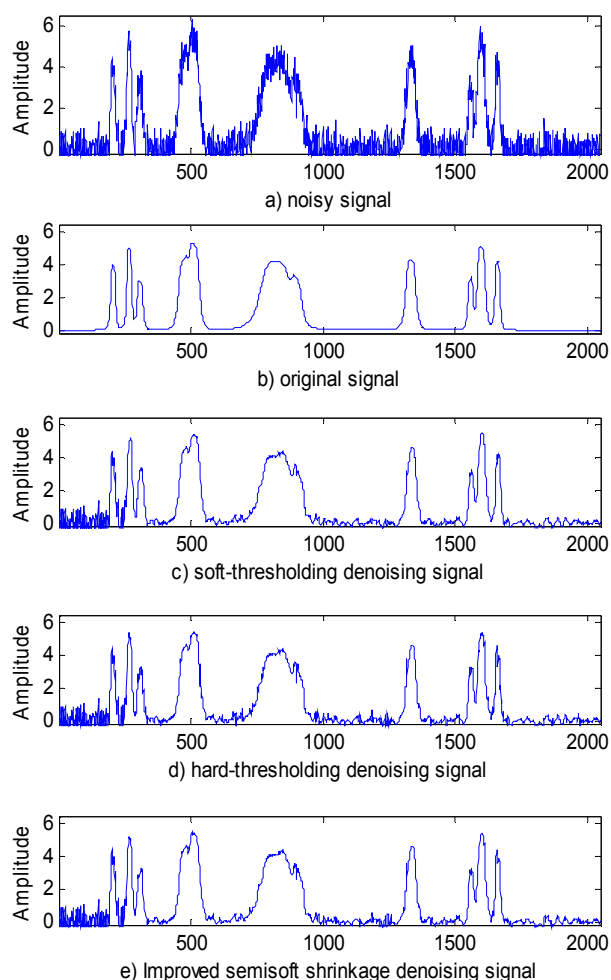

Fig.10. Simulation denoising results for the Bumps signal.

Table 3. Comparison of different denoising method.

	Bumps signal	
	RMSE	SNR(dB)
No denoising	0.4406	12.6139
Soft thresholding	0.2870	16.3368
Hard thresholding	0.2327	18.1565
Improved semisoft shrinkage	0.2231	18.5242

Denoising simulation results for the Bumps signal are shown in Fig.10. From the visual quality, the noise reduction effect of the soft-thresholding method is the best, followed by improved semisoft shrinkage method, the hard-thresholding method is the worst. Real-time denoising performance of Bumps signal is shown in Table 3. From Table 3., we can see that RMSE, SNR of semisoft shrinkage method are optimal, followed by the hard-thresholding method, the soft-thresholding method is the worst.

Through analysis of real-time noise reduction effect of Bumps, Doppler and HeaviSine signal, we can find:

1) Only from the visual quality, the denoising effect of the soft-thresholding method is the best, but the performance criteria may not be the best, and may actually be the worst.

2) For signals whose continuity is better, such as Doppler and HeaviSine signal, the soft-thresholding method is significantly better than the hard-thresholding method, regardless of the visual quality or the performance criteria.

3) For signals whose continuity is poor, such as Bumps signal, although denoising effect of the soft-thresholding method is superior to the hard-thresholding method from visual quality, the hard-thresholding method actually is obviously superior to the soft-thresholding method from the comparison of quantitative performance criteria.

4) Whether the continuity of actual signal is good or bad, the denoising effect of improved semisoft shrinkage method is the best from performance criteria, and visual quality is suboptimal. Considering both the performance criteria and visual quality, the improved semisoft shrinkage method proposed is better than the soft-thresholding and the hard-thresholding method, and has a wider application scope.

5. CONCLUSIONS

Wavelet threshold denoising method is one of the primary methods of wavelet denoising. For applying wavelet threshold denoising method in real-time noise reduction application, an improved semisoft shrinkage real-time denoising method based on lifting wavelet transform was presented. This method, which adopts moving data window technology, makes real-time wavelet denoising processing possible. Wavelet transform based on lifting implementation greatly enhances the computation speed. The hyperbolic threshold function and recursive method to calculate every level wavelet threshold better ensure the dynamic behavior of the system and further enhance computation efficiency. At last, this method is applied to denoising simulation for Doppler, Bumps and HeaviSine signal, the simulation results prove the effectiveness of this proposed method.

APPENDIX

A. Lifting implementation algorithm.

The lifting implementation algorithm of wavelet transform is written in Table 4.

Table 4. Algorithm implementation of wavelet lifting transform.

Algorithm 1: The lifting implementation algorithm of forward wavelet transform

Step1: Lazy wavelet transform

$$e_l^0 = x_{2l}$$
$$o_l^0 = x_{2l+1} \quad, l = 0,1,\cdots, N/2-1$$

Step2: Lifting and dual lifting

For $i = 1$ to m

$$\begin{cases} e_l^i = e_l^{i-1} - \sum_k u_k^i d_{l-k}^{i-1} \\ o_l^i = o_l^{i-1} - \sum_k p_k^i e_{l-k}^i, \qquad l = 0,1,\cdots, N/2-1 \end{cases}$$

Step3: Scale correction

$$\begin{cases} a_l = e_l^m / K \\ d_l = K o_l^m \end{cases}$$

a_l, d_l denote as the lth level wavelet approximation and detail coefficients.

Algorithm 2: The lifting implementation algorithm of inverse wavelet transform

Step1: Scale correction

$$\begin{cases} e_l^m = K a_l \\ o_l^m = d_l / K \end{cases}$$

Step2: Lifting and dual lifting

For $i = m$ to 1

$$\begin{cases} o_l^{i-1} = o_l^i + \sum_k p_k^i e_{l-k}^i \\ e_l^{i-1} = e_l^i + \sum_k u_k^i o_{l-k}^{i-1}, \qquad l = 0,1,\cdots, N/2-1 \end{cases}$$

Step3: Inverse lazy wavelet transform

$$x_{2l} = e_l^0$$
$$x_{2l+1} = o_l^0 \quad, l = 0,1,\cdots, N/2-1$$

B. *Formulas for test signals.*

Doppler:

$$f(t) = \sqrt{t(t-1)} \sin(2\pi(1+\varepsilon)/(t+\varepsilon)), \varepsilon = 0.05$$

HeaviSine:

$$f(t) = 4\sin 4\pi t - sign(t-0.3) - sign(0.72-t)$$

Bumps:

$$f(t) = \sum h_j K((t-t_j)/\omega_j) \qquad K(t) = (1+|t|^4)^{-1}$$
$$(t_j) = (0.1, 0.13, 0.15, 0.23, 0.25, 0.4, 0.44, 0.65, 0.76, 0.78, 0.81)$$
$$(h_j) = (4, -5, 3, -4, 5, -4.2, 2.1, 4.3, -3.1, 5.1, -4.2)$$
$$(\omega_j) = 0.01*(0.5, 0.5, 0.6, 1, 1, 3, 1, 1, 0.5, 0.8, 0.5)$$

REFERENCES

[1] Mallat, S.G. (1989). A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11 (7), 674-693.

[2] Xu, Y., Weaver, J.B., Healy, D.M., Lu, J. (1994). Wavelet transform domain filters: Aspatially selective noise filtration technique. *IEEE Transactions on Image Processing*, 3 (6), 747-758.

[3] Mallat, S.G., Hwang, W.L. (1992). Singularity detection and processing with wavelet. *IEEE Transactions on Information Theory*, 38 (2), 617-643.

[4] Donoho, D.L. (1995). De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41 (3), 613-627.

[5] Sweldens, W. (1996). The lifting scheme: A custom-design construction of biorthogonal wavelet. *Applied and Computational Harmonic Analysis*, 3 (2), 186-200.

[6] Sweldens, W. (1998). The Lifting Scheme: A construction of second generation wavelet. *SIAM Journal of Mathematical Analysis,* 29 (2), 511-546.

[7] Daubechies, I., Sweldens, W. (1998). Factoring wavelet transform into lifting steps. *Journal of Fourier Analysis and Application*, 4 (3), 245-267.

[8] Liu, Y.H., Huang, X.S. (2009). Lifting wavelet de-noising for SINS alignment. *Journal of System Simulation*, 21 (19), 6013-6016.

[9] Abid, M., Cagnazzo, M., Pesquet-Popescu, B. (2010). Image de-noising by adaptive lifting schemes. In *2nd European Workshop on Visual Information Processing*, 108-113.

[10] Xia, R., Meng, K., Qian, F., Wang, Z.L. (2007). Online wavelet denoising via a moving window. *Acta Automatica Sinica*, 33 (9), 897-901.

[11] Lv, P., Lai, J.Z., Ling, D. (2009). Research on threshold algorithm in real-time wavelet de-noising of FOG. *Journal of Projectiles, Rockets, Missiles and Guidance*, 29 (1), 18-22.

[12] Wang, F.Y., Liu, G., Yuan, G.N. (2009). Wavelet threshold-based sea clutter signal denoising. *Radar & ECM*, 2, 21-26.

[13] Bamberger, R.H., Eddins, S.L., Nuri, V. (1994). Generalized symmetric extension for size-limited multirate filter banks. *IEEE Transactions on Image Processing*, 3 (1), 82-87.