Quantum Stochastic Filtering

Laxmidhar Behera * and Indrani Kar †
Department of Electrical Engineering,
Indian Institute of Technology, Kanpur,
208 016, UP, INDIA
* lbehera@iitk.ac.in † indranik@iitk.ac.in

***Abstract*— This paper presents a new paradigm for stochastic filtering by modeling the unified response of a neural lattice using the Schroedinger wave equation. The model is based on a novel concept that a quantum object mediates the collective response of a neural lattice. The model is referred as recurrent quantum neural network (RQNN). The RQNN model has been simulated in two different ways. In one case the potential field of the Schroedinger wave equation is linearly modulated and in the other case the potential field of the Schroedinger wave equation is nonlinearly modulated. It is shown that the proposed quantum stochastic filter can efficiently denoise signals such as DC, sinusoid, amplitude modulated sinusoid and speech signals embedded in very high Gaussian and non-Gaussian noises. Performance of linearly modulated RQNN compares well with traditional techniques such as Kalman filter and wavelet filter. However, preliminary results show that nonlinearly modulated RQNN performs much better when compared with traditional techniques. For example, nonlinearly modulated RQNN model denoises a DC signal 1000 times more accurately in comparison to a traditional Kalman filter. The most important fact is that the proposed quantum stochastic filter does not make any assumption about the shape and nature of the signal and noise when denoising a signal. In a sense, the proposed quantum stochastic filter is a step forward towards intelligent filtering.**

I. Introduction

The systemic understanding of cognitive processes, consists of the formulation and solution of three fundamental problems, in the design of intelligent machines that 'intelligently' observe, predict and interact with the objects of the physical world. These problems can be viewed as the system identification problem, the stochastic filtering problem, and the adaptive control problem. This paper addresses the issue of intelligent stochastic filtering problem.

Instead of considering one of the conventional neural architectures [1], [2], [3], [4], [5], an alternative neural architecture 'recurrent quantum neural network' (RQNN) is proposed for intelligent stochastic filtering. To this end, we envision a neural lattice actuating a spatio-temporal potential field to a given stimulus. In this paper, the neural lattice is confined to a one dimensional spatial structure where each neuron is a simple computational unit. A hypothetical quantum object then mediates this spatio-temporal potential field. The idea is to represent the collective information processing of neurons by a unified quantum activation function. The proposed recurrent quantum neural network provides a framework in which the *pdf* information of an observed stochastic variable $y(t)$ can be transferred to the $\psi(.)$ function, the solution of the Schroedinger wave equation, in an unsupervised manner. The motivation to select Schroedinger wave equation as the quantum activation function came from the work by Robert Dawes [6], [7].

Interestingly, the closed form dynamics of the recurrent quantum neural network (equation (12)) closely resembles a nonlinear Schroedinger wave equation with cubic nonlinearity studied in quantum electrodynamics ([8]):

$$i\hbar\frac{\partial\psi(x,t)}{\partial t} = \left(-\frac{\hbar^2}{2m}\nabla^2 - \frac{e^2}{r}\right)\psi(x,t) + e^2\int\frac{\psi(x,t)\mid\psi(x',t)\mid^2}{\mid x-x'\mid}dx' \tag{1}$$

where $2\pi\hbar$ is the Planck's constant, $\psi(x,t)$ is the wave (probability amplitude) function associated with the electron at space-time point(x,t), m is the electron mass, e is the elementary charge and r is the magnitude of $\mid x \mid$. Also nonlinear Schroedinger wave equations with cubic nonlinearity of the form $\frac{\partial}{\partial t}\mathcal{A}(t) = c_1\mathcal{A} + c_3 \mid \mathcal{A} \mid^2 \mathcal{A}$, where c_1 and c_3 are constants, frequently appear in nonlinear optics [9] and in the study of solitons [10], [11]. At the moment, the proposed RQNN is based on single dimensional Schroedinger wave equation and this nonlinear form exhibits soliton property for some range of parameters. We have been able to use this property for stochastic filtering.

Two types of RQNN have been considered here. First one uses a linear neural circuit to set up the potential field in which the quantum object is dynamically excited, while the second one uses a nonlinear neural circuit. This fundamental change in the architecture has resulted in two new phenomena. The wave packets, $f(x,t) = \mid \psi(x,t) \mid^2$ are moving like a particle. Here $\psi(x,t)$ is the solution of the nonlinear Schroedinger wave equation. The other very interesting observation is that movements of wave packets while tracking a fixed target are not continuous but discrete. These observations remind us one of the saccadic movements of eye [12]. In a recent paper, we have explored this idea to provide a theoretical model of eye tracking [13].

In simulation section, it is demonstrated that linearly modulated RQNN can efficiently denoise complex signals without any assumption about the nature and shape of both signal and noise. Accuracy of denoising a DC signal at 0dB noise has been compared with a Kalman filter and it has been shown that the proposed quantum stochastic filter can be 1000 times more accurate.

II. The proposed conceptual model of RQNN

A conceptual framework concerning the functioning of the proposed recurrent quantum neural network is given in figure 1. In this figure, it is shown that a quantum process models average behavior (collective response) of a neural lattice. The time evolution of the collective response ψ is described by the Schroedinger wave equation(SWE):

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

where symbols such as i and ∇ carry usual meaning in the context of Schroedinger wave equation.

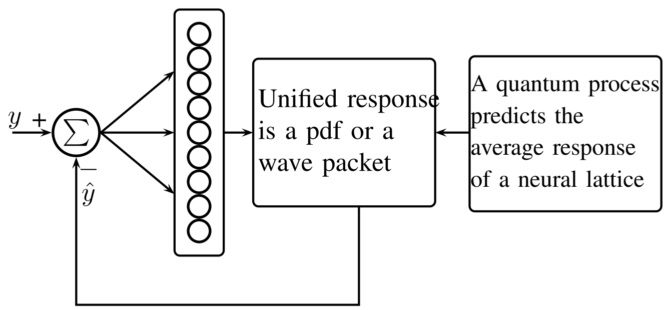

Fig. 1. Conceptual framework for the Recurrent Quantum Neural Networks

Here a neuronal lattice sets up a spatial potential field $V(x)$. A quantum process described by a quantum state ψ which mediates the collective response of a neuronal lattice evolves in this spatial potential field $V(x)$ according to (2). As $V(x)$ sets up the evolution path of the wave function, any desired response can be obtained by modulating the potential field properly. But here we only focus on the stochastic filtering application which is described in the next section.

III. A Stochastic Filter using RQNN

According to R.S. Bucy [14], every solution to a stochastic filtering problem involves the computation of time varying probability density function *(pdf)* on the state space of the observed system. Using the same concept, an architecture of RQNN is proposed where this *pdf* information of a stochastic variable can be transferred to $\psi(.)$, the wave amplitude function, of the Schroedinger wave equation. Along the spatio-temporal evolution of $\psi(.)$ function, the actual *pdf* is computed using the fact that the square of the modulus of the ψ function, the solution of this wave equation, is also a *pdf* which is denoted as $\rho(.)$ in the subsequent sections. The proposed RQNN architecture is shown in figure 2.

A. RQNN with Linear Modulation

When a linear neural circuit modulates the potential field, it is expressed as

$$V(x,t) = \zeta(U(x,t) + G(\mid\psi\mid^2)) \qquad (3)$$

where

$$U(x,t) = -K(x,t)y(t) \qquad (4)$$

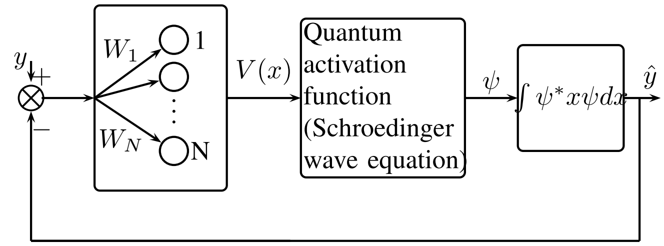

Fig. 2. A stochastic filter using linearly modulated RQNN

$$G(\mid\psi\mid^2) = K(x,t)\int x\rho(x,t)dx \qquad (5)$$

$$\rho(x,t) = \mid\psi(x,t)\mid^2 \qquad (6)$$

Since the potential field term in equation (2) is a function of $\psi(x,t)$, the Schroedinger wave equation that describes the stochastic filter is nonlinear. In contrast to artificial neural networks studied in literature, in our model the neural lattice consisting of N neurons is described by the state $\psi(x,t)$ which is the solution of equation (2). Simultaneously, the model is recurrent as the dynamics consists of a feedback term $G(.)$. When the estimate $\hat{y}(t)$ is the actual signal, then the signal that generates the potential field for the Schroedinger wave equation, $\hat{\nu}(t)$, is simply the noise that is embedded in the signal. If the statistical mean of the noise is zero, then this error correcting signal $\hat{\nu}(t)$ has little effect on the movement of the wave packet. Precisely, it is the actual signal content in the input $y(t)$ that moves the wave packet along the desired direction which, in effect, achieves the goal of the stochastic filtering. It is expected that the synaptic weights evolve in such a manner so as to drive the ψ function to carry the exact information of the *pdf* of the observed stochastic variable $y(t)$.

1) Learning and Estimation: The nonlinear Schroedinger wave equation given by equation (2) exhibits soliton property, i.e. the square of $\mid\psi(x,t)\mid$ is a wave packet which moves like a particle. The importance of this property is explained as follows. Let the stochastic variable $y(t)$ be described by a Gaussian probability density function $\rho(x,t)$ with mean κ and standard deviation σ. Let the initial state of equation (2) correspond to zero mean Gaussian probability density function $\rho'(x,t)$ with standard deviation σ'. As the dynamics evolves with on-line update of the synaptic weights $K(x,t)$, the probability density function $\rho'(x,t)$ should ideally move toward the *pdf*, $\rho(x,t)$, of the signal $y(t)$. Thus the filtering problem in this new framework can be seen as the ability of the nonlinear Schroedinger wave equation to produce a wave packet solution that glides along with the time varying *pdf* corresponding to the signal $y(t)$.

The synaptic weights $K(x,t)$ which is a $N\times 1$ dimensional vector is updated using Hebbian learning algorithm.

$$\frac{\partial K(x,t)}{\partial t} = \beta\nu(t)\rho(x,t) \qquad (7)$$

where $\nu(t) = y(t) - \hat{y}(t)$. $\hat{y}(t)$ is the filtered estimate of the actual signal $y_a(t)$. We compute the filtered estimate as Maximum Likelihood Estimator *(MLE)*:

$$\hat{y}(t) = \int x\rho(x,t)dx \tag{8}$$

In the stochastic filter model given by Dawes ([6]), the author used a inverse filter in the feedback. Using this model we could not move the wave packet and the author agreed to this finding in our personal correspondence. We will see later in this paper that the wave packet moves in the required direction in our new model.

B. RQNN with Nonlinear Modulation

In this case the potential field of the Schroedinger wave equation is set up using a nonlinear neural circuit. The architecture is shown in figure 3. With this modification the potential field can now be expressed as:

$$V(x,t) = \sum_{i=1}^{n} W_i(x,t)\phi_i(\nu(t)) \tag{9}$$

where $\phi_i(.)$ is a Gaussian Kernel function and n represents the number of such Gaussian functions to describe the nonlinear map that represent the synaptic connections. $\nu(t)$ represents difference between the input signal y and the filtered estimate $\hat{y}$. The Gaussian kernel function is taken as:

$$\phi_i(\nu(t)) = exp(-(\nu(t) - g_i)^2) \tag{10}$$

where g_i is center of the i^{th} Gaussian function, ϕ_i. This is chosen from input space through uniform random sampling.

The filtered estimate is calculated as:

$$\hat{y}(t) = \int x(t)\rho(x,t)dx \tag{11}$$

Based on this estimate, the wights are changed thus establishing a new potential field for the next time evolution.

The closed form dynamics of this recurrent quantum neural network as described in figure 3 thus becomes:

$$\begin{aligned} i\hbar\frac{\partial\psi(x,t)}{\partial t} &= -\frac{\hbar^2}{2m}\nabla^2\psi(x,t) + \\ &\zeta G\left(y(t) - \int x \mid \psi(x,t) \mid^2 dx\right)\psi(x,t) \end{aligned} \tag{12}$$

where $G(.)$ is a Gaussian kernel map introduced to nonlinearly modulate the spatial potential field that excites the dynamics of the quantum object as described earlier. In fact $\zeta G(.) = V(x,t)$ where $V(x,t)$ is given in equation (9).

1) Learning and Estimation: In equation (12), the unknown parameters are weights $W_i(x,t)$ associated with Gaussian kernel. It is expected that the synaptic weights evolve in such a manner so as to drive the ψ function to carry the exact information of the *pdf* of the observed stochastic variable $y(t)$. To achieve this goal the weights are updated using the following learning algorithm

$$\frac{\partial W_i(x,t)}{\partial t} = \beta\phi_i(\nu(t))\rho(x,t) \tag{13}$$

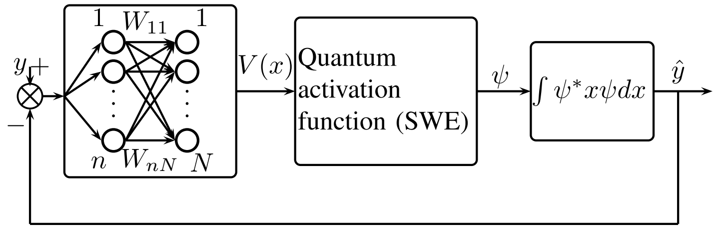

Fig. 3. RQNN architecture with nonlinear neural circuit

where $\nu(t) = y(t) - \hat{y}(t)$.

IV. Integration of the Schroedinger wave equation

The nonlinear Schroedinger wave equation is – from the mathematical point of view – a partial differential equation in two variables: x and t. In an abstract sense receptive fields of N neurons span the entire distance along the x-axis. The equation (2) is converted to the finite difference form by dividing the x-axis into N mesh points so that x and t are represented as follows:

$$x_j = j\triangle x \quad t_n = n\triangle t \tag{14}$$

where j varies from $-N/2$ to $+N/2$. The finite-difference form of equation (2) is expressed as:

$$\begin{aligned} i\frac{\psi(x,t+\triangle t) - \psi(x,t)}{\triangle t} &= \\ -\frac{\psi(x+\triangle x,t) - 2\psi(x,t) + \psi(x-\triangle x,t)}{2m\triangle x^2} &+ V(x)\psi(x,t) \end{aligned} \tag{15}$$

Here, we have assumed that $\hbar = 1$. For convenience, we represent $\psi(x_j, t_n + \triangle t)$ as ${\psi_j}^{n+1}$, $\psi(x_j, t_n)$ as ${\psi_j}^n$ and $\psi(x_j - \triangle x, t_n)$ as ${\psi_{j-1}}^n$. With these representations, equation (15) reads as:

$${\psi_j}^{n+1} = {\psi_j}^n + i\triangle t\frac{{\psi_{j+1}}^n - 2{\psi_j}^n + {\psi_{j-1}}^n}{2m\triangle x^2} - i\triangle tV_j{\psi_j}^n \tag{16}$$

Rewriting this equation in a matrix form one gets:

$$F^{n+1} = F^n - i\triangle tH'F^n \tag{17}$$

where the Hamiltonian H' is defined as:

$$H' = -\frac{\hbar^2}{2m}\frac{d^2}{dx^2} + V(x) \tag{18}$$

Subsequently

$$F^{n+1} = UF^n \quad \text{where} \quad U = I - i\triangle tH' \tag{19}$$

Since it is required that the norm of F is $F^*F = 1$, U must be an orthonormal operator. Since U in equation (19) does not have such a property, in our simulation we impose the normalization after every step. At this point we do not stress too much on the accuracy of the integration, rather the emphasis is on the verification of the conceptual framework presented in this paper.

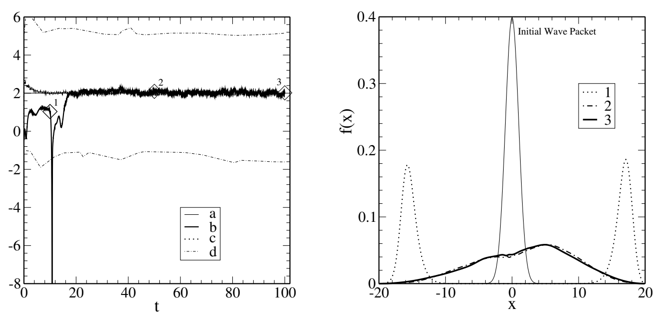

Fig. 4. (*left*)Tracking of a dc signal of strength 2.0 with 6 dB SNR, Gaussian noise: 'a' represents actual dc signal, 'b' represents signal estimation using RQNN and 'c' represents signal estimation using Kalman filter while the noise envelope is represented by the curve 'd'; (*right*) The snapshots of wave packets corresponding to the marker points (1,2,3) placed in the *left* figure. These marker points correspond to 10, 50 and 100 secs respectively

A. Selection of parameters

The nonlinear equation (12) involves four external parameters: $\hbar$, m, ζ and β. The last parameter β is necessary to update the synaptic weight vector $K(x,t)$. For simplicity, the parameter $\hbar$ is taken as unity and the other three parameters are tuned accordingly. Looking at the complexity of equation (12), we used a genetic algorithm (GA) based on the concept of univariate marginal distribution algorithm (UMDA) ([15], [16]) to select near-optimal parameters.

V. Simulation Results

In this section we present the simulation results to filter different signals using RQNN.

A. Tracking of a DC signal embedded in Gaussian Noise

Here we present the filtering result of a DC signal embedded in high Gaussian noise. We also compare the result with the performance of a Kalman filter [17] designed for the same purpose.

It should be noted that the Kalman filter has the *a priori* knowledge that the embedded signal is a dc signal while the RQNN is not provided with this knowledge. The Kalman filter also makes use of the fact that the noise is Gaussian and estimates the variance of the noise based on this assumption. Thus it is expected that the performance of Kalman filter will degrade as the noise becomes non-Gaussian. In contrast, RQNN does not make any assumption about the noise.

The strength of the dc signal is taken as $y_a(t) = 2.0$ embedded in Gaussian noise with SNR values 20, 6 and 0 dB respectively. The near optimal values of the parameters β, m and ζ have been found using univariate marginal distribution

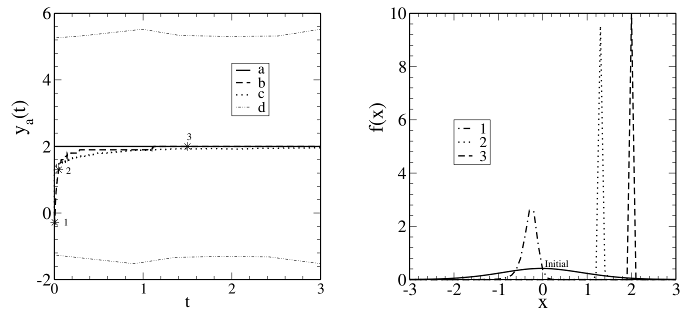

Fig. 5. (*left*) Tracking of a dc signal of strength 2 embedded in a noise of 6 dB SNR: 'a' represents the actual signal, 'b' represents the tracking using RQNN model and 'c' represents the tracking using a Kalman filter. The noise envelope is represented by the curve 'd'; (*right*) The snapshots of wave packets corresponding to the marker points (1,2,3) placed in the *left* figure.The solid line represent the initial wave packet assigned to Schroedinger wave equation.

algorithm. For RQNN with linear modulation of the potential field, the different parameter values are

$$\beta = 0.08 \; m = 0.10 \; \zeta = 480 \; \hbar = 1.0 \; \triangle x = 0.1 \; \triangle t = 0.01 \tag{20}$$

whereas for nonlinear modulation of the potential field,

$$\beta = 0.86 \; m = 2.5 \; \zeta = 2000 \; \hbar = 1.0 \; \triangle x = 0.1 \; \triangle t = 0.001 \tag{21}$$

The number of neurons along the x-axis is fixed at $N = 400$. In addition we select a parameter γ which represents the number of iterative steps that are required for the response of the wave equation to reach a steady state to an instant sample of the external stimulus. In other words, given an input signal sample, $y(t)$, the finite-difference equation model (19) is iteratively updated γ times before the next signal sample is presented. In our simulation, this parameter was found to be $\gamma = 2$ for the linear case and $\gamma = 8$ for the nonlinear case heuristically.

Figure 4 shows the tracking result for SNR value of 6 dB when a linear neural circuit is used to set up potential field of the RQNN model. The performance of RQNN compares well with the performance of the Kalman filter. Figure 5 shows the tracking result for the same SNR value but with a nonlinear modulation of the potential field of the Schroedinger wave equation. It can be easily seen from this figure that the performance of RQNN is much better than the Kalman filter.

Next we explain the principle of filtering using RQNN through the movement of the wave packet. A wave packet is defined as $\psi^*\psi$ where ψ is the solution of equation (12). In each figure, corresponding to marker points in *left* plot, snapshots of wave packets are shown in the *right* plot.

The initial wave packet (at $t = 0$) is always a zero mean Gaussian in all our simulations. For the linear case, it is observed that the wave packet was splitted into two halves (snapshot 1 at $t = 10$ secs) and again the spiltted packets got merged (snapshots 2 and 3 at $t = 50, 100$ secs respectively).

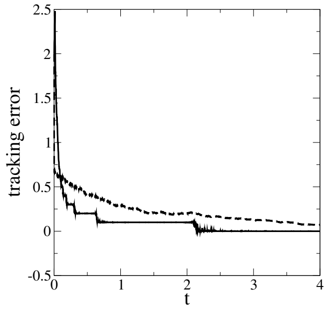

Fig. 6. Continuous line represents tracking error using nonlinear RQNN model while the broken line represents tracking error using Kalman filter

TABLE I
PERFORMANCE COMPARISON FOR DC SIGNAL ($y(t) = 2$)

SNR	Kalman filter RMSE	linear RQNN RMSE	nonlinear RQNN RMSE
20dB	0.015	0.014	0.000040
6dB	0.037	0.060	0.000062
0dB	0.090	0.150	0.000090

Finally the rigid packet moves toward mean 2 (as the actual signal to be tracked has a strength 2) but with a high variance whereas in the nonlinear case there is no such splitting of the wave packet, movement is discrete and variance is also very less. To better appreciate the nature of the wave packet movement in the second case, error plot is shown in figure 6. It can be seen from this figure that the tracking is not continuous but consists of jumps and rests. In other words the soliton behavior of the wave packet is much pronounced in this case.

The filtering performance of RQNN with the Kalman filter is compared in terms of root mean square error (RMSE) which is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{k}(y_a(k) - \hat{y}(k))^2} \tag{22}$$

where k refers to signal samples and N refers to total number of signal samples after the RQNN settles down to a steady state. The performance comparison is shown in Table I for the Gaussian noise. It is very clear that linear RQNN performance is very much comparable with the Kalman filter where nonlinear RQNN outperforms the Kalman filter, given the fact that Kalman filter performs optimally for the linear systems with Gaussian observation noise. Moreover it is seen that the nonlinear RQNN performs equally well for all noise levels whereas performance of Kalman filter degrades with increase in noise level.

B. Filtering of sinusoidal signals

The sinusoid signal is taken as

$$y_a(t) = 2sin(2\pi 10t) \tag{23}$$

Like the case of filtering a dc signal, the actual signal is embedded in noise to generate signals with SNR 20dB and 6dB noise respectively. For the linear case, selected values of parameters associated with the nonlinear Schroedinger wave equation are:

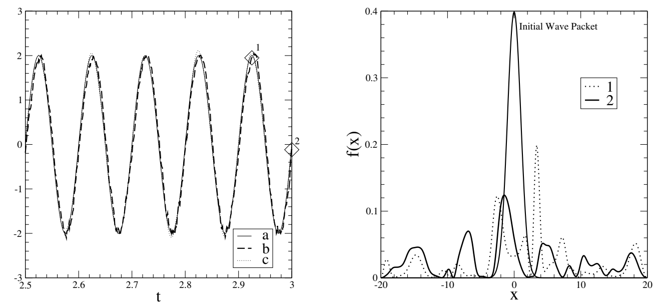

Fig. 7. (*left*) Tracking of a sinusoid signal $y_a(t) = 2sin(2\pi 10t)$ with 20 dB SNR, Gaussian noise; the signal 'a' is the actual signal, the signal 'b' is estimated by RQNN and the signal 'c' is estimated by wavelet filter. (*right*) The snapshots of two wave packets corresponding to the marker points (1,2) placed in the *left* figure. These marker points correspond to 2.925 and 3 secs respectively.

$$\beta = 0.79\ m = 0.40\ \zeta = 83.0\ \hbar = 1.0\ \triangle x = 0.1\ \triangle t = 0.001 \tag{24}$$

The number of neurons along the x-axis is taken as $N = 400$. However the parameter γ is taken 40. The filtering results are shown in *left* plot of the figure 7 for 20dB SNR. For the sinusoid's we compare the performance of RQNN filter with a Daubechis wavelet filter with four vanishing moments, keeping level of decomposition equal to three [18], [19]. The implementation of Daubechis filter was done using MATLAB wavelet toolbox, version 6.5. The snapshots of wave packets corresponding to marker points in the *left* plot are shown in *right* plot of the same figure. With the parameter values given in (24), the shape of wave packets 1 and 2 do not look like Gaussian. However, these wave packets assume a Gaussian shape if the parameter values are taken as per equation (20) used for dc signal filtering.

For the nonlinear case, near optimal parameter values are found as

$$\beta = 0.01;\ \ m = 1.75;\ \ \zeta = -250; \tag{25}$$

Rest of the parameters are taken same as in case of the linear one. The result for 20 dB Gaussian noise is in shown in figure 8

However in this case the Gaussian shape of the wave packet is preserved. In the above figure, the snapshots of wave packets are plotted for three different instances of time indicated by the

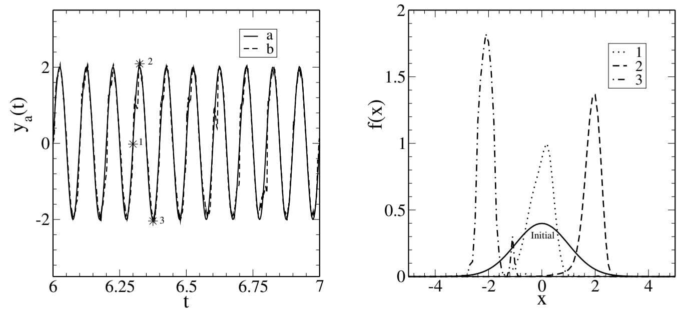

Fig. 8. (*left*) Tracking of a sinusoidal signal embedded in 20 dB Gaussian noise: 'a' res presents the actual signal, 'b' represents the tracking using RQNN model; (*right*) The snapshots of wave packets corresponding to the marker points (1,2,3) placed in the *left* figure. The solid line represent the initial wave packet assigned to Schroedinger wave equation.

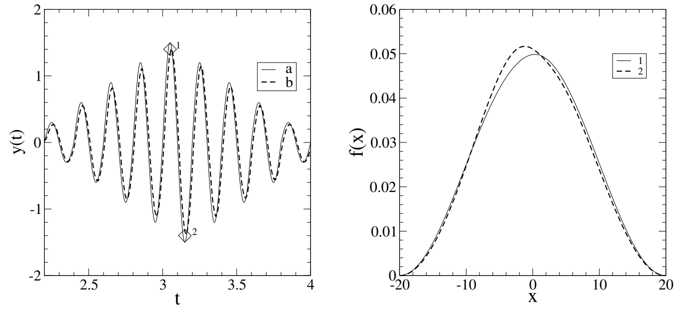

Fig. 9. (*left*) Tracking of amplitude modulated sinusoid signal embedded in 20 dB Gaussian noise: 'a' represents the actual signal and 'b' represents the tracking by the RQNN; (*right*) Snapshots of wave packets corresponding to marker points shown in the *left* plot.

marker points (1,2,3) as shown in the trajectory tracking. When the signal is at position 1, the corresponding wave packet has a mean at 0. When the signal is at position 2, the corresponding wave packet has a mean at +2, and the mean of the wave packet moves to -2 when the signal goes to position 3 that means the wave packet is moving back and forth with mean between -2 to +2 according to the signal value. If we compare the resulting wave packet movement with its linear counterpart as in figure 7, it is clearly seen that the wave packets are moving like a particle along the x-axis in the nonlinear case whereas such movement is not observed in the linear one.

C. Amplitude modulated signals and speech signals

In this section we present the filtering results for complex signals like amplitude modulated sinusoid's and speech signals using the linear RQNN model.

1) Amplitude modulated signals: Amplitude modulated and frequency modulated signals are normally used in coding and transmission of data and appear corrupted at the receiver's end by channel noise ([20]). For simulation purpose, we have selected the frequency of the carrier signal to be a sinusoid of frequency $5Hz$, although in reality they are very high frequency signals. The amplitude was modulated by superimposing a triangular variation of frequency $0.5Hz$. Thus the expression for the composite signal $y_a(t)$ is:

$$y_a(t) = a(t){\cdot}sin(2\pi 5t)\ ; \quad a(t) = \begin{cases} 1.5t & 0 \le t \le 1 \\ 1.5\,(2-t) & 1 \le t \le 2 \end{cases} \tag{26}$$

where $a(t)$ is periodic with period $0.5Hz$. The amplitude modulated sinusoid signal $y_a(t)$ in equation (26) was immersed in Gaussian noise. The variance of the Gaussian noise was set according to the $20dB$ and $6dB$ SNR measurement. The values selected for the parameters of the Schroedinger wave equation using UMDA are as follows:

$$\beta = 0.11 \quad m = 0.015 \quad \zeta = 84.31 \quad \hbar = 1.0 \tag{27}$$

The number of neurons along the x-axis are taken as $N = 400$. The parameters for the finite-difference equation used for integration are selected as:

$$\triangle x = 0.1 \quad \triangle t = 0.001 \tag{28}$$

The parameter γ is selected as 100. The tracking of the desired signal $y_a(t)$ is shown in figure9. It can be observed that the tracking is very smooth and accurate. Snapshots of wave packets are shown in the same figure correponding to marker points shown in *left* plot. It can be observed that the *pdf* does not split and slides along the x-axis back and forth like a particle.

2) Speech signals: Speech signals are degraded in many ways that limit their effectiveness for communication. One major source of noise in the speech signal is *Channel Noise* which is a major concern, especially in speech recognition systems. Since the RQNN estimates the *pdf* of the incoming signal at every instant, if the incoming signal is corrupted by zero-mean noise, then the RQNN must be able to filter out that noise. Working on this hypothesis, we added zero-mean Gaussian noise with variance equal to the square of the instantaneous amplitude of the speech signal. The peak amplitude of the speech signal was normalized to 1.0. For tracking the speech signals, the number of neurons along the x-axis is taken as $N = 400$. The parameters for the finite-difference equation used for integration are selected as:

$$\triangle x = 0.1 \quad \triangle t = 0.001 \tag{29}$$

The values selected for the parameters of the Schroedinger wave equation using UMDA are as follows:

$$\beta = 0.16 \quad m = 0.015 \quad \zeta = 27.45 \quad \hbar = 1.0 \tag{30}$$

The parameter γ was selected as 800. The tracking for a particular period of the speech signals selected from the database is shown in figure 10 and the snapshots of the wave packet at 2 time instants are also shown in the same figure.

It is evident from figure 10 that the RQNN does track the *pdf* of the input signal at every instant. The wave packet does

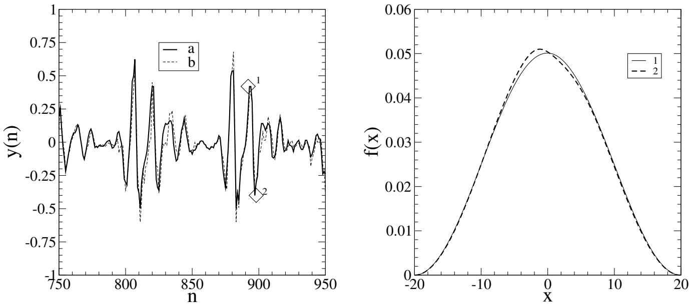

Fig. 10. speech file NU_24streetaddr.wav: *(left)* Filtering of speech signal immersed in Gaussian noise: 'a' represents the actual speech signal and 'b' represents the tracking by the RQNN;*(right)* Snapshots of wave packets at marker points (1,2) shown in the *left* plot.

not split and it maintains an approximate Gaussian nature. It moves slightly along the x-axis like a particle maintaining its soliton property. By estimating the actual signal as the mean of the *pdf* at every instant, we can filter out the corrupting noise added to the actual signal. In addition, we reconverted the tracked speech signal and the noisy speech signal to the *WAV* format. On listening to these signals, we could verify that the RQNN filtering does improve the quality of the input noisy speech signal if corrupted by zero-mean Gaussian noise.

VI. Conclusion

The paper presented an alternative neural information processing architecture where a quantum process mediates the collective response of a neural lattice having spatial structure. The proposed RQNN is governed by a single dimensional non-linear Schroedinger wave equation. The nonlinear Schroedinger wave equation which resulted because of recurrent structure of the network exhibited soliton property - a property that defines the particle type of movement of the wave packet. The interesting feature of this model is the movement of the wave packet along the *pdf* of the observed stochastic variable. Two different variants of RQNN have been used in this paper, first one used a linear neural circuit for setting up the potential field of the Schroedinger wave equation while the second one used a nonlinear neural circuit to do the same. Given no *a priori* knowledge about the noise, the proposed RQNN has very efficiently filtered the signals embedded in high noise. The surprising result is that nonlinearly modulated RQNN denoises a DC signal 1000 times more accurately as compared to a classical Kalman filter without any external constraint. The Kalman filter uses the *a priori* knowledge that the signal is dc and the noise is Gaussian while RQNN makes no such assumptions about the noise. Beside the dc and the sinusoid's, very sophisticated signals such as amplitude modulated and speech signals have been effectively filtered. Thus proposed quantum stochastic filter can be termed as intelligent when compared to classical counterparts since denoising is accomplished without any kind of assumption about the signal and noise. This is precisely the advantage of estimating *pdf* using Schroedinger wave equation.

We expect that this paper will motivate many researchers to explore this new approach to design quantum learning systems. One may like to compare the performance of RQNN with other existing adaptive algorithms such as LMS and RLS.

References

[1] M. A. Cohen and S. Grossberg, "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks," *IEEE Trans Syst, Man and Cybernetics*, vol. 13, pp. 815–826, 1983.
[2] S. Amari, "Field theory of self-organizing neural nets," *IEEE Trans SMC*, vol. SMC-13, no. 5, pp. 741–748, Sept/Oct 1983.
[3] L. Behera, M. Gopal, and S. Chaudhury, "On adaptive control of a robot manipulator using inversion of its neural emulator," *IEEE Trans Neural Networks*, vol. 7, no. 6, pp. 1401–1414, 1996.
[4] L. Behera, S. Chaudhury, and M. Gopal, "Applications of self-organizing neural networks in robot tracking control," *IEE Proceedings Control Theory and Applications*, vol. 145, no. 2, pp. 134–140, 1998.
[5] D. J. Amit, *Modeling brain function*, Springer-verlag,Berlin/Heidelberg, 1989.
[6] R. L. Dawes, "Quantum neurodynamics: Neural stochastic filtering with the schroedinger equation," *IJCNN Proceedings*, p. 133, July 1992.
[7] Robert L. Dawes, *Rethinking Neural Networks:Quantum Fields and Biological Data*, chapter Advances in the theory of quantum neurodynamics, Erlbaum, Hillsdale, N.J., 1993.
[8] Sanjay Gupta and R.K.P. Zia, "Quantum neural networks," *Journal of Computer and System Sciences*, vol. 63, no. 3, pp. 355–383, 2001.
[9] R.W. Boyd, *Nonlinear Optics*, Academic Press, 1991.
[10] E. Atlee Jackson, *Perspectives of Nonlinear Dynamics*, Cambridge, 1991.
[11] I. Bialynicki-Birula and J. Mycielski, "Nonlinear wave mechanics," *Annals of Physics*, vol. 100, pp. 62–93, 1976.
[12] A. T. Bahill and L. Stark, "The trajectories of saccadic eye movements," *Scientific American*, vol. 240, pp. 84–93, 1979.
[13] L. Behera, I. Kar, and A. C. Elitzur, "A recurrent quantum neural network model to describe eye tracking of moving targets," *Foundations of Physics Letters*, 2005, Accepted for publication in quarter III.
[14] R. S. Bucy, "Linear and nonlinear filtering," *IEEE Proceedings*, vol. 58, no. 6, June 1970.
[15] L. Behera, *New Optimization Techniques in Engineering*, chapter Parametric Optimization of a Fuzzy Logic Controller for NonlinearDynamical Systems using Evolutionary Computation, McGraw Hill, 2002.
[16] H. Muehlenbein and T. Mahnig, *Foundations of Real-World Intelligence*, chapter Evolutionary Computation and Beyond, CSLI Publications, 2001.
[17] M. S. Grewal and A. P. Andrews, *Kalman Filtering : Theory and Practice Using MATLAB*, Wiley-Interscience, 2001.
[18] I. Daubechis, *Ten Lectures on Wavelets", Society for Industrial and Applied Mathematics*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.
[19] S. Mallat, *A Wavelet Tour of Signal Processing*, Acedemic Press, 1998.
[20] Simon Haykin, *Communication Systems*, John Wiley and Sons, Inc., 4 edition, 2001.