Thresholding Neural Network for Adaptive Noise Reduction

Xiao-Ping Zhang, *Member, IEEE*

***Abstract*—In this paper, a type of thresholding neural network (TNN) is developed for adaptive noise reduction. New types of soft and hard thresholding functions are created to serve as the activation function of the TNN. Unlike the standard thresholding functions, the new thresholding functions are infinitely differentiable. By using the new thresholding functions, some gradient-based learning algorithms become possible or more effective. The optimal solution of the TNN in a mean square error (MSE) sense is discussed. It is proved that there is at most one optimal solution for the soft-thresholding TNN. General optimal performances of both soft and hard thresholding TNNs are analyzed and compared to the linear noise reduction method. Gradient-based adaptive learning algorithms are presented to seek the optimal solution for noise reduction. The algorithms include supervised and unsupervised batch learning as well as supervised and unsupervised stochastic learning. It is indicated that the TNN with the stochastic learning algorithm scan be used as a novel nonlinear adaptive filter. It is proved that the stochastic learning algorithm is convergent in certain statistical sense in ideal conditions. Numerical results show that the TNN is very effective in finding the optimal solutions of thresholding methods in an MSE sense and usually outperforms other noise reduction methods. Especially, it is shown that the TNN-based nonlinear adaptive filtering outperforms the conventional linear adaptive filtering in both optimal solution and learning performance.**

***Index Terms*—Adaptive noise reduction, image denoising, mean square error (MSE), optimal estimation, thresholding, thresholding neural network (TNN), wavelet transforms.**

I. Introduction

NOISE reduction is a traditional problem in signal processing as well as many applications in the real world. Conventional linear system adaptive filtering techniques have been widely used in adaptive noise reduction problems. However, because of the linearity of the operation, the filter cannot change the intrinsic property of the original noised signal, such as regularity, etc., Indeed, the linear filter is a kind of linear manipulation of the spectrum of a signal because the complex exponential function $e^{j\omega}$ is the eigenfunction of a linear system. Therefore, it is difficult to suppress the noise and keep the signal using linear filters when the spectrum of a signal is somewhat wideband and nonstationary, which is the usual case. For example, some transient impulses can cause wideband components in the signal. The linear filter tends to eliminate or keep both noise and this type of important component because both of them may have similar appearance in spectrum. Also, the FIR (finite impulse response) filter based noise reduction techniques in the transform domain have been investigated [1], [2]. However, since the transformation used is usually linear, the overall filtering is equivalent to a linear filter. The convergence speed of the adaptive linear filter may be improved in the transform domain, however, the optimal noise reduction performance is the same as the conventional time domain linear filtering.

Recently, wavelet thresholding methods proved to be powerful tools for denoising problems [3]–[12]. The main purpose of these methods is to estimate a wide class of functions in some smoothness spaces, such as Besov space and Triebel space, etc., from their corrupted (by additive Gaussian noise) versions. The main wavelet thresholding scheme is the soft-thresholding [5]. This technique is effective because the energy of a function with some smoothness is often concentrated on few coefficients while the energy of noise is still spread in all coefficients in the wavelet domain. The Donoho's wavelet soft-thresholding method achieves asymptotically near optimal in the meaning of minimax mean square error (MSE) over a wide range set of functions with certain smoothness. However, it often tends to oversmooth the function and thus remove some important high-frequency components. In many signal processing applications, we need to search for the optimal minimum MSE solution using *a priori* information for a specific signal. The optimal minimax solution often has only theoretical meaning in such cases because it may be far from the optimal solution for a specific practical problem. Another natural thresholding scheme called the hard-thresholding has also been tested and reported to have better MSE performance in some simulations [7].

The questions are: 1) What are the optimal solutions of thresholding methods in an MSE sense? i.e., what is the best achievable noise reduction performance of the thresholding methods? 2) In what situation, can the thresholding noise reduction methods perform better than linear filter-based methods? 3) How can the optimal solutions of thresholding methods be achieved in real applications? It is noted that all the current thresholding methods cannot completely adapt to the optimal solution for a given specific signal.

In this paper, a new type of thresholding neural network (TNN) for noise reduction in various applications is developed. New types of smooth soft-thresholding and hard-thresholding activation functions are presented. They make many gradient-based learning algorithms feasible. The optimal solution of soft-thresholding methods in an MSE sense is discussed. We prove that there is at most one optimal solution for soft-thresholding in an MSE sense. The optimal solutions of different

Manuscript received October 11, 1999; revised September 21, 2000 and February 14, 2001.

The author is with the Department of Electrical and Computer Engineering, Ryerson Polytechnic University, Toronto, ON M5B 2K3, Canada (e-mail: xpzhang@ieee.org).

Publisher Item Identifier S 1045-9227(01)03572-X.

methods are investigated. Subsequently, the gradient-based learning algorithms of TNNs are presented to seek the optimal solution in various situations and applications. It is also shown that the presented TNN can be used in real-time time-scale or time-frequency adaptive noise reduction. Several numerical examples are given. The results show that the presented TNN and its learning algorithms are very effective in finding the optimal solutions of thresholding methods in an MSE sense in various noise reduction applications.

This paper is organized as follows. Section II reviews the basic concepts and results of thresholding methods. In Section III, the thresholding neural network (TNN) and new types of thresholding functions are presented. The optimal solutions of the thresholding methods are discussed in Section IV. The learning algorithms for different applications are introduced in Section V. Section VI presents several numerical examples to demonstrate our methods. Finally, Section VII concludes the paper.

II. Soft-Thresholding and Hard-Thresholding

A. Noise Reduction Problem and Thresholding Methods

The general noise reduction problem can be formulated as follows. Assuming the real signal is x and the observed signal $y = x + n$, where n is the noise, we can obtain an estimate $\hat{x} = f(y)$ of the real signal x from the observed signal y. The objective of noise reduction is to reduce the noise in y and make the estimate $\hat{x}$ as close to x as possible. The commonly used criteria to measure the closeness is the error energy $E||\hat{x} - x||^2$, i.e., mean square error (MSE). Note that in the above formulation the signal x can be a finite data sample set, or an infinite data sample set generated by a real-time stochastic process. In the later case, a real-time adaptive estimation method may be necessary.

For simplification of analysis, first we consider the former case, i.e., the signal is a finite data sample set, denoted by a vector $\boldsymbol{x} = [x_0, x_1, \ldots, x_{N-1}]^{\mathrm{T}}$, and then the observed signal is

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

i.e.,

$$y_i = x_i + n_i, \quad i = 0, \ldots, N-1 \tag{2}$$

where $\boldsymbol{n}$ is a noise data vector. Gaussian white noise with distribution $N(0, \sigma^2)$ is commonly assumed for n. In the following discussion, we will first stick to these simplifications and then show that our results can be generalized to the general stochastic process.

The objective of noise reduction is to estimate the real signal $\boldsymbol{x}$ from $\boldsymbol{y}$ to minimize the MSE risk

$$J(\hat{x}, x) = \frac{1}{2} E||\hat{\boldsymbol{x}} - \boldsymbol{x}||^2 = \frac{1}{2} \sum_{i=0}^{N-1} E(\hat{x}_i - x_i)^2 \tag{3}$$

where $\hat{x} = f(y)$.

For linear noise reduction, the estimate $\hat{x}$ is a linear combination of observed data samples y_i. The estimation operator $f(\cdot)$ is linear. For nonlinear noise reduction, the operator $f(\cdot)$ is nonlinear. The thresholding methods use nonlinear operations to reduce the noise. The most commonly used thresholding functions are the soft-thresholding function and the hard-thresholding function. The standard soft-thresholding function is defined as

$$\eta_s(x,t) \triangleq \operatorname{sgn}(x)(|x| - t)_+ = \begin{cases} x + t, & x < -t \\ 0, & |x| \le t \\ x - t, & x > t \end{cases} \tag{4}$$

where $t \ge 0$ is the threshold. And the standard hard-thresholding function is defined as

$$\eta_h(x,t) \triangleq \begin{cases} x, & |x| > t \\ 0, & |x| \le t \end{cases} \tag{5}$$

These two functions are shown in Fig. 1.

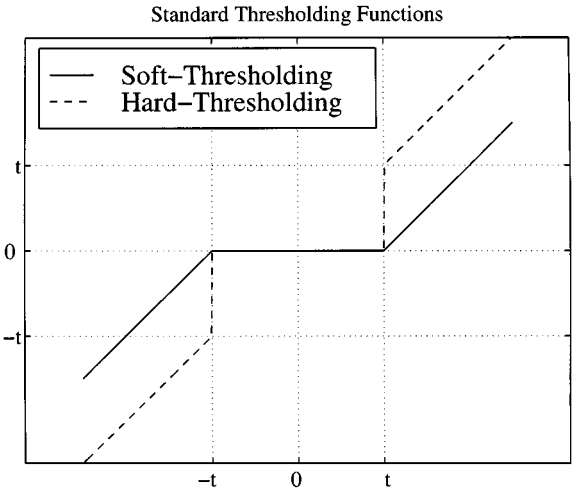

Fig. 1. The standard soft and hard thresholding functions.

In thresholding methods, the observed data samples with smaller (than t) values are suppressed and the samples with larger (than t) values are kept. Therefore, when the smaller samples are dominated by noise components, the thresholding operation can suppress the noise in observed data samples.

B. Wavelet Thresholding

Recently, soft-thresholding in the wavelet transform domain has been studied in statistical estimation problems and proved to have many good mathematical properties [5]. There are three steps in the standard wavelet thresholding method [5].

1) Apply the discrete wavelet transform (DWT) to the observed data vector $\boldsymbol{y}$ and obtain the empirical wavelet coefficients.
2) Apply the nonlinearity—soft-thresholding to the empirical wavelet coefficients, where a universal threshold $t = \sigma\sqrt{2\log(N)/N}$ is chosen.
3) Use the inverse DWT on thresholded wavelet coefficients and obtain the estimated function vector $\hat{\boldsymbol{x}}$.

The basic idea of the wavelet thresholding method is that the energy of a signal (with some smoothness) is often concentrated on few coefficients while the energy of noise is spread among all coefficients in the wavelet domain. Therefore, the nonlinear soft-thresholding function tend to maintain few larger coefficients representing the signal while reducing the

noise coefficients to zero in the wavelet domain. It is proved [5] that the above soft-thresholding based wavelet thresholding denoising method achieves a "noise free" property (i.e., the estimated signal is at least as smooth as the original signal) and is asymptotically near optimal in the meaning of minimax MSE over a wide range of smoothness classes. A universal threshold is intuitively expected to uniformly remove the noise since the white noise still has the same variance over different scales in the transform domain.

Scale-dependent thresholds can be used in Step 2) of the wavelet thresholding scheme so that the denoising result can adapt to the local smoothness of the function. A scale-dependent threshold selection procedure called the *SureShrink* [3] is proposed based on *Stein's Unbiased Risk Estimate* (SURE). It proved to be smoothness-adaptive in a near minimax sense. However, *SureShrink* no longer has a "noise free" property since SURE risk is just an estimation of the MSE.

Note that the MSE is the most commonly used criteria in signal processing applications. In wavelet thresholding research for function estimation, it is also often used as the criteria to select the threshold [6], [7]. *SureShrink* approach uses MSE as the risk of estimation. Therefore, in this paper, we use the MSE to evaluate the noise reduction performance.

From the basic idea of the thresholding method, we can reasonably infer that the thresholding method can reduce noise in a transform domain, as long as the transform can concentrate the signal energy but spread the noise energy in the transform domain.

III. Thresholding Neural Network

A. Neural Network Structure

We construct a type of thresholding neural network (TNN) to perform the thresholding in the transform domain to achieve noise reduction. The neural network structure of the TNN is shown in Fig. 2.

The transform in TNNs can be any linear orthogonal transform. The linear transform performed on observed data samples can change the energy distribution of signal and noise samples. By thresholding, the signal energy may be kept while the noise is suppressed. For a specific class of signal, the appropriate linear transforms may be selected to concentrate signal energy versus noise, and then a good MSE performance can be achieved. Here the thresholding functions are employed as nonlinear activation functions of the neural network. The inverse transform is employed to recover the signal from the noise-reduced coefficients in the transform domain. Specifically, since most signals have some kinds of regularities and the wavelet transform is a very good tool to efficiently represent such characteristics of the signal, the wavelet transform is often a suitable linear transform in TNNs.

Note that there are several orthogonal channels in the transform domain. We denote the set of coefficients at channel j as I_j. The different thresholds t_j are used in different orthogonal channels and they are independent, i.e., the thresholds of different orthogonal channels can be optimized independently.

It is also worth pointing out that although the term "neural network" is used, the TNN is different from the conventional multilayer neural network. In TNNs, a fixed linear transform is used and the nonlinear activation function is adaptive, while in conventional multilayer neural networks, the activation function is fixed and the weights of the linear connection of input signal samples are adaptive. We use the term "neural network" because the TNN has some basic elements similar to a conventional neural network, i.e., interconnection of input signal samples, nonlinear activation functions, and adaptivity to a specific input, etc. In addition, it is possible to change the fixed linear transform in Fig. 2 to an adaptive linear transform. In this way, both the weights of linear connections of input signal samples and the nonlinear activation function are adaptive, and then the conventional multilayer neural network techniques may be incorporated. This will be a meaningful exploration we are going to pursue in the future.

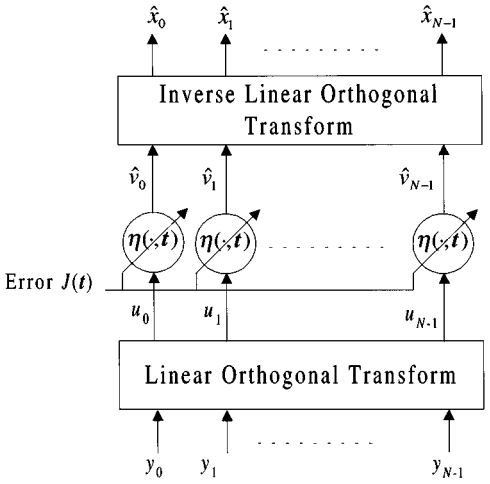

Fig. 2. The structure of thresholding neural network (TNN).

B. Newsoft and Hard Thresholding Functions

It is well known that it is very important for a neural network to have good learning algorithms [13]. Most learning algorithms of a neural network employ the gradients and higher derivatives of the network activation function. In addition, high-order differentiable activation functions make a neural network have better numerical properties. It is desired that the activation function of a neural network can be differentiated and has high-order derivatives so that the gradient-based learning algorithms can be developed. However, the standard soft-thresholding function is only weakly differentiated and does not have any high-order derivative. The standard hard-thresholding function is a discontinuous function and cannot be differentiated at all. The author's previous work presented a new type of soft-thresholding function which has second order weak derivatives and proved to be useful [14]. In the following, we will present new types of smooth soft-thresholding and hard-thresholding functions which are infinitely differentiable.

The new type of soft-thresholding function is constructed as follows:

$$\eta_{st}(x,t) = x + \frac{1}{2}\left(\sqrt{(x-t)^2+\lambda} - \sqrt{(x+t)^2+\lambda}\right) \quad (6)$$

where t is the threshold and $\lambda > 0$ is a user-defined (fixed) function parameter. Obviously, the soft-thresholding functions

$\eta_{st}(x,t)$ have all higher order derivatives for $\lambda > 0$. Note that when $\lambda = 0$, $\eta_{st}(x,t)$ is just the standard soft-thresholding function $\eta_s(x,t)$. The new thresholding functions with different parameter λ are shown in Fig. 3(a). It can be seen that the new thresholding functions perform the operations similar to the standard soft-thresholding function. Therefore, similar thresholding effects of the estimate using the new thresholding functions can be expected.

The new type of hard-thresholding function is motivated by the sigmoid function [13]. It is constructed as follows:

$$\eta_{ht}(x,t) = \left(\frac{1}{1+\exp\left\{\frac{-x+t}{\mu}\right\}} - \frac{1}{1+\exp\left\{\frac{-x-t}{\mu}\right\}} + 1 \right) x \tag{7}$$

where t is the threshold and $\mu > 0$ is a user-defined (fixed) function parameter. It is also easy to see that the new hard-thresholding functions have all higher order derivatives for $\mu > 0$. When $\mu \to 0$, $\eta_{ht}(x,t)$ is just the standard hard-thresholding function $\eta_h(x,t)$, i.e., $\lim_{\mu \to 0} \eta_{ht}(x,t) = \eta_h(x,t)$. The new hard-thresholding functions with different parameter μ are shown inFig. 3(b). It can be seen that the new hard-thresholding functions also have the thresholding effect similar to the standard hard-thresholding function. However, by using new hard-thresholding functions $\eta_{ht}(x,t)$, it becomes possible to construct a gradient-based learning algorithm for the TNN. Furthermore, the SURE risk, which employs the second derivatives of the estimate, can be utilized in unsupervised learning of the TNN by using the new type of hard-thresholding function. These will be shown in Section V-B1.

When the gradient of the thresholding functions with respect to the threshold t is employed in the learning algorithm, new soft-thresholding functions have better adjustability since they have nonzero derivatives for all t. The standard soft-thresholding function does not have adjustability when $|x| < t$ since it has zero function values and derivatives for $|x| < t$. This is one reason why the new type of thresholding function has better numerical properties than the standard thresholding function in adaptive learning processes.

Apparently, the larger are the function parameters λ and μ in (6) and (7), the more adjustability the new thresholding functions have, since they will have larger derivatives when $|x| < t$. However, the thresholding ability of the new functions decreases when λ and μ are too large. Actually, when $\lambda, \mu \to \infty$, the new thresholding functions become a linear function and they have no thresholding ability at all. We suggest selecting a small λ and μ so that the new thresholding functions are good approximations of the standard thresholding functions and practically keep all good properties of standard thresholding techniques.

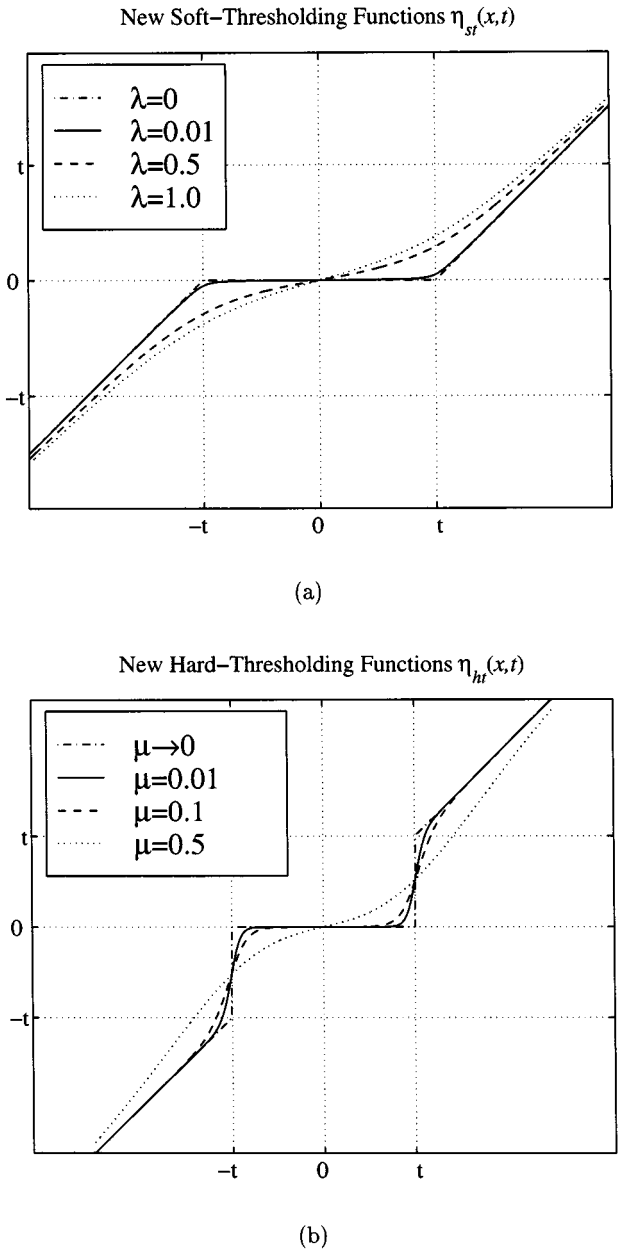

Fig. 3. New thresholding functions. (a) New soft-thresholding functions. (b) New hard-thresholding functions.

IV. On Optimal Performance of the Thresholding Neural Network

For any method dealing with the noise reduction problem, we want to get the bottom line of its performance. That is, we want to ask the questions: What is the best performance of this method? What are the properties of the optimal solution of the method? How can the optimal solution of the method be achieved? These are natural questions when we evaluate a method since the best performance defines all the potential of the method, and it is our objective to achieve the best performance of the method in practice. In this section, the optimal solution of the TNN and its properties will be discussed. The learning algorithms of the TNN to achieve the optimal solution will be presented in the next section.

Since the orthogonal linear transform used in the TNN preserves the signal energy, the MSE of the estimation in the transform domain is equal to the MSE of the estimation in the time domain. Furthermore, since the thresholds of different orthogonal channels are independent and the thresholds of different orthogonal channels can be optimized independently, we will only analyze the optimal solution of one channel in the transform domain in the following, without loss of generality.

A. The Optimal Solution of the Soft-Thresholding

In many nonlinear optimization problems, such as training of an artificial neural network, a very troublesome issue is that there may be more than one local optimum. This often makes it difficult to find the global optimal solution of the problem. However, in the following, we will prove that when using the standard soft-thresholding function as the activation function, the TNN has only global optimal solution.

Theorem 1: Suppose x_i are signal samples, n_i are Gaussian white noise samples with i.i.d. distribution $N(0,\sigma^2)$ and the observed noise corrupted signal samples are $y_i = x_i + n_i$, $i = 0,\ldots,N-1$. Let $\hat{x}_i$ be the noise reduction output using the standard nonlinear soft-thresholding method, i.e.,

$$\hat{x}_i = \eta_s(y_i, t) \tag{8}$$

where $t \geq 0$. Define the risk function to be the MSE, i.e.,

$$J(t) = \frac{1}{2}E\left\{\sum_i \varepsilon_i^2\right\} \tag{9}$$

with $\varepsilon_i = \hat{x}_i - x_i$, and denote the optimal solution

$$t^* = \arg\min_t J(t). \tag{10}$$

There exists at most one optimal solution $t^* \geq 0$ and $t^* = 0$ if and only if the noise level $\sigma = 0$.

See Appendix A for proof.

Remarks:

1) Theorem 1 shows that if there is an optimal threshold solution for MSE risk $J(t)$, then it is unique. There is no local minima problem when the gradient-based optimization algorithms are used.
2) The results in Theorem 1 hold for the thresholding of the data at one channel in the transform domain of the TNN. Since the threshold parameters at different channels in the transform domain are adjusted independently as we mentioned before, Theorem 1 also holds for the whole TNN.
3) Note that the conclusion in Theorem 1 is not a trivial result of convexity. Although $\eta_s(x,t)$ is convex with respect to $t > 0$, the risk function $J(t)$ is not convex. The properties of $J(t)$ depend on the distribution of the noise, i.e., the conclusion does not hold for any distribution. It is still an open question to verify the validity of the Theorem 1 for some other commonly used noise distributions.
4) The conclusion in Theorem 1 is derived from the standard piecewise soft-thresholding function. Since the new soft-thresholding functions in (6) have properties similar to the standard soft-thresholding function, it is reasonable to expect that the conclusion in Theorem 1 also holds for the new type of soft-thresholding function in (6).
5) This conclusion does not hold for hard-thresholding activation functions. It is easy to find a numerical counter-example to verify this.

B. Soft-Thresholding, Hard-Thresholding, and Linear Filtering

Unlike linear filtering, it is not tractable to find the analytic optimal solution of the nonlinear thresholding method, as we can see from the proof of Theorem 1. In the following, with the help of numerical analysis, the optimal performances of various thresholding methods are analyzed and compared.

As can be seen, for both thresholding methods, $J(t) = \sum_i J(t|x_i)$. For soft-thresholding, $J_s(t) = \sum_i J_s(t|x_i)$ (here we will use subscript s to represent different activation functions), where

$$\begin{aligned}
J_s(t|x) =& \frac{1}{2}E\left\{\varepsilon_i^2|x\right\} \\
=& \frac{1}{2}\Big[\int_{t-x}^{\infty}(\xi - t)^2 p_n(\xi)d\xi \\
&+ x^2 \cdot \int_{-t-x}^{t-x} p_n(\xi)d\xi \\
&+ \int_{-\infty}^{-t-x}(\xi + t)^2 p_n(\xi)d\xi\Big] \\
=& \frac{1}{2}\Bigg\{\sigma^2 + t^2 + (x^2 - \sigma^2 - t^2) \\
&\times [F(t-x) - F(-t-x)] + \frac{\sigma}{\sqrt{2\pi}} \\
&\times \left[(t+x)\exp\left[-\frac{(t+x)^2}{2\sigma^2}\right]\right. \\
&\left. + (t-x)\exp\left[-\frac{(t-x)^2}{2\sigma^2}\right]\right] \\
&- \sqrt{\frac{2}{\pi}}\sigma t\left[\exp\left[-\frac{(t+x)^2}{2\sigma^2}\right]\right. \\
&\left.\left. + \exp\left[-\frac{(t-x)^2}{2\sigma^2}\right]\right]\right\}
\end{aligned} \tag{11}$$

where $F(x) \triangleq \int_{-\infty}^{x} p_n(\xi)d\xi$ is the distribution function of the additive Gaussian noise n.

Similarly, for hard-thresholding activation function (5), $J_h(t) = \sum_i J_h(t|x_i)$, where

$$\begin{aligned}
J_h(t|x) =& \frac{1}{2}E\left\{\varepsilon_i^2|x\right\} \\
=& \frac{1}{2}\Big[\int_{t-x}^{\infty}\xi^2 p_n(\xi)d\xi + \cdot \int_{-t-x_i}^{t-x_i}\xi^2 p_n(\xi)d\xi \\
&+ \int_{-\infty}^{-t-x_i}\xi^2 p_n(\xi)d\xi\Big] \\
=& \frac{1}{2}\Bigg\{\sigma^2 + (x^2 - \sigma^2)[F(t-x) - F(-t-x)] \\
&+ \frac{\sigma}{\sqrt{2\pi}}\left[(t+x)\exp\left[-\frac{(t+x)^2}{2\sigma^2}\right]\right. \\
&\left.\left. + (t-x)\exp\left[-\frac{(t-x)^2}{2\sigma^2}\right]\right]\right\}.
\end{aligned} \tag{12}$$

Then the MSE risk difference between soft-thresholding and hard-thresholding is

$$\begin{aligned}
\Delta J_{sh}(t|x) =& J_s(t|x) - J_h(t|x) \\
=& \frac{1}{2}\Bigg\{t^2[F(-t+x) - F(-t-x)] - \sqrt{\frac{2}{\pi}}\sigma t \\
&\times \left[\exp\left[-\frac{(t+x)^2}{2\sigma^2}\right] + \exp\left[-\frac{(t-x)^2}{2\sigma^2}\right]\right]\Bigg\}.
\end{aligned} \tag{13}$$

Apparently, the MSE risks of soft-thresholding and hard-thresholding depend on the signal energy distribution and the signal-to-noise-ratio (SNR). There is no simple relationship between them due to the nonlinearity. Therefore, the general properties of these risks are investigated using numerical methods as follows. Without loss of generality, we assume the noise level $\sigma^2 = 1$ (i.e., let the signal and threshold level be normalized by the SNR). The soft-thresholding risk (11), hard-thresholding risk (12) and their difference (13) are shown in Fig. 4(a)–(c), respectively. The grayscale in the figures is normalized according to the MSE risk. In Fig. 4(c), the absolute value of the risk difference is shown and the positive, negative and zero regions are indicated. From the figures we can see that when the threshold t is much larger than x and n, both soft-thresholding and hard-thresholding threshold all data samples to zero, which corresponds to the zero zone inFig. 4(c). When the signal x is relatively larger than the threshold t and noise n, the hard-thresholding will have better MSE performance, which corresponds to the positive zone in Fig. 4(c). Generally, the MSE of the estimate of large signal samples is dominant in whole MSE risk. This can justify the reported results [7] that in most cases the hard-thresholding gives better performance than soft-thresholding, as far as the optimal solution of the method is concerned.

Note that when the signal power is comparable with the noise, the soft-thresholding method seems to give better MSE risk. However, in such a case both thresholding methods tend to suppress both signal and noise. In Fig. 5, the optimal thresholds t for conditional MSE risks (11) and (12) are shown. When the signal power is comparable with the noise power, the optimal t is large and tends to threshold both signal and noise to zero. Practically, there is no useful signal after such thresholding.

Ho wever, since the hard-thresholding MSE risk function usually has many local minima, it is difficult to find the global optimal threshold for hard-thresholding methods. In addition, since the standard hard-thresholding function is discontinuous, it is even difficult to find a local minimum for it. In wavelet thresholding methods [5], it is also proved that the soft-thresholding method may keep better smoothness of the true signal than the hard-thresholding method.

Since the nonlinear thresholding methods are used to replace the linear filter for noise reduction applications, it is of interest to compare them with a linear operation on the signal in terms of the optimal MSE risk. For comparison, here we consider the simplest linear operation for noise reduction, i.e.,

$$\hat{x} = ay \tag{14}$$

where a is the linear parameter. The conditional MSE risk $J_l(a|x)$ can be written as

$$J_l(a|x) = \frac{1}{2}E(ay - x)^2. \tag{15}$$

It is easy to obtain that the optimal parameter a for $J_l(a|x)$ is $a_{opt} = x^2/(\sigma^2 + x^2)$ and the optimal conditional MSE risk is $J_{l,opt}(a|x) = \sigma^2 x^2/2[(\sigma^2 + x^2)]$.

The optimal conditional MSE risk of the above linear operator is plotted in Fig. 6. For comparison, the numerical optimal MSE risks of (11) and (12) are also shown in Fig. 6. The noise level is still normalized. It can be seen that the optimal conditional MSE risk of this simple linear operator is better than both

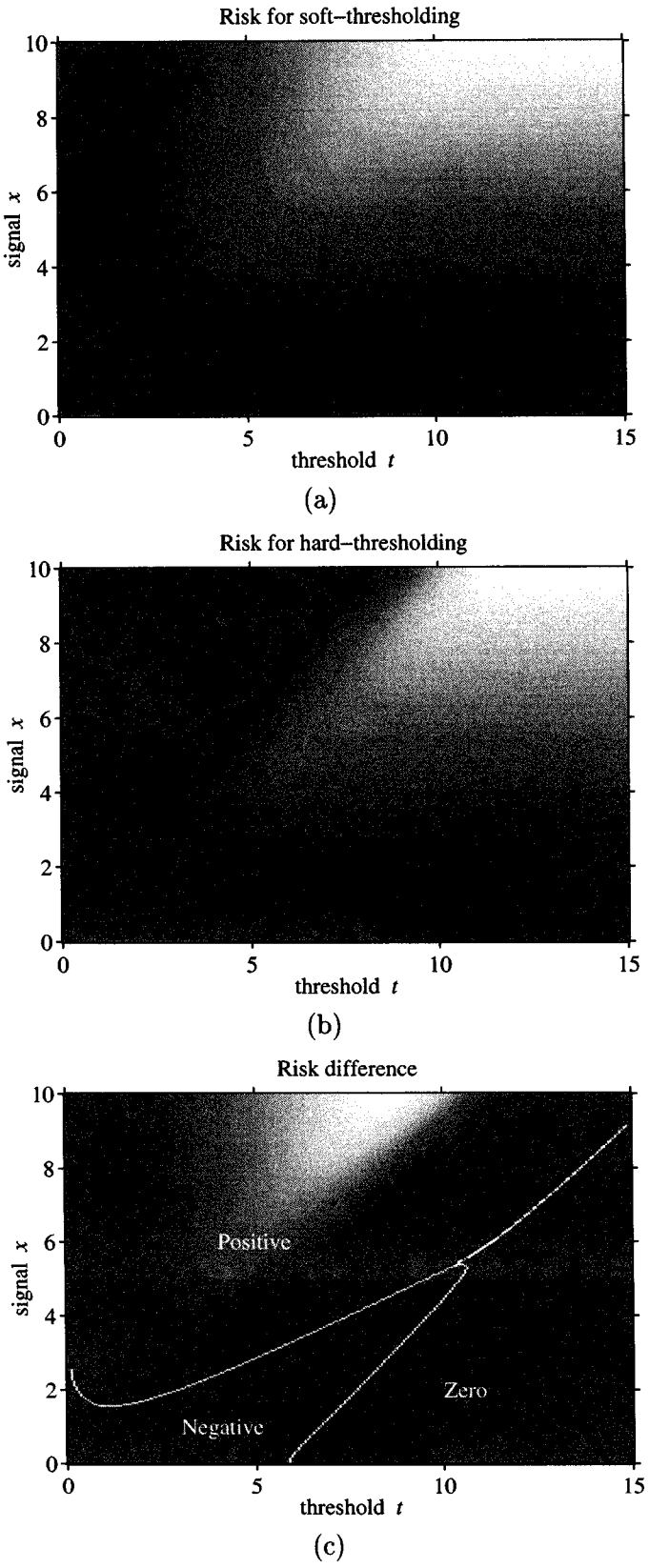

Fig. 4. Risks of soft-thresholding and hard thresholding with normalized noise power $\sigma^2 = 1$. (a) Soft-thresholding. (b) Hard-thresholding. (c) Risk difference between soft-thresholding and hard-thresholding.

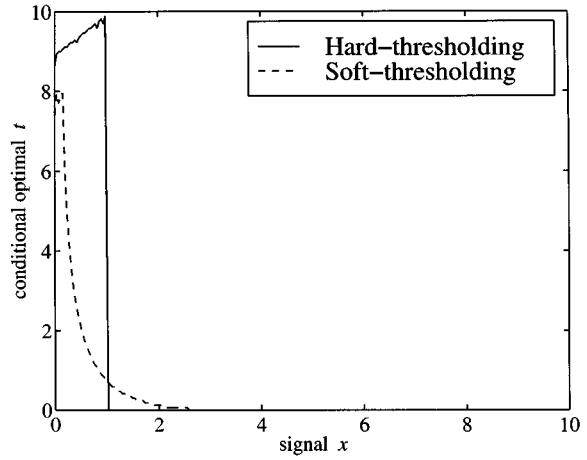

Fig. 5. Conditional optimal thresholds for hard-thresholding (solid line) and soft-thresholding (dashed line) for a given signal value x with normalized noise power $\sigma^2 = 1$.

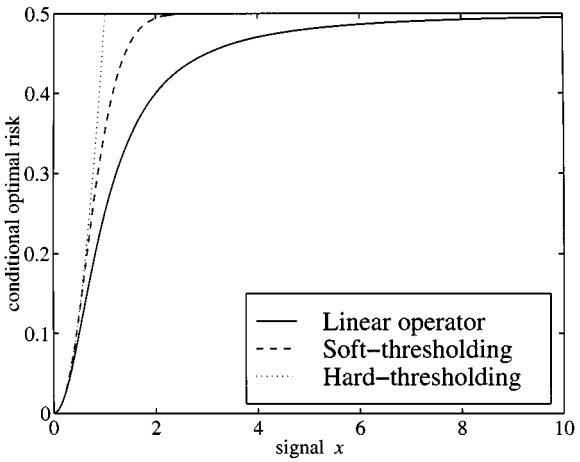

Fig. 6. Conditional optimal risks for linear operator (solid line), soft-thresholding (dashed line) for a given signal value x with normalized noise power $\sigma^2 = 1$.

soft-thresholding and hard-thresholding. However, this only indicates that for a constant signal, the linear operator performs better. When the signal energy is concentrated on only few samples, the linear method still keeps the noise on small signal samples while the thresholding method removes the small signal samples as well as the noise on them. By nonlinear operation, the thresholding method exhibits better MSE performance. This can be illustrated by an example in an extreme case as follows.

Suppose the signal is $x_i = A\delta(i), i = 0, \ldots, N-1$ and the noise is Gaussian noise, $N(0, \sigma^2)$, where δ is a Kronecker δ. If $A \gg \sigma$ and a relative large threshold $\sigma \ll t \ll A$ is selected, the MSE risks of soft-thresholding and hard-thresholding are $J_s(t) \approx \sigma^2 + t^2$ and $J_h(t) \approx \sigma^2$, respectively. For the linear operator (14), the optimal MSE risk is $J_{l,opt}(a) = (N\sigma^2 A^2/[2(N\sigma^2 + A^2)]$. Actually, if a is a vector, i.e., a general linear filter is used, the above MSE risk is still optimal when the signal spectrum is white. This can be proved by using optimal Wiener filtering [15], [16]. It can be seen that when N is relatively large, the optimal MSE risk of this linear operator is much inferior to the thresholding methods.

From the above analysis, we can infer that, compared to linear methods, the thresholding methods are more effective only when the signal energy is concentrated on few signal samples and the local SNR is relatively large. In such cases, generally the optimal MSE risk of the hard-thresholding method is superior to the optimal MSE risk of the soft-thresholding method, however, it is much easier for the soft-thresholding method to achieve its optimal MSE performance since it only has global optima.

The presented TNN indeed combines the linear method and thresholding methods. The linear orthogonal transform performs as local matched filters to concentrate the signal energy in the transform domain. The transform bases are also employed to maintain the desired structure of the recovered signals. For examples, the Fourier transform maintains the harmonic components of the recovered signal structure and the wavelet transform maintains some local regularity of the recovered signal structure.

V. Adaptive Learning Algorithms of TNNs toward the Optimal Solution

A. Supervised Learning with the Reference

In this section, we discuss the supervised learning algorithms of the TNN. In supervised learning, a reference signal serves as a teacher to evaluate the MSE in the learning process. The following two cases are considered.

1) The Original Signal x is Known: This case seems impractical since the original signal is usually unknown in practice. The objective of the TNN is to estimate the original signal. However, the scheme can actually be used in two occasions: 1) When the original signal is used as a reference signal, the TNN can give us a numerical method to calculate the optimal thresholds. Note that the global optimal thresholds of the thresholding method can not be calculated analytically by close form expression. 2) When there are some known training signal sequences of a relatively stationary signal, this scheme can be used to train the TNN first and then the trained TNN can be used to process the subsequent unknown signal sequences.

Since the soft-thresholding function $\eta_s(x,t)$ is weakly differentiable in Stein's sense [17], $\boldsymbol{t}_{opt}$ for (3) can be calculated numerically by a gradient-based optimization algorithm. In learning step k, the parameter $\boldsymbol{t}$ can be adjusted as follows:

$$\boldsymbol{t}(k+1) = \boldsymbol{t}(k) - \Delta\boldsymbol{t}(k) \tag{16}$$

and

$$\Delta\boldsymbol{t}(k) = \boldsymbol{\alpha}(k) \cdot \frac{\partial J(\boldsymbol{t})}{\partial \boldsymbol{t}}\bigg|_{\boldsymbol{t}=\boldsymbol{t}(k)} = \boldsymbol{\alpha}(k) \cdot \sum_{i=0}^{N-1} \varepsilon_i \cdot \frac{\partial \hat{v}_i}{\partial \boldsymbol{t}}\bigg|_{\boldsymbol{t}=\boldsymbol{t}(k)} \quad cr \tag{17}$$

where $\varepsilon_i = \hat{v}_i - v_i$ is the estimation error, $\boldsymbol{\alpha}(k) = \mathrm{diag}[\alpha_1(k), \alpha_2(k), \ldots, \alpha_P(k)]$ is the learning rate matrix at each step and α_j is the learning rate for parameter t_j. Note

$$\frac{\partial \hat{v}_i}{\partial t_j} = 0, \quad i \notin I_j \tag{18}$$

i.e., the parameter t_j only depends on the wavelet coefficients at scale j. This makes the above algorithm computationally efficient. This optimization procedure can be used as the learning

algorithm of TNN's depicted in Fig. 2. The given signal series $\boldsymbol{y}$ and the reference signal series $\boldsymbol{y'} = \boldsymbol{x}$ are used as the training set.

2) A Reference Noisy Signal $y' = x + n'$ can be Generated: In practice, the original signal x is usually unknown and cannot be used as the reference signal y' in Fig. 2. However, we may know more than one noise corrupted version of a signal. For example, in adaptive echo cancellation applications, we have two measurements for the same source signal [16], [17]. Also, in some applications, we may have an array of sensors and each sensor may give us one corrupted version of the signal.

In such cases, a practical denoising scheme is developed. In this scheme, two noise corrupted signal y and y' are produced from the same signal x plus uncorrelated noise n and n', i.e.,

$$\begin{aligned} y_i =& x_i + n_i \\ y'_i =& x_i + n'_i, i = 0, \ldots, N-1. \end{aligned} \tag{19}$$

In this case, the error between the estimate $\hat{\boldsymbol{x}}$ and the reference signal $\boldsymbol{y'}$ is $\boldsymbol{\varepsilon'} = \hat{\boldsymbol{x}} - \boldsymbol{y'}$. The MSE risk becomes

$$\begin{aligned} J'(\boldsymbol{t}) =& \frac{1}{2}E\{\|\boldsymbol{\varepsilon'}\|^2\} = \frac{1}{2}E\{\|\hat{\boldsymbol{x}} - \boldsymbol{y'}\|^2\} \\ =& \frac{1}{2}E\{\|\hat{\boldsymbol{x}} - \boldsymbol{x} - \boldsymbol{n'}\|^2\} \\ =& \frac{1}{2}\left[E\{\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2\} + E\{\|\boldsymbol{n'}\|^2\} + 2E\{(\hat{\boldsymbol{x}} - \boldsymbol{x})^{\mathrm{T}} \cdot \boldsymbol{n'}\}\right] \\ =& \frac{1}{2}E\{\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2\} + \frac{1}{2}E\{\|\boldsymbol{n'}\|^2\} \\ =& J(\boldsymbol{t}) + \frac{1}{2}E\{\|\boldsymbol{n'}\|^2\}. \end{aligned} \tag{20}$$

Note that $E\{(\hat{\boldsymbol{x}} - \boldsymbol{x})^{\mathrm{T}} \cdot \boldsymbol{n'}\} = 0$ because n' is uncorrelated with x and n. As can be deduced from (20), when the parameter $\boldsymbol{t}$ is adjusted so that $J'(\boldsymbol{t})$ is minimized, $J(\boldsymbol{t})$ in (10) is also minimized accordingly. That means, when selecting y' in (19) as the reference signal, the neural network structure in Fig. 2 can also be used to adjust the parameter $\boldsymbol{t}$ to minimize the risk $J(\boldsymbol{t})$.

Theorem 2: Suppose the reference noisy signal $y' = x + n'$ is selected as in (19), i.e., the reference error series is $\varepsilon'_i = \hat{x}_i - y'_i$ and the risk is $J'(t) = (1/2)E\{\sum_i (\hat{x}_i - y'_i)^2\}$. Suppose

$$t'^* = \arg\min_t J'(t)$$

then $t'^* = t^*$ and both the errors ε'_i and ε_i can be used to seek t^* for minimizing $J(t)$ in adaptive algorithms.

Proof: First, from (20) we know that $t'^* = t^*$. The error ε'_i is an instantaneous error for $J'(t)$ and can be used to adaptively seek the t'^*, which is the same as t^*.

This is to say, it is equivalent to use the reference signal y' or the true signal x as the reference signal in terms of minimizing the MSE of the estimation. Also note that the optimal solution t^* does not depend on the SNR of the reference noisy signal [See (20)].

B. Unsupervised Learning-Only the Received Signal y is Known

In many practical occasions, it is hard to obtain any available reference signal and only the received noisy signal y is known. Yet it is still desired to recover the original true signal x. Since there is no other reference signal available for the evaluation of the estimation error, the estimation error has to be estimated from the received signal y itself in an unsupervised fashion. Then we may construct an unsupervised learning algorithm to minimize this error. Based on this principle, the following two unsupervised learning algorithms for the TNN are developed.

1) SURE Based Learning Algorithms: In practice, the noise variance σ^2 is usually known or easy to be estimated. In such instances, there is a good method to estimate the estimation error of the true signal for additive Gaussian noise. The *Stein's Unbiased Risk Estimate* (SURE) is an unbiased estimator of the MSE [17]. For noise suppression problem as in (2), assume the noise variance has been normalized to $\sigma^2 = 1$, without loss of generality. Suppose an estimation operator $f(\cdot)$ is used to estimate the true signal x, i.e., $\hat{x} = f(y)$. Define

$$\boldsymbol{g}(\boldsymbol{y}) \triangleq \hat{f}(\boldsymbol{y}) - \boldsymbol{y} \tag{21}$$

where $\boldsymbol{g} = [g_0, g_1, \cdot, g_{N-1}]^{\mathrm{T}}$ is a function from $\mathcal{R}^N$ to $\mathcal{R}^N$. Stein [17] showed that when $\boldsymbol{g}(\boldsymbol{y})$ is weakly differentiable

$$E\|\hat{f}(\boldsymbol{y}) - \boldsymbol{x}\|^2 = N + E\left\{\|\boldsymbol{g}(\boldsymbol{y})\|^2 + 2\nabla_y \cdot \boldsymbol{g}(\boldsymbol{y})\right\} \tag{22}$$

where $\nabla_y \cdot \boldsymbol{g}(\boldsymbol{y}) = \sum_{i=0}^{N-1} \partial g_i / \partial y_i$. The *Stein's Unbiased Risk Estimate* (SURE) is defined as

$$J_{\mathrm{SURE}}(t) \triangleq N + \|\boldsymbol{g}(\boldsymbol{y})\|^2 + 2\nabla_y \cdot \boldsymbol{g}(\boldsymbol{y}). \tag{23}$$

where t is the threshold parameter when the thresholding functions are used as the estimation operator $f(\cdot)$. Clearly, it is an unbiased estimator of the MSE risk $J(t)$. Note that for the TNN, the above SURE risk can be calculated for each channel in the transform domain.

Then the SURE risk can be used as the objective function of the TNN and the gradient-based adaptive learning algorithm can be used to minimize this objective function. However, the estimation operator $f(\cdot)$ has to have at least second-order derivatives to obtain the gradient of the SURE risk $J_{\mathrm{SURE}}(t)$. Neither the standard soft-thresholding function $\eta_s(x,t)$ nor the standard hard-thresholding function $\eta_h(x,t)$ has second-order derivatives. Note that the new proposed thresholding functions $\eta_{st}(x,t)$ and $\eta_{ht}(x,t)$ in (6) and (7) are infinitely differentiable and thus can be used. The gradient of the SURE risk can then be calculated as follows:

$$\frac{\partial J_{\mathrm{SURE}}(t)}{\partial t} = 2\sum_{i=0}^{N-1} g_i \cdot \frac{\partial g_i}{\partial t} + 2\sum_{i=0}^{N-1} \frac{\partial^2 g_i}{\partial y_i \partial t}. \tag{24}$$

where $g_i = \eta(y_i, t) - y_i$ and the thresholding function $\eta(\cdot)$ may be $\eta_{st}(x,t)$ or $\eta_{ht}(x,t)$. Then the gradient-based adaptive learning steps (16)–(17) can be employed in this unsupervised learning process.

Also note that Donoho's method [3] uses the standard soft-thresholding function and selects a threshold t^S in a finite set $\{y_0, y_1, \ldots, y_{N-1}\}$. Therefore the selected threshold t^S is a sub-optimal threshold for the SURE risk.

2) Learning by Cross-Validation: It is possible to "create" a reference signal from the received signal y itself by cross-validation. Then the estimation error can be estimated and the gradient-based adaptive learning algorithm as in (16) and (17) can be used. This scheme is similar to the cross-validation wavelet shrinkage method proposed by Nason [6], where the wavelet transform and standard thresholding functions are used.

The received signal samples y can be divided into even samples y_e and odd samples y_o by a subsampling process. Then y_e can be the reference signal of y_o to calculate the estimation error in Fig. 2 and vice versa. This is reasonable because the features

of coefficients of y_e and y_o in the transform domain are usually very similar. For example, for an oversampled bandlimited signal y with a sampling rate doubling the critical Nyquist sampling rate, the amplitude spectra of y_e and y_o are completely the same and there are only some phase differences between their spectra. The similar properties also exist for other orthogonal linear transforms. We suggest the use of necessary magnitude or phase compensation when calculating the cross-validation error. Note that this cross-validation learning algorithm does not need to know any *a priori* information of the noise.

C. Time Adaptive Stochastic Learning-Nonlinear Adaptive Filtering

1) Time Adaptive Stochastic Learning Algorithms: In the above adaptive learning algorithms, we assume that the signal samples are finite and all samples of the received signal y are used in learning process. However, in real-time adaptive signal processing applications, the signal is usually time-varying random process and only the past samples of the received signal are known. It is necessary for the TNN to track the changes of the signal in real-time and continually seek the optimum in some statistical sense. The previous analysis and learning algorithms indeed can be generalized to this case.

For stochastic signals, the noise suppression problem can be formulated as follows with a slight modification of (2). Assume that a random signal x is transmitted over a channel to a sensor that receives the signal with an additive uncorrelated noise n. The received signal y is given by

$$y = x + n. \tag{25}$$

Let $\hat{x}$ denote the signal after noise suppression. Now the MSE risk can be written as

$$J(\hat{x}, x) = \frac{1}{2}E\{\epsilon^2\} = \frac{1}{2}E\{(\hat{x} - x)^2\} = \frac{1}{2}E\{(\hat{x}_i - x_i)^2\}, \forall i. \tag{26}$$

Here we assume the signal x is an ergodic stochastic process, which is a commonly used assumption in adaptive filtering profiles [19]. The only difference between (26) and (3) is that (26) uses the mathematical expectation instead of the average (summation) of finite samples in (3). By replacing the average with the mathematical expectation, it is easy to show that the Theorem 1 and its proof still hold for this stochastic case, i.e., the above MSE risk has at most one minimum with respect to the threshold parameter t. Also, we can rewrite (17) in the gradient based adaptive learning algorithm as:

$$\Delta \boldsymbol{t}(k) = \boldsymbol{\alpha}(k) \cdot \frac{\partial J(\boldsymbol{t})}{\partial \boldsymbol{t}}\bigg|_{\boldsymbol{t}=\boldsymbol{t}(k)} = \boldsymbol{\alpha}(k) \cdot E\left\{\epsilon_i \cdot \frac{\partial \hat{v}_i}{\partial \boldsymbol{t}}\right\}\bigg|_{\boldsymbol{t}=\boldsymbol{t}(k)} \tag{27}$$

where $\epsilon_i = \hat{v}_i - v_i$ is the estimation error, $\boldsymbol{\alpha}(k) = \text{diag}[\alpha_1(k), \alpha_2(k), \ldots, \alpha_P(k)]$ is the learning rate matrix at each step and α_j is the learning rate for parameter t_j. Similarly, one can easily prove that (20) and Theorem 2 can be generalized to real-time stochastic signals. Therefore, after replacing the summation with the mathematical expectation in adaptive learning, all above-mentioned supervised and unsupervised learning algorithms can be generalized to the real-time stochastic signal model depicted in (25).

Nevertheless, the mathematical expectation of the error is difficult to obtain in practice. Hence, we suggest the use of an LMS-like scheme, i.e., we use instantaneous square error risk $J_i(\hat{x}, x) = 1/2\epsilon^2$ to approximate the true risk $J(\hat{x}, x)$ in (26) in all of the adaptive supervised and unsupervised learning algorithms we previously developed. The thresholding parameter $\boldsymbol{t}$ can then be adjusted adaptively by

$$\Delta \boldsymbol{t}(k) = \boldsymbol{\alpha}(k) \cdot \frac{\partial J_i(\boldsymbol{t})}{\partial \boldsymbol{t}}\bigg|_{\boldsymbol{t}=\boldsymbol{t}(k)}. \tag{28}$$

Especially, for SURE based algorithms, the SURE risk can be used to estimate the MSE instantaneously in the transform domain

$$J_{i,\text{SURE}}(\boldsymbol{t}) \triangleq 1 + g^2(u_i) + 2 \cdot \frac{\partial g(u_i)}{\partial u_i}. \tag{29}$$

This stochastic learning process is depicted in Fig. 7. In this case, the linear orthogonal transform should be implemented in real-time. Usually, we can use a specific filter bank depending on the transform.

By using above time adaptive stochastic learning algorithms, we actually achieve a type of nonlinear time adaptive filtering tool.

2) Convergence of the Stochastic Learning Algorithm: The analysis of an adaptive nonlinear system is generally difficult. Here we only analyze the convergence property of the proposed stochastic learning algorithm in a tangible situation. The analysis of the algorithm will be based on stationary signals, although our nonlinear adaptive filtering methods are designed to track nonstationary random input. This idealization is commonly used so that the analysis becomes relatively tractable. Furthermore, we assume the standard soft-thresholding function is used. In the following theorem, we show that in such an ideal situation, the algorithm is convergent in certain statistical sense.

Theorem 3: For the stochastic signal model described in (25) and (26), assume the following learning algorithm is used:

$$t(i+1) = t(i) - \Delta t(i) \tag{30}$$

when $t(i) \geq 0$

$$\begin{aligned}\Delta t(i) =& \alpha(i) \cdot \frac{\partial \hat{x}_i}{\partial t} \cdot \epsilon_i \\ =& \begin{cases} \alpha(i) \cdot [n_i + t(i)], & y_i < -t(i) \\ 0, & |y_i| \leq t(i) \\ -\alpha(i) \cdot [n_i - t(i)], & y_i > t(i) \end{cases}\end{aligned} \tag{31}$$

and when $t(i) < 0$

$$\Delta t(i) = \begin{cases} \alpha(i) \cdot [n_i + t(i)], & y_i < t(i) \\ 2\alpha(i)t(i), & |y_i| \leq -t(i) \\ -\alpha(i) \cdot [n_i - t(i)], & y_i > -t(i). \end{cases} \tag{32}$$

Here $\alpha(i) > 0$ is the learning rate of each step.

If there exists optimal t^* as in (10), then

$$\lim_{i \to \infty} E\{t(i)\} = t^* \tag{33}$$

when $\alpha(i)$ is suitably selected, i.e., the above learning algorithm is convergent in the mean. See Appendix B for proof.

Note that (32) constructed such that the negative t can be handled smoothly when $\Delta t(i)$ is too large and then the limit is unbiased. In practice, we can select $\alpha(i)$ small enough so that $t(i)$ is nonnegative. Although it is not easy to select a proper series $\alpha(i)$, the difference between $t(i)$ and t^* will approach a small number when $\alpha(i)$ is small enough. From (30) and (31), it is easy to see that if $|t(i) - t^*| > \delta$, when $\alpha(i) < \delta/\epsilon_i$, $t(i)$ will

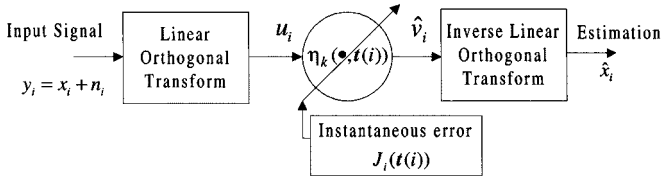

Fig. 7. Stochastic learning for TNNs.

be adjusted toward t^* until $|t(i) - t^*| \leq \delta$. Therefore, the algorithm is still very practical even if a series $\alpha(i)$ cannot be found to make the algorithm convergent in certain mathematical sense. The results in Theorem 3 also hold for the transform domain in TNNs. This is simply because each threshold t_j in $\boldsymbol{t}$ is only dependent on coefficients series and MSE at channel j. Each t_j is adjusted independently from the others.

Apparently, if the reference signal as in (19) is used, Theorem 3 still holds for the same reasons in Theorem 2.

3) Time-Scale or Time-Frequency Adaptive Noise Reduction: The above stochastic adaptive learning algorithms are capable of tracking the time-varying nonstationary features of the signal and noise toward the minimum MSE risk. Therefore, we can also use the above stochastic adaptive learning algorithms for a finite length signal to track local changes within the signal. In this case, we can take advantage of the time-varying local estimation error instead of the overall average. Then the threshold parameter t is dependent on not only different channels in the transform domain but also time position, i.e., it is fully adaptive with respect to the time and different channels. If the wavelet-based or Fourier-based linear orthogonal transform is used, the time-scale or time-frequency adaptive noise reduction by TNN's is indeed achieved [19].

VI. Examples

A. Adaptive Noise Reduction When all Signal Samples are Known

Five test signals are generated using software WaveLab [20]. Four of them are commonly used in denoising literature [3], namely, Blocks, Doppler, Bumps and HeaviSine. Their signal length is 1024 samples. To test the adaptivity of the algorithm, we intentionally create a new test signal, which is a combination of two completely different signals – Blocks signal and Doppler signal. Its signal length is 2048 samples. The test signals are shown as in Fig. 8(a)–(e). The SNR is 7 dB and the noise variance is normalized to 1, i.e., the MSE of the noisy signal y is 1. For comparison, the noise reduction results of different methods are tested and the discrete wavelet transform (DWT) is used as the linear transform in the TNN. Daubechies 8-tap least asymmetrical wavelet filters are used. The largest scale of the DWT is set to be $M = 6$ in the experiments, i.e., there are $M + 1 = 7$ orthogonal channels in the transform domain of the TNN.

Some typical adaptive noise reduction methods based on the TNN are tested. The results of different noise reduction schemes are shown in Table I. The information and techniques used by different methods are also indicated in Table I. Two commonly used wavelet thresholding denoising methods are also given for comparison: "VisuShrink" is a universal wavelet thresholding method [5]; "SureShrink" is an optimized hybrid scale dependent thresholding scheme based on SURE risk [3], which has the best MSE performance among conventional thresholding denoising methods. To clearly illustrate the results, Fig. 9 plots the estimation error for the signal "Blocks-Doppler", i.e., $\hat{x} - x$. The results of other test signals are not plotted here due to space limitations.

TNN-OPT0 and TNN-OPT1 are the supervised TNN methods with the standard soft-thresholding $\eta_s(x,t)$ and the new soft-thresholding function $\eta_{st}(x,t)$, respectively. They both use the true signal as reference and converge to MSE minima. For soft-thresholding functions, the found minima can only be the global minima. Therefore, they actually represent the optimal MSE performance of the thresholding scheme. Note that here we take an empirical function parameter value $\lambda = 0.01$ for the new soft-thresholding function $\eta_{st}(x,t)$[1] This new soft-thresholding function is used in all following TNN methods unless otherwise specified. TNN-hard shows a local minimum found by the TNN using a new hard-thresholding function $\eta_{ht}(x,t)$ with an empirical function parameter value $\mu = 0.01$. Obviously, this local minimum is usually not as good as soft-thresholding. In experiments, we found that the TNN using hard-thresholding is very easy to get trapped in local minima.

TNN-ref is a supervised TNN method using noisy reference $y' = x + n'$. Here noise n' is set to have the same variance as n. The MSE performance is very close to the optimal, as expected. And yet this is a very practical adaptive noise reduction method since a noisy reference y' is often easy to obtain. TNN-sure is an unsupervised TNN method based on SURE risk while the noise variance is known or can be well estimated. We see that the MSE result is also rather good. Note that only new soft-thresholding functions can be used in this method since SURE risk is used. When there is no additional *a priori* information available but the received noisy signal y itself, the TNN method using cross-validation can be used, denoted as TNN-CV. As can be expected, the results of TNN-CV is not as good as that of the other methods that use additional *a priori* information. However, TNN-CV is still a good method when no additional *a priori* information is available.

Last, TNN-TS employs a supervised stochastic learning method of the TNN and achieves time-scale adaptive noise reduction. The reference signal is again $y' = x + n'$. We see that it usually gives the best MSE performance, even better than the optimal MSE performance the nontime-adaptive thresholding methods can achieve (see TNN-OPT0 and TNN-OPT1). However, for signal "HeaviSine," it does not give better MSE performance than the optimal nontime-adaptive solutions. This can be justified by the fact that the "HeaviSine" signal is pretty smooth and has little abrupt changes. Therefore, the time adaptivity of the algorithm cannot help to improve the MSE performance.

For comparison we also calculated the optimal MSE performance of a 16-tap linear filter. The original true signal x is used as a reference and the solution of the Wiener-Hoff equation –

[1]The performances of the numerical tests in this paper are reasonably invariant in the neighborhood of the selected empirical values of function parameters λ and μ.

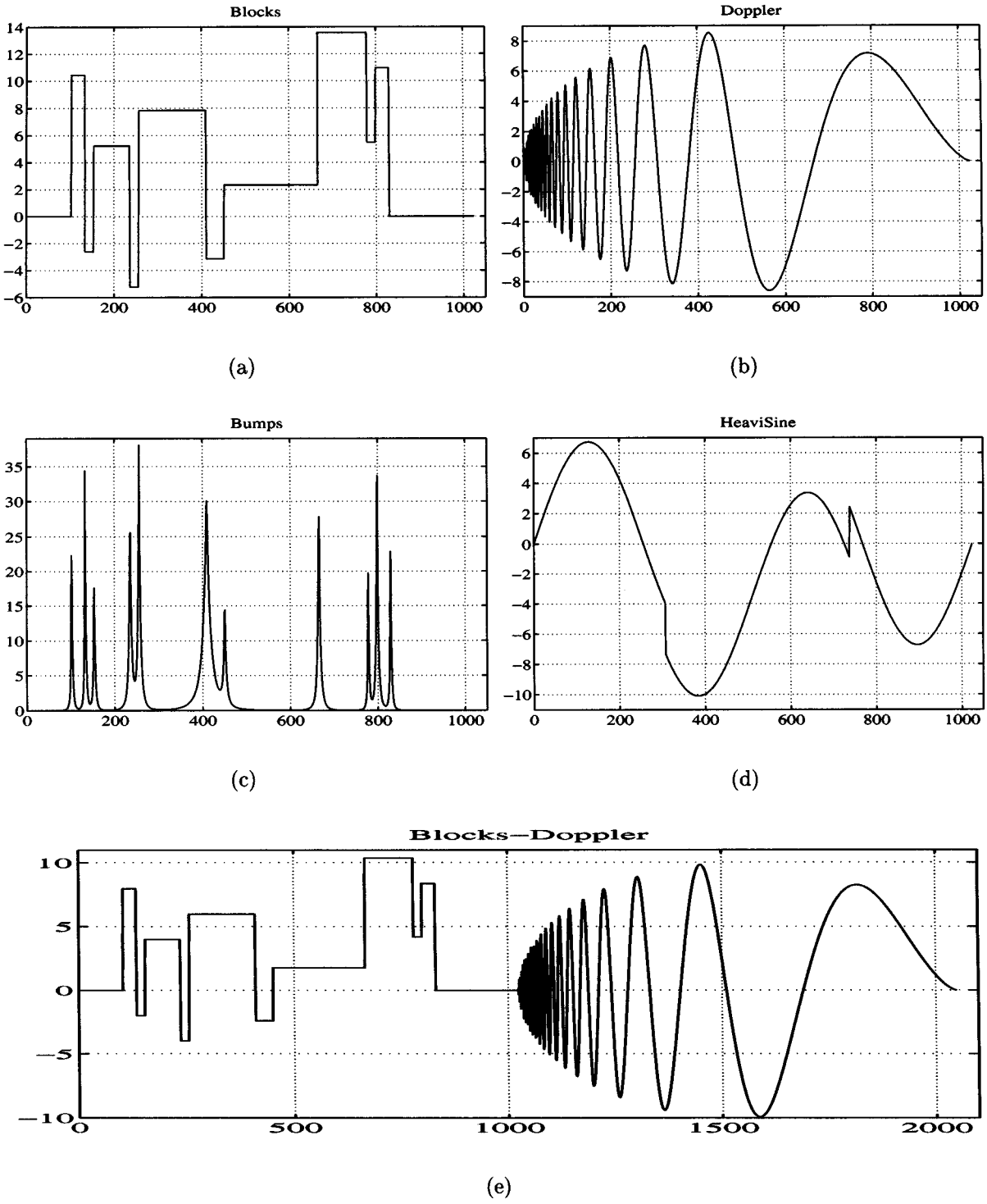

Fig. 8. Test Signals. (a) Blocks. (b) Doppler. (c) Bumps. (d) HeaviSine. (e) Blocks-Doppler.

Wiener filter is obtained. The corresponding MSE is shown in the last column in Table I.

From the results, we see that almost all adaptive noise reduction methods perform better than conventional wavelet thresholding methods and the Wiener filter in terms of MSE. From the plots, we observe that compared to nonadaptive conventional methods, the TNN-based methods can better adapt to fast changes of the local features and usually preserve better fine structure (high-frequency part) of the signal.

In numerical experiments, we also observed that in cases where both standard and the new thresholding functions can be used, the methods using the new thresholding functions have similar optimal MSE performance to the ones using standard thresholding functions, but the adjustability and robustness of the learning algorithms using the new thresholding functions are much better and they usually give better learning results. This is because the new thresholding functions have nonzero derivatives for all t, while the

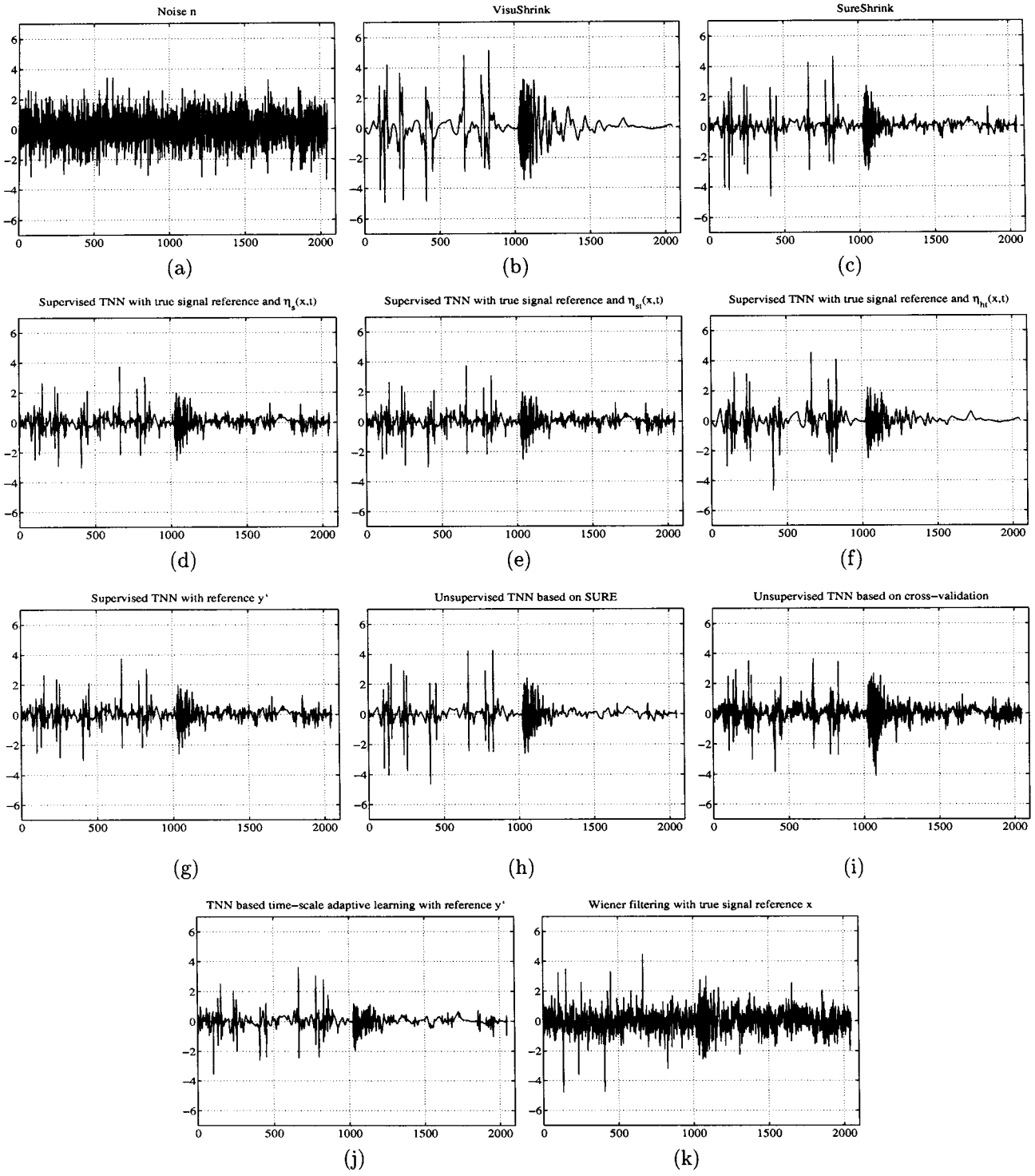

Fig. 9. Numerical results of the error signals for different noise reduction methods applying on Blocks-Doppler signal (a) Original error–noise n (b) VisuShrink (c) SureShrink (d) TNN-OPT0: Supervised TNN with true signal reference, using standard soft-thresholding function $\eta_s(x,t)$ (e) TNN-OPT1: Supervised TNN with true signal reference, using new soft-thresholding function $\eta_{st}(x,t)$ with $\lambda = 0.01$ (f) TNN-hard: Supervised TNN with true signal reference, using new hard-thresholding function $\eta_{ht}(x,t)$ with $\mu = 0.01$ (g) TNN-ref: Supervised TNN with the noisy reference signal $y' = x + n'$ using $\eta_s(x,t)$ (h) TNN-sure: Unsupervised TNN based on SURE (i) TNN-CV: Unsupervised TNN based on cross-validation (j) TNN-TS: TNN-based time-scale adaptive learning with reference $y' = x + n'$. (k) Linear: Wiener filtering with true signal reference x.

standard soft-thresholding function has zero function values and derivatives when $|x| < t$. Therefore, when $t > |x|$, the learning algorithm based on the standard soft-thresholding function completely loses its adjustability, while the learning algorithm based on the new functions still knows where to go.

Note that except for the TNN using hard-thresholding functions, all other algorithms are generally not sensitive to the ini-

TABLE I
MSR RISK FOR DIFFERENT DENOISING SCHEMES

	VisuShrink	SureShrink	TNN-OPT0	TNN-OPT1	TNN-hard	TNN-ref	TNN-sure	TNN-CV	TNN-TS	Linear
reference true signal x			✓	✓	✓					✓
known σ^2	✓	✓					✓			
standard soft-thresholding	✓	✓	✓							
new soft-thresholding				✓		✓	✓	✓	✓	
new hard-thresholding					✓					
reference $y' = x + n'$						✓			✓	
cross-validation								✓		
Time-Scale Adaptive									✓	
MSE(Blocks)	1.0304	0.4289	0.2685	0.2682	0.3941	0.2847	0.4010	0.6774	0.2427	0.5755
MSE(Doppler)	0.5242	0.2527	0.2121	0.2121	0.2009	0.2162	0.2811	0.3619	0.1915	0.4977
MSE(Bumps)	1.2580	0.4513	0.3653	0.3648	0.4298	0.3690	0.5198	0.6268	0.3109	0.8103
MSE(HeaviSine)	0.1125	0.0900	0.0786	0.0735	0.0626	0.0737	0.0845	0.1109	0.0796	0.2077
MSE (Blocks-Doppler)	0.7931	0.3087	0.2170	0.2170	0.2800	0.2184	0.2777	0.3963	0.1673	0.5527

tial value of the TNN. The learning rate defines the convergence speed and accuracy of all the algorithms.

B. Adaptive Noise Reduction in Real-Time Adaptive Filtering

In real-time adaptive filtering, only past samples of the received signal are known. The objective is to use these past samples to track the changes of the signal in real time and continually seek the optimum. Therefore, the time adaptive stochastic learning algorithms described in Section V-C1 can be used.

The test results for the signal "Blocks-Doppler" are presented here to illustrate the adaptive filtering, since it is highly nonstationary. The signal is assumed to be a stochastic real time series, i.e., only past samples are available at each time. Two stochastic learning algorithms for the TNN are tested, i.e., supervised stochastic learning with noisy reference $y' = x + n'$ and unsupervised stochastic learning based on SURE risk. In both algorithms, the new soft-thresholding function $\eta_{st}(x, t)$ with $\lambda = 0.01$ is used. Again, the DWT and the same setup as in the above subsection are used. Then the optimal MSE performance calculated for TNN-OPT1 can be treated as the optimal convergent MSE performance of these statistical learning algorithms.

For comparison, the LMS (Least-mean-square) based linear adaptive filtering method [16] is also used for the same signal. Clearly, the linear adaptive method needs a reference signal. To be justified, we use the same noisy signal y' as the reference signal for the linear adaptive method. The length of the adaptive linear filter is set to be 16. We tried different learning rates and selected the best one for the LMS based linear adaptive method. It is easy to show that the optimal MSE performance of this adaptive linear filter is the Wiener filtering result as in Table I.

The learning performances of different adaptive filtering methods are shown in Fig. 10. Note that the MSE at each time i is calculated by taking the mean value of the squared error using the adaptive parameters (thresholds or linear filter coefficients) at time i to filter all samples. The MSEs at time $i = 2048$ of different methods are shown in Table II.

From the results, we can see that the TNN-based methods converge rather fast toward the optimal solutions of the methods. In addition, we see that when using linear adaptive filtering, there are abrupt changes of MSE at the time of abrupt changes of the signal. Apparently, such sudden changes are not desired in adaptive systems, although they seem inevitable. For TNN-based methods, there is no such phenomena. The adaptive process is rather smooth in terms of MSE. This is because the abrupt changes (singularities) are caught on all scales in the transform domain [21]. Their energy is spread to all channels and therefore will not cause the sudden change of adaptive systems.

Indeed, more numerical simulations of other signals also show that the TNN based methods perform much better than linear adaptive filtering in both learning performance and the optimal MSE performance.

C. Spatial-Scale Adaptive Image Denoising

Image denoising is an often-encountered application in real world. In the following, an image denoising example using a real-world image is presented to illustrate the application of the TNN. The 256×256 "cameraman" image is used as the test image, with additive independent, identically distributed (i.i.d.) Gaussian noise. The original clean image is shown in Fig. 11(a). Two noisy images are generated with same noise variance. One of them is used as a reference image y'. TNN-TS method presented in the preceding Section VI-A is used. This is a supervised stochastic learning method of the TNN. For two-dimensional images, it becomes a *spatial-scale adaptive* image denoising method instead of *time-scale adaptive* for the one-dimensional signal. The same wavelets as in Section VI-A are used and the largest scale of the two-dimensional DWT is set to $M = 3$ in the experiments. The new soft-thresholding function $\eta_{st}(x, t)$ with $\lambda = 0.01$ is used. The algorithm is tested for noisy images with different noise variances. The peak-signal-to-noise-ratio (PSNR) results are shown in Table III. The first column is the original PSNRs of noisy images. Note that the new spatial-scale adaptive image denoising method is still denoted as "TNN-TS" in the table.

For comparison, Table III also shows the results of the nonadaptive conventional wavelet schemes. They are calculated using functions provided in Matlab Wavelet Toolboxes. As in Section VIA, "VisuShrink" is the universal soft-thresholding

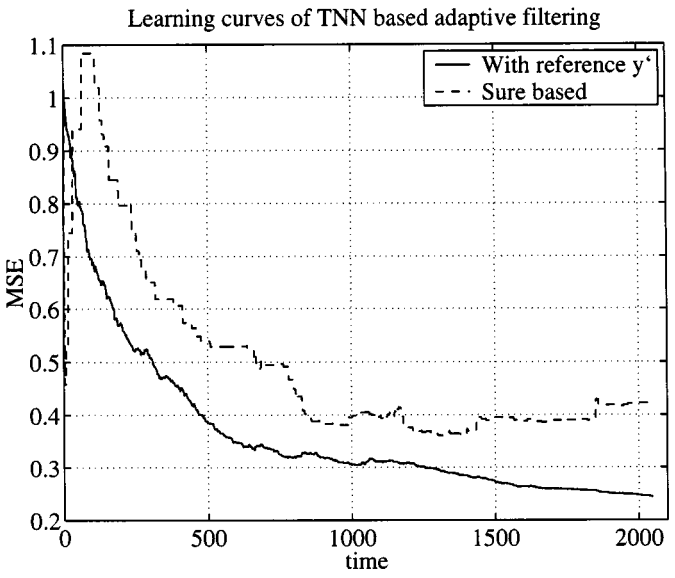

Fig. 10. Learning curves of various adaptive filtering methods. (a) TNN-based adaptive filtering. (b) Linear adaptive filtering (LMS algorithm). Note that the scales of (a) and (b) are different.

TABLE II
MSE AT (CONVERGENT) TIME $i = 2048$ FOR DIFFERENT ADAPTIVE FILTERING SCHEMES

TNN with noisy reference y'	SURE based TNN	Linear (LMS) with reference y'
0.2441	0.4233	0.7942

denoising technique and "SureShrink" is a SURE risk-based scale dependent denoising technique [3], [5]. The column "Wiener" represents the denoising results by Wiener filter, which is the optimal solution of the linear filtering technique.

As can be seen, the TNN based spatial-scale adaptive image denoising has the best performance in terms of PSNR improvement, especially when the PSNR of the original noisy image is high. This can be expected since the amplitudes of the few coefficients representing the signal in the transform domain are much higher than those coefficients representing the noise, and more of the signal energy can be preserved when cutting off all the coefficients with a threshold.

Fig. 11(b) shows the noisy image with PSNR $= 20$ dB (the first row in Table III. The denoised images using different methods are shown in Fig. 11(c)–(f). Apparently, the TNN based spatial-scale adaptive denoising method gives the best visual result as well as the best PSNR improvement.

VII. CONCLUSION

In this paper, we developed a new type of TNN structure for adaptive noise reduction, which combines the linear filtering and thresholding methods. We created new types of soft and hard thresholding functions to serve as the activation function of TNNs. Unlike the standard thresholding functions, the new thresholding functions are infinitely differentiable. By using these new thresholding functions, some gradient-based learning algorithms become possible and the learning process becomes more effective.

We then discussed the optimal solution of the TNN in the MSE sense. It is proved that there is at most one optimal solution for the soft-thresholding TNN. The general optimal performances of both soft and hard thresholding TNNs are analyzed and compared to the linear noise reduction method. It is shown that the thresholding noise reduction methods are more effective than linear methods when the signal energy concentrates on few coefficients in the transform domain. It is indicated that the hard-thresholding may have many local minima. Although the optimal MSE performance of hard-thresholding may be superior to that of soft-thresholding, the soft-thresholding is still more practical since its optimal solution is much easier to find.

Gradient-based adaptive learning algorithms are presented to seek the optimal solution for noise reduction. The algorithms include supervised and unsupervised batch learning as well as supervised and unsupervised stochastic learning. The optimal solution of the TNN can be found by supervised learning with the true signal as a teacher. A practical supervised learning scheme using a noisy signal y' as a reference is developed and proved to be as effective as using the true signal as a reference. In unsupervised learning, two learning schemes are developed. The SURE risk and cross-validation are used to estimate the MSE of the TNN, respectively. Depending on *a priori* information, different learning algorithms can be selected.

By estimating the instantaneous MSE of the TNN, the stochastic learning algorithms can be developed. It is indicated that the TNN with the stochastic learning algorithms can be used as a novel real-time nonlinear adaptive filter. It is proved that the stochastic learning algorithm is convergent in certain statistical sense in ideal conditions.

Numerical results are given for different noise reduction algorithms including conventional wavelet thresholding algorithms and linear filtering method. It is shown that almost all the TNN-based adaptive noise reduction algorithms perform much better than the others in terms of MSE. It is also shown that the more

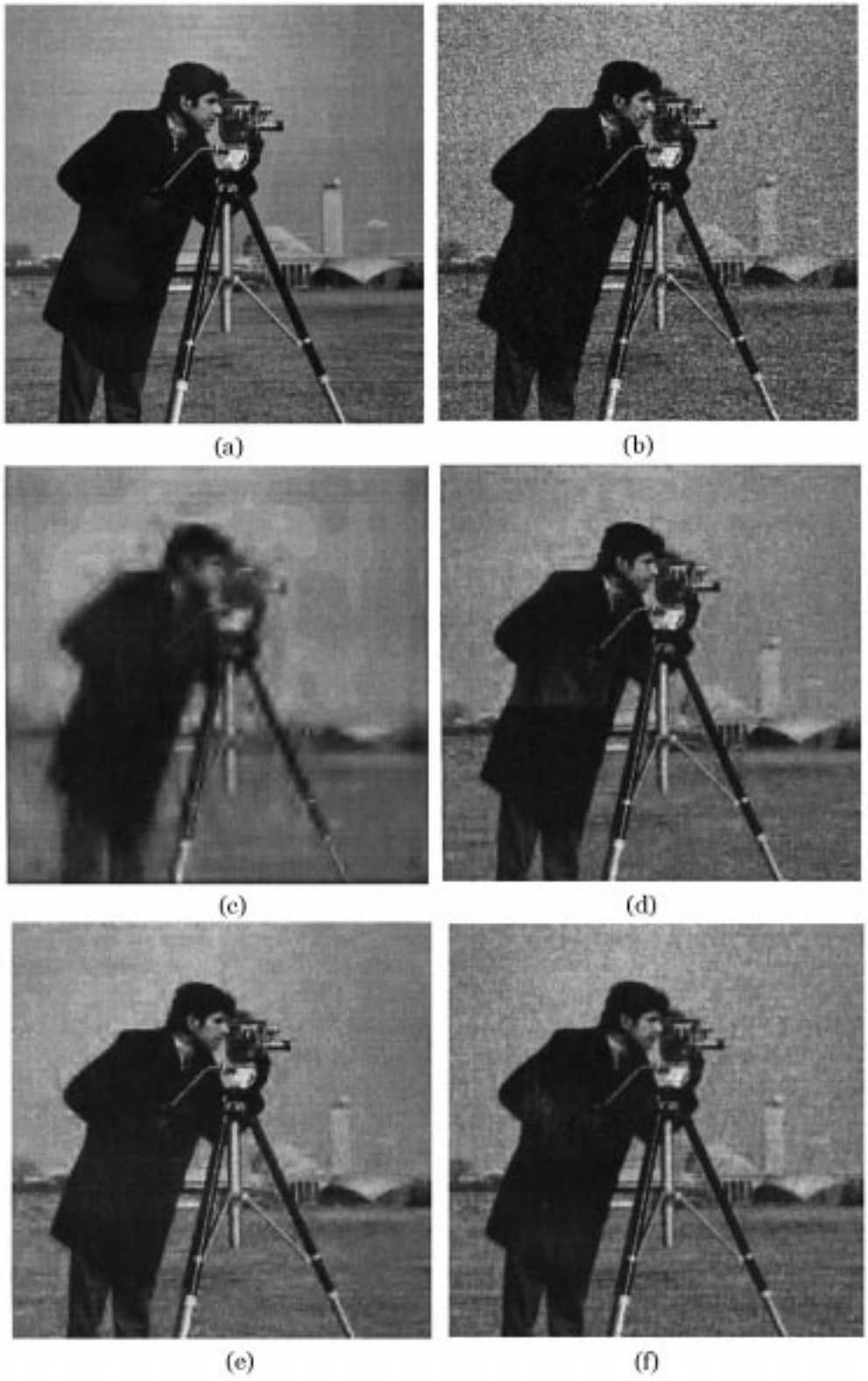

Fig. 11. (a) Original. (b) Noisy (PSNR $= 20$ dB). (c) VisuShrink. (d) SureShrink. (e) TNN-based spatial-scale adaptive filtering. (f) Wiener filtering.

a priori information is used, the better is the performance. Notably, the time-scale adaptive noise reduction method based on stochastic learning often performs even better than the optimal solutions of conventional thresholding noise reduction schemes. The TNN-based nonlinear adaptive filtering methods are also tested and compared with the traditional linear filtering method. It is shown that the new methods outperform the linear adaptive filtering in both the optimal solution and the learning performance. With a real world application – image denoising, we construct a spatial-scale adaptive image denoising algorithm based on the TNN. The new image denoising scheme shows superior performance in both PSNR and visual effect in our tests.

TABLE III
THE PSNRS (DB) OF DIFFERENT DENOISING METHODS. THE FIRST COLUMN IS THE ORIGINAL PSNRS OF NOISY IMAGES

Noisy	VisuShrink	SureShrink	TNN-TS	Wiener
20.0159	20.4768	25.7450	26.6423	26.2899
25.0290	22.4583	28.9996	29.9728	29.1181
30.0057	24.6496	32.6900	33.7289	32.4476
34.9942	26.7662	36.7637	37.8609	36.3116
39.9867	28.8212	41.0362	41.9908	40.6440

In this paper, we show that the proposed TNN is a very effective new tool for adaptive noise reduction of both deterministic and stochastic signals. Various noise reduction algorithms can be constructed based on the TNN. Further investigation on the detailed setup of TNNs and applications is worth being pursued.

APPENDIX A
PROOF OF THEOREM 1

Proof: First rewrite (8) as

$$\hat{x}_i = \begin{cases} y_i + t, & y_i < -t \\ 0, & |y_i| \le t \\ y_i - t, & y_i > t. \end{cases}$$

Let $J(t|x_i) = J(t)|_{x_i} = 1/2E\{\varepsilon_i^2|x_i\}$, i.e., $J(t) = \sum_i J(t|x_i)$. Note $J(t|x_i)$ is a *conditional expected value* [22]. From (9),

$$\begin{aligned} J(t|x_i) =& \frac{1}{2}E\{\varepsilon_i^2|x_i\} \\ =& \frac{1}{2}\big[P(y_i < -t)\cdot E\{(n_i - t)^2|y_i < -t\} \\ &+ P(|y_i| \le t)\cdot E\{x_i^2||y_i| \le t\} \\ &+ P(y_i > t)\cdot E\{(n_i + t)^2|y_i > t\}\big] \\ =& \frac{1}{2}\left[\int_{t-x_i}^{\infty}(\xi - t)^2 p_n(\xi)d\xi + x_i^2\cdot\int_{-t-x_i}^{t-x_i} p_n(\xi)d\xi \right. \\ &\left. + \int_{-\infty}^{-t-x_i}(\xi + t)^2 p_n(\xi)d\xi\right] \end{aligned}$$

Taking derivative with respect to t, we obtain[2]

$$\begin{aligned} h(t|x_i) \triangleq& \frac{\partial J(t|x_i)}{\partial t} \\ =& \int_{-\infty}^{-t-x_i}(\xi + t)p_n(\xi)d\xi - \int_{t-x_i}^{\infty}(\xi - t)p_n(\xi)d\xi \\ =& \int_{-\infty}^{-x_i}\xi p_n(\xi - t)d\xi - \int_{-x_i}^{\infty}\xi p_n(\xi + t)d\xi \\ =& P(y_i > t)\cdot E\{(n_i + t)|y_i > t\} \\ &- P(y_i < -t)\cdot E\{(n_i - t)|y_i < -t\}. \end{aligned}$$

Let $h(t) \triangleq \sum_i h(t|x_i) = \sum_i(\partial J(t|x_i)/\partial t) = (\partial J(t)/\partial t)$, then $h(t^*) = 0$ must hold. Since n_i is Gaussian distribution, its probability density function (pdf) is $p_n(\xi) = (1/\sqrt{2\pi}\sigma)\exp\{-(\xi^2/2\sigma^2)\}$. Subsequently

$$\begin{aligned} \frac{\partial h(t|x)}{\partial t} =& \frac{1}{\sigma^2}\int_{-\infty}^{-x}\xi(\xi - t)p_n(\xi - t)d\xi \\ &+ \frac{1}{\sigma^2}\int_{-x}^{\infty}\xi(\xi + t)p_n(\xi + t)d\xi \\ =& \frac{1}{\sigma^2}\left[\int_{-\infty}^{-x}\xi^2 p_n(\xi - t)d\xi + \int_{-x}^{\infty}\xi^2 p_n(\xi + t)d\xi\right] \\ &- \frac{t}{\sigma^2}\left[\int_{-\infty}^{-x}\xi p_n(\xi - t)d\xi - \int_{-x}^{\infty}\xi p_n(\xi + t)d\xi\right] \\ =& \frac{1}{\sigma^2}\left[\int_{-\infty}^{-x}\xi^2 p_n(\xi - t)d\xi + \int_{-x}^{\infty}\xi^2 p_n(\xi + t)d\xi\right] \\ &- \frac{t}{\sigma^2}h(t|x). \end{aligned}$$

Then

$$\begin{aligned} \frac{\partial h(t)}{\partial t} =& \frac{1}{\sigma^2}\sum_i\left[\int_{-\infty}^{-x_i}\xi^2 p_n(\xi - t)d\xi \right. \\ &\left. + \int_{-x_i}^{\infty}\xi^2 p_n(\xi + t)d\xi\right] - \frac{t}{\sigma^2}h(t). \end{aligned}$$

Therefore, if we let t_z denote any zero of $h(t)$, i.e., $h(t_z) = 0$, then

$$\begin{aligned} \left.\frac{\partial h(t)}{\partial t}\right|_{t=t_z} =& \frac{1}{\sigma^2}\sum_i\left[\int_{-\infty}^{-x_i}\xi^2 p_n(\xi - t_z)d\xi \right. \\ &\left. + \int_{-x_i}^{\infty}\xi^2 p_n(\xi + t_z)d\xi\right]dx - \frac{t}{\sigma^2}h(t_z) \\ =& \frac{1}{\sigma^2}\sum_i\left[\int_{-\infty}^{-x_i}\xi^2 p_n(\xi - t_z)d\xi \right. \\ &\left. + \int_{-x_i}^{\infty}\xi^2 p_n(\xi + t_z)d\xi\right]dx \\ >& 0. \end{aligned}$$

That means: 1) All zeros of $h(t)$ must be the minimum points of $J(t)$ because

$$\left.\frac{\partial^2 J(t)}{\partial t^2}\right|_{t=t_z} = \left.\frac{\partial h(t)}{\partial t}\right|_{t=t_z} > 0 \tag{34}$$

i.e., $t^* = t_z$. 2) Function $h(t)$ increases monotonically in the neighborhood of t_z. Since $h(t)$ is a continuous differentiable function of t, $h(t)$ cannot cross axis t more than once. Therefore, $h(t)$ has at most one zero, i.e., there is at most one minimum solution t^* for $J(t)$.

On the other hand, function $h(t|x)$ can be rewritten as

$$\begin{aligned} h(t|x) =& -\int_x^{\infty}\xi p_n(\xi + t)d\xi - \int_{-x}^{\infty}\xi p_n(\xi + t)d\xi \\ =& -\sigma^2\left[p_n(t - x) + p_n(t + x)\right] \\ &+ t\left[\int_x^{\infty}p_n(\xi + t)d\xi + \int_{-x}^{\infty}p_n(\xi + t)d\xi\right]. \end{aligned} \tag{35}$$

Obviously, if the noise level is zero, then $t^* = 0$. If the noise level is not zero, then $\forall\, t \le 0$, $h(t|x) < 0$, i.e., $h(0) < 0$. Since $t \ge 0$ and $(\partial h(t)/\partial t)|_{t=0} = 1 > 0$, then $t^* > 0$.■

APPENDIX B
PROOF OF THEOREM

Proof: From (30) and (31), we obtain

$$\begin{aligned} E\{t(i+1)|t(i), x_i\} =& t(i) - \alpha(i) \\ &\times\left[\int_{-\infty}^{-t(i)-x_i}(\xi + t(i))p_n(\xi)d\xi \right. \\ &\left. - \int_{t(i)-x_i}^{\infty}(\xi - t(i))p_n(\xi)d\xi\right] \\ =& t(i) - \alpha(i)\cdot h(t(i)|x_i) \end{aligned} \tag{36}$$

[2]Note that: $(\partial/\partial t)\int_{b(t)}^{a(t)}p(\xi,t)d\xi = p(b(t),t)(\partial b(t)/\partial t) - p(a(t),t)(\partial a(t)/\partial t) + \int_{b(t)}^{a(t)}(\partial p(\xi,t)/\partial t)d\xi$.

when $t(i) \geq 0$. From (30) and (32), we obtain

$$\begin{aligned} E\{t(i+1)|t(i),x_i\} =& t(i) - \alpha(i) \\ & \times \left[\int_{-\infty}^{t(i)-x_i} (\xi + t(i)) p_n(\xi) d\xi \right. \\ & \quad - \int_{-t(i)-x_i}^{\infty} (\xi - t(i)) p_n(\xi) d\xi \\ & \left. + 2t(i) \int_{t(i)-x_i}^{-t(i)-x_i} p_n(\xi) d\xi \right] \\ =& t(i) - \alpha(i) \\ & \times \left[\int_{-\infty}^{-t(i)-x_i} (\xi + t(i)) p_n(\xi) d\xi \right. \\ & \quad \left. - \int_{t(i)-x_i}^{\infty} (\xi - t(i)) p_n(\xi) d\xi \right] \\ =& t(i) - \alpha(i) \cdot h(t(i)|x_i) \end{aligned} \tag{37}$$

when $t(i) < 0$.

Hence

$$\begin{aligned} E\{t(i+1)|t(i)\} &= \int p_s(x) \cdot E\{t(i+1)|t(i),x\} dx \\ &= t(i) - \alpha(i) \cdot h(t(i)). \end{aligned} \tag{38}$$

Suppose there exists t^*. Then $|E\{t(i+1)|t(i)\} - t^*| = |t(i) - t^* - \alpha(i) \cdot h(t(i))|$. Since t^* is unique and $\partial h(t)/\partial t|_{t=t^*} > 0$ (see (34)), we obtain

$$\begin{cases} h(t) < & 0, t > t^* \\ h(t) < & 0, t > t^* \end{cases} . \tag{39}$$

Hence, when $t(i) \leq t^*$, $t^* - E\{t(i+1)|t(i)\} = t^* - t(i) - \alpha(i) \cdot |h(t(i))|$. If $\alpha(i)$ is selected such that $0 < \alpha(i) = \beta_i((t(i) - t^*)/h(t(i)))$ with constant $0 < \beta_i < 1$, we obtain

$$0 \leq t^* - E\{t(i+1)|t(i)\} = (1 - \beta_i)[t^* - t(i)]. \text{ if } t(i) \leq t^*. \tag{40}$$

On the other hand, if $t(i) \geq t^*$, then $E\{t(i+1)|t(i)\} - t^* = t(i) - t^* - \alpha(i) \cdot h(t(i))$. If $\alpha(i)$ is selected such that $0 < \alpha(i) = \beta_i((t(i) - ^*)/h(t(i)))$ with the same β_i as above, we obtain

$$\begin{aligned} 0 \leq E\{t(i+1)|t(i)\} - t^* =& (1 - \beta_i)[t(i) - t^*] \\ \text{if} \quad & t(i) \geq t^* \end{aligned} \tag{41}$$

Therefore, if a positive $0 < \alpha(i) = \beta_i(t(i) - t^*)/h(t(i))$ is selected, we obtain

$$\begin{aligned} |E\{t(i+1)\} - t^*| =& \left| \int_{-\infty}^{\infty} p_{t(i)}(t(i)) \right. \\ & \left. \times E\{t(i+1)|t(i)\} dt(i) - t^* \right| \\ =& \left| - \left\{ \int_{-\infty}^{t^*} p_{t(i)}(t(i)) \right. \right. \\ & \left. \left. \times [t^* - E\{t(i+1)|t(i)\}] dt(i) \right\} \right. \\ & + \left\{ \int_{t^*}^{\infty} p_{t(i)}(t(i)) \right. \\ & \left. \left. \times [E\{t(i+1)|t(i)\} - t^*] dt(i) \right\} \right| \\ =& (1 - \beta_i) \left| - \left\{ \int_{-\infty}^{t^*} p_{t(i)}(t(i)) \right. \right. \\ & \left. \left. \times [t^* - t(i)] dt(i) \right\} \right. \\ & + \left\{ \int_{t^*}^{\infty} p_{t(i)}(t(i)) \right. \\ & \left. \left. \times [t(i) - t^*] dt(i) \right\} \right| \\ =& (1 - \beta_i) \left| \left\{ \int_{-\infty}^{\infty} p_{t(i)}(t(i)) \right. \right. \\ & \left. \left. \times [t(i) - t^*] dt(i) \right\} \right| \\ =& (1 - \beta_i) |E\{t(i)\} - t^*|. \end{aligned} \tag{42}$$

Since $0 < (1 - \beta_i) < 1$, it follows

$$\lim_{i \to \infty} |E\{t(i)\} - t^*| = 0. \tag{43}$$

■

REFERENCES

[1] S. Hosur and A. H. Tewfik, "Wavelet transform domain adaptive FIR filtering," *IEEE Trans. Signal Processing*, vol. 45, pp. 617–630, Mar. 1997.

[2] N. Erdol and F. Basbug, "Wavelet transform based adaptive filters: analysis and new results," *IEEE Trans. Signal Processing*, vol. 44, pp. 2163–2171, Sept. 1996.

[3] D. L. Donoho and I. M. Johnstone, "Adaptating to unknown smoothness via wavelet shrinkage," *J. Amer. Statist. Assoc.*, vol. 90, no. 432, pp. 1200–1224, 1995.

[4] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard, "Wavelet shrinkage: Asymptopia?," *J. Roy. Statist. Soc. B.*, vol. 57, no. 2, pp. 301–337, 1995.

[5] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inform. Theory*, vol. 41, pp. 613–627, May 1995.

[6] G. P. Nason, "Wavelet shrinkage by cross-validation," *J. Roy. Statist. Soc. B*, vol. 58, pp. 463–479, 1996.

[7] A. Bruce and H.-Y. Gao, "Understanding WaveShrink: Variance and bias estimation," *Biometrika*, vol. 83, no. 4, 1996.

[8] C. Bielza and B. Vidakovic, "Time adaptive wavelet denoising Tech. Rep," Institute of Statistics and Decision Sciences, Duke University, 1996.

[9] R. R. Coifman and D. L. Donoho, "Translation-invariant denoising Tech. Rep ," Stanford Univ., Dept. Statist., 1995.

[10] M. Lang, H. Guo, J. E. Odegard, C. S. Burrus, and R. O. Wells, "Noise reduction using an undecimated discrete wavelet transform," *IEEE Signal Processing Lett.*, vol. 3, no. 1, pp. 10–12, 1996.

[11] I. M. Johnstone and B. W. Silverman, "Wavelet threshold estimators for data with correlated noise," *J. Roy. Statist. Soc. B*, vol. 59, pp. 319–351, 1997.

[12] F. Abramovich, T. Sapatinas, and B. W. Silverman, "Wavelet thresholding via a Bayesian approach," J. Roy. Statist. Soc. B, vol. 60, 1998, to be published.

[13] S. Haykin, *Neural Network: A Comprehensive Foundation*, 2 ed. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[14] X.-P. Zhang and M. Desai, "Adaptive denoising based on SURE risk," *IEEE Signal Processing Lett.*, vol. 10, no. 5, pp. 265–267, Oct. 1998.

[15] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, ser. Prentice-Hall Signal Processing Series. Englewood Cliffs, NJ: Prentice-Hall, 1985.
[16] S. Haykin, *Adaptive Filter Theory*, 3 ed, ser. Prentice-Hall Information and System Science Series. Englewood Cliffs, NJ: Prentice-Hall, 1996.
[17] C. Stein, "Estimation of the mean of a multivariate normal distribution," *Ann. Statist.*, vol. 9, no. 6, pp. 1135–1151, 1981.
[18] P. M. Clarkson, *Optimal and Adaptive Signal Processing*, ser. Electronic Engineering Systems Series. Boca Raton, FL: CRC, 1993.
[19] X. -P. Zhang and Z. Q. Luo, "A new time-scale adaptive denoising method based on wavelet shrinkage," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Phoenix, AZ, Mar. 1999.
[20] D. L. Donoho *et al.*. (1996) WaveLab .701. Stanford University. [Online]. Available: http://www-stat.stanford.edu/~wavelab/
[21] S. Mallat and W. L. Hwang, "Singularity detection and processing with wavelets," *IEEE Trans. Inform. Theory*, vol. 38, pp. 617–643, Mar. 1992.
[22] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, ser. McGraw-Hill Series in Systems Science. New York: McGraw-Hill, 1965.

Xiao-Ping Zhang (M'97) was born in Xinjiang, China, in 1972. He received the B.S., M.S., and Ph.D. degrees from Tsinghua University, Beijing, China, in 1992, 1993, and 1996, respectively, all in electronic engineering.

He was a Research Associate in High-Speed Signal Processing Laboratory and VHSIC Laboratory in Tsinghua University from 1990 to 1996. From 1996 to 1998, he was a Postdoctoral Fellow at the University of Texas, San Antonio, and then at the Beckman Institute, the University of Illinois at Urbana-Champaign. He held research and teaching positions at the Communication Research Laboratory, McMaster University, in 1999. From 1999 to 2000, he was a Senior DSP Engineer at SAM Technology at San Francisco, CA, and a Consultant at San Francisco Brain Research Insitute. He joined the Department of Electrical and Computer Engineering, Ryerson Polytechnic University, as an Assistant Professor in fall 2000. His research interests include neural networks and signal processing for communications, multimedia, and biomedical and bioinfomatics applications.

Dr. Zhang received Mei Yiqi Fellowship, Tsinghua University, in 1996. He was a recipient of Science and Technology Progress Award by State Education Commission of China for his significant contribution in a National High-Tech Project, in 1994.