A Stochastic Filtering Algorithm Using Schrödinger Equation

JIN Fu-Jiang^{1, 2} WU Hao-Han¹ LAI Lian-You² WANG Liang¹

Abstract This paper provides a new adaptive algorithm for single-step prediction by modeling the potential field of a one dimension Schrödinger wave equation using neural network. This new architecture is referred to as the recurrent quantum neural network (RQNN). The RQNN can filter the signal embedded with non-stationary noise without any priori knowledge of the shape of the signal and statistical properties of the noise. We compared the simulation results of the RQNN with a classical adaptive stochastic filter-RLS. It is shown that the RQNN is much more efficient in denoising signals embedded with Gaussian stationary, non-Gaussian stationary and Gaussian nonstationary noise such as DC, sinusoid, staircase and speech signals. The RQNN can enhance the signal to noise rate (SNR) by 20 dB, which is more than 10 dB given by the traditional technology when it denoising sinusoid signal.

Key words Adaptive filtering, quantum mechanics, recurrent quantum neural network (RQNN)

Citation Wu Hao-Han, Jin Fu-Jiang, Lai Lian-You, Wang Liang. A stochastic filtering algorithm using Schrödinger equation. Acta Automatica Sinica, 2014, 40(10): 2370-2376 DOI 10.3724/SP.J.1004.2014.02370

In modern communication, control and signal processing fields, the signals are almost always embedded with noises and the signal is also $\text{stochastic}^{[1-4]}$. So how to estimate the actual signal is very important. The classical stochastic filter such as the Wiener filter and Kalman filter cannot estimate the actual signal from the non-stationary noises [1-4]. The performance of the traditional adaptive filtering algorithm, such as the LMS, RLS and etc., is also very limited. We must find other adaptive stochastic filtering algorithm that can estimate the actual signal much more accurate. As we already know that the Kalman filter has been proved the best linear filter^[2], we should introduce nonlinear filtering algorithm. This paper provides a new nonlinear filtering algorithm by introducing the Schrödinger wave equation (SWE).

As we all know that the quantum mechanics is the best physical theory in the micro world for its introducing some basic assumptions^[5], i.e., the discontinuous energy of everything and the Schrödinger wave equation. These assumptions could not be proved just by the math until now. In quantum mechanics, the evolution of a micro object can be described by the Schrödinger wave equation which is the same as the Newton's second law in the macro low speed world^[5]. We also know that electronic signal is the results of a large number of electron transfers, which is a micro phenomenon, so their stochastic evolution could be described by the Schrödinger wave equation. If we can modulate the potential field of the Schrödinger wave equation properly, we can describe the evolution of the electronic accurately, then we can describe the signal accurately, which is the principle of the recurrent quantum neural network $(RQNN)^{[5]}$. Now our problem is how to modulate the potential field properly.

Bucy said that every solution to a stochastic filtering problem involves the computation of a time varying probability density function (PDF) on the state space of the observed system^[2], which in this paper is the x. In [6], Dawes gave the original theoretical architecture of the RQNN which was a parametric avalanche stochastic filter. In [7-9], Behera et al. improved Dawes's architecture by introducing the maximum likelihood estimate (MLE) to it. In [10-13], it was used in practice. This paper improves Behera's neural network with a new one which can insure the system is always steady to any parameters. This paper uses a recurrent neural network to modulate it and the weights of the neural network are updated by an unsupervised learning algorithm which is a variant of the classical Hebbian learning algorithm.

The remainder of the paper is organized into 6 sections. Section 1 describes the physical meaning of Schrödinger wave equation and the conceptual framework for the RQNN. Section 2 describes the principle of the RQNN. Section 3 describes the numerical implementation of the whole RQNN system. Section 4 describes how to select the parameters. Section 5 describes the simulation results and discussion. Section 6 concludes the paper.

Schrödinger wave equation and concep-1 tual framework for RQNN

The Schrödinger wave equation^[5] is

$$i\hbar \frac{\partial \psi(x,t)}{\partial t} = -\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} \psi(x,t) + V(x,t)\psi(x,t) \qquad (1)$$

where $2\pi\hbar$ is the Plank's constant; i is imaginary unit $\sqrt{-1}$; $\psi(x,t)$ (a vector in the Hilbert space) is the wave function or probability amplitude function whose square is the probability density function (PDF) of a quantum object at the space-time point (x,t); m is the mass of the object and V(x,t) is the potential energy. Equation (1) is one of the basic assumptions of quantum mechanics. It describes the evolution of a quantum object as the Newton's second law does in the macro world. Equation (1) also indicates that the quantum mechanics would never tell the clear position of a micro object except the PDF of it, which is the essence of quantum mechanics. According to (1), the evolution path of wave function can be set up by V(x, t), which is modulated by a recurrent neural network in this paper. A conceptive framework of the RQNN is shown in Fig. 1.



Fig. 1 Conceptual framework of the RQNN

Manuscript received October 16, 2013; accepted May 15, 2014 Supported by National Natural Science Foundations of China (61273069, 61203040), Fujian Significant Technology Projects for Cooperation of Industries and Universities (2013H61010135, 2011H6019), the Important Project of Technology Plan of Fujian Province (2009H0033), the Fundamental Research Funds for the Central Universities (JB-ZR1204) Recommended by Associate Editor LI Shao-Yuan

^{1.} College of Information Science and Engineering, Huaqiao Univer-ity, Xiamen 361021, China 2. College of Mechanical Engineering sity, Xiamen 361021, China and Automation, Huaqiao University, Xiamen 361021, China

In Fig. 1, y(t) is the input signal or the observed signal which is polluted by unknown noise. $\hat{y}(t)$ is the estimated signal by the MLE algorithm. $y(t) - \hat{y}(t)$ is the error signal which modulates the potential field for the Schrödinger wave equation. The range of the observed signal is defined as the observation space O_y . The space variable x in (1) is an arbitrary element of O_y but y(t) is just a certain element of O_y at a certain time t. In this paper, an imaginary quantum object is placed in O_y and the state of the object is described by wave function $\psi(x, t)$ which is controlled by V(x, t). Further describing of the RQNN will be done in the next section.

2 Theoretical analysis

The detailed schematic diagram is shown in Fig. 2



Fig. 2 Schematic diagram for RQNN in detail

In Fig. 2, y(t) is the observed signal embedded with unknown noise with a statistically zero mean; f(x, t) is a variance variable Gaussian kernel function modulated by y(t)and e(t), which is expressed as

$$f(x,t) = \exp\left(\frac{-(1+\alpha_1 \cdot e(t)^2)(y(t)-x)^2}{2\sigma^2}\right)$$
(2)

This means that the center of the kernel function will move with the changing of the observed signal, which is very important to tracking the actual signal embedded in the observed signal. At the same time, the variance of f(x,t)would change along with the changing of e(t), which could solve the contradiction between the stability and accuracy of the filter. We use the neural network to adjust the shape of the potential field again.

$$V(x,t) = \varsigma W(x,t) f(x,t) - \alpha_2 \cdot \rho(x,t)$$
(3)

If we regard f(x,t) as the input quantity and $1 + (e(t))^2$ as the output quantity of the neural network, a variant of the Hebbian learning algorithm^[10-13] can be used to update the weight of the network dynamically as

$$\frac{\partial W(x,t)}{\partial t} = -\beta_d W(x,t) + \beta f(x,t)(1 + (e(t))^2) \quad (4)$$

where

$$e(t) = y(t) - \hat{y}(t) \tag{5}$$

$$\hat{y}(t) = \arg \max_{x} \rho(x, t)$$
 (6)

In (4), β is the learning rate and β_d is the de-learning rate^[14-15]. Use de-learning to forget the previous information, as the input signal is not stationary. The values of the weights W(x, t) may keep growing indefinitely without

de-learning, which may cause the unstablility of the system. Both the Kernel function and the neural network are used to provide Schrödinger wave equation with a suitable potential field. Especially, the kernel function makes it sure that the wave packet would not pirate, that is to say, the quantum object moves like a particle or the soliton characteristic is very good. The network increases the tracking capability of the RQNN dynamically. Of cause they interact each other any time anywhere. The function of the potential field in the Schrödinger wave equation is to drive the wave function to the most suitable location which is the lowest place in energy of the potential field^[5]. Fortunately, it cannot be finished instantaneously, that is to say, it is a process of evolution which in fact is the most natural way of evolution of the actual signal from the physical point of view. It does not change too fast from the mathematic point of view. So the noise, which has too little contact with itself or changes too $fast^{[1-4]}$, would be filtered without too much loss of the actual signal which is strongly correlated and won't change too fast. In other words, the Schrödinger wave equation is a related extractor. This is the reason why the RQNN can be a stochastic filter. At last the RQNN provides us a new way to estimate the PDF of the noise if we modulate the potential field in a more proper way. After we get the wave equation, we can get the PDF.

3 Numerical implementation of the whole RQNN system

Section 2 explained the principle of RQNN. As the RQNN must use a digital computer to solve all the equations introduced in section 2, we have to give the numerical implementation of the RQNN. The discrete framework of the RQNN is shown in Fig. 3.



Fig. 3 Discrete framework of RQNN

In a discrete system, the arguments should be expressed as

$$t = n \cdot dt, \quad n = 0, 1, 2, \cdots$$

$$x = \min(x) + k\Delta x \quad k = 0, 1, 2, \cdots, N-1$$

$$\boldsymbol{X} = [\min(x) + \Delta x, \min(x) + 2\Delta x, \cdots, \min(x) + (N-1)\Delta x]^{\mathrm{T}}$$

where dt is the sampling interval. The discrete kernel function is a vector of dimension N.

$$\boldsymbol{f}(\boldsymbol{X},n) = \exp(-(1+\alpha_1 e(n)^2)(\boldsymbol{X}-y(n)).^2/2\sigma^2) \quad (7)$$

Where the ".²" means each element of a matrix square. Then the potential field function should be

ι

$$V(n+1) = \boldsymbol{W}(n+1) \cdot \boldsymbol{f}(\boldsymbol{X}, n) - \alpha_2 \boldsymbol{\rho}(n)$$
(8)

where

Y

$$\boldsymbol{W}(n+1) = (1 + \beta_d \Delta t) \boldsymbol{W}(n) + \beta \boldsymbol{f}(\boldsymbol{X}, n) (1 + e(n)^2) \quad (9)$$

$$e(n) = y(n) - \hat{y}(t) \tag{10}$$

Where "..." means multiplying elements of two matrices in the same position. Then we can get $\boldsymbol{\psi}(n+1)$ by Schrödinger wave equation which will be discussed later. Then we could get the PDF

$$\boldsymbol{\rho}(n+1) = |\boldsymbol{\psi}(n+1)|.^2 \tag{11}$$

Use maximum likelihood estimation (MLE) to estimate the actual signal as

$$\stackrel{\wedge}{y}(n+1) = (\arg \max_{1 \le i \le N} \boldsymbol{\rho}(n+1)_i) \cdot \Delta x + \min(x) \quad (12)$$

Where $\boldsymbol{\rho}(n+1)_i$ is the *i*th member of vector $\boldsymbol{\rho}(n+1)$ and we should insure that $\sum_{i=1}^{N-1} \boldsymbol{\rho}(n+1)_i \Delta x = 1$. Now let us concentrate on the numerical form of the Schrödinger wave equation (SWE). As the actual processor does not have plural handling capability and the Schrödinger wave equation is a complex equation, we should transform it into the form of a real number equation. Assume the *j*th member of vector $\boldsymbol{\psi}(n)$ is $\psi_j^n = \psi_{-} \operatorname{real}_j^n + \mathrm{i}\psi_{-} \operatorname{imag}_j^n$. Then the SWE could be realized as

$$i\hbar \frac{\psi_j^{n+1} - \psi_j^n}{\Delta t} = -\hbar^2 \frac{\psi_{j+1}^n - 2\psi_j^n + \psi_{j-1}^n}{2m \left(\Delta x\right)^2} + V_j^n \psi_j^n, \quad (13)$$
$$\psi_1^n = \psi_N^n = 0, \quad j = 1, 2, \cdots, N$$

Here we regard Δt as a new parameter which is different from dt generally. The form of real number equation is

$$\psi_{-}\mathrm{real}_{j}^{n+1} = \psi_{-}\mathrm{real}_{j}^{n} - \frac{\Delta t\hbar}{2m\left(\Delta x\right)^{2}} \left(\psi_{-}\mathrm{imag}_{j+1}^{n} - 2\psi_{-}\mathrm{imag}_{j}^{n} + \psi_{-}\mathrm{imag}_{j-1}^{n}\right) + \frac{\Delta t}{\hbar} V_{j}^{n} \psi_{-}\mathrm{imag}_{j}^{n} \qquad (14)$$

$$\psi_{\perp} \operatorname{imag}_{j}^{n+1} = \psi_{\perp} \operatorname{imag}_{j}^{n} + \frac{\Delta t \hbar}{2m \left(\Delta x\right)^{2}} \left(\psi_{\perp} \operatorname{real}_{j+1}^{n} - 2\psi_{\perp} \operatorname{real}_{j}^{n} + \psi_{\perp} \operatorname{real}_{j-1}^{n}\right) - \frac{\Delta t}{\hbar} V_{j}^{n} \psi_{\perp} \operatorname{real}_{j}^{n} \qquad (15)$$

According to Taylor series, the order of the truncation error is $(O((\Delta t)^2) + O((\Delta x)^2))$ in (13). As any digital calculator has the finite word length effect, the rounding error would be accumulated. The error accumulation may make the results beyond recognition, which is so called unstable. If Δx and Δt obey a proper condition to vanish the initial error as time goes on, equation (13) converges to equation (1). The properly condition is the so called stable condition and it has already been proved in [16]. It is

$$\frac{2\hbar\Delta t}{m(\Delta x)^2} + \frac{V(x,t)\Delta t}{\hbar} \le 2 \tag{16}$$

Thanks to the de-learning rate β_d , the item $\frac{V(x,t)\Delta t}{\hbar}$ can be ignored. Thus, (16) can be simplified as $\frac{\hbar\Delta t}{m(\Delta x)^2} \leq \frac{1}{4}$.

4 How to select parameters

Although the RQNN architecture proposed in this paper is not easy to be unstable to the parameters, the values of parameters are also very important to the performance of it. Here we add another new parameter γ to adjust the conflict between tracking performance and smooth performance. The new one is defined as the iteration time for each single observed signal. The others have been introduce in the above sections such as β , β_d , Δt , m and ς . We use the traversal method to find the proper γ first, then we use generation algorithm (GA) to find the others. As the word length limit of the digital computer, the mass of the quantum object m and the Plank's constant $2\pi\hbar$ have been enlarged in this paper.

5 Simulation results and discussion

In this section, we will compare the performance of RQNN adaptive filtering algorithm with a traditional linear adaptive algorithm, RLS. We will test them in both stationary and non-stationary noise environments. The observed signal is always

$$y(t) = y_a(t) + n(t) \tag{17}$$

Where $y_a(t)$ is the actual signal and n(t) is the noise signal in all the simulation cases.

5.1 DC signal denoising

Here we set up the noise with stationary and nonstationary high Gaussian noise n(t). In the stationary case, the SNRs are 20 dB, 6 dB and 0 dB, respectively. The optimal values of the parameters are dt = 0.001, $\Delta t =$ 0.001, $\beta = 0.57$, m = 0.327, $\varsigma = 0.859$, $\beta_d =$ 0.671, $\gamma = 1$, $\alpha_1 = 1.794$, $\alpha_2 = -1.253$, $\hbar = 1.0$, $\Delta x =$ 0.1, $\min(x) = -20$, N = 400.

The actual DC signal is $y_a(t) = 2$, $0 \le t \le 20$ s. The results for the RQNN and RLS to filter the signal whose SNR is 6 dB with stationary noise are shown in Figs. 4 and 5, respectively.



Fig. 4 RQNN filtering results for 6 dB DC signal ("a" represents the raw signal, "b" represents the estimated signal by RQNN, "c" represents the actual signal.)



Fig. 5 RLS filtering result for 6 dB DC signal ("a" represents the raw signal, "b" represents the estimated signal by RLS, "c" represents the actual signal.)

From Figs. 4 and 5 we can see that the accuracy of RQNN adaptive filtering is much better, but the RQNN needs more time to find the actual signal which will be explained in the end of this section. For a more accurate comparison between the performances of the two filtering algorithms, we give the output signal to noise rate (SNR) directly for the input SNR 0 dB, 6 dB and 20 dB, respectively. The output SNRs in filtering the DC signal by RQNN and RLS are shown in Table 1.

 Table 1
 Performance comparison for DC signal in stationary environment

Input SNR	RLS SNR	RQNN SNR
0 dB	10.06 dB	44.1445 dB
6 dB	$19.9625~\mathrm{dB}$	$47.7262~\mathrm{dB}$
20 dB	33.68 dB	$62.8978~\mathrm{dB}$

It is very clear that the performance of the RQNN is much better than that of the traditional adaptive algorithm. Now let us compare their performances in the nonstationary noise environment. Figs. 6 and 7 display the results of RQNN and RLS in filtering DC signal which is embedded with non-stationary signal. The input SNR is 6 dB.



Fig. 6 RQNN for non-stationary noise ("a" represents the raw signal, "b" represents the estimated signal by RQNN, "c" represents the actual signal.)



Fig. 7 RLS for non-stationary case ("a" represents the raw signal, "b" represents the estimated signal by RLS, "c" represents the actual signal.)

The output SNRs of the two filtering algorithms are shown in Table 2.

 Table 2
 Performance comparison for DC signal in non-stationary environment

Input SNR	RLS SNR	RQNN SNR	
$7.4011~\mathrm{dB}$	$21.0653~\mathrm{dB}$	$48.0523~\mathrm{dB}$	

From Fig. 6, Fig. 7 and Table 2, we can see that the results of RQNN are not only more accurate but also much more smooth than the results of RLS. Compare Fig. 6 with Fig. 4, we can see that the non-stationary noise environment has little effect on the result of RQNN, that is, RQNN has better adaptability.

5.2 Sinusoid signal denoising

The original sinusoid signal is

$$y_a(t) = 2\sin(2\pi \times 20 \times t), \quad 0 \le t \le 0.4 \,\mathrm{s}$$
 (18)

The values of parameters of RQNN are dt = 0.0001, $\Delta t = 0.001$, $\beta = 0.57$, m = 0.327, $\varsigma = 0.859$, $\beta_d = 0.671$, $\gamma = 43$, $\alpha_1 = 1.794$, $\alpha_2 = -1.253$, $\hbar = 1.0$, $\Delta x = 0.1$, $\min(x) = -20$, N = 400.

The number of points along the x-axis is taken as N = 400. The SNR of the input signal is 6 dB. The result is shown in Figs. 8 and 9.



Fig. 8 RQNN for sinusoid signal ("a" represents the raw signal, "b" represents the estimated signal by RQNN, "c" represents the actual signal.)



Fig. 9 RLS for sinusoid signal ("a" represents the raw signal, "b" represents the estimated signal by RLS, "c" represents the actual signal.)

The SNRs are shown in Table 3.

 Table 3
 Performance comparison for sinusoid signal

Input SNR	RLS SNR	RQNN SNR	
6 dB	$16.2746~\mathrm{dB}$	$28.4722~\mathrm{dB}$	

To see the accuracy of the RQNN from another angle, we plot the power spectral densities (PSDs) of the actual signal, observed signal and estimated signal, respectively in Fig. 10.



Fig. 10 The PSD comparision ((a) The PSD of actual signal; (b) The PSD of observed signal; (c) The PSD of estimated signal)

From Fig. 10, we see that not only the PSD of the estimated signal is very similar to the PSD of the actual signal but also a lot of noise is attenuated. That is to say, the filtering result is very accurate. A non-stationary noise case is shown in Fig. 11 and Fig. 12.

The output SNRs of the two filtering algorithms are shown in Table 4.

From Fig. 8, Fig. 9, Fig. 11, Fig. 12 and Table 4, we can also see that the conclusion obtained in the DC signal case is also suitable to the sinusoid signal case. That is to say, RQNN has better adaptability.



Fig. 11 RQNN filtering sinusoid signal in non-stationary noise ("a" represents the raw signal, "b" represents the estimated signal by RQNN, "c" represents the actual signal.)



Fig. 12 RLS filtering sinusoid signal in non-stationary noise ("a" represents the raw signal, "b" represents the estimated signal by RLS, "c" represents the actual signal.)

Table 4 Performance comparison for sinusoid signal in non-stationary environment

Input SNR	RLS SNR	RQNN SNR	
$7.8372~\mathrm{dB}$	$16.6084~\mathrm{dB}$	$26.0483~\mathrm{dB}$	

5.3 Non-Gaussian noise for sinusoid signal denoising

As the RQNN does not need any assumption of statistical properties of noise except the zero mean, it should perform very well for any zero mean non-Gaussian noise. Here we use the sinusoid signal embedded with uniformly distributed noise. The observe signal is

 $y(t) = y_a(t) + n(t), \quad 0 \le t \le 0.4 \,\mathrm{s}$

where

(19)

$$y_a(t) = 2\sin(2\pi \times 20 \times t), \quad 0 \le t \le 0.4 \,\mathrm{s}$$
 (20)

n(t) is the uniform noise signal. The values of parameters of RQNN are dt = 0.0001, $\Delta t = 0.001$, $\beta = 0.57$, m = 0.327, $\varsigma = 0.859$, $\beta_d = 0.671$, $\gamma = 43$, $\alpha_1 = 1.794$, $\alpha_2 = -1.253$, $\hbar = 1.0$, $\Delta x = 0.1$, $\min(x) = -20$, N = 400.

The non-Gaussian noise case is shown in Fig. 13.



Fig. 13 RQNN filter staircase signal ("a" represents the raw signal, "b" represents the estimated signal by RQNN, "c" represents the actual signal.)

The SNR of the raw signal is 7.713 dB and the SNR of the filtered signal is about 24.5 dB. From Fig. 13, we can see that the filtered signal is also very smooth. These verify that the RQNN does not need any assumption of the noise, which has been detailed in Section 2.

5.4 Staircase signal denoising

As the staircase signal changes very quickly at some specific point and many stochastic filters such as LMS cannot track this signal. This part we do not compare the performance of RQNN with RLS in detail, but just show that the RQNN can also track the staircase signal. The SNR of the raw signal is set 20 dB. The values of RQNN's parameters are dt = 0.001, $\Delta t = 0.001$, $\beta = 0.57$, m =0.327, $\varsigma = 0.859$, $\beta_d = 0.671$, $\gamma = 90$, $\alpha_1 = 1.794$, $\alpha_2 =$ -1.253, $\hbar = 1.0$, $\Delta x = 0.1$, $\min(x) = -20$, N = 400.

The result of filtering staircase signal is shown in Fig. 14.



Fig. 14 RQNN filter staircase signal ("a" represents the raw signal, "b" represents the estimated signal by RQNN, "c" represents the actual signal.)

The output SNR is about 28.304 dB. We can see from Fig. 14 that its track property is very good if we select proper parameters.

5.5 Speech signal denoising

The speech signal is quite the same as white noise and changes very fast. The traditional adaptive filter cannot work. Let's select the values of the parameters as follows: dt = 1.2207e - 04, $\Delta t = 100 \times dt$, $\beta = 0.57$, m =

0.327, $\varsigma = 0.859$, $\beta_d = 0.671$, $\gamma = 1$, $\alpha_1 = 1.794$, $\alpha_2 = -1.253$, $\hbar = 1.0$, $\Delta x = 0.1$, $\min(x) = -20$, N = 400.

The result of filtering speech signal is shown in Fig. 15.



Fig. 15 RQNN filter speech signal ("a" represents the raw signal, "b" represents the estimated signal by RQNN, "c" represents the actual signal.)

The output SNR is 1.9442 dB with raw SNR = -0.3 dB. The result also means that the tracking property of RQNN is much better than that of any traditional filter. RLS which has the best tracking property in traditional filters does not work in the speech signal case with the current sampling frequency 1/dt. It is necessary to talk about the new added parameter γ which has already been defined in section 4. Here we explain the function of it. We can see from the result of each figure that a big γ means the output signal changes much faster than the lower SNR. In the filtering field, tracking signal always is the most important thing. This is the reason why we should find a proper γ first, then use GA to find other parameters.

6 Conclusion

This paper presents a new architecture for stochastic filtering, in which a one dimensional Schrödinger wave equation that can abstract the strong relevant parts in the observed signal embedded with weakly correlated noise signal is introduced to denoise the stationary and non-stationary weakly correlated noise signal. The potential field of the Schrödinger wave equation is set up with a variable variance Gaussian kernel function and a linear neural network whose weights are updated by a variant form of the Hebbian learning algorithm. The Gaussian kernel function whose center changes with the observed signal maintains the moving of the wave packet and the neural network maintains the soliton property of the wave packet, which is the premise of the moving of the wave packet and also provides the probability of noise PDF online estimation. We use the new architecture of RQNN to denoise the DC signal, sinusoid signal, staircase signal and speech signal in the stationary and non-stationary noise environments without the prior knowledge about the shape of the actual signal or properties of the noise. The result of the RQNN is much better both in the accuracy and the adaptivity compared to the RLS filter. In the future, we hope that the RQNN is not only a stochastic filter but also a non-stationary noise PDF estimator.

Acknowledgement

The authors would like to acknowledge Laxmidhar Behera for his selfless help.

References

- Feng Bo, Ma Hong-Bin, Fu Meng-Yin, Wang Shun-Ting. A framework of finite-model Kalman filter with case study: MVDP-FMKF algorithm. Acta Automatica Sinica, 2013, 39(8): 1246-1256 (in Chinese)
- 2 Busy R S. Linear and nonlinear filtering. Proceedings of the IEEE, 1970, 58(6): 854–864
- 3 Ge Quan-Bo, Li Wen-Bin, Xu Zi. Centralized fusion algorithms based on EKF for multisensor non-linear systems. Acta Automatica Sinica, 2013, 36(6): 816-825 (in Chinese)
- 4 Zhang Yong-Gang, Huang Yu-Long, Wu Zhe-Min, Li Ning. A high order unscented kalman filtering method. Acta Automatica Sinica, 2014, 40(5): 838-848 (in Chinese)
- 5 Cohen C, Diu B, Laloë F [Author], Liu Jia-Mo, Chen Xing-Kui [Translator]. Quantum Mechanics. Beijing: Higher Education Press, 1987.
- 6 Dawes R L. Quantum neurodynamics: neural stochastic filtering with the schroedinger equation. In: Proceedings of the 1992 International Joint Conference on Neural Networks. Baltimore, MD USA: IEEE, 1992. 133–140
- 7 Behera L, Kar I. Quantum stochastic filtering. In: Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics. The Big Island, Hawaii, USA: IEEE, 2005. 2161–2167
- 8 Behera L, Kar I, Elitzur A C. A recurrent quantum neural network model to describe eye tracking of moving targets. Found of Physics Letters, 2005, **18**(4): 357–370
- 9 Behera L, Kar I, Elitzur A C. The Emerging Physics of Consciousness: Recurrent Quantum Neural Network and Its Applications. Berlin: Springer, 2006.
- 10 Gandhi V, Arora V, Behera L, Prasad G, Coyle D H, McGinnity T M. A recurrent quantum neural network model enhances the EEG signal for an improved brain-computer interface. In: Proceedings of the 2011 IET Seminar on Assisted Living. London, UK: IET, 2011. 1–6
- 11 Gandhi V, Prasad G, Coyle D H, Behera L, McGinnity T M. A novel EEG signal enhancement approach using a recurrent quantum neural network for a brain-computer interface. In: Technically Assisted Rehabilitation. Berlin, Germany: 2011
- 12 Gandhi V, Prasad G, Coyle D H, Behera L, McGinnity T M. Quantum neural network-based EEG filtering for a braincomputer interface. *IEEE Transactions on Neural Networks* and Learning Systems, 2014, **25**(2): 278–288
- 13 Gandhi V, McGinnity T M. Quantum neural network based surface EMG signal filtering for control of robotic hand. In: Proceedings of the 2013 International Joint Conference on Neural Networks. Dallas, USA: IEEE, 2013. 1-7
- 14 Haykin S [Author], Shen Fu-Rao, Xu Ye, Zheng Jun, Chao Jing [Translator]. Neural Networks and Learning Machines (Third Edition). Beijing: China Machine Press, 2011.
- 15 Zhang Yong-Gang, Wang Cheng-Cheng, Wei Ye, Li Ning, Zhou Wei-Dong. A spatially distributed variable tap-length strategy over adaptive networks. Acta Automatica Sinica, 2014, 40(7): 1355-1365 (in Chinese)
- 16 Kuang Xiao-Jing, Wu Xian-Liang, Huang Zhi-Xiang, Wang Dao-Ping. Solving time-dependent Schrödinger equation based on FDTD method. In: Proceedings of the 2009 National Conference on Microwave and Millimeter Wave. Xi'an, China: Publishing House of Electronics Industry, 2009. 990–993

WU Hao-Han Master student at College of Information Science and Engineering, Huaqiao University. His main research interest is quantum stochastic filtering algorithm. Corresponding author of this paper. E-mail: wuhaohan123@sina.com **JIN Fu-Jiang** Professor at College of Information Science and Engineering, Huaqiao University. His reservent interest covers control and optimization of complex industrial processes. E-mail: jinfujiang@163.com

LAI Lian-You Ph. D. candidate at College of Mechanical Engineering and Automation, Huaqiao University. His research interest covers quantum stochastic filtering and stochastic model of sea wave. E-mail: kaikaixinxinlly@163.com

WANG Liang Master student at College of Information Science and Engineering, Huaqiao University. His reserved interest covers CT image reconstruction. E-mail: wl15856255232@163.com