# Quantum Recurrent Neural Networks for Filtering

A thesis submitted to the Department of Computer Science,
University of Hull
in partial fulfilment of the requirements for the degree of
Doctor of Philosophy
by

Woakil Uddin Ahamed
M Sc. (Mathematics), University of Hull, 2005
M Sc. (Mathematics), University of Dhaka, 1991

February 2009

*Dedicated to my late parents who have sacrificed their entire lives for my well-being.*

# Acknowledgement

This thesis is the result not only of my efforts but also of all those people who during my years in Hull University have helped and provided me the emotional and physical supports. This thesis is as much theirs as is mine. First of all I would like to express my heartiest gratitude to Dr Chandra Kambhampati for his active and genius supervision. I am indebted and grateful to the University of Hull for awarding the Faculty of Science scholarship to carry out this research. Mayur Sarangdhar also deserves appreciation for his help in technical matters and stimulus discussions. I am deeply grateful to my wife, Eva, and children Wasee and Intan, for their understanding and constant support throughout this endeavour. Finally, I would also like to express my heartiest appreciation to my all family members, though living far, for their encouragements and vivacious words to move on.

# Abstract

The essence of stochastic filtering is to compute the time-varying probability density function (*pdf*) for the measurements of the observed system. In this thesis, a filter is designed based on the principles of quantum mechanics where the Schrödinger wave equation (SWE) plays the key part. This equation is transformed to fit into the neural network architecture. Each neuron in the network mediates a spatio-temporal field with a unified quantum activation function that aggregates the *pdf* information of the observed signals. The activation function is the result of the solution of the SWE. The incorporation of SWE into the field of neural network provides a framework which is so called the Quantum Recurrent Neural Network (QRNN). A filter based on this approach is categorized as *intelligent filter*, as the underlying formulation is based on the analogy to real neuron.

In a QRNN filter, the interaction between the observed signal and the wave dynamics are governed by the SWE. A key issue, therefore, is achieving a solution of the SWE that ensures the stability of the numerical scheme. Another important aspect in designing this filter is in the way the wave function transforms the observed signal through the network. This research has shown that there are two different ways (a normal wave and a calm wave, Chapter-5) this transformation can be achieved and these wave packets play a critical role in the evolution of the *pdf*. In this context, this thesis have investigated the following issues: existing filtering approach in the evolution of the *pdf*, architecture of the QRNN, the method of solving SWE, numerical stability of the solution, and propagation of the waves in the well. The methods developed in this thesis have been tested with relevant simulations. The filter has also been tested with some benchmark chaotic series along with applications to real world situation. Suggestions are made for the scope of further developments.

# Table of Contents

# List of figures

# List of tables

# List of symbols:

| | |
|---|---|
| $T$ | Total time interval |
| $t$ , $k$ | Any instant of time |
| $\Delta t$ | Small change of time |
| $\Re$ | Real number |
| $x_k$ | Input vector at certain instant $k$ |
| $y_k$ | Output vector at certain instant $k$ |
| $\Delta x$ | Small change of space |
| $\Omega$ | Set or event of a set |
| $E[\cdot]$ | Statistical expectation |
| $F$, $H$ | Transformation matrices |
| $R$ , $Q$ | Covariance matrices |
| $\psi(\vec{r};t)$ | Wave function at position $\vec{r}$ at time $t$ |
| $\mathcal{H}$ | Hamiltonian matrix |
| $V$ | Potential function |
| $\xi$ | Potential field excitation |
| $h$ | Planck constant |
| $\hbar$ | Planck constant divided by $2\pi$ |
| $m$ | Mass of the quantum object |
| $\kappa$ | Wave momentum (or velocity) |
| $L$ | Length of the well (or box) |
| $N$ | Number of neurons in the network |
| $\eta$ | Learning rate |

# Chapter 1

# Introduction

## 1.1 Introduction

A common problem in modelling, and control of dynamical system is to estimate the true value of a system state where the measurements or signals of the system are noisy. Filtering is a mathematical algorithm or device through which the true value of a system state is extracted or estimated from these noisy measurements. With model based methods, the key to designing a filter is the availability of the model that represents the dynamics of the system and the relationship between the system state and the measurement processes. These dynamics of the system are often time-varying and nonlinear while their mathematical representations are often linear and at times time-invariant. Designing a filter becomes complex when an accurate description of the system is not available and thus based on the nature of the available model, stochastic filters are formulated using probabilistic measures. The key feature of filtering is in the manner in which the probability density function (*pdf*) evolves over time.

## 1.2 Model based approach

In order to analyse and estimate the state of a dynamic system, it requires two important models: First, a model describing the evolution of the state with time (called the state process), and, second, a model describing the noisy measurements to the state (called the measurement process). These models may be linear or nonlinear. Filtering algorithm

is designed combining these two processes to estimate the current state of the system. The Kalman Filter (KF) is a popular algorithm and is based on a linear model (Kalman 1960; Kailath 1968; Andrew 1970; Rhodes 1971; Julier & Uhlmann 1997; Haykin 2001; Mohinder & Angus 2001, Arulampalam *et al.* 2002). This filter performs a conditional probability density propagation in which the system and measurement noises are white and Gaussian (Maybeck 1979, 1982). The performance of the filter is measured by a criterion (e.g., cost function) and is optimal with respect to this criterion. One of the key features of the Kalman filter, which enables it to be optimal, is that it incorporates all the available information to estimate the current state of the system. The information incorporated includes

- knowledge of the system and measurement dynamics,
- statistical description of the system noises, measurement errors, uncertainties in the model dynamics, and
- information on the initial conditions of the states or variables of interests.

If the system dynamics and measurement processes are linear, and the Gaussian assumption is relaxed then the Kalman filter can be shown to be the best (minimum error variance) optimal filter out of the class of linear unbiased filters (Maybeck 1979). In most practical applications, the system dynamics and measurement equations are not linear. In such cases, nonlinear representations of the models (e.g., Extended Kalman Filters (or EKF)) are used. The optimal solutions to these filters require that the model be linearized and a complete description of the conditional probability density function (or *pdf*) is maintained (Maybeck 1979; Merwe *et al.* 2000). Unfortunately, this complete description requires a potentially unbounded number of parameters, such as moments, to be evaluated (Maybeck 1982; Julier & Uhlmann 2000). Despite these difficulties, the EKFs have been widely accepted and are used as standard tools for the nonlinear

filtering techniques. Details of these two filters (KF and EKF) are presented in Chapter-3 of this thesis.

Apart from the above, a number of other filtering methods have been proposed e.g., the Particle Filter (PF), and the Unscented Kalman Filter (UKF). The Particle Filter is based on the Monte Carlo method which uses a sequential importance sampling (SIM) and re-sampling approach (Carpenter *et al*. 1999; Liu *et al*. 2001; Haykin 2001, Arulampalam *et al.* 2002). Objective of this filter is to track a variable of interest as it evolves over time. The basis of this approach is to construct a sample (also known as particle) based representation of the entire *pdf* which represents the posterior distribution of the state variables by a system of particles (or samples) that evolves and adapts recursively as new information becomes available. In practice, a large number of particles (or samples) are required to provide an adequate approximation to the conditional *pdf*. This is why in high-dimensional applications this filter is rarely used. A problem that arises in the use of particle filter is the depletion of the particle (or sample) population after a few iterations and the resultant updating parameters. This problem can be overcome by a re-sampling method (Liu *et al*. 2001; Rekleitis 2003). However, for a particular application, in a particle filter the decision on the number of particles is crucial.

In the extended Kalman filter, the distribution of the state variable is approximated by a set of Gaussian random variables which are then propagated analytically through a linear system. Often, this introduces a large error in the evolution of the *pdf* causing suboptimal performance or even divergence of the filter (Julier & Uhlmann 2000; Haykin 2001). The UKF addresses this issue by using a deterministic sampling approach called the "unscented transformation". In the EKF, the state process consists of two terms; a nonlinear deterministic term and an uncertainty term. The UKF filter considers a set of points that are deterministically selected from the Gaussian

approximation to the initial *pdf*. These points are then propagated through the true nonlinear function and the parameters of the Gaussian density are re-evaluated. It has been found that, for some cases (Orderud 2005), this filter gives better results than the EKF. At a superficial level, the computational requirements of the UKF and EKF appear to be the same. However, the UKF is computationally more accurate in that the derivatives need not be evaluated in order to construct the Jacobian or Hessian matrices. These two matrices are, however, required for the EKF (Maybeck 1979; Wan *et al.* 2000; Haykin 2001).

## 1.3 Data driven approach

In contrast to the model based approach (mentioned above), filtering algorithms have also been designed where the nominal representations of the state and measurement processes are not required. For example, algorithms based on neural networks and neuro-fuzzy networks (Lee *et al.* 1997; Mandic & Chambers 2000; Haykin 2001; Lin *et al.* 2004). The advantage of these methods is that they are purely data-driven, generic, and can be applied with little or no a *priori* knowledge of the observed system.

Filters based on neural approaches are categorised as *Intelligent Filters*, since the underlying formulation is based on the analogy to a real neuron and neuronal networks. In neuroscience literature, it has been suggested that real neurons are fired in a distributed manner, and that these are then aggregated to provide a coherent representation of the true state (Koch & Segev 1989; Haykin 1994; Dorffner 1997, Husmeier 1999a, 199b). It is argued that although filters based on neural networks work along similar lines there is an underlying mechanistic feature in the neural network itself that make them not sufficiently intelligent (Penrose 1994, Dorffner 1997).

In this thesis, a filtering algorithm is devised based on the properties of the Schrödinger wave equation (SWE). This equation plays an important role in quantum mechanics and has a number of built-in properties which are being used in physics, electronics, and chemistry, as described in Figure-1.1 (Nave 2006), in a microscopic level in finding out the location of a quantum object or ionic movement at certain time references.

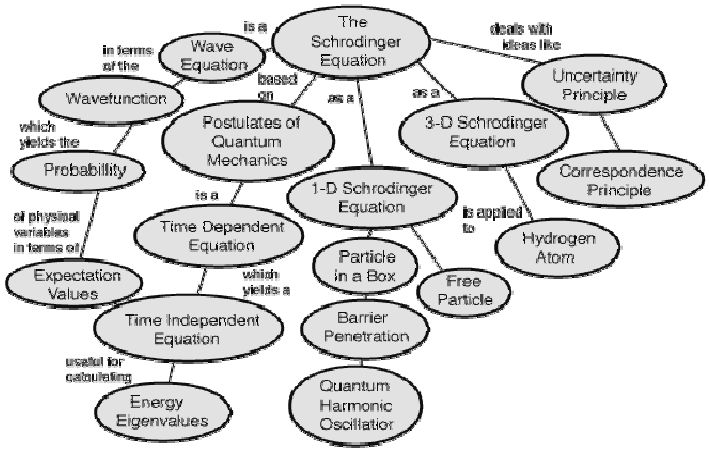



Figure-1.1: Schrödinger equation in various fields of Quantum mechanics

As can be seen from Figure-1.1, SWE has the ability of predicting the behaviour of a dynamical system in a way that this equation works as a wave equation in terms of the wave function which predicts analytically and precisely the probability of events or outcomes (Schiff 1968; Feynman 1986). In this thesis, the filtering algorithm is developed based on this property of SWE. It is argued that an integration of quantum principles in a neural network paradigm would provide a better framework in developing the filtering algorithm. Some important advantages of this filter, as will be shown in later chapters, are that

- It does not require a *priori* information on the nature of noises present in the measurement processes
- It has a simple single layer network architecture

As a result of these features, a recurrent neural network has been designed incorporating the quantum principles of the Schrödinger wave equation (SWE). For this purpose, the equation has been transformed to fit into the recurrent neural network framework with a nonlinear term (Dawes 1989a, 1989b, 1989c, 1992, 1993; Behera *et al.* 2004, 2005a, 2005b). It has been suggested that an extended nonlinear form of the SWE would provide a framework for approximating the *pdf* of the observed system. The resulting network is then referred as the Quantum Recurrent Neural Network (QRNN). Due to the transformation of the SWE, it becomes important to examine the properties of the equation so that an accurate description of the measurements can be found. Indeed this thesis along with Dawes (1989a, 1989b, 1989c, 1992, and 1993) and Behera *et al.* (2004, 2005a, and 2005b) is an attempt to this.

Although filters based on the SWE have been developed, their full analysis, especially the solution technique of the SWE, architecture of the network, learning algorithm and design parameters, are still limited. These issues can be analysed by examining the network dynamics in terms of the differential structure of the SWE and its associated parameters. A schematic diagram depicting the filter mechanism is shown in Figure-1.2.

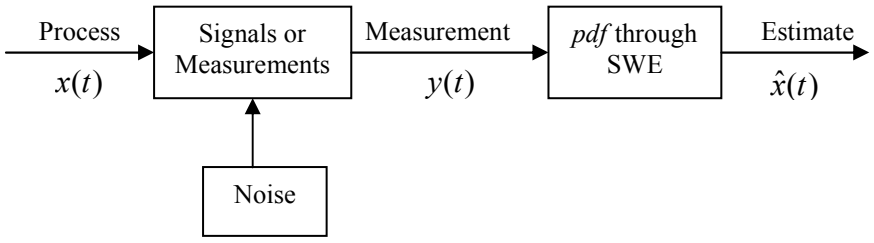



Figure-1.2: A schematic diagram of the QRNN filter

## 1.4 Objectives

A nonlinear filtering algorithm based on the Schrödinger wave equation (SWE) under a neural network paradigm has been proposed by Dawes (1989a, 1989b, 1989c, 1992, and

1993) where SWE has been transformed into a set of equations which can represent a recurrent neural network. However, to solve the SWE an explicit method was introduced. Although the performance of the filter with this method is good, the stability of the solution has not been ensured. Another important aspect that has not been addressed is the way the wave packets propagate in the well. In this context, the aim of this research is to investigate key issues regarding the development of the QRNN filter based on the SWE. For this reason, this thesis looks into the following issues: *the construction of the quantum recurrent neural network, solution techniques of the SWE, connection and contribution of the parameters of the SWE itself on the evolution of the pdf, dynamics of propagating wave packets in the well, learning algorithm and convergence of error*.

### 1.4.1 Solution of SWE

The SWE can be solved in many different ways that mainly depend on the requirements of the application and the problem formulation. It can be seen from Figure-1.1 that each of these problems results in different forms of the solution. In the case of application of SWE within a neural network framework, this equation must be solved in such a way that the resultant architecture represents a neural network with its activation functions and connections. It is also important that an appropriate numerical procedure is selected to solve the SWE so that the filter is *robust and also maintains the essential functionality of a neural network* (see Chapter–4). It is well known that partial differential equations have a better class of solutions, in terms of stability and boundedness of the solutions, if a numerical method based on implicit scheme is used. However, in the case of filtering applications, the numerical procedure is not straightforward in that these are essentially multi-loop schemes. Here, an inner loop is concerned with the discretisation of the state space in-order to obtain the Hamiltonian

whilst the outer loop deals with the dynamics of the observed system (Ahamed & Kambhampati 2008). The discretisation carried out in the inner loop is crucial in that this process enables the incorporation of localization of the wave and decides on the nature of the potential field. These determine the bound on some of the parameters of the filter. The outer loop deals with the integration of the dynamics of the real process with that of the quantum filter in order to establish the prediction error and learning mechanisms. The integration of these two loops will provide the basis for the evolution of the *pdf* (Ahamed & Kambhampati 2008).

In the first instance the solution of the SWE depends on the choice of some constants. In physics such parameters have a well defined meaning. In filtering applications, these parameters have to be scaled as well as appropriately tuned. However, tuning of these values heuristically may lead to an unacceptable solution. On the other hand, a detailed study/analysis has to be undertaken to observe their affect on the solution space. Here, the localization techniques have been used to determine the precise magnitude of the parameters, or at the very least to determine the bounds from within which a value can be picked up (see Chapter-4).

### 1.4.2 Network architecture and issues

A neural network consists of a set of interconnected nodes called neurons. These neurons when interconnected by a well-defined topology can be trained to perform certain tasks. Often, the number of such neurons in the network and the topology are decided *a priori* and this decision is based on the experience of the designer. Indeed, no satisfactory solution has been found as yet, as too few neurons can cause under-fitting, and too many causes over fitting. QRNN is a single layer network, the connectivity of the neuron is dependent on the nature of the potential field (a term in the SWE which will be apparent in Chapter-4) – this eliminates the necessity of making a decision on

the topology. It is suggested that a single layer quantum recurrent neural network in a spatio-temporal field will be able to learn the *pdf* information coherently, where each neuron act as a simple computational unit and will be able to transfer the observed stochastic variable to the wave function of the SWE in a unsupervised manner (Behera *et al.* 2005b). In fact, the potential field of the SWE is transformed (see Chapter-4) so that the network can receive input via the potential field. This potential field is added to the Hamiltonian which acts upon the wave function for the evolution of the wave in time and space (see Chapter–4).

The neural network is trained using training rules or learning methods. For the QRNN filter, the Hebbian learning rule is preferred as with this rule the network can be trained in an unsupervised manner.

### 1.4.3 Propagation of waves

Numerical solution of SWE is implemented in a grid system called well (or box). Each node in the grid acts as a neuron and the solution of SWE in this grid represents the wave function whose modulus-squared is defined as the probability density function (*pdf*) for the observed signal. The solution of SWE requires initial conditions through which the wave packet initiates the propagation. However, discretisation of SWE in the grid system introduces limitations on the choice of values for the parameters of the SWE, especially on the propagation of the wave packet. Practically, these issues dictate the ability of the filter to extract the signal and to be able to understand the underlying properties of the signal.

## 1.5 Aim and objectives of this research

It was mentioned earlier that the QRNN filter is developed based on the properties of the SWE. Concept of this filtering algorithm is relatively new and was first developed

by Dawes (Dawes 1989, 1992) where each neuron in the network mediates a spatio-temporal field with a unified quantum activation function that aggregates the *pdf* information of the observed signal. Although there have been works in the application of this filter (e.g., Behera and Sundaram 2004) a number of key issues and questions have not been explored. Thus, the objective of this thesis is to investigate the construction of the quantum recurrent neural network, solution techniques of the SWE, connection and contribution of the parameters of the SWE itself on the evolution of the *pdf*, dynamics of propagating wave packets in the well, learning algorithm and convergence of error. Moreover, as a result of these investigations, a new method for the propagation of the wave packets in the well is developed. To achieve this, focus is given on the following specific issues:

- Investigation of the SWE and filter development
- Numerical procedure for the solution of the SWE
- Design issues of the filter and wave packet propagation
- Training the neural network and the learning process
- Evolution of the *pdf* under quantum mechanical properties, and
- Sensitivity of the parameters involved

In the next few chapters the development of the quantum recurrent neural network will be discussed along with a brief description of the existing filter constructions. The general framework for the QRNN filter along with procedures for the numerical solution of the SWE is presented. Framework for implementing the QRNN filter is discussed in the subsequent sections of Chapter-4. Design issues of the filter and propagation of the wave packets in the potential well are discussed in Chapter-5. Simulation results are presented in Chapter-6. Applications of the QRNN filter are

shown in Chapter-7. Conclusion and future development of the filter is discussed in Chapter-8. It is expected that tuning the parameters and their sensitivities to solution of the SWE, methods of propagating waves in the well would help see further insight to the QRNN filter.

# Chapter 2

# Preliminaries: Filtering and Neural Networks

## 2.1 Introduction

Filtering (or estimation), is closely connected to control, information, and communication theory. A common problem in dynamical system is to filter the stochastic process. Filter refers to an algorithm that can estimate the state of a system from its observations (or signals). There are number of applications that require estimation of current state such as localization of an object in space using radar, spotting underground or underwater object using sonar, screening and diagnosis of cardiovascular diseases using sensory data from medical devices. In order to control a stochastic plant (a plant could mean a dynamic system that is time varying, such as mobile platform, haptic devices) with incomplete and noisy state information, to compute the joint information between a random variable and stochastic process, all these require the solution of a related filtering problem.

Generally, the filtering problem is defined as follows: at a certain period of time $T$, a measurement $y(k),\ k \in T$ is observed form a system. Filtering is concerned with the recovery of the current system state $x(k)$ or an approximation to $x(k)$ from this measurement $y(k)$. To estimate the system state, there are three important points to consider. First, filter is concerned with obtaining the information about $x(k)$ at time $k$. Second, the information is available at time $k$, not at some later time and third,

measurement right up to, but not after, time $k$ is used. An example of such filter could be the voice signal which is modulated to a high frequency carrier and then transmitted to the receiver. This received signal when demodulated, it is filtered to recover the signal as well as possible.

In this chapter, definitions of the stochastic process along with a brief exposition of neural network and their components, structures and mechanisms will be presented. In the next few sections preliminaries, those are relevant and necessary for the design and development of a filter is presented. At the outset, definition of a dynamic system and relationship between the state and measurement processes are discussed.

## 2.1 Dynamical system

Often dynamical systems are presented by difference equations (for discrete time) or differential equations (for continuous time). The difference equation

$$x(k+1) = f(x(k), v_1(k), k) \text{ with } k = 0,1,...,n, \tag{2.1}$$

where $f$ is may be nonlinear $n-$dimensional continuously differentiable function, is known as *nonlinear stochastic vector difference equation*. If the noise $v_1(k)$ is absent in the equation (2.1) then it turns into a difference equation in deterministic nature and in such a case $x(k)$ represents the solution. The measurement process $y(k)$ and the state process $x(k)$ are related by the equation

$$y(k) = g(x(k), v_2(k), k). \tag{2.2}$$

where $v_2(k)$ is known as measurement noise. The state process $x(k)$ is not directly observed but information concerning $x(k)$ is obtained from the observation $y(k)$. Equations (2.1) and (2.2) are known as the state and the measurement processes of a system respectively. These two equations together form a filtering model. The state of

the system is estimated from the measured variables. For example, a filtering model may be defined as:

$$x(k+1) = a + \sin(b\pi k) + cx(k) + v_1(k) \text{ (State process)} \tag{2.3}$$

$$y(k) = lx^2(k) + v_2(k) \text{ (Measurement process)} \tag{2.4}$$

where $a$, $b$, $c$, and $l$ are scalar parameters. Random variables $v_1(k)$ and $v_2(k)$ are drawn from some specified distributions. Given only the measurement $y(k)$, the objective here would be to combine state and measurement information to estimate (or extract) the underlying system state $x(k)$. It can be seen that equations (2.1) and (2.2) are both stochastic processes as they are governed under the influence of time varying random noises.

The equivalent continuous time filtering model can be written as follows

$$\dot{x} = f(x, v_1) \text{ (State process)} \tag{2.3a}$$

$$y = g(x, v_2) \text{ (Measurement process)} \tag{2.4b}$$

## 2.3 Definitions

### 2.3.1 Stochastic process

A stochastic process $x(k)$ is a family of random variables indexed with parameter set $k \in T$ which is, in general, referred to time. If $T = \{1,2,...\}$ then the random variable $x(k)$ is called a stochastic process in *discrete time* (that is, a sequence of random variables). If $T = [0,\infty)$ then the random variable $x(k)$ is called a stochastic process in continuous time.

For example, a stochastic process $x(k), k \geq 0$ in continuous time may be defined by the equation $x(k) = a\sin(2\pi k) + b$, where $a$ or $b$, or both of them are random variables.

### 2.3.2: Density function

The word probability refers to a basic entity associated with any random variable. The relationship between the probability density function $\rho_X(x)$ of a random variable $X$ and the distribution function $F_X(x)$ is given by

$$\rho_X(x) = \frac{dF_X(x)}{dx} \tag{2.5}$$

The density function exists if the distribution function is differentiable or the number of points at which the distribution function is not differentiable is countable (or defined). Equation (2.5) can also be written as

$$F_X(x) = \int_{-\infty}^{x} \rho_X(\xi) d\xi \tag{2.6}$$

The distribution function, $F_X(x)$, is interpreted as a mapping from the real line to the interval [0, 1] and so $F_X(x) = P(X \le x)$. The quantity $\rho_X(\xi)d\xi$ is interpreted as the probability that the random variable falls between $\xi$ and $\xi + d\xi$.

### 2.3.3: Characteristic function

A random variable $X$ may be specified in terms of its characteristic function which is defined by

$$\varphi_X(u) \triangleq E\{\exp(iuX)\}, \text{ where } i^2 = -1. \tag{2.7}$$

One important property of this function is that it is just the Fourier transform of its density function

$$\varphi_X(u) = \int_{-\infty}^{+\infty} \exp(iux)\rho_X(x)dx. \tag{2.8}$$

Therefore, if the characteristic function is absolutely integrable, the inverse Fourier transformation

$$\rho_X(x) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \exp(-iux)\varphi_X(u)du \tag{2.9}$$

gives the density function.

### 2.3.4: Gaussian random variables

A random variable $X$ is said to be Gaussian (or normally distributed) if its density function is given by

$$\rho_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)^2\right], \tag{2.10}$$

where mean $m = E(X)$, and variance $\sigma^2 = variance\ (X)$.

### 2.3.5: Finite dimensional distribution

The joint distribution function of the random variables $x(k_1),...,x(k_n)$ for any finite set $\{k_i\} \in T$ is called a finite dimensional distribution of the process. A stochastic process can be characterised by specifying the finite dimensional distribution:

$$F(x(k_1),...,x(k_n)) \text{ for all finite set } \{k_i\} \in T. \tag{2.11}$$

The meaning of (2.11) is that with this distribution function it is possible to answer many probabilistic questions such as the expectation, variance, covariance, conditional density, and correlations about the random variables $\{x(k), x(\tau)\}$ where $k, \tau \in T$. A stochastic process can be characterised by specifying the joint density function

$$\rho(x(k_1),...,x(k_n)) \text{ for all finite set } \{k_i\} \in T \tag{2.12}$$

or the joint characteristic function

$$\varphi_{x(k_1),\ldots,x(k_n)}(u_1,\ldots,u_n) \tag{2.13}$$

for all finite set $\{k_i\} \in T$ .

### 2.3.6: Expectation

The expectation (or probability-weighted average) of a continuous random variable $X$ is defined by

$$E(X) = \int_{-\infty}^{+\infty} x\rho_X(x)dx, \tag{2.14}$$

where $\rho_X(x)$ is the probability density function.

### 2.3.7: Conditional density

The conditional density $\rho_{X|Y}(x \mid y)$ of $X$ given that $Y$, where $x$ is a realization of the random variable $X$ and $y$ is a realization of random variable $Y$, is defined by

$$\rho_{X|Y}(x \mid y) = \frac{\rho_{X,Y}(x,y)}{\rho_Y(y)} \tag{2.15}$$

If random variable $X$ and $Y$ are independent then

$$\rho_{X|Y}(x \mid y) = \rho_X(x) \tag{2.16}$$

### 2.3.8: Conditional expectation

The conditional expectation of a random variable $X$ given the random variable $Y$ is defined by

$$E(X \mid Y) = \int x\rho_{X|Y}(x \mid y)dx \tag{2.17}$$

Properties of the conditional densities and conditional expectation are as follows: If the random variables $X$ and $Y$ are jointly distributed then

i. $\rho_{X|Y}(x \mid y) \geq 0$ (2.18)

ii. $\int \rho_{X|Y}(x \mid y)dx = 1$ (2.19)

iii. $E\{\rho_{X|Y}(x \mid y)\} = \int \rho_{X|Y}(x \mid y)\rho_Y(y)dy$ (2.20)

### 2.3.9: Markov process

A stochastic process $\{x(k), k \in T\}$ is called Markov process if the probability law of the next state depends only on the current state and not on all the previous states. The conditional density function of the Markov process can be written as

$$\rho(x(k_n) \mid x(k_{n-1}),...,x(k_1)) = \rho(x(k_n) \mid x(k_{n-1})) \quad (2.21)$$

### 2.3.10: Optimal filter

An optimal filter (or an optimal estimator) is a computational algorithm that provides an estimate of a variable of interest, by optimising a given cost function. The cost function is often defined as the mean-squared-error (MSE). For example, a criterion may be that the MSE should be as small as possible. The word optimal is subjective in the sense that it is the best possible results among any other results that can be obtained or achieved.

### 2.3.11: sigma-filed

Let $\Omega$ be a non-empty set. A $\sigma$ - field $\Im$ on $\Omega$ is a family of subsets of $\Omega$ such that

i) the empty set $\phi$ belongs to $\Im$;

ii) if $A$ belongs to $\Im$, so does the complement $\Omega \setminus A$;

iii) if $A_1$, $A_2$, … is a sequence of sets in $\Im$, then their union $A_1 \cup A_2 \cup ...$ also belongs to $\Im$

### 2.3.12: Filtration

A sequence of $\sigma$ - fields $\Im_1, \Im_2, ..., \Im_T$ on $\Omega$ such that $\Im_1 \subset \Im_2 \subset ... \subset \Im_k$ is called filtration, where $\Im_T$ represents the knowledge at time $k$ and contains all events $A$ such

that at time $k$ it is possible to decide whether or not $A$ has occurred. As $k$ increases, there will be more such events $A$ and the knowledge about the system will be richer.

### 2.3.13: Wiener Process

The Wiener process (or Brownian motion) is a stochastic process $X(k) \in \Re$ defined for $k \in [0, \infty)$ such that

i) $X(0) = 0$ almost sure;

ii) the sample paths $k \to X(k)$ are almost sure continuous;

iii) for any finite sequence of times $0 < k_1 < ... < k_n$ and Borel sets $A_1$, $A_2$, …, $A_n \subset \Re$

$$P\{X(k_1) \in A_1, ..., X(k_n) \in A_n\}$$
$$= \int_{A_1} \cdots \int_{A_n} \rho(k_1, 0, x_1)\rho(k_2 - k_1, x_1, x_2) \cdots \rho(k_n - k_{n-1}, x_{n-1}, x_n) dx_1 \cdots dx_n ,$$

where $\rho(k, x, y) = \frac{1}{\sqrt{2\pi k}} \exp\left( - \frac{(x-y)^2}{2k} \right)$ (2.22)

defined for any $x$, $y \in \Re$ and $k > 0$ is called the transition density.

## 2.4 Stochastic filtering

The essence of stochastic filtering is the computation of time-varying probability density function for the state of the observed system (Bucy & Joseph 1968; Bucy 1970; Andrew 1970; Julier & Uhlmann 1997). This is because the conditional probability embodies all the statistical information of the state and in the initial conditions. A method, called the *innovation method (or approach)*, for stochastic filtering is described by Kailath (Kailath 1968, 1970, 1998). It is a loop where the corrections for the estimates are made based on prediction errors. In this approach all information, which is currently available, are represented by the error (that is the difference between the observed signal and the estimated signal). This error, called the prediction error and its

statistical history provide the means to reduce the error both in the immediate future and in the statistical future. In the immediate future, the error is then used to control the evolution of the computational model so that the probability density function flows into a distribution that minimizes the discrepancy between the expectation (estimated value) and the observation (Dawes 1993). Figure-2.1 shows such a predictor-error-corrector loop.

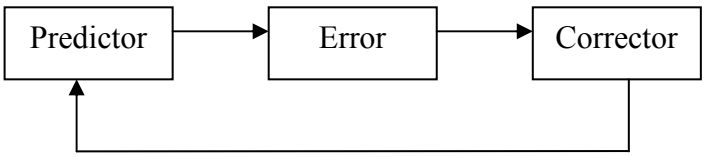



Figure-2.1: Predictor-error-corrector loop

Earlier researchers have evaluated the conditional density function considering the measurement as Markov process (Bucy & Joseph 1968; Stratonovich 1959). However, often signals presented by sensors are not Markov processes but they are Martingale Processes. Martingale Process has a complex structure which is linked to the specification of filtration. Filtration describes the information available from knowledge of the process as it evolves over time. This increasing structure provides a home in the present time for all the probabilistic information which is contained in the past history of the observations and thus making the past history accessible for the operation in the present.

A loop similar to the predictor-error-corrector is developed using the Schrödinger wave equation (SWE) within a neural network framework that incorporate $\sigma$-field of increasing structure in the evolution of the conditional probability density function (Dawes 1989a, 1989b, 1989c, 1992). The difference between the innovation approach for Kalman filter and the QRNN filter is that in the Kalman filter the probability density is assumed to be Gaussian and so it can completely be characterized by mean and

covariance matrix. In QRNN filter, the probability density is not assumed to be Gaussian. Given a wave packet, modulus-squared of this wave is defined as the probability density function. The dynamics of the wave is governed by the SWE. The density is used to compute the conditional expectation of the current measurement. The difference between the expectation and the measurement is fed forward where it is used to *control the flow* of the wave function and hence the evolutions of the probability density function.

## 2.5 Neural network and its elements

Artificial neural networks are computational algorithms whose development has taken inspiration from some aspects of the way it is understood that the biological nervous system functions. It resembles the biological nervous system in two respects:

- Knowledge is acquired by the network through learning process,
- Interneuron connection strengths known as synaptic weights are used to store the knowledge.

Cells in the biological neuron receive incoming impulses via *dendrites* (receiver) by means of chemical processes. If the number of incoming impulses exceeds certain threshold value the neuron discharges it off to other neurons through its synapses which determine the impulse frequency to be fired off. Analogous to this biological process, the processing unit or neuron of an artificial neural network consists of three main components

- Synaptic weights connecting the neurons
- Summation function within the neurons and
- Activation (or transfer) function that produce output for the neuron

Synaptic weights characterize themselves with their strength (a numerical value) which corresponds to the importance of the information coming from each neuron. In other

words, the information is encoded in these strength-weights. The summation function is used to calculate a total input signal by multiplying their synaptic weights and summing up all the products. Figure-2.2 below shows a typical neuron and its output, where $x_i$ and $w_i$ with $1 \le i \le n$ are input data and synaptic weights respectively, $u$ represents summation within the node and $f(u)$ is the activation function (or transfer function). The activation function transforms the summed up input signal received from the summation function into an output.

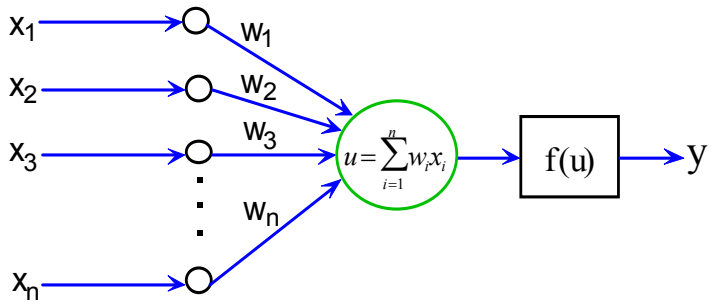



Figure-2.2: A typical neural network

The activation function can be either linear or non-linear. A particular activation function is chosen to satisfy some specification of the problem that the neuron is attempting to solve. There are varieties of activation functions such as sigmoid function as shown in equation (2.23).

$$f(u) = \frac{1}{1+e^{-u}} \tag{2.23}$$

In the last decade, artificial neural networks have been applied in many real world situations and found to be effective in solving complex, large-scale problems arising in the fields of engineering, business, medical and biological sciences. The design of a complete neural network to solve a particular problem requires various considerations such as:

- Problem identification, that is, whether the problem can be solved under neural network paradigm
- Nature of input-output space, that is, defining input-output space and their structures
- Network architecture, that is, the number of neurons and layers, connections and weight vectors in the network
- Learning algorithms, that is, whether the learning algorithm would be feed-forward, back-propagation or recurrent
- Activation (or transfer) function consideration (e.g., sigmoid or tangent function) in order to get the output of the neurons.

Appropriate combination of all these factors leads to the development of the artificial neural network that would be able to solve a problem under consideration. It is important to note that the adaptability, reliability and robustness of a neural network depend on the source, range, quantity, and quality of the signal or data.

### 2.5.1 Recurrent neural network

A recurrent neural network is an artificial neural network which has feedback loop, that is, some of its outputs are connected back to its inputs. The feedback connections, as shown in Figure-2.3, originate from its hidden neurons as well as output neurons. The presence of the feedback loops has a profound impact on the learning capability of the network and its performance. Moreover, the feedback loops involve in particular branches composed of unit-delay (denoted by $z^{-1}$) which results in a nonlinear dynamic behaviour by virtue of the nonlinear nature of the neuron. In fact, nonlinear dynamics plays an important role in the storage function of a recurrent network. The basic principle of recurrent structure is inspired by various ideas in statistical physics and they share the distinctive features such as nonlinear computing units, symmetric synaptic

connections and abundant use of feedback (Haykin 1994, 2001). All these characteristics are exemplified by the Hopfield network – a recurrent neural network that embodies a profound physical principle namely that of storing information in a dynamically stable configuration. Mathematical representation of this network is as follows:

$$x(k+1) = -\beta x(k) + W\sigma(x(k)) + \Gamma u(k) \tag{2.23}$$

$$y(t) = Cx(k) \tag{2.24}$$

where $x \in \Re^n$ is a state vector, $W$ is the weight matrix, $\beta$ is the feedback matrix (usually a diagonal matrix), $\Gamma$ is a vector describing the weighting on the input, $\sigma(x(k))$ is a transfer function, and $y(k)$ is the output of the network where $C$ is a transformation operator. Goal of this network is to minimize the error function

$$error = \sum_{i=1}^{n-1}[x(i) - y(i+1)] \tag{2.25}$$

where $x(i)$ is the $i-$th state and $y(i+1)$ is the observation at $i+1$.

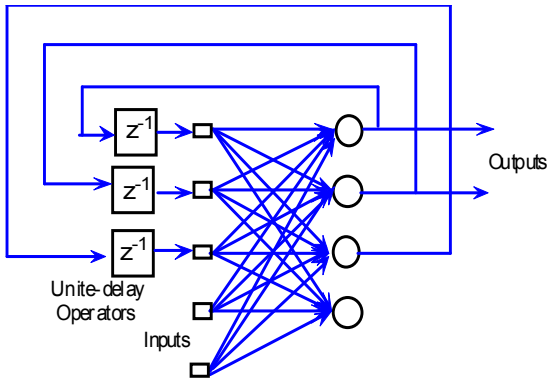



Figure-2.3: A typical recurrent neural network structure

## 2.6 Conclusion

In this chapter the basic ideas of stochastic process and relevant definitions, stochastic filtering, neural network along with its structure and recurrent neural network have been

discussed. The fundamental concept of the neural network is that neurons are interconnected to form a purposeful network that can gain ability to mimic a system if the network is trained by examples under certain roles or algorithms. Since recurrent neural network has the ability to learn the nonlinear dynamics, therefore, applying the training method based on the extended Kalman filter can solve the nonlinear filtering problem (Haykin 2001). However, in this thesis the nonlinear stochastic filtering problem is devised through the evolution of the probability density function using the algorithm based on the quantum recurrent neural network (discussed in Chapter-4). In recent years, numbers of papers have been contributed in the evolution of the *pdf* (or the covariance matrix) using the concepts of neural networks (Singhal & Wu 1989; Puskorius & Feldkamp 1991; MacKay 1992; Husmeier & Taylor 1999a; Husmeier 1999b; Haykin 2001; Leung & Chan 2003).

A major problem with neural network in practical applications is the impact on the generalization performance caused by over fitting, especially when the input dimension is large and the available training data set is sparse. That is, although a network might be able to learn the underlying dynamics of a given training set, may be incapable of generalising to a new set of data which has not been exposed during training. In fact, many recent developments in neural network have focused on this problem (e.g., Bishop 1995). However, with an appropriate combination of neurons and their interconnections, and with an appropriate set of learning rules it may be possible to develop an efficient computational algorithm for the problem under consideration.

Neural network which is considered in this thesis will broadly be single layer where the entire bunch of neurons will be treated in a spatial coordinate on the real line where the SWE acts as a mediator on the evolution of the *pdf* maintaining the characteristic of the recurrent neural network structure.

In Kalman filter, the dynamical system is described by a model in which the system and measurement noises are white and Gaussian. The word whiteness implies that the noise values are uncorrelated in time. That is, if the value of the noises is known now this knowledge does not help anyway in predicting the noise values at any other future time. Under these three restrictions (linearity, whiteness, and Gaussianity), the Kalman filter is a recursive algorithm through which the conditional expectation of the input signal is evaluated. In the next chapter, two general filters common in the literature such as linear (Kalman filter) and nonlinear (extended Kalman filter) filters will be discussed.

# Chapter 3

# Linear and Nonlinear Filters

## 3.1 Introduction

In Chapter-2 (Section–2.4) it was stated that the solution to the stochastic filtering problem is to compute the conditional probability density function (or *pdf*) based on the past and current measurements of the observed system. Thus, along with the state process $x(t)$ and the measurement process $y(t)$ the conditional probability density function, denoted by $\rho\big(x(t)\,|\,y(t)\big)$, forms the basis for the solution of the stochastic filtering problem. This is because given an initial condition $x(t_0)$, the *pdf* evolves in such a way that it embodies all the statistical information about $x(t)$ which is contained in the available measurements. Using this density along with a minimization procedure, such as mean-square-error, the expected mean of the state provides the estimate of the system state.

In this chapter, both linear and nonlinear models are considered for the filtering problem. If the dynamics of the model are linear then the state of the system is estimated based on the evolution of the conditional density function. It is assumed that the state process $x(t)$ and the measurement process $y(t)$ are jointly Gaussian and statistical properties of the noise processes are known. For nonlinear dynamics, a linearization technique is applied to transform the nonlinear model into a linear model.

In the next section, linear filter and its structure along with its extension to nonlinear filtering algorithms are discussed.

## 3.2 Linear filter (The Kalman filter)

In the last chapter a general model for filtering problem was presented (see Section-2.2, equations (2.1) and (2.2)) from which a linear stochastic differential equation that describes the state process with a discrete time measurement process can be written as follows (Kalman 1960; Kailath 1968; Bucy 1970; Rhodes 1971; Maybeck 1979, 1982):

$$\dot{x}(k) = F(k)x(k) + G(k)w(k) \qquad \text{(State equation)} \qquad (3.1)$$

$$y(k_i) = H(k_i)x(k_i) + v(k_i) \qquad \text{(Measurement process)} \qquad (3.2)$$

where

$x(\cdot,\cdot)$ is an $n$-vector state process of which one sample would generate a state time history,

$F(\cdot)$ is an $n \times n$ system dynamic matrix,

$G(\cdot)$ is an $n \times s$ noise matrix,

$y(\cdot,\cdot)$ is an $m$-vector discrete time measurement process of which one sample provides a particular measurement time history,

$H(\cdot)$ is an $m \times n$ measurement matrix,

$w(\cdot,\cdot)$ is an $s$-vector white Gaussian process noise, and

$v(\cdot,\cdot)$ is an $m$-vector white Gaussian measurement noise.

The state noise process $w(\cdot,\cdot)$ has the following statistics

$$E[w(k)] = 0$$
$$\mathrm{E}[w(k)w'(l)] = \begin{cases} Q(k) & if \quad k = l, \\ 0 & if \quad k \neq l, \end{cases} \tag{3.3}$$

and the measurement noise process $v(\cdot,\cdot)$ has the following statistics

$$E[v(k_i)] = 0$$
$$\mathrm{E}[v(k_i)v'(k_j)] = \begin{cases} R(k_i) & if \quad k_i = k_j, \\ 0 & if \quad k_i \neq k_j, \end{cases} \tag{3.4}$$

for all $k_i, k_j \in T$, where $Q(k)$ is an $s \times s$ symmetric positive semi-definite matrix for all $k \in T$, and $R(k)$ is an $m \times m$ symmetric and positive definite matrix for all $k_i \in T$. In the above equations (3.4) and (3.5), $(\cdot)'$ denotes the transposition of the matrix or vector $(\cdot)$.

In order to initialize the filter, the initial conditions $x(k_0)$ are required (Kailath 1968; Bucy 1970; Rhodes 1971; Maybeck 1979, 1982). Generally, it is modelled as an $n$-vector random variable which is normally distributed and is completely described by its mean $\hat{x}(k_0)$ and covariance $P(k_0)$ as:

$$E[x(k_0)] = \hat{x}(k_0)$$
$$E[(x(k_0) - \hat{x}_0)(x(k_0) - \hat{x}_0)'] = P(k_0) \tag{3.5}$$

The initial state process $x(k_0)$, the noise process $w(\cdot,\cdot)$, and the measurement noise process $v(\cdot,\cdot)$ are all assumed to be uncorrelated and independent of each other.

In order to develop this filtering algorithm to estimate the system state, it is required to combine the measurement data taken from the system and the information provided by the state process along with the statistical information of the uncertainties together so that an optimal filter can be evolved. This would be done here by adopting the Bayesian

measure to find the conditional probability density of the system state conditioned on the entire history of the measurements. Once this is accomplished then the optimal estimate will be defined as the mean (or mode, median) under certain criterion such as minimum mean-square-error so that conditional estimate converges to that estimated value.

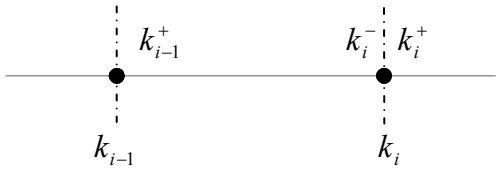



Figure-3.1: Time reference for the density propagation

## 3.2.1 Derivation of the Kalman Filter (KF)

Assume that the measurement data is taken and processed at time $k_{i-1}$ and is denoted by $y(k_{i-1}, \omega_j) = y_{i-1}$. With a Bayesian measure, it is now required to propagate the probability density of the state $x(k_{i-1})$ conditioned on the entire history up to time $k_{i-1}$, that is $\rho_{x(k_{i-1})|y(k_{i-1})}(\xi \mid y_{i-1})$ to the next measurement time $k_i$ (see Figure-3.1) to generate $\rho_{x(k_i)|y(k_i)}(\xi \mid y_i)$. To do this it is assumed that the conditional probability density $\rho_{x(k_{i-1})|y(k_{i-1})}(\xi \mid y_{i-1})$ is Gaussian and is defined by

$$\rho_{x(k_{i-1})|y(k_{i-1})}(\xi \mid y_{i-1}) = \left[ (2\pi)^{n/2} \left| P(k_{i-1}^{+}) \right|^{1/2} \right]^{-1} \times \exp[\bullet] \tag{3.6}$$

where $[\bullet] = \left[ -\frac{1}{2} \left\{ \xi - \hat{x}(k_{i-1}^{+}) \right\}' P^{-1}(k_{i-1}^{+}) \left\{ \xi - \hat{x}(k_{i-1}^{+}) \right\} \right]$.

The conditional mean $\hat{x}(k_{i-1}^{+})$ and the conditional covariance $P(k_{i-1}^{+})$ are defined as

$$\hat{x}(k_{i-1}^{+}) \overset{\Delta}{=} E[x(k_{i-1}) \mid y(k_{i-1}) = y_{i-1}] \text{ and} \tag{3.7}$$

$$P(k_{i-1}^{+}) \overset{\Delta}{=} E\left[ \left\{x(k_{i-1}) - \hat{x}(k_{i-1}^{+})\right\}\left\{x(k_{i-1}) - \hat{x}(k_{i-1}^{+})\right\}' \mid y(k_{i-1}) = y_{i-1} \right] \tag{3.8}$$

It can be observed that the conditional covariance defined in (3.8) is equal to the unconditional covariance. This is because the covariance recursion does not depend on the actual values of the measurement taken, and thus can be computed without knowledge of the realised measurement $y_i$. For this reason it is possible to precompute the time history of the covariance by using $\hat{x}(k_i^{+})$ as the optimal estimate of the system state at time $k_i$.

The solution $x(k_i)$ of the model presented in equation (3.1) can be written as

$$x(k_i) = \Phi(k_i, k_{i-1})x(k_{i-1}) + w_d(k_{i-1}) \text{ with} \tag{3.9}$$

$$w_d(k_{i-1}) = \int_{k_{i-1}}^{k_i} \Phi(k_i, \tau)G(\tau)d\beta(\tau), \tag{3.10}$$

where $\Phi$ is an $n \times n$ state transitional matrix.

Since (3.9) describes the state $x(k_i)$ as a linear combination of $x(k_{i-1})$ and $w_d(k_{i-1})$ then the conditional probability density for $x(k_i)$, written as $\rho_{x(k_i)|y(k_{i-1})}(\xi \mid \zeta_{i-1})$ will be Gaussian provided the conditional probability density for $x(k_{i-1})$ and $w_d(k_{i-1})$, written as $\rho_{x(k_{i-1}),w_d(k_{i-1})|y(k_{i-1})}(\xi, \eta \mid \zeta_{i-1})$ is Gaussian. It is in fact Gaussian because $w_d(k_{i-1})$ is independent of $x(k_{i-1})$ and $y(k_{i-1})$ according to the description of the model (3.1).

For the state $x(k_i)$, the time propagation from $k_{i-1}^{+}$ to $k_i^{-}$ (see Figure-3.1) can be computed as follows. The conditional mean for the relation (3.9) can be expressed as:

$$E\left[x(k_i) \mid y(k_{i-1}) = y_{i-1}\right] = E\left[\Phi((k_i, k_{i-1})x(k_{i-1}) + w_d(k_{i-1}) \mid y(k_{i-1}) = y_{i-1}\right]$$

$$= \Phi(k_i, k_{i-1})E\left[x(k_{i-1}) \mid y(k_{i-1}) = y_{i-1}\right]$$

$$+ E\left[w_d(k_{i-1}) \mid y(k_{i-1}) = y_{i-1}\right] \tag{3.11}$$

In (3.11), since $w_d(k_{i-1})$ is independent of $y(k_{i-1})$, therefore, the conditional mean turns into unconditional mean and this mean is zero as of the description of the model (3.1). Thus the conditional mean stands at

$$E\left[x(k_i) \mid y(k_{i-1}) = y_{i-1}\right] = \Phi(k_i, k_{i-1}) E\left[x(k_{i-1}) \mid y(k_{i-1}) = y_{i-1}\right]. \tag{3.12}$$

If the conditional mean in (3.12) is denoted by $\hat{x}(k_i^-)$ before the measurement $y(k_i) = y_i$ is taken, that is

$$\hat{x}(k_i^-) \overset{\Delta}{=} E\left[x(k_i) \mid y(k_{i-1}) = y_{i-1}\right], \tag{3.13}$$

then combining the notation presented in (3.7) and the term in (3.13), the time propagation for the conditional mean can be written as

$$\hat{x}(k_i^-) = \Phi(k_i, k_{i-1}) \hat{x}(k_{i-1}^+). \tag{3.14}$$

Similarly, defining $P(k_i^-)$ as a conditional covariance for the state $x(k_i)$ before the measurement $y(k_i) = y_i$ is taken, the conditional covariance can be written as

$$P(k_i^-) \overset{\Delta}{=} E\left[ \left\{x(k_i) - \hat{x}(k_i^-)\right\}\left\{x(k_i) - \hat{x}(k_i^-)\right\}' \mid y(k_{i-1}) = y_{i-1} \right], \tag{3.15}$$

and the conditional covariance propagation from time $k_{i-1}^+$ to $k_i^-$ (see Figure-3.1) is written as

$$P(k_i^-) = \Phi(k_i, k_{i-1}) P(k_{i-1}^+) \Phi'(k_i, k_{i-1})$$

$$+ \int_{k_{i-1}}^{k_i} \Phi(k_i, \tau) G(\tau) Q(\tau) G'(\tau) \Phi'(k_i, \tau) d\tau \tag{3.16}$$

It can be observed from (3.14) that if $\hat{x}(k_i^-)$ is used as an estimate of $\hat{x}(k_i)$ before the new measurement $y(k_i)$ is taken, then the difference $\left\{x(k_i) - \hat{x}(k_i^-)\right\}$ is simply the error

for a particular set of measurement history $y(k_i) = y_{i-1}$ and so $P(k_i^-)$ is the conditional covariance of the state as well as the error. Using above definitions (given in (3.14) and (3.16)), the propagation of density function from time $k_{i-1}^+$ to time $k_i^-$ (see Figure-3.1) with current measurement available at a time $k_i$, that is $y(k_i, \omega_j) = y_i$, the new density function written as $\rho_{x(k_i)|y(k_i)}(\xi \mid y_i)$ can be defined.

**Table-3.1: Algorithmic structure of the Kalman filter**

*The optimal state estimate is propagated from the measurement time* $k_{i-1}$ *to* $k_i$*. The entire algorithm can be describe as follows:*

*Initialization: For* $i = 0$,

$$\hat{x}(k_0) = E[x(k_0)]$$

$$P(k_0) = E\left[\{x(k_0) - \hat{x}(k_0)\}\{x(k_0) - \hat{x}(k_0)\}'\right]$$

*Computation: For* $i = 1,2,...,$

*State estimate propagation*

$$\hat{x}(k_i^-) = \phi(k_i, k_{i-1})\hat{x}(k_{i-1})$$

*Error covariance propagation*

$$P(k_i^-) = \Phi(k_i, k_{i-1})P(k_{i-1}^+)\Phi'(k_i, k_{i-1}) + \int_{k_{i-1}}^{k_i} \Phi(k_i, \tau)G(\tau)Q(\tau)G'(\tau)\Phi'(k_i, \tau)d\tau$$

*Kalman gain Matrix*

$$M(k_i) = P(k_i^-)H'(k_i)[H(k_i)P(k_i^-)H'(k_i) + R(k_i)]^{-1}$$

*State estimate update*

$$\hat{x}(k_i^+) = \hat{x}(k_i^-) + M(k_i)[y(k_i) - H(k_i)\hat{x}(k_i^-)]$$

*Error covariance update*

$$P(k_i^+) = P(k_i^-) - M(k_i)H(k_i)P(k_i^-)$$

*where* $M(k)$ *is called the Kalman Gain matrix and* $P(k)$ *is called a posterior covariance matrix. The negative/positive sign in the superscript indicates the prior/posterior information for the respective notations.*

Using Bayes' rule repeatedly and the same analogy as described above for the propagation of the density function from one time reference to the next, the final conditional mean $\hat{x}(k_i^+)$ at time $k_i^+$ (see Figure-3.1) can be rewritten as (Maybeck 1979, 1982)

$$\hat{x}(k_i^+) = \hat{x}(k_i^-) + P(k_i^-)H'(k_i)\left[H(k_i)P(k_i^-)H'(k_i) + R(k_i)\right]^{-1}\left[y_i - H(k_i)\hat{x}(k_i^-)\right], \quad (3.17)$$

and the covariance update as

$$P(k_i^+) = P(k_i^-) - P(k_i^-)H'(k_i)\left[H(k_i)P(k_i^-)H'(k_i) + R(k_i)\right]^{-1} H(k_i)P(k_i^-). \quad (3.18)$$

With the definition of Kalman gain matrix, the algorithm is summarised in Table-3.1.

### 3.2.2 The KF algorithmic structure

The filter discussed above is based on the linear stochastic model with discrete time measurement process. There are number of assumptions have been made in developing this filter such as the noise processes are Gaussian, uncorrelated and that the state $x(k)$ and the measurement process $y(k)$ are jointly Gaussian. As a result, the state of the system is completely determined by its mean and covariance matrices. Conditional probabilities are propagated together from the same initial time by assuming that they are Gaussian and because of this, the algorithm maintains the conditional density of the state conditioned on the measurement taken. The state estimate $\hat{x}(k_i^+)$ is optimal. This is because this estimate is not only the conditional mean but also the conditional mode as it maximizes the conditional density of the state $x(k)$ conditioned on the entire history of measurements. This is also a recursive algorithm. Because the optimal state estimate $\hat{x}(k_i^-)$ is propagated from measurement time $k_{i-1}$ to measurement time $k_i$ and once the measurement is taken at time $k_i$ then the estimate is updated by defining the

Kalman gain $M(k_i)$ and employing it in both the mean $\hat{x}(k_i^+)$ and covariance relation $P(k_i^+)$. Moreover, the algorithm acts like a predictor-error-corrector loop as mentioned earlier (see Section-2.4). This is because from the best estimate $\hat{x}(k_i^-)$ it is possible to compute $[H(k_i)\hat{x}(k_i^-)]$, which is the best prediction of what the measurement at the time $k_i$ would be before the actual measurement is taken. The measurement error, difference between predicted estimate and the current measurement, $[y_i - H(k_i)\hat{x}(k_i^-)]$, is then computed. This error, known as *innovation process*, is then multiplied by the Kalman gain to obtain the correction term which is then added to $\hat{x}(k_i^-)$ to compute $\hat{x}(k_i^+)$.

In summary, the Kalman filter algorithm requires defining both the structure of the model for the system and description of the uncertainty. The structure of the model is established by $F$, $\Phi$, and $G$ while the uncertainties are specified by the initial conditions $\hat{x}(k_0)$, $P(k_0)$, $Q$, and $R$.

Given all these advantages, however, this filter (if implemented as shown in Table-1) has number of well documented numerical difficulties (Robert 1971). In updating the conditional mean $\hat{x}(k_i^+)$ it requires the inversion of $n \times n$ matrix (see equations 3.17 and 3.18). This is possible only when matrices $P$ and $R$ are positive definite so that inverses of theses two matrices exists. This computation can be reduced if the measurement vector is significantly smaller than the state vector, that is if $m << n$. Moreover, if the measurements are very accurate that is the eigenvalues of $R$ are very small compared to those of $P(k_i^-)$ then the error covariance matrix can have negative eigenvalues and because of this the filter may diverge. To overcome this difficulty, an alternative formulation has been developed. This method is called the square root

Kalman filter (Mehra 1971; Maybeck 1979, 1982; McGee & Schmidt 1985). This filter is capable of propagating and updating the state estimate and error covariance matrix through square root matrix or inverse square root matrix. This formulation has helped increase the numerical accuracy and thus reduced the numerical difficulties in implementations of the Kalman filter.

## 3.3 Nonlinear filter (The Extended Kalman Filter)

In the last section, filter is developed based on the linear stochastic differential equation that represented the dynamics of the system state and the measurement relation. It has been shown that under the assumption that the state and measurement noise processes are Gaussian and uncorrelated, the filter is able to estimate the state optimally. However, linear models are not always useful, especially when the dynamics of the system are strongly nonlinear. In such a case development of a nonlinear filter is essentially important. For this, a nonlinear stochastic differential equation that describes the state and measurement relation is defined by

$$\dot{x}(k) = f\big(x(k),k\big) + G(k)w(k) \qquad \text{(State equation)} \qquad (3.19)$$

$$y(k_i) = h\big(x(k_i),k_i\big) + v(k_i) \qquad \text{(Measurement process)} \qquad (3.20)$$

where

$x(\cdot,\cdot)$ is an $n$-vector state process,

$f(\cdot,\cdot)$ is an $n$-vector function, which is assumed to be Lipschitz in its state variable and piecewise continuous in its time domain

$G(\cdot)$ is an $n \times s$ noise matrix,

$h(\cdot,\cdot)$ is an $m$-vector of functions,

$w(\cdot,\cdot)$ is an $s$-vector zero mean white Gaussian noise, and

$v(\cdot,\cdot)$ is an $m$-vector zero mean white Gaussian measurement noise.

It is assumed that the state noise process $w(\cdot,\cdot)$ and the measurement noise process $v(\cdot,\cdot)$ have the same statistics as defined in relations (3.3) and (3.4).

Given the above nonlinear model, the objective here is to develop an algorithm to estimate the system state. The idea for the development of this filter is to linearize the nonlinear part of the model about a nominal state trajectory at each instant of time and then apply the linear filtering technique discussed in the previous section (Section-3.2.1). This filter is known as the Extended Kalman Filter (EKF) (Smith and Schmidt 1961; Andrew 1970; Schmidt 1981; Maybeck 1982; McGee and Schmidt 1985; Julier and Uhlmann 1997).

### 3.3.1 Derivation of the EKF

Let $x_{nom}(k)$ be the nominal state trajectory at time $k \in T$ that start at an initial condition $x_{nom}(k_0) = x_{nom}(0)$ and satisfy the deterministic equation

$$\dot{x}_{nom}(k) = f\left(x_{nom}(k), k\right). \tag{3.21}$$

Similarly, the nominal measurement sequence, associated with the nominal state trajectory, would be

$$y_{nom}(k_i) = h\left(x_{nom}(k_i), k_i\right) \tag{3.22}$$

The deviation of the state from the assumed nominal trajectory can be written as $[x(k) - x_{nom}(k)]$ for all $k \in T$ which is also a stochastic process and satisfy the perturbed equation defined by

$$[\dot{x}(k) - \dot{x}_{nom}(k)] = f(x(k), t) - f(x_{nom}(k), k) + G(k)w(k). \tag{3.23}$$

This equation can be expanded about the nominal state $x_{nom}(k)$ using Taylor series as

$$[\dot{x}(k) - \dot{x}_{nom}(k)] = \left.\frac{\partial f(x,k)}{\partial x}\right|_{x=x_{nom}(k)} [x(k) - x_{nom}(k)] + h.o.t + G(k)w(k) \tag{3.24}$$

which in turn can be written as a first order approximation of the form

$$\delta\dot{x}(k) = F[k; x_{nom}(k)]\delta x(k) + G(k)w(k) \tag{3.25}$$

where $\delta x(k)$ represents the first order approximation of the process $[x(k) - x_{nom}(k)]$, and $F[k; x_{nom}(k)]$ is the $n \times n$ matrix consists of the partial derivatives of $f(\cdot)$ with respect to its state variable evaluated along the nominal state as defined by

$$F[k; x_{nom}(k)] = \left.\frac{\partial f(x,k)}{\partial x}\right|_{x=x_{nom}(k)} \quad \text{or} \tag{3.26}$$

$$F(\cdot) = \left[\begin{array}{ccc} \frac{\partial f_1(x,k)}{\partial x_1} & \cdots & \frac{\partial f_1(x,k)}{\partial x_n} \\ \vdots & \frac{\partial f_i(x,k)}{\partial x_i} & \vdots \\ \frac{\partial f_n(x,k)}{\partial x_1} & \cdots & \frac{\partial f_n(x,k)}{\partial x_n} \end{array}\right]_{x=x_{nom}(k_i)} \tag{3.26a}$$

Now comparing equations (3.23) and (3.25), it can be observed that the solution to the equation (3.25) is a viable approximation to the solution of the equation (3.23) provided that the deviations from the nominal state trajectory are small enough for the higher order terms in (3.24) to be negligible.

In a similar manner, the measurement deviation at each time $k_i$ can also be established using equations (3.20) and (3.22) as

$$[y(k_i) - y_{nom}(k_i)] = h(x(k_i), k_i) - h(x_{nom}(k_i), k_i) + v(k_i) \tag{3.27}$$

**Table-3.2: Algorithmic structure of the Extended Kalman filter**

*Given the nonlinear system described by (3.19) and (3.20), the following steps are required to develop the EKF*

**Step1***: Construction of matrices*

$$F(k;\hat{x}(k/k_i)) = \left.\frac{\partial f(x,k)}{\partial x}\right|_{x=\hat{x}(k/k_i)}, \quad H[k_i;\hat{x}(k_i^-)] = \left.\frac{\partial f(x,k_i)}{\partial x}\right|_{x=\hat{x}(k_i^-)}$$

*which are Jacobians of the form (3.26a) and (3.29a) described earlier.*

**Step 2***: The measurement update incorporates the measurement* $y(k_i,\omega_j) = y_i$ *by means of*

$$M(k_i) = P(k_i^-)H'[k_i;\hat{x}(k_i^-)]\left[H[k_i;\hat{x}(k_i^-)]P(k_i^-)H'[k_i;\hat{x}(k_i^-)] + R(k_i)\right]^{-1}$$

$$\hat{x}(k_i^+) = \hat{x}(k_i^-) + M(k_i)\left[y_i - h[\hat{x}(k_i^-),k_i]\right]$$

$$P(k_i^+) = P(k_i^-) - M(k_i)H[k_i;\hat{x}(k_i^-)]P(k_i^-)$$

*This estimate is propagated to the next sample time* $k_{i+1}$ *by integrating*

$$\dot{\hat{x}}(k/k_i) = f[\hat{x}(k/k_i),k]$$

$$\dot{P}(k/k_i) = F[k;\hat{x}(k/k_i)]P(k/k_i) + P(k/k_i)F'[k;\hat{x}(k/k_i)] + G(k)Q(k)G'(k)$$

*from time* $k_i$ *to* $k_{i+1}$ *with initial condition*

$$\hat{x}(k_i/k_i) = \hat{x}(k_i^+)$$

$$P(k_i/k_i) = P(k_i^+).$$

*After integrating,* $\hat{x}(k_{i+1}^-)$ *and* $P(k_{i+1}^-)$ *are defined as*

$$\hat{x}(k_{i+1}^-) = \hat{x}(k_{i+1}/k_i)$$

$$P(k_{i+1}^-) = P(k_{i+1}/k_i)$$

*to use in the next measurement update. For the first interval from* $k_0$ *to* $k_1$*, the value* $\hat{x}(k_0)$ *and* $P(k_0)$ *is considered as the initial value. The time propagation relations are expressed as follows*

$$\hat{x}(k_{i+1}^-) = \hat{x}(k_i^+) + \int_{k_i}^{k_{i+1}} f[\hat{x}(k/k_i),k]dk$$

$$P(k_{i+1}^-) = \Phi[k_{i+1},k_i;\hat{x}(\tau/k_i)]P(k_i^+)\Phi'[k_{i+1},k_i;\hat{x}(\tau/k_i)] + \int_{k_i}^{k_{i+1}} \Phi[k_{i+1},k_i;\hat{x}(\tau/k_i)]G(k)Q(k)G'(k)\Phi'[k_{i+1},k_i;\hat{x}(\tau/k_i)]dk$$

*where* $\Phi[k_{i+1},k_i;\hat{x}(\tau/k_i)]$ *denotes the state transition matrix which is associated with* $F[\tau;\hat{x}(\tau/k_i)]$ *for all* $\tau \in [k_i,k_{i+1})$.

Linearization of this relation gives the measurement perturbation model as

$$\delta y(k_i) = H[k_i; x_{nom}(k_i)]\delta x(k_i) + v(k_i) \tag{3.28}$$

where $\delta y(k_i)$ represents an approximation of the process $[y(k_i) - y_{nom}(k_i)]$, and $H[k_i; x_{nom}(k_i)]$ is the $m \times n$ matrix consists of the partial derivatives of $h(\cdot)$ with respect to its state variable evaluated along the nominal state as defined by

$$H[k_i; x_{nom}(k_i)] = \left.\frac{\partial h(x, k_i)}{\partial x}\right|_{x=x_{nom}(k_i)} \quad \text{or} \tag{3.29}$$

$$H(\cdot) = \left[\begin{array}{ccc} \frac{\partial h_1(x,k)}{\partial x_1} & \cdots & \frac{\partial h_1(x,k)}{\partial x_n} \\ \vdots & \frac{\partial h_i(x,k)}{\partial x_i} & \vdots \\ \frac{\partial h_n(x,k)}{\partial x_1} & \cdots & \frac{\partial h_n(x,k)}{\partial x_n} \end{array}\right]_{x=x_{nom}(k_i)} \tag{3.29a}$$

Having these two equations (3.25) and (3.28) as a linearized model for the nonlinear dynamics described by (3.19) and (3.20), the nonlinear filter can be developed using liner filtering technique about a priori nominal state $x_{nom}(\cdot)$ evaluating $F[k; x_{nom}(k)]$ and $H[k_i; x_{nom}(k_i)]$. It is assumed that these derivatives are exists. The input measurement for this filter at time $k_i$ would be the difference $[y(k_i) - y_{nom}(k_i)]$ and output of such a filter would be the optimal estimate of $\delta x(k)$ for all $k \in T$. With this linearization, the Extended Kalman filter is summarised in Table-3.2.

## 3.3.2 The EKF algorithmic structure

The filter developed above is based on the model that describes the system state through a nonlinear stochastic differential equation and a discrete time nonlinear measurement function. Noise processes were assumed to be zero mean white Gaussian processes. The

key is that the state equations are repeatedly linearized about each estimate once it has been computed. As soon as a new state estimate is made, a new and better state trajectory is incorporated into the estimation process and this is how the assumption is validated that deviation from the nominal trajectory are small enough to allow the linear technique to be employed.

It can be observed that with the introduction of two Jacobian matrices, the linearization process is achieved through the first order approximation of the Taylor series expansion. These approximations, however, introduce a large error in the true posterior mean and covariance of the transformed state variable which leads to sub-optimal performance and sometimes divergence of the filter (Schlee *et al.* 1967; Sorenson 1970; Robert 1971; McGee and Schmidt 1985; Bair 1993; Magnus *et al.* 2000; Bar-Shalom *et al.* 2001; Haykin 2001; Mohinder & Angus 2001). This is because of the truncation in Taylor series expansion or nontrivial representation of the covariance matrices about the noise. In practice, the use of extended Kalman filter has well known drawbacks (Julier & Uhlmann 1995). First, linearization can produce highly unstable filter if the time step for the integration is not sufficiently small. Secondly, the derivation of the Jacobian matrices is nontrivial in most applications and often leads to significant implementation difficulties. And thirdly, a small time step interval implies a high computational overhead as the number of calculations increases for the generation of the Jacobian, updates of state estimate, and covariance matrices.

To overcome numerical difficulties of the EKF, the immediate approach was the square-root filter. This filter gives better numerical stability and update of the covariance matrix at the expense of added computational complexities (Andrews 1968, Thornton 1976) some of which were finally refined (Kailath 1984). The other notable developments for the nonlinear filtering were the Unscented Kalman Filter (UKF)

(Julier & Uhlmann 1995) and the Particle Filter (Carpenter *et al.* 1999; Merwe *et al.* 2000; Arulampalam *et al.* 2005). The UKF addresses the linearization issues of the EKF by using a minimal set of carefully chosen sample points that capture the posterior mean and covariance up to second order accurately. However, the computational burden of the UKF remains same as that of the EKF (Haykin 2001).

The Particle Filter (also known as Sequential Monte Carlo Methods) is a simulation technique. The objective of this filter is to track a variable of interests as it evolves over time by constructing a sample-based representation of the entire *pdf*. The filter represents the posterior distribution of the state variables by a system of particles that evolves and adapts recursively as new information becomes available (Carpenter *et al*. 1999). The advantage of the filter is that with sufficient particles the filter approaches optimal estimate accurately. But a large numbers of particles are required to provide adequate approximation for which an expensive computational procedure becomes inevitable.

## 3.4 Conclusion

In this chapter two filtering algorithms; one based on the linear model (Kalman filter) and the other based on the nonlinear model (extended Kalman filter) have been discussed. These filters are optimal in the sense that they incorporate all available information, regardless of their precision, to estimate the current state of the system. It has been observed that in either case the evolution of the conditional probability density function (and consequently the conditional mean and conditional covariance) combines all measurement information and a priori knowledge about the system to produce optimal estimate in such a way that the error is minimized (Maybeck 1979, 1982).

The linear filter estimates the system state through computing the conditional mean and covariance matrix in a recursive manner. Estimation of the state by this filter is not only

the conditional mean but also the conditional mode as it maximises the conditional density of the state conditioned on the entire history of measurements. The covariance matrix measures the uncertainty in the estimates. Since the state estimate $\hat{x}(k_i^+)$ is an optimal estimate, therefore, $P(k_i^+)$ represents not only the state covariance but also the covariance of the error committed by the estimate.

Filtering algorithm for the nonlinear model is developed based on the assumption that the deterministic part of the nonlinear state process can be linearized about a nominal state trajectory. Computationally this filter has an advantage that the linear filtering method can be applied after the model is linearized. The disadvantage is that the filter may cause large magnitude of error if the true value of the state differs significantly from the nominal state trajectory. Assumptions in developing this filter are fairly tight and the computational efforts associated with matrices increases exponentially as the number of state increases.

In this thesis, a filter has been proposed which is based on the principles of quantum mechanics under neural network paradigm where the Schrödinger wave equation plays the key parts. This filter, called the QRNN filter, is outlined in the next chapter.

# Chapter 4

# Quantum Recurrent Neural Network and its construction

## 4.1 Introduction

An artificial neural network consists of a set of interconnected neurons which are trained to learn the behaviour of a given system or process. Information regarding the process (or system) is stored in the weights associated with the interconnections. The number of neurons, learning rules, number of layers and nature of interconnections, results in a number of different possible neural network architectures for a given problem. Of these, an architecture which is suitable for stochastic filtering was discussed in references (Dawes 1989a, 1989b, 1989c, 1992, 1993, Sanjay and Zia 2001, Behera *et al.* 2005b). This architecture requires a neural lattice which is confined to a single dimensional structure. The idea is that each neuron in the network mediates a spatio-temporal field with a unified quantum activation function that aggregates the *pdf* information of the observed signal. The activation function is the result of the Schrödinger wave equation (SWE). This equation has manifold applications in quantum mechanics and particle physics (see Chapter-1, Figure-1.1). Transformation of this equation into a neural network results in the Quantum Recurrent Neural Network (QRNN). The architecture of this network and how this network is used for filtering are outlined in this chapter along with examining the following features:

- numerical procedures for solving the Schrödinger wave equation,
- numerical stability of the solution,
- training, and learning schemes for the QRNN,
- evolution of the *pdf* with the QRNN filter.

## 4.2 The Schrödinger wave equation

The use of the Schrödinger wave equation is well established in quantum mechanics (Schiff 1968; Peleg *et al.* 1998). The time-dependent form of this equation in vector notation is given by

$$i\hbar \frac{\partial \psi(\vec{r},t)}{\partial t} = -\frac{\hbar^2}{2m}\nabla^2 \psi(\vec{r},t) + V(\vec{r},t)\psi(\vec{r},t) \tag{4.1}$$

where $\hbar$ is the universal constant (i.e., the Planck's constant divided by $2\pi$), $i$ is the imaginary unit, $\nabla^2$ is the Laplace operator, $\psi(\vec{r},t)$ is the wave function at space-time point $(x,y,z;t)$, $m$ is the mass of the quantum particle, and $V(\vec{r},t)$ is called the potential function (or field). The constant $\hbar$ is related to wave length $\lambda$, and momentum (or velocity) $\kappa$ by $\hbar = \kappa\lambda/2\pi$. Equation (4.1) is a homogenous complex-valued partial differential equation. The solutions of this equation have the following three properties:

- $\psi(\vec{r},t)$ can interfere with itself so that it can account for the results of diffraction,
- $\psi(\vec{r},t)$ is large in magnitude where the quantum particle is *likely to be* and small elsewhere, and
- $\psi(\vec{r},t)$ to be considered as describing the behaviour of a single particle of mass $m$.

The key with these properties is that the wave packet $\psi(\cdot)$ is regarded as a measure of probability of finding a particle of mass $m$ at a particular position $\vec{r}$ with respect to the origin of its region. Since the probability must be real and nonnegative, therefore, the product of $\psi(\cdot)$ and its complex conjugate is defined as the probability density function (*pdf*) (Schiff 1968; Bialynicki-Birula & Mycielski 1976; Feynman 1986; Peleg *et al.* 1998). Interpretation of this *pdf* is that it localizes the position of a particle in that this product itself represents a *pdf* for the location of the particle in the region. This *pdf* is given by

$$P(\vec{r},t) = \psi(\vec{r},t) \times \overline{\psi(\vec{r},t)} = \left|\psi(\vec{r},t)\right|^2 \tag{4.2}$$

where $\overline{\psi(\cdot)}$ denotes complex conjugate of $\psi(\cdot)$.

Equation (4.1) is integrable over an interval of interest and this integration can be carried out on a discrete lattice set out in a well such as the one shown in Figure-4.1 (details of this figure will be apparent in Section-4.4 below, Chapter-5 and Chapter-6).

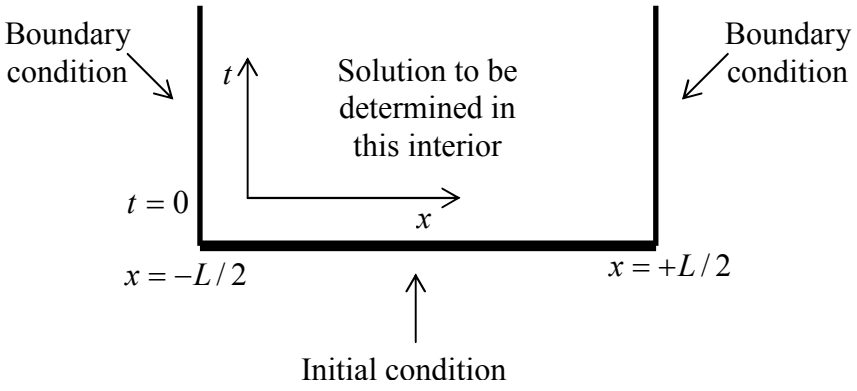



Figure-4.1: A well (box) for the solution of the SWE

It is mentioned earlier that the Schrödinger wave equation has a wave-like solution. As the solutions of SWE evolve over time, their wave-like envelopes (the modulus-squared) disperse and finally diminish which can be seen as a rippling effect of the wave. This causes the particles to lose their identity. Thus the utility of time-dependent

Schrödinger wave equation as a model of coherent particle motion only holds over short time intervals (Dawes 1989a, 1989b, 1989c, 1992, 1993). To offset this, equation (4.1) is transformed by modifying the potential function. The potential function $V(\vec{r},t)$ explains the force field in which particles defined by the wave function are constrained to move. Equation (4.1) is transformed by adding a nonlinear component with the potential function to obtain an extended nonlinear form of the Schrödinger wave equation (Dawes 1989a, 1989b, 1989c, 1992, 1993) which is given by

$$i\hbar\frac{\partial\psi(\vec{r},t)}{\partial t} = -\frac{\hbar^2}{2m}\nabla^2\psi(\vec{r},t) + \left[U(\vec{r},t) + G\left(|\psi|^2\right)\right]\psi(\vec{r},t) \tag{4.3}$$

where $U(\vec{r},t)$, and $G\left(|\psi|^2\right)$ are defined in equations (4.6) and (4.7) respectively. Effect of the nonlinear component (i.e., $G\left(|\psi|^2\right)$) is that it positions the dispersion in the potential field that acts like a shadow of the envelope of $\psi(\cdot)$. Therefore, whereas the dispersion tends to make the particle spread outwards this nonlinear potential shadow tends to make the particle collapse inward.

The position of wave particle in the wave function is defined by $\vec{r}$ with $\psi(x,y,z;t)$. For a single dimensional space the resulting equation is written as

$$i\hbar\frac{\partial\psi(x,t)}{\partial t} = -\frac{\hbar^2}{2m}\nabla^2\psi(x,t) + \left[U(x,t) + G\left(|\psi|^2\right)\right]\psi(x,t), \tag{4.4}$$

Based on this equation the single layer Quantum Recurrent Neural Network (QRNN) is designed. The equation (4.4) is simply referred as the SWE. This equation has an important property which is known as soliton (Schiff 1968; Dawes 1989a, 1989b, 1989c, 1992, 1993). That is, the wave packets propagate in the well in a way that the wave particles collide with each other without changing their shapes, velocity and remain localized. This property is important for the QRNN filter in that it would

*preserve the shape of the wave in the well*. For two or more dimensions, the form of nonlinearity that will produce true soliton solutions is not known. However, many important properties of wave mechanics are known to hold by the resulting SWE described by equation (4.3) (Schiff 1968; Bialynicki-Birula & Mycielski 1976; Boyd 1992).

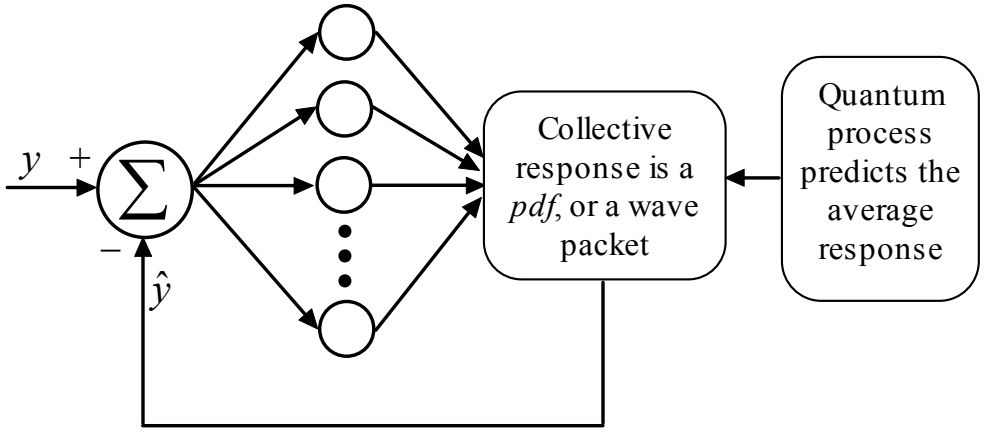



Figure-4.2: Neural lattice of the QRNN

## 4.3 Neural dynamics and architecture of the filter

Let the input signal be represented by a stochastic variable $y$ and let a single dimensional neural lattice be defined in which each node of the lattice acts like a processing element or neuron. Suppose the nodes of this lattice are initialized by an initial wave function $\psi(x,0)$ whose group momentum is $\kappa$. Now, if the dynamics of the wave function are derived by the SWE then by definition the modulus-squared of this wave function would localize the particle in the vicinity of the node at $x = x_0$ at time $t = 0$ with a group momentum of $\kappa$. Suppose a *controlled-input* is given to SWE through $U(x,t)$ of the potential function then SWE drives the wave particles around the trajectory of the input. Suppose that each neuron of the lattice represents some state of the measured (or observed) signal and let each of these neurons receive signals through synaptic connections that represents observations being in state $x$. By doing so, the

modulus-squared $|\psi(x,t)|^2$ would represents an estimate of probability of occurrence which is associated with the observation coded into the neuron located at $x$ at time $t$. Figure-4.2 shows one such neural lattice whose neurons are excited by *controlled-input* that reaches each neuron through the synaptic connections.

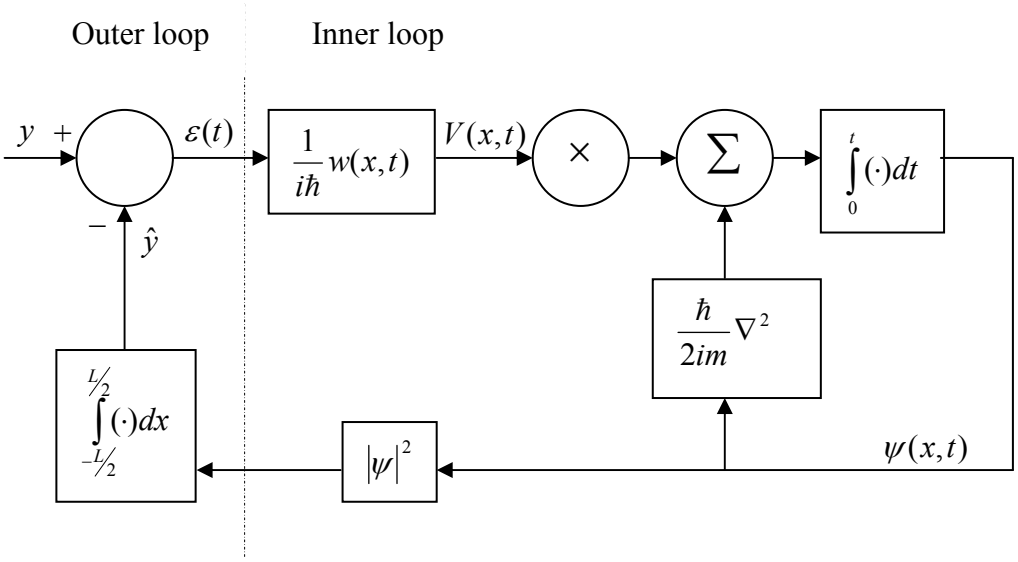



Figure-4.3: Control diagram of the QRNN filter

Thus far, the estimate of the probability density is based solely on the initial condition. This process can be generalized for a continuing series of observations and is done by transforming the estimation error (that is, difference between the estimate and the current measurement) in such a way that the resulting *controlled-input* drives the SWE toward the neurons whose codes are in the best agreement in the sense that the mean-squared-error is the least. In this way, the error is fed back through the potential field as a new measurement becomes available. This is what known as the *innovation approach* mentioned in Chapter-2 (Section-2.4). To transform the estimation errors and to accomplish other computational requirements such as updating the network weights, the following architecture (Figure-4.3) is developed. This architecture will later be referred to as the QRNN filter which is associated with the SWE described in equation (4.4).

The architecture of the filter shows a double loop procedure. In the outer loop the filter receives a signal $y(t)$ that excites an array of one-dimensional neural field (see Figure-4.1) through a set of synaptic weights which are represented by the time varying synaptic weights $w(x,t)$. The key is in the manner in which the error (difference between the current estimate and current observation) is transformed into the wave function. This is done in the inner loop by the activation of the spatial field of the neurons where the state of the quantum object evolves according to SWE. The unified dynamics of the neural field consisting of $N$ neurons is described by the nonlinear Schrödinger wave equation

$$i\hbar\frac{\partial\psi(x,t)}{\partial t}=\left[-\frac{\hbar^2}{2m}\nabla^2+\xi\left(U(x,t)+G\left(|\psi|^2\right)\right)\right]\psi(x,t) \tag{4.5}$$

where the parameter $\xi$ provides an additional excitation to the quantum object and has to be appropriately selected. The potential field consists of two terms, they are defined as follows

$$U(x,t)=-w(x,t)y(t) \tag{4.6}$$

where $w(x,t)$ is the synaptic weight and

$$G\left(|\psi|^2\right)=w(x,t)\int_{-\infty}^{+\infty}x|\psi(x,t)|^2\,dx\,. \tag{4.7}$$

Thus equation (4.5) can be rewritten as

$$i\hbar\frac{\partial\psi(x,t)}{\partial t}=\left[-\frac{\hbar^2}{2m}\nabla^2+\xi w(x,t)\left(\int x|\psi(x,t)|^2\,dx-y(t)\right)\right]\psi(x,t) \tag{4.8}$$

The equation (4.8) is recurrent because of the term $G\left(|\psi|^2\right)$ and the neural field that consists of $N$ neurons is described by the state function $\psi(x,t)$, which is in turn the

solution of the SWE. Since $|\psi(x,t)|^2$ represents the density, this value is used to compute the expectation (in the sense of statistical mean) of the state $x$ at time $t$.

## 4.4 The Crank-Nicholson scheme

The partial differential equation (PDE) is represented by a set of difference equations which in turn allows a numerical solution to the PDE. The idea of doing so is that the partial derivatives are replaced by a relationship between functional values and mash (or lattice) points of some grid system (shown in Figure-4.4) using Taylor series expansion and hence the PDE can be approximated by a set of algebraic equations. This process is known as the discretisation process. If there are no rounding errors in solving the algebraic equations then their exact solution is obtained at each of the mash points. The essential concept of defining stability is that the numerical process should not cause any small perturbations introduced through rounding at any stage to grow and ultimately dominate and distort the solution. There are variety of differencing schemes such as explicit scheme which includes backward, forward, central difference, and the Crank-Nicolson schemes. Numerical solution of a PDE requires a correct choice of differencing schemes since the structure of the PDE itself and the differencing scheme of the PDE may impose restrictions on the choice of parameter values and/or on the region of variables of interest. A particular differencing scheme can be analysed to check if the scheme will produce a stable solution for the problem at hand (Press *et al.* 1992; Evans *et al.* 2000).

From the mathematical point of view, the time-dependent single dimensional Schrödinger wave equation is a partial differential equation describing the dynamics of the wave packet in the presence of a potential field which is given by

$$i\hbar\frac{\partial\psi(x,t)}{\partial t}=\left[-\frac{\hbar^2}{2m}\nabla^2+\xi\left(U(x,t)+G\left(\left|\psi\right|^2\right)\right)\right]\psi(x,t) \tag{4.9}$$

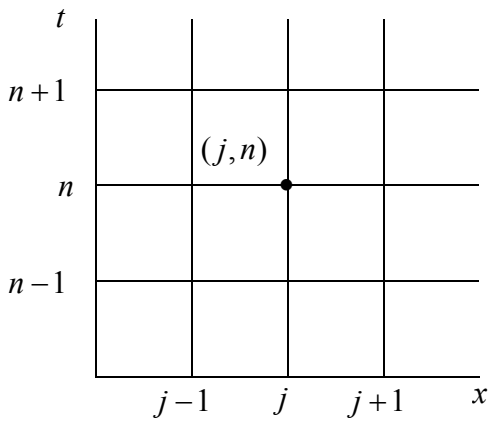



Figure-4.4: A grid (or mesh) system

Equation (4.9) can also be written as

$$i\hbar\frac{\partial\psi}{\partial t}=-\frac{\hbar^2}{2m}\frac{\partial^2\psi}{\partial x^2}+V(x,t)\psi \tag{4.10}$$

where

$$V(x,t)=\xi\left(U(x,t)+G\left(\left|\psi\right|^2\right)\right) \tag{4.11}$$

is called the potential field in the Schrödinger wave equation. Because of the complex structure of this equation, it is intended that the numerical solution meet the following two requirements

- stability of the recursive numerical scheme,
- an accuracy of up to second order in both space and time.

The initial wave function $\psi(x,0)$ satisfies the boundary conditions $\psi\to 0$ as $x\to\pm\infty$. Discretizing equation (4.10) at the spatial point $x_j$, where $j=0,1,2,...,N$, and at the time instant $t^{n+1}$, where $n=0,1,2,...,T$, the *Implicit Scheme* (using central differences) results in the following

$$i\hbar\left[\frac{\psi_j^{n+1}-\psi_j^n}{\Delta t}\right]=-\frac{\hbar^2}{2m}\left[\frac{\psi_{j+1}^{n+1}-2\psi_j^{n+1}+\psi_{j-1}^{n+1}}{(\Delta x)^2}\right]+V_j^n\psi_j^{n+1} \tag{4.12}$$

This difference equation is required to solve the coupling of $\psi_j^{n+1}$ for various $j$ and the values at time level $n$ have to be stored to find the values at time level $n+1$. Rearranging the terms, the scheme (4.12) can be written as

$$\left[\frac{\hbar^2}{2m}\frac{\Delta t}{(\Delta x)^2}\right]\psi_{j-1}^{n+1}-\left[\frac{\hbar^2}{2m}\frac{\Delta t}{(\Delta x)^2}-i\hbar+\Delta tV_j\right]\psi_j^{n+1}+\left[\frac{\hbar^2}{2m}\frac{\Delta t}{(\Delta x)^2}\right]\psi_{j+1}^{n+1}=i\hbar\psi_j^n \tag{4.13}$$

or in simplified form

$$A\psi^{n+1}=\psi^n \tag{4.14}$$

where $A$ is a tri-diagonal matrix which forms with the coefficients of $\psi_{j-1}^{n+1}$, $\psi_j^{n+1}$, and $\psi_{j+1}^{n+1}$. To study the numerical stability of the solution of the differencing scheme it is required to analyze the eigenvalues of the matrix $A$. Using Von Neumann stability (Press *et al.* 1992) analysis, assume that

$$\psi_j^n=\gamma^n\sin(\omega j) \tag{4.16}$$

where $\omega=l\Delta x$ with $l$ is a real spatial wave number and $\gamma=\gamma(l)$ is a complex number that depends on $l$. The key feature of (4.16) is that separating the temporal from the spatial dependence, time dependence of a single eigen-mode is the successive integer powers of the complex number $\gamma$. Thus, the difference scheme is unstable (i.e., exponentially growing modes) if $|\gamma(l)|>1$ for some $l$. The number $\gamma$ is called the amplification factor. To find the value of $\gamma$, substituting (4.16) into (4.12) gives

$$\gamma=\frac{1}{1+i\left[\frac{2\hbar}{m}\frac{\Delta t}{(\Delta x)^2}\sin^2\left(\frac{\omega}{2}\right)+\frac{\Delta t}{\hbar}V_j^n\right]} \tag{4.17}$$

Thus the discretized scheme (4.12 or 4.13) of the Schrödinger wave equation is unconditionally stable if the absolute value of $\gamma$ is less than or equal to one and hence the stability condition is

$$|\gamma| = \left| \frac{1}{1 + i\left[ \frac{2\hbar}{m} \frac{\Delta t}{(\Delta x)^2} \sin^2\left( \frac{\omega}{2} \right) + \frac{\Delta t}{\hbar} V_j^n \right]} \right| \leq 1 \tag{4.18}$$

It is clear from (4.18) that the absolute value of $\gamma$ is always less than or equal to one regardless the values of $\hbar$, $m$, $\Delta t$, and $\Delta x$ and ***hence the implicit scheme is unconditionally stable***. However, the scheme is not unitary (Press *et al.* 1992; Evans *et al.* 2000; Sun *et al.* 2006) – a property that requires that inverse of a matrix is equal to its conjugate transpose. Thus, underlying physical problem is solved if the total probability of finding the quantum object somewhere is unity. This can formally be represented by the modulus-squared norm of the wave function $\psi$ as

$$\int_{-\infty}^{\infty} |\psi|^2 \, dx = 1 \tag{4.19}$$

The initial wave function $\psi(x,0)$ is normalized to satisfy this equation (see Section-4.6). The Schrödinger wave equation (4.10) then guarantees that this condition is satisfied at all times. Writing equation (4.10) in operator form, we have

$$i\hbar \frac{\partial \psi}{\partial t} = \mathcal{H}\psi \tag{4.20}$$

where

$$\mathcal{H} = -\frac{\hbar}{2m} \frac{\partial^2}{\partial x^2} + \frac{1}{\hbar} V(x,t) \tag{4.21}$$

is known as Hamiltonian operator. The closed form solution of (4.20) is given by

$$\psi(x,t) = e^{-i\mathcal{H}t}\psi(x,0) \tag{4.22}$$

where $\psi(x,0)$ is the initial wave packet and the exponential operator is defined by its power series expansion. Applying the Cayley's form (Seborg *et al.* 1989; Press *et al.* 1992; Garcia 1994) for the finite-difference representation of $e^{-i\mathcal{H}t}$, equation (4.22) can be written as

$$\psi_j^{n+1} = \left(1 + \frac{1}{2} i\mathcal{H}\Delta t\right)^{-1} \left(1 - \frac{1}{2} i\mathcal{H}\Delta t\right) \psi_j^n \tag{4.23}$$

Replacing $\mathcal{H}$ by its finite-difference approximation, a complex-valued tri-diagonal system is obtained and is required to be solved. This method is stable, unitary, and second order accurate in space and time. Equation (4.23) is solved iteratively using the initial wave packet at $x_j$, where $j$ represents the number of space interval and at the time instant $t^{n+1}$, where $n$ represents the number of time intervals in the well.

Implementation of the QRNN filter in this way requires that the numerical scheme (defined in equation 4.23) be implemented with an initial wave packet which will act as an initial condition for the solution of the SWE and the wave packet is required to be normalised. This is sharp contrast to the procedure outlined by the earlier authors (see Behera *et al.* 2005a, Dawes 1989b). In their procedure, it requires that the Hamiltonian be normalised at every time instant (see Behera *et al.* 2005a). It is well known fact that the solution of a partial differential equation requires understanding the dynamics of its variables. In the case of SWE, these variables are the space step size $\Delta x$, and the time step size $\Delta t$. It is to be noted that the space step size is linked to the size of the Hamiltonian and time step size is linked to time interval of the incoming measurements. The disadvantage of their procedure is that if the space step size decreases then the number of neurons will increase consequently the size of the Hamiltonian will increase

causing a significant numerical difficulty in implementing the filter. This may also lead to potentially instability of the filter. The procedure outlined in this thesis has overcome these difficulties (see Ahamed and Kambhampati 2008).

## 4.5 Implementation of the filter

Since $|\psi|^2$ is interpreted as probability density function, this *pdf* is the estimate of the *pdf* of the input signal $y(t)$ as defined by

$$y(t) = y_a(t) + \mu(t) \tag{4.24}$$

where $y_a(t)$ is the actual signal, and $\mu(t)$ is the noise. This stochastic signal excites an array of $N$ neurons spatially located along the $x-axis$ after being pre-processed by the synapses. Let the signal $y(t)$ be represented by the Gaussian probability density function $f(x,t)$ with mean $\lambda_y$ and variance $\sigma^2$ and let the initial condition for solving the equation (4.23) be represented by the Gaussian probability density function $\hat{f}(x,t)$ with mean $\hat{\lambda}_y$ and variance $\hat{\sigma}$. As the dynamics of the wave evolves, the *pdf* $\hat{f}(x,t)$ moves toward the *pdf* $f(x,t)$ of the signal $y(t)$ through the updated synaptic weights $w(x,t)$ and hence the transformation of the input signal to the wave packet takes place. The time varying synaptic weights $w(x,t)$ are updated using Hebbian learning rule:

$$\frac{\partial}{\partial t} w(x,t) = \eta \varepsilon(t) |\psi(x,t)|^2 \tag{4.25}$$

where $\varepsilon(t) = y(t) - \hat{y}(t)$, with $\hat{y}(t)$ being the filter estimate of the actual signal $y_a(t)$. This estimate is defined by

$$\hat{y}(t) = \int_{-L/2}^{+L/2} x f(x,t) dx \,. \tag{4.26}$$

Given the wave packets in discretised form the above expectation is computed using the Riemann integral (Capiński and Kopp 2004). From the filter formulation it can be observed that there are a number of important parameters, which need to be tuned. These are $\Delta x$, $\Delta t$, $\hbar$, $m$, $\eta$, $\xi$. Of these $\Delta x$ and $\Delta t$ are from the discretization process used for the numerical procedure to solve the SWE. The others are for the filter itself, and the learning process. Of these there are two important parameters which play an important role in the SWE namely the universal constant, $\hbar$ and the mass of the quantum object, $m$. However, considering the constant $\hbar$ as unit value, the mass of the quantum object is required to be tuned. It is possible to use a variety of algorithms to tune the values of $m$, $\xi$, and $\eta$ (Behera *et al.* 2005b). The advantage of using the proposed scheme is that the solution becomes independent of the mass, *m*, of the quantum object (see equation (4.21)). This in turn reduces the number of independent design parameters to a choice of $\xi$ and $\eta$, since the others are either set to unity or the scheme is independent of their values. Another advantage of the scheme is that the potential function is always bounded and hence the normalization property is maintained throughout the process for all time. However, all of these are dependent on the ability to normalize the initial wave function.

## 4.6 Normalization of the initial wave function

For the evolution of the wave packet and at the same time keeping the solution stable it is required to normalize the initial wave packet. However, to be a valid wave function it must have the following properties

- The wave function must be a solution of the Schrödinger wave equation
- It is continuous and differentiable everywhere

- It is a single valued function i.e., the function is either one-to-one or many-to-one
- $\psi \to 0$ as $x \to \pm\infty$.

Any equation of the form $e^{i\theta}$ would satisfy the SWE so long as the potential function remains bounded over time (Schiff 1968). The general form of the plane wave is given by:

$$\psi(\vec{r},t) = \int_{-\infty}^{+\infty} a(\vec{\kappa}) \exp[i(\vec{\kappa}.\vec{r} - \omega t)] d^3\kappa \tag{4.27}$$

where $\omega$ is a function of the propagation number $\vec{\kappa}$. This equation has all the properties mentioned above. The dispersion relation $\omega(\vec{\kappa})$ determines all physical properties such as phase and group velocities (that is, velocity of the centre of the wave packet) of the wave. Considering the wave propagating along $x-axis$, the normalized single dimensional wave function with mean zero and standard deviation of $\sigma_x$ is given by (see Appendix-A)

$$\psi(x,0) = \frac{1}{\sqrt{\sigma_x \sqrt{\pi}}} e^{ik\cdot x} \times e^{-\left(\frac{x^2}{2\sigma_x^2}\right)} \tag{4.28}$$

Considering equation (4.28) as an initial wave packet (which is known as the Gaussian wave packet), question may arise whether, under QRNN, the solution of SWE can be explained as probability amplitude. Interestingly, the answer is yes. The reason is that the potential field is defined to accommodate the neural lattice (see equations (4.6) and (4.7)). This potential field produces a numerical value (see equation (4.11)) and is transformed into the Hamiltonian (e.g., equation 4.21) which acts upon the wave function for the evolution of the wave packet in space and time. In QRNN filter, modulus-squared of this evolutionary wave is evaluated as probability density function.

**Table-4.1: Algorithmic structure of the QRNN filter**

**Step-1:**

*Initialization:*

*Hamiltonian Matrix*

$$\mathcal{H} = -\frac{\hbar}{2m}\frac{\partial^2}{\partial x^2} + \frac{1}{\hbar}V(x)$$

*and setting up the initial value for the wave function*

**Step-2:**

*Computation:*

*Wave function* $\psi$ *using Crank-Nicholson scheme*

$$\psi_j^{n+1} = \left(1 + \frac{1}{2}i\mathcal{H}\Delta t\right)^{-1}\left(1 - \frac{1}{2}i\mathcal{H}\Delta t\right)\psi_j^n$$

*Wave packet as pdf*

$$pdf = |\psi|^2$$

*Updating the weights using Hebbian learning rule*

$$\frac{\partial}{\partial t}w(x,t) = \eta\varepsilon(t)|\psi(x,t)|^2$$

## 4.7 Algorithmic structure for the QRNN filter

The quantum recurrent neural network is developed exploiting the dynamics of the Schrödinger wave equation of the form

$$i\hbar\frac{\partial\psi(x,t)}{\partial t} = \left[-\frac{\hbar^2}{2m}\nabla^2 + \xi\left(U(x,t) + G\left(|\psi|^2\right)\right)\right]\psi(x,t)$$

and the filtering problem is solved by identifying the time varying *pdf* of the observed stochastic variable $y(t)$ which is transformed into $\psi(\cdot)$, the wave function of the Schrödinger wave equation, in an unsupervised manner *without a prior information about the noise*. The algorithmic structure of the filter is shown in Table-4.1.

## 4.8 Conclusion

In this chapter the formulation of the quantum recurrent neural network has been discussed along with its mathematical construction. Schrödinger wave equation has a complex structure. Therefore, in developing the numerical scheme attention is given to maintain stability of the solution so that the iterative scheme does not propagate any rounding error that can dominate the solution space producing undesirable results. The differencing scheme stated in equation (4.12) is called implicit rather than explicit (Press *et al.* 1992; Garcia 1994). The meaning is that the term $\psi^{n+1}$ is not given directly in terms of $\psi^{n}$ but some algebraic techniques are required in order to obtain the wave function at the time step $n+1$ from that at $n$. However, this equation is avoided in computing the wave function on the subsequent state on the ground that it is not unitary. Unitary is the characteristic of the SWE which ensures that the normalization of the wave function does not change over time. For this reason a relatively simple unitary approximation is developed which is provided by the Cayley's form described in equation (4.23). The solution of the SWE is carried out using this equation with initial conditions and initial wave packet. The Gaussian wave packet is selected as an initial wave packet and is normalized. However, initial conditions together with the boundary conditions impose limitations on the choice of values of the parameters that arise due to the differencing scheme and propagation of the wave in the well. This will be discussed in the next chapter.

# Chapter 5

# Design Issues of the QRNN Filter

## 5.1 Introduction

An algorithm for the implementation of the QRNN filter has been outlined in Chapter-4 in which the solution of the Schrödinger wave equation (SWE) plays the important part. This equation was discretised for the numerical scheme to be implemented. However, this discretisation introduces limitations on the choice of values for the parameters of the SWE and there remains certain design issues which need to be emphasized. These deal with:

- Initialization of the wave packet,
- Selection of values for the parameters in order to solve the SWE and to satisfy the constraints imposed in the numerical strategy, and
- The size of the potential well in order to localize and propagate the wave packet.

These issues dictate the ability of the filter to extract the signal and to be able to understand the underlying properties of the wave propagation. As will be shown, there are specific situations, depending on the length of the well, the discretisation (number of neurons) and the window size, when a so called *calm wave* (see Section-5.4.2) can be defined. Presence of this *calm wave* in the well indicates that the filter has enough knowledge about the signal and hence provides better results. Essentially, the presence

of the *calm wave* indicates that the ends of the wave packet are fixed to the extremity of the well (see Section-5.4).

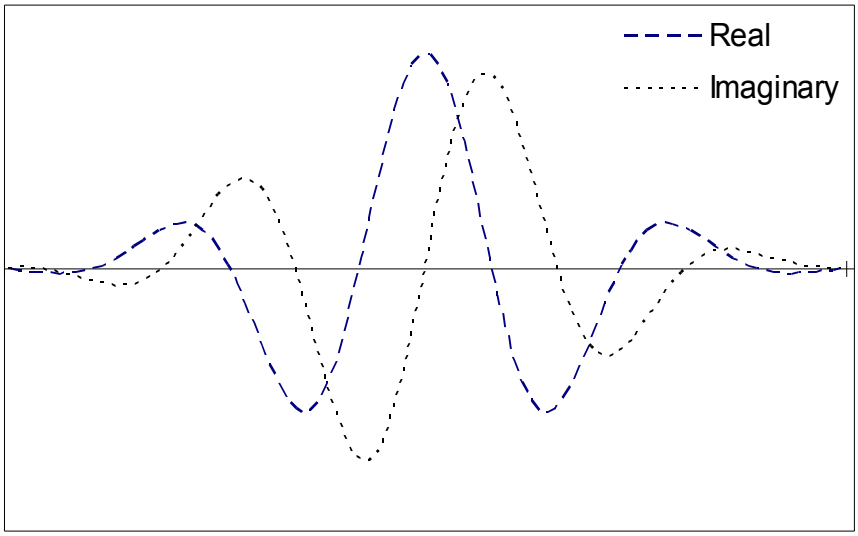



Figure-5.1: The initial wave packet

## 5.2 Initialization of the filter

Initial conditions are important for the solution of the differential equations. However, in the case of the partial differential equation, this becomes ever more critical as these would define which variables are dynamic (in this case, time-dependent). In QRNN filter, the initial wave function at time $t = 0$ is given by (see Section-4.6)

$$\psi(x,0) = \frac{1}{\sqrt{\sigma\sqrt{\pi}}} e^{i\kappa \cdot x} \cdot e^{-\left(\frac{(x-x_0)^2}{2\sigma_0^2}\right)} \tag{5.1}$$

There are two features with this initialization. The first is to govern the speed, this is done by the term $e^{ikx}$, where $k$ is the average momentum of the initial wave. The second deals with the size of the spread. The wave is centered at $x = x_0$ with a spread in $x$ governed by $\sigma_0$. The initial wave is normalized in order to satisfy the constraints given by equation 4.19 (see also Section-4.6).

Initialization of the wave in this way ensures that the wave function is finite everywhere and that the probabilities are bounded and confined to the closed interval *[0, 1]* and is single valued. As a result, there will not be any multiple values of the probability. The wave function, $\psi \to 0$ as $x \to \infty$. In QRNN filter, the wave function $\psi(\cdot)$ is required to be zero at the boundary of the well and this will form the boundary condition for the wave to propagate. With these boundary conditions for the solution of the SWE, Figure-5.1 shows the graph for the initial wave packet.

## 5.3 Parameters in the QRNN filter

There are various parameters which are required to be tuned for the implementation of the QRNN filter. This is essential since the wave packet must contain and propagate in the well. These parameters will ultimately dictate the wave to produce desired results and they are categorised as follows

a) ***SWE parameters***: The parameters of the SWE itself, i.e., mass of the quantum particle $m$ and the universal constant $\hbar$,

b) ***Discretisation parameters***: Parameters due to the discretisation of the SWE, i.e., length of the well $L$, number of mash points $N$, length of the mesh $\Delta x$, and time interval in the grid $\Delta t$, and

c) ***Neural parameters***: Parameters due to neural network, i.e., number of neurons, connection weights $w(x,t)$ in the network, potential field excitation term $\xi$, and learning rate $\eta$.

Selection of values for these parameters and their internal relations are discussed below. Values for the parameters of the second and third categories (discretisation parameters and neural parameters) will be discussed under the localization of the wave packet.

### 5.3.1 Values for the SWE parameters

For simplicity and numerical efficiency, values for the SWE parameters are set as follows. The universal constant $\hbar$ is set to unity and the value of $m$ is set to $\frac{1}{2}$ so as to get the coefficient of the first term in the Hamiltonian (equation 4.21) as unity. The physical meaning of $m$ is the mass of the signal which does not exists in a practical sense, thus it can take other values as well. However, a very large value of *m* would cause the Hamiltonian matrix to become very small. This in turn means, if the value of *m* approaches infinity the Hamiltonian matrix would be trivial resulting in the incorrect solutions of the SWE. A very small value of $m$ will cause the Hamiltonian matrix to explode causing the potential field (equation 4.11) to have no effect on the solution of the SWE and thus the evolution of the *pdf*. The value of $m$ cannot be negative or zero as it results in a Hamiltonian matrix which is undefined.

### 5.3.2 Localization of the wave packet

At first it is required to decide the length of the well where the wave packet is generated. If the length of the well is *L*, then the number of grid points $N$ and length of the grid $\Delta x$ is related as follows:

$$\Delta x = \frac{L}{N} \tag{5.2}$$

This relation gives the number of neurons in the network, which is represented by $N$. It is important to note from equation (5.2) that parameters in the second set are related to the parameters in the third set especially the number of neurons which is critical in designing the neural network. The length of the well is chosen in a manner such that it is possible to generate a smooth wave packet inside the well which can move forward or backward meeting boundary conditions yet remaining completely within the walls.

The wave packet is generated in a well of length $L$ and the wave packet must vanish at the edge of this well. As a result, values of $x_0$ and $\sigma_0$ (equation 5.1) must be chosen so that they meet the boundary conditions of $\psi(\cdot)$ (i.e., $\psi(0,0)$ and $\psi(L,0)$ are to be essentially zero) and at the same time the wave remains within the well. Let, $x_0 = \frac{L}{2}$, then the initial wave packet (5.1) becomes

$$\psi(x,0) = \frac{1}{\sqrt{\sigma\sqrt{\pi}}} e^{i\kappa \cdot x} \cdot e^{-\left(x-\frac{L}{2}\right)^2 / 2\sigma_0^2} \tag{5.3}$$

Now let, $x = 0$ and $\sigma_0 = \frac{L}{20}$ (that is, let 5% of the total length of the well) then the initial wave packet (5.3) becomes

$$\left|\psi(0,0)\right| = \frac{1}{\sqrt{L/20\sqrt{\pi}}} e^{-25}, \tag{5.3a}$$

which is close zero and thus clearly vanishes at the edge of the well. It can also be shown that when $x = L$ and $\sigma_0 = \frac{L}{20}$ then the initial wave packet (5.3) becomes

$$\left|\psi(L,0)\right| = \frac{1}{\sqrt{L/20\sqrt{\pi}}} \times 1 \times e^{-25}, \tag{5.3b}$$

which is close zero and thus clearly vanishes at the other edge of the well. However, there are two more restriction that must be imposed on the wave. The first one is how far the wave would travel and the second one is what would be the spread of the wave over the course of time as it is time dependent.

The wave packet must not travel so far that it hits the wall of the well. This can be ensured by letting the centre of the packet, which starts at $x = \frac{L}{2}$, move no further to the right than $x = L$. This is, in turn, accomplished by the requirement that the average velocity of the wave packet be

$$\kappa \approx \frac{mL}{2T}$$

where $T$ is the total time required for the wave packet to move from $x = \frac{L}{2}$ to $x = L$, which gives

$$\kappa \approx \frac{mN \times \Delta x}{2T} \tag{5.4}$$

This relates the value of the momentum in terms of the length and number of grid points in the well. The second one is concerned with the fact that the wave packet spreads in the course of time. Therefore, the spread of the wave must be arranged in a way that the reflected and transmitted wave packets continue to be completely within the well. The spread of the wave packet at time $t$ is given by (Goldberg 1967; Schiff 1968)

$$\sigma(t) = \sigma_0 \sqrt{1 + \frac{\hbar^2 t^2}{4m^2 \sigma_0^4}} \tag{5.5}$$

where $\sigma_0$ is the initial spread (can be selected as described earlier, see equation 5.3a) of the wave packet. Since the second term under squared root is reasonably small therefore the spread at certain moment will not be significantly greater than the initial value. Equation (5.4) and (5.5) provide guidelines for the choice of values for the parameters. However, for the implementation of QRNN filter, the value for the momentum is selected to be one as this choice would provide a constant speed of the wave. The spread of the wave is evaluated from the error (difference between the input signal and the estimate of the past measurement) which varies from time to time. Thus, the relation in (5.5) provides an approximation rule for the spread of the wave. The time varying spread of the wave depends on the spread of the error. These two parameters (i.e., $\kappa$, $\sigma_0$) play the critical part in the design of the filter.

Other parameters required to be tuned are the weights for the neural network which is selected as a vector values of a set of random variables within a certain range such as Gaussian random variables between [-1 1], the learning rate which usually within the interval of [0 1] and the values for the potential field excitation is set to one.

It can be realized from the above discussion that the conditions that arise due to localisation of the wave impinge on the magnitude of the potential and also to the size of the well. Indeed the bounds on the size of the potential well are also taken into account by ensuring that the spread of the signals are such that they are well within the well size, i.e., $|L| > \sigma^2$; where $L$ is the length of the well, in some sense represents the upper limit of the incoming signal. This is similar to the situation in neural networks, where the normalization is done to ensure that trained network is able to generalize.

## 5.4 Propagation of wave in the well

In quantum processes, the propagation of the wave packet in the well is important. In QRNN filter it takes on added significance since the wave packet dictates the evolution of the *pdf*. If the wave packet is completely reflected (i.e., bounces) of the wall of the well (see Figure-5.2) it implies either that due to lack of sufficient information to the initialization has resulted in the wave packet having an inappropriate movement with a spread, or that the network has not learnt the signal well. As more information comes in, this is resolved and it can be seen that the wave settles down in that the ends are fixed to the walls (see Figure-5.3). Such a wave is termed as a *calm wave* in this thesis. Indeed if there was sufficient a priori knowledge the filtering process could have initialized into a *calm wave*.

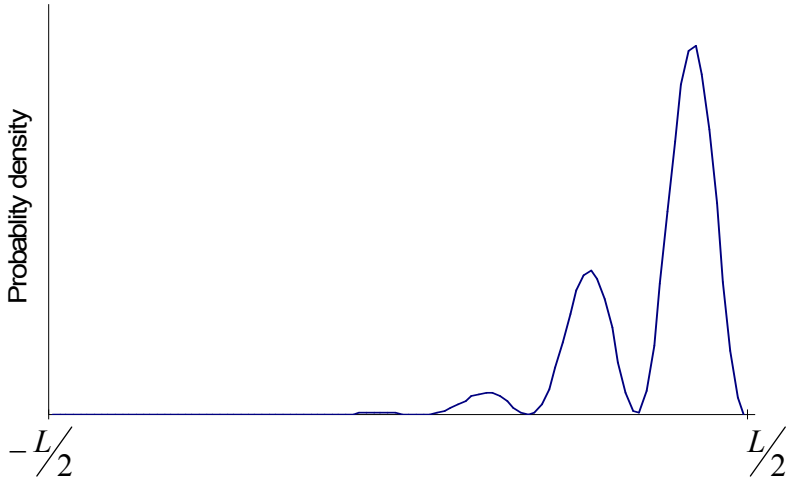



Figure-5.2: A snapshoot of the transmitted (or reflected) wave packet

The wave packet can also be propagated in the well without this *calm wave*. This wave will be referred as the *normal wave* in this thesis. In both cases, the wave can be treated in two different ways. One by allowing the wave to spread in the well with immediate past estimated value as a mean and absolute value of the error as a current spread, and the other by switching the spread of the wave off but maintaining the initial spread as a spread and the immediate past estimate as a mean for the next wave packet. Logical issues behind these are that once the localization of the wave (see Section-5.3.2) is done then it is possible to control the wave packet by selecting the various parameters within the constraints to propagate the wave packet in the localized well to achieve the expected or desired results. Details of propagating these waves are discussed in the next sections.

### 5.4.1 Normal wave

The initial wave is propagated in the well by considering the initial estimate and initial spread as a centre and spread of the wave packet respectively. Once the initial wave packet is launched then an estimate is made for the input signal and this estimate is used as a centre of the wave for the next input signal. The error is calculated by subtracting the current estimate from the new measurement and absolute value of the error is taken

as a spread of the wave to measure the next input signal. Evolution of the *pdf* in this way has similarity, in some sense, with the Kalman gain matrix (see Table-3.1). In the Kalman filter, the optimal estimate at certain time $t_2$ is equal to the best prediction of its value at time $t_2^-$ (that is, just before the measurement is actually taken) plus a correction term of an optimal weighting value times the difference between the actual measurement taken at $t_2$ and the best prediction.

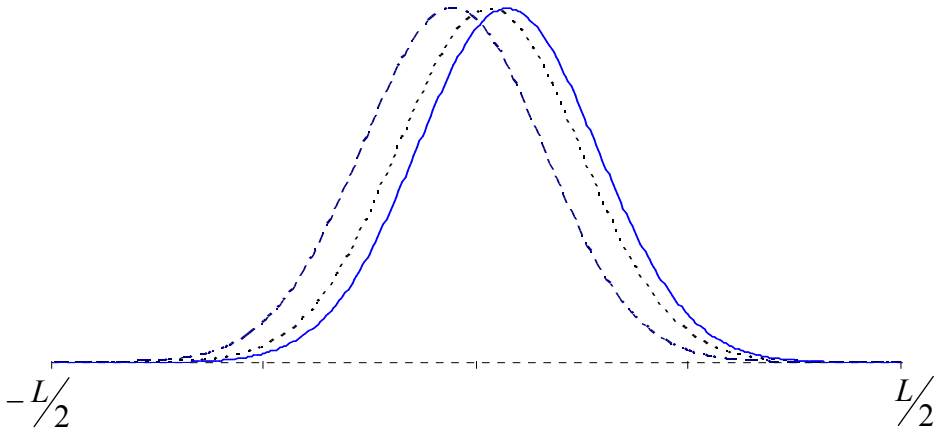



Figure-5.3: Snapshots of the normal wave

In the QRNN filter, the estimation emerges almost in the same way where the initial estimate is set to zero and then with the first input measurement the absolute value of the difference between the estimate and the input measurement is taken to update the initial estimate by combining the *pdf* information from the initial wave packet. This process continues until all measurements are exposed to the network. The pseudo-code for this wave is summarised in Table-5.1.

A complication that may arise in propagating this wave is that the coefficient of the initial wave packet (see Equation 5.1). Since the time varying error is taken as spread ($\sigma$) of the wave packet, therefore, the term $\frac{1}{\sqrt{\sigma\sqrt{\pi}}}$ may explode when $\sigma$ is very

small or approaches zero. This problem can be overcome by resetting or switching off the term for the subsequent development (see Section-6.4.1).

**Table-5.1: Pseudo-code for the normal wave**

```
Initialize all parameters
Launch initial wave packet in the well
For i = 1:T
    Store previous estimate
    Estimate the current state using the wave packet
    Input new measurement
    Evaluate the error
    Update the network & Hamiltonian matrix
    Solve SWE with new Hamiltonian and generate a wave packet
End
```

### 5.4.2 The Calm wave

Another way of propagating the wave in the well is introducing the *calm wave*. The concept of calm wave is that at the outset a wave packet is created with an initial value of zero and the wave remains *calm* and so the name. Number of wave particles in the *calm wave* is the same as the number of neurons in the network. As the wave propagates in the well the *calm wave* gets perturbed. The perturbation here refers to the situation that when each measurement is given as input to the network a wave packet is generated in the well which in turn gives the distribution of the input signal. The signal receives a weighted estimate and stores this value in the *calm wave* in an iterative manner. Thus the *calm wave* starts to have perturbation. The weighted estimate is determined using the equation (4.26) and the error is evaluated which constitutes the potential function. This error is then fed back to the network to update the network parameters thus it works like an ensemble of information in the evolution of the probability density

function. It is to be noted that the *calm wave* is not a wave but it is a moving window of measurement information. The *calm wave* is perturbed from the centre of the well and the input signal is spread through both sides of the well symmetrically as shown in Figure-5.4(a–c).

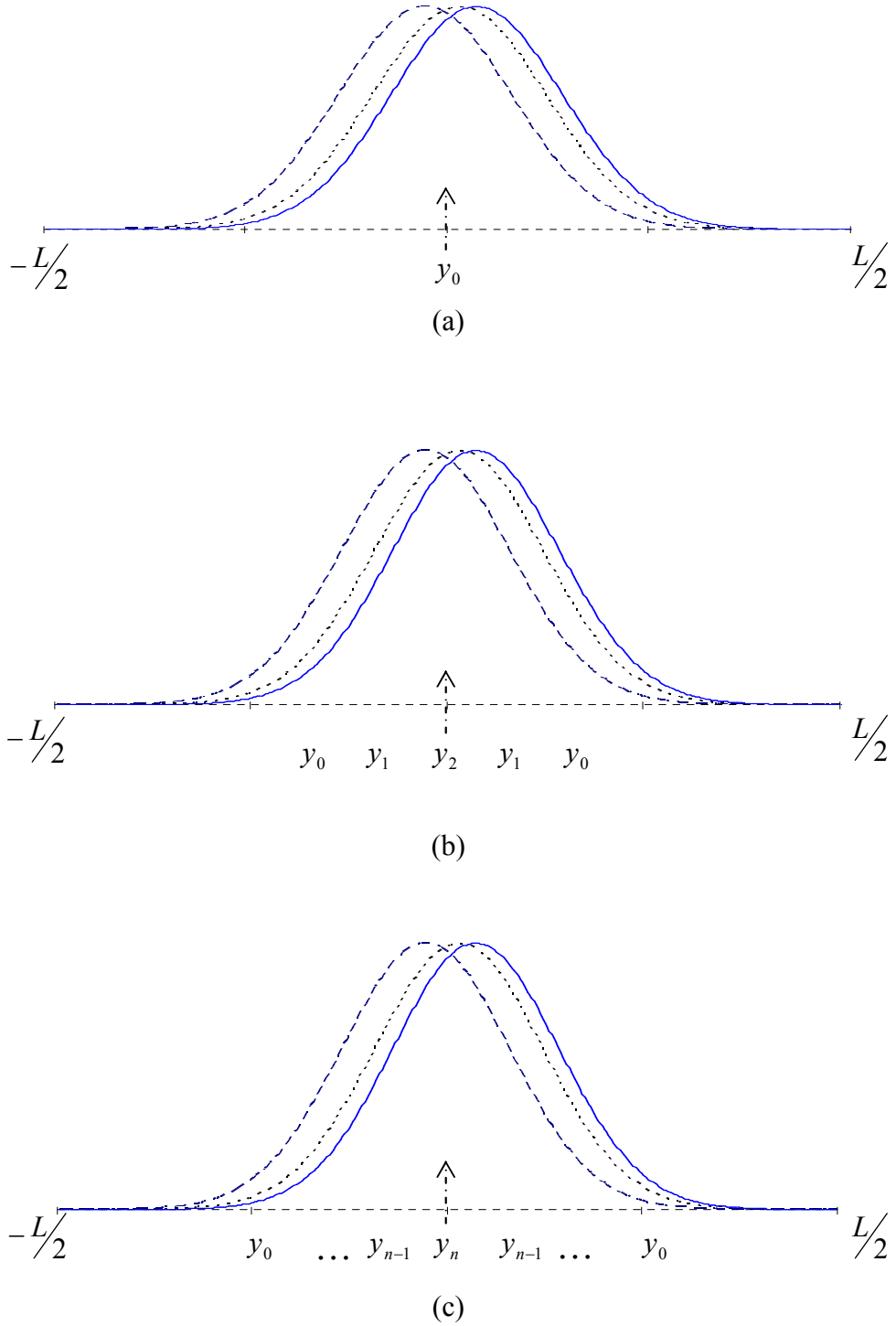



Figure-5.4(a–c): Signal input to the calm wave

The explanation of why it works is the recurrent structure of the network where the ensemble of wave packet enters to the network as a potential function along with the new measurement signal. This perturbed wave is then tilted towards the pressure exerted by the potential function and moves forward and backward with constant speed (see Figure-5.3) according to the new measurement signal. In this way the networks not only maintain the previous history but also pass the information for the evolution of the *pdf* in the statistical future. It is mentioned in Chapter-2 (Section-2.4) that stochastic signal can be treated with a process called Martingale Process which is linked to some specifications known as filtration. Although the exact structure of the filtration seems to be complicated to fit into the QRNN filter, the *calm wave* resembles the necessary structure of the sigma-algebra of Martingale Process in the sense that it acts as a moving window containing and carrying increasing information of the past measurements. The pseudo-code for this wave is summarised in Table-5.2.

**Table-5.2: Pseudo-code for the calm wave**

```
Initialize all parameters
Launch calm wave
Launch initial wave packet in the well
For i = 1:T
    Store previous estimate
    Estimate the current state using the wave packet
    For k = 1:N
        Perturb the calm wave
    End
    Input new measurement
    Evaluate the error
    Update the network & Hamiltonian matrix
    Solve SWE with new Hamiltonian and generate a wave packet
End
```

A complication that may arise here is the allocation of huge amount of memory for the *calm wave*. It can be seen (see Figure-6.9), nonetheless, the propagated wave along with the input signal is able to generate the *pdf* and produces expected results.

## 5.5 Conclusion

Initialization of the various parameters of the SWE and how to tune those values to achieve desired results have been discussed in this chapter. Tuning of the parameters is important in that they dictate the propagation of the wave packet in the well. It has been emerged that the wave packets can be generated in two different ways in the well in order to get a *normal wave* or a *calm wave*. For the *normal wave*, a single value of the signal is measured while in the *calm wave* more than one value of the signal is used. These wave packets play the core part in the evolution of the *pdf*. In the next chapter, values of these parameters will be selected according to the specification developed in this chapter and two types of wave packets will be propagated in the well for the implementation of the QRNN filter.

# Chapter 6

# Evaluating Performance by Varying Design Parameters

## 6.1 Introduction

In Chapters-4 & 5 a filter based on the quantum recurrent neural networks was developed. In Chapter-4, it was shown that the *pdf* evolves from an initial wave packet and is able to learn the dynamics of the signal in an unsupervised manner. Using this initial value of the wave the filter predicts and estimates the current value of the state which is then compared with the measurements to estimate the error. This error is then used to drive the Schrödinger wave equation (SWE) in order to evolve the *pdf* for the next time step. Essentially, the error becomes the potential function for the SWE which acts upon the wave function to evolve the wave in space and time. The estimate of the state is obtained from the wave packet as described in equation (4.26). The error is then used to update the weights of the network. In order to contain the wave in the well, a number of restrictions have been imposed on the choice of values for the parameters of the SWE. These, along with the way in which the wave packets are propagated in the well, have been discussed in Chapter-5. In this chapter, the QRNN filter is tested on a number of test signals. In the next section normalization of measurement signals, methods of assessing performance of the filter, and the framework for discussion of simulation results are outlined.

## 6.2 Normalization and performance criterion

In most data driven techniques, the data is often normalized. This normalization is done in order to ensure that (a) there is uniformity in the data space, (b) data falls within the range of the values permissible for the model. In this application this normalization is even more critical, because there are restrictions on the nature of the wave packet (see Chapter-4, Section-4.6, and Chapter-5, Section-5.3) and also to ensure that the waves are well within the potential (in other words, the range of data must be within the potential well). The following formula has been used in normalizing the signal:

$$\textit{Normalized signal} = \textit{[signal} - \textit{average(signal)]/[max(signal)} - \textit{min(signal)]} \tag{6.1}$$

The performance of the filter is judged based on the value of the Root-Mean-Square-Error (RMSE) that measures average magnitude of the error. This is done using the following formula

$$RMSE = \sqrt{\frac{1}{T}\sum_{i=1}^{T}\left(y_a(i) - \hat{y}(i)\right)^2} \tag{6.2}$$

where $y_a(\cdot)$ represents the actual signal and $\hat{y}(\cdot)$ is the estimated output, $T$ denotes the total number of measurements processed. The errors (difference between the input signal and estimated output) are squared before they are averaged so that the RMSE gives a relatively higher weight to large errors (see equation (6.2)). A lower value of the RMSE is used (in the sense of optimality) as indicator for the better performance of the filter.

## 6.3 Framework for discussion of the results

Given the localization and the propagation of the wave in the well the results of the simulations are discussed within the following framework:

- How the wave packets generated in the well are able to capture the dynamics of the signal in terms of the evolution of the *pdf*?
- How good the performance of the filter as is judged in terms of RMSE?
- Is the QRNN filter stable and converge?

All of these will be discussed in the simulation and results section along with the analysis of the tests results.

## 6.4 Simulation and results

The QRNN filter is tested using the signals sampled with sinusoidal, shifted sinusoidal, amplitude modulated, and mixed sinusoidal signals. They are defined as follows:

$$y(t) = 2\sin(2\pi t) + \mu(t) \tag{6.3}$$

$$y(t) = 2 + 2\sin(2\pi t) + \mu(t) \tag{6.4}$$

$$y(t) = f(t)\sin(2\pi t) + \mu(t) \tag{6.5}$$

where $f(t)$ is the time varying amplitude and is defined by

$$f(t) = \begin{cases} 1.5t & 0 \le t \le 5 \\ 1.5(10-t) & 5 < t \le 10 \end{cases}$$

$$y(t) = 2\sin(20\pi t) + 2\sin(10\pi t) + \mu(t) \tag{6.6}$$

where $\mu(t)$ is random noise with zero mean. These signals are representative to many forms of physical process such as tracking eye movements. Signals are sampled at the rate of 10 samples per cycle, 100 samples per cycle, and 1000 samples per cycle. The reason for these different rates is to conform with the integration of the SWE, where length of the time interval is required. Noise strengths are measured in terms of the Signal to Noise Ratio (SNR) in decibel (dB) and taken to be 6dB, 10dB, 20dB, and 30dB. Altogether a total of 48 tests are performed. A typical graph of the first 500

samples (sinusoidal signal (6.3)) with 100 samples per cycle and noise strength of 10dB is shown in Figure-6.1. To verify the performance of the QRNN filter, two different experiments are performed on these 48 sets of measurements. In these experiments, it is assumed that the filter has no *a priori* knowledge of the noise. This is one of the *key features* of the QRNN filter mentioned in Chapter-1 (Section-1.3).

The first experiment is performed considering values for the parameters that fall within the constraints discussed in Chapter-5. In this thesis, these values are considered to be the standard values for the development of the QRNN filter. The second experiment is performed changing values for the length of the well (and hence the number of neurons in the network, spread of the wave proportionately) keeping values for other parameters (such as mass $m$, the universal constant $\hbar$, wave momentum $\kappa$, neural parameters $\xi$, and $\eta$) constant. In either of these experiments, waves are propagated in the well using both normal and calm waves. Details of all theses experiments are as follows.

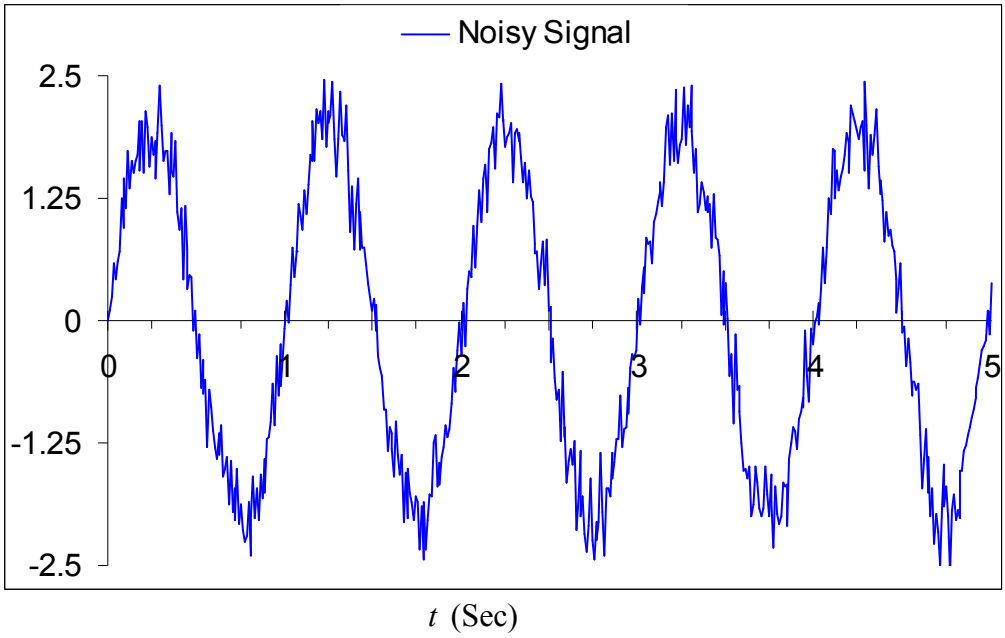



Figure-6.1: Noisy sinusoidal signals (un-normalized) of SNR 10dB

### 6.4.1 Experiment-1:

#### I. Normal wave

This experiment is performed using the normal wave and values for the three different parameter sets are selected as follows:

a) ***SWE parameters***: The universal constant $\hbar = 1$ and mass of the quantum object $m = \frac{1}{2}$,

b) ***Discretisation parameters***: The length of the well is taken as $L = 2$. The reason of taking this value is that the measurement is normalized (equation (6.1)) within the interval [-0.5 0.5] and it is necessary to have a smooth passage for the wave to move forward and backward to reach the peak of the signal within the well. The initial spread of the wave (standard deviation) is selected to be $\sigma_0 = 0.6$ which will keep the spread and amplitude of the wave well enough inside the well, $\Delta x = 0.01$, and $\Delta t = 0.1$ (for the first set of measurements), $\Delta t = 0.01$ (for the second set of measurements) and $\Delta t = 0.001$ (for the third set of measurements), and

c) ***Neural parameters***: Number of neurons (which also represents the grid points for the implicit scheme) $N = \frac{L}{\Delta x} = 200$, the weight $w(k,t)$ vector is filled in with random numbers drawn from the Gaussian distribution with zero mean and standard deviation of one, the learning parameter $\eta = 0.5$, and potential function excitation $\xi = 1$.

With these values of the parameters, the single layer Quantum Recurrent Neural Network is designed (Figure-4.2, Chapter-4) and the network is trained using the Hebbian learning rule (see equation (4.25)). Table below (Table-6.1) shows the

performance of the QRNN filter for all four different signals with different noise strength for the sampling frequency of 100 samples per cycle. Results for other sampling frequencies are shown in Appendix-B (in Table-6.1a: 10 samples per cycle, and in Table-6.1b: 1000 samples per cycle). The results presented in Table-6.1 are also been shown in Figure-6.2.

| Signals/Strengths | Root-mean-square-error (RMSE) | | | |
|---|---|---|---|---|
| | 6 dB | 10 dB | 20dB | 30dB |
| Sinusoidal | 0.13746 | 0.10374 | 0.05545 | 0.04012 |
| Shifted sinusoidal | 0.13753 | 0.10470 | 0.05739 | 0.04146 |
| Amplitude modulated | 0.05884 | 0.04343 | 0.02488 | 0.02001 |
| Mixed sinusoidal | 0.11653 | 0.08820 | 0.04617 | 0.03162 |

Table-6.1: RMSE of various signals and noise strengths (Normal wave)
(Sampling rate (SR): 100 samples per cycle)
$L = 2,\ N = 200,\ \Delta x = 0.01,\ \sigma = 0.6,\ \xi = 1,\ \eta = 0.5,$
$m = 0.5,\ \hbar = 1,\ \kappa = 1,\ \Delta t = 0.01.$

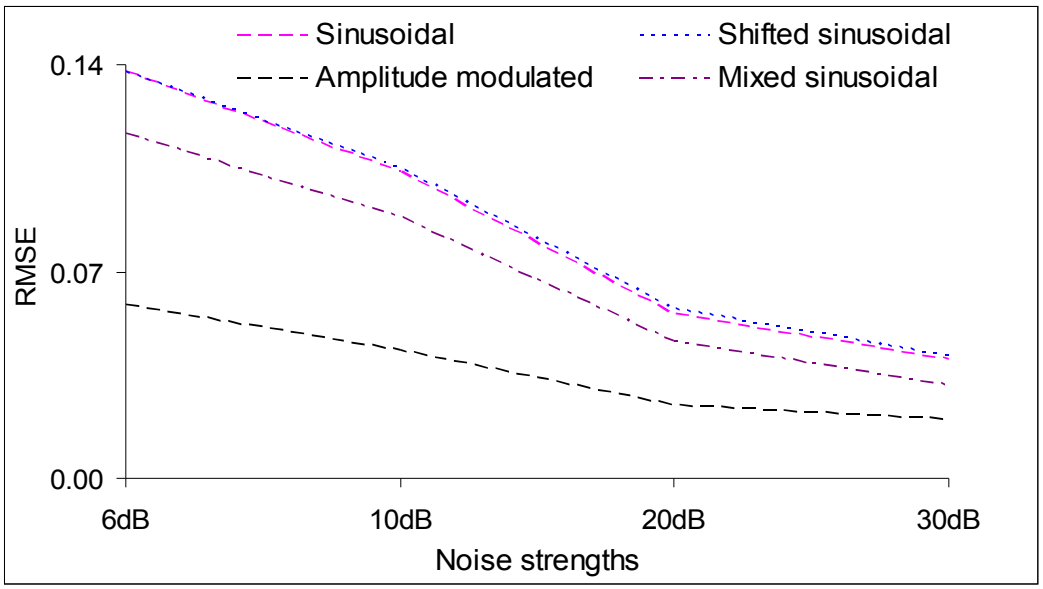



Figure-6.2: Noise strengths vs. RMSE (Normal wave)

From the above table (and Table-6.1a, Table-6.1b in the Appendix-B) it can be seen that as the signal-to-noise ratio (SNR) increases the performance of the filter improves. This is obvious because higher SNR means lower magnitude of noise and thus smaller RMSE. It can also be observed that the RMSE varies from signal to signal even for the same noise strengths. This is because of changes in the pattern (or orientation) of the signals. For example, in the amplitude modulated signal (signal equation (6.5)) the RMSE is considerably low (see Figure-6.2). The reason for this is that of the magnitude of errors at the tail ends are small (see Figure-6.3) and so the RMSE.

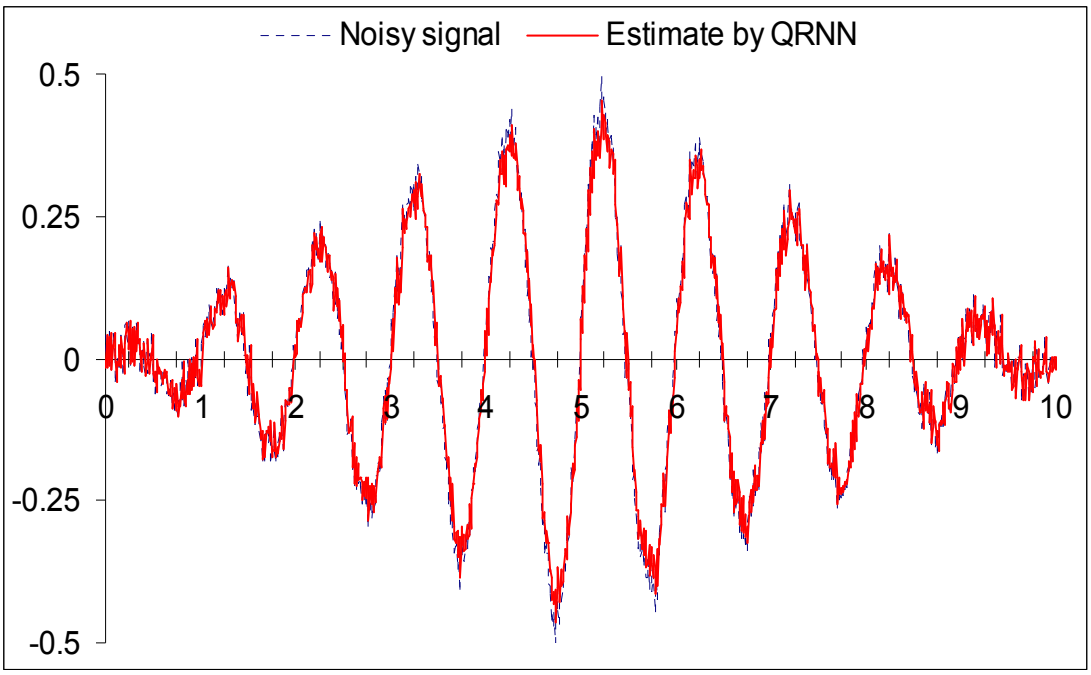



Figure-6.3: Noisy and estimated amplitude modulated signals (SNR: 10dB)
L = 2, N = 200, $\Delta x = 0.01$, $\sigma = 0.6$, $\xi = 1$, $\eta = 0.5$,
m = 0.5, $\hbar = 1$, $\kappa = 1$, $\Delta t = 0.01$.

To take a closer view of the table (Table-6.1) a specific case is selected for further analysis. This is the sinusoidal signal (signal equation (6.3)) with SNR of 10dB. The noisy signal along with its estimated signal is shown in Figure-6.4. The filtering error is shown in Figure-6.6. It can be seen from Figure-6.4 that the QRNN filter is able to learn the distribution of the signal; however, it has not been able to filter out the noise. By

magnifying for the first second, the results are presented in Figure-6.5. It was mentioned earlier that the reason for not being able to remove noise from the signal is due to the first term in the initial wave packet (see Section-5.4.1). This term is $\frac{1}{\sqrt{\sigma\sqrt{\pi}}}$.

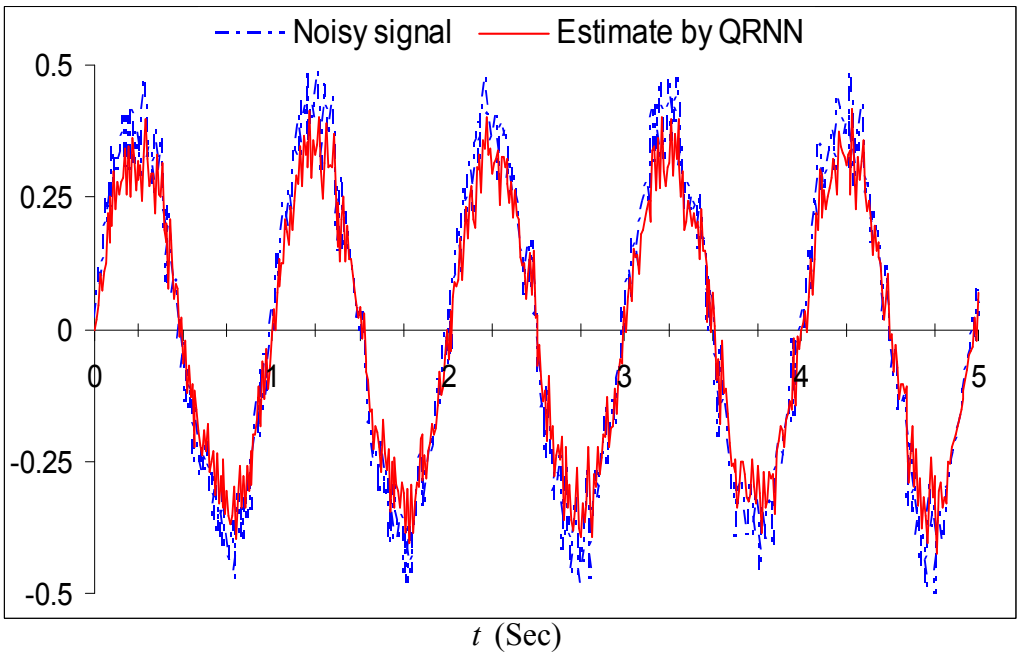



Figure-6.4: Noisy and estimated sinusoidal signals (SNR: 10dB)
L = 2, N = 200, $\Delta x = 0.01$, $\xi = 1$, $\eta = 0.5$,
m = 0.5, $\hbar = 1$, $\kappa = 1$, $\Delta t = 0.01$.

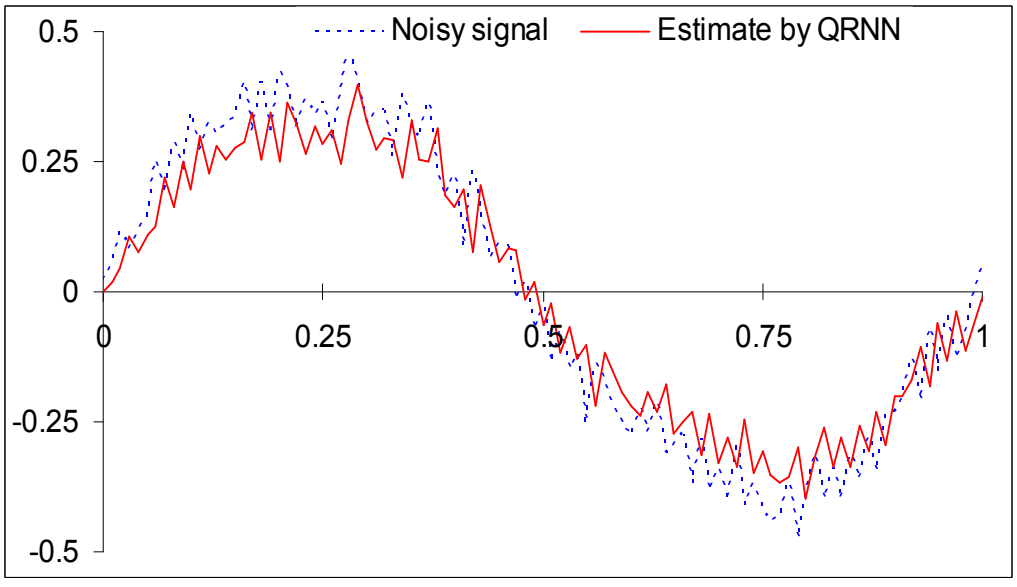
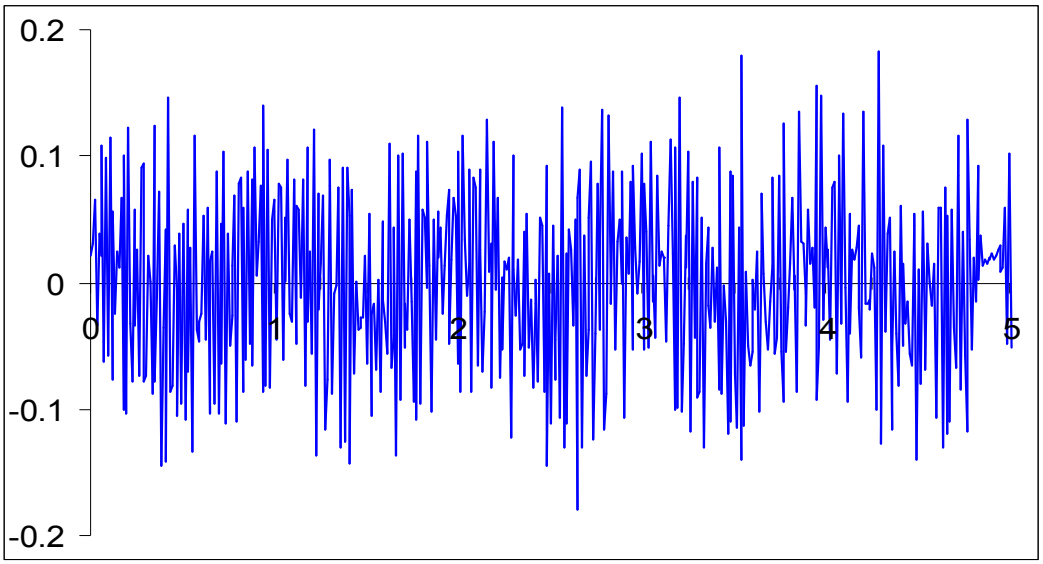
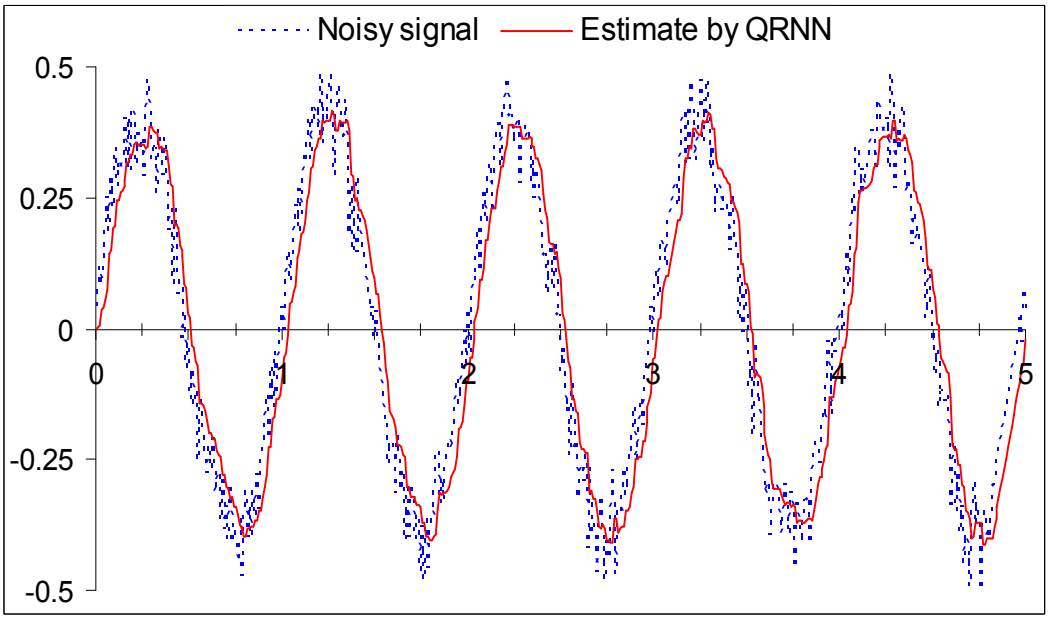



Figure-6.5: A segment of Figure-6.4




$t$ (Sec)

Figure-6.6: Time varying error with normal wave (SNR: 10dB)




$t$ (Sec)

Figure-6.7: Noisy and estimated sinusoidal signals (SNR: 10dB)

$L = 2, N = 200, \Delta x = 0.01, \sigma_0 = 0.6, \xi = 1, \eta = 0.5,$
$m = 0.5, \hbar = 1, \kappa = 1, \Delta t = 0.01.$

It should be noted that the spread of the wave $\sigma$ is time varying and its value depends on the magnitude of the error. If this error becomes smaller (i.e., approaches to zero)

then $\frac{1}{\sqrt{\sigma\sqrt{\pi}}}$ increases (or explodes) which in turn causes the wave to exceed the threshold limit of one in the well. To overcome this problem this term is reset to the initial value if the value of $\sigma$ goes below a certain limit (e.g., $\sigma < 0.02$) after the initial wave packet is launched. It has been observed that in such a case the wave packet remains within the threshold value of one and be able to remove the noise from the signal. This has been shown in Figure-6.7 as well as the first one second in Figure-6.8.

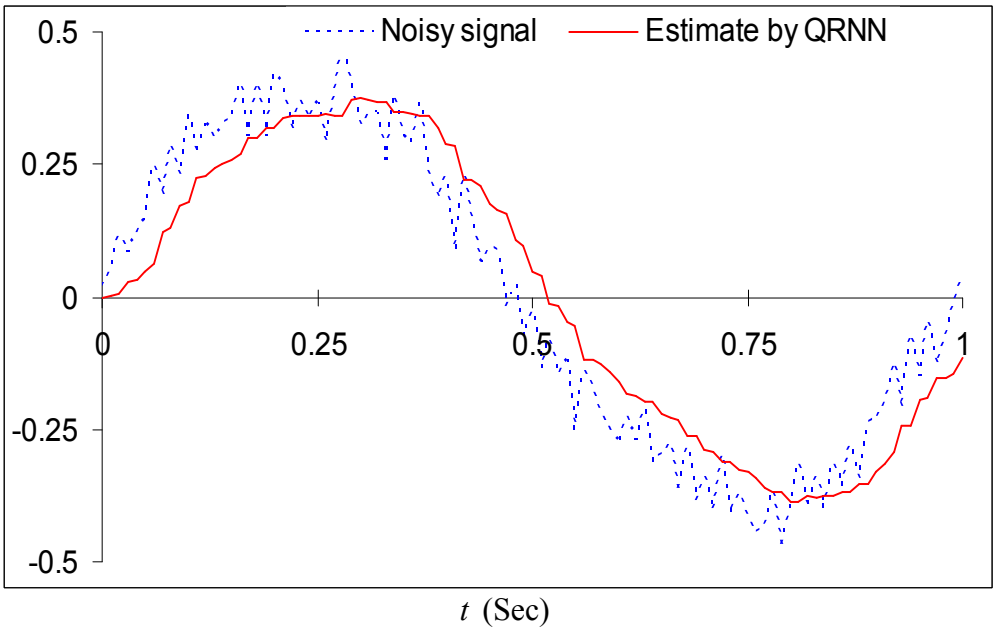



Figure-6.8: A segment of Figure-6.7

To see further how good the estimation is and whether the estimated signal has achieved the actual amplitude, the un-normalized actual (noise free) signal and their estimated signals (first five seconds) are shown in Figure-6.9. The signal is tracked with a RMSE of 0.02222. It can be seen from the graph that the estimated signal has achieved the required amplitude of the actual signal but in some cases it failed to do so. This is probably because of the spread of the wave packet, weights adjustment in the network, and/or truncation error in the evolution of the *pdf*.

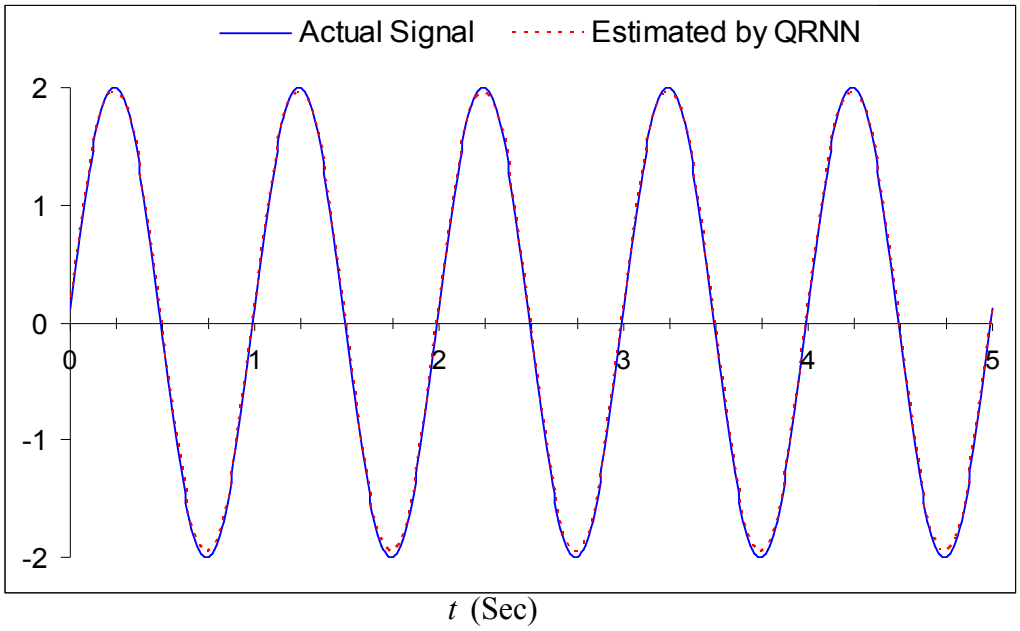



Figure-6.9: Un-normalized actual and estimated sinusoidal signals (SNR: 10dB)

## II. Calm wave

Results shown above were obtained using the normal wave. Another way of removing noise from the signal is to use a wave packet, which remains calm at the initial stage (see Section-5.4.2). For this test, the same set of signals is used as in the earlier test. Values of the parameters are selected as follows:

a) ***SWE parameters***: The universal constant $\hbar = 1$ and mass of the quantum object $m = \frac{1}{2}$,

b) ***Discretisation parameters***: The length of the well is taken as $L = 20$ with $\Delta x = 0.1$. The reason of taking this value is that the periodic signal (in this case, equation (6.3)) has a period of 100 samples per cycle. For the better performance of the filter, it is necessary to have doubled the number of neurons in the network to implement this calm wave. The initial spread of the wave is selected to be $\sigma_0 = 0.6$ which will keep the spread and amplitude of the wave well enough inside the well, $\Delta t = 0.1$ (for the first

set of measurements), $\Delta t = 0.01$ (for the second set of measurements) and $\Delta t = 0.001$ (for the third set of measurements), and

c) ***Neural parameters***: Number of neurons (which also represents the grid points for the Crank-Nicolson scheme) $N = \frac{L}{\Delta x} = 200$, the weight $w(k,t)$ vector is filled in with random numbers drawn from the Gaussian distribution with zero mean and standard deviation of one, the learning parameter $\eta = 0.5$, and potential function excitation $\xi = 1$.

With these values for the parameters, the QRNN filter is trained. Table-6.2 below shows the performance of the filter for all four different signals of various noise strengths for the sampling frequency of 100 samples per cycle. The table is also presented through graphs in the Figure-6.11. Results for other sampling frequencies are shown in Appendix-B (in Table-6.2a: for 10 samples per cycle, and in Table-6.2b: for 1000 samples per cycle).

| Signals/Strengths | Root-mean-square-error (RMSE) | | | |
|---|---|---|---|---|
| | 6 dB | 10 dB | 20dB | 30dB |
| Sinusoidal | 0.13125 | 0.10843 | 0.08727 | 0.08594 |
| Shifted sinusoidal | 0.13384 | 0.11140 | 0.08891 | 0.08633 |
| Amplitude modulated | 0.05975 | 0.05397 | 0.05021 | 0.05016 |
| Mixed sinusoidal | 0.10181 | 0.08100 | 0.05779 | 0.05470 |

Table-6.2: RMSE of various signals and noise strengths (Calm wave)

L = 20, N = 200, $\Delta x = 0.1$, $\sigma_0 = 0.6$, $\xi = 1$, $\eta = 0.5$,

m = 0.5, $\hbar = 1$, $\kappa = 1$, $\Delta t = 0.01$.

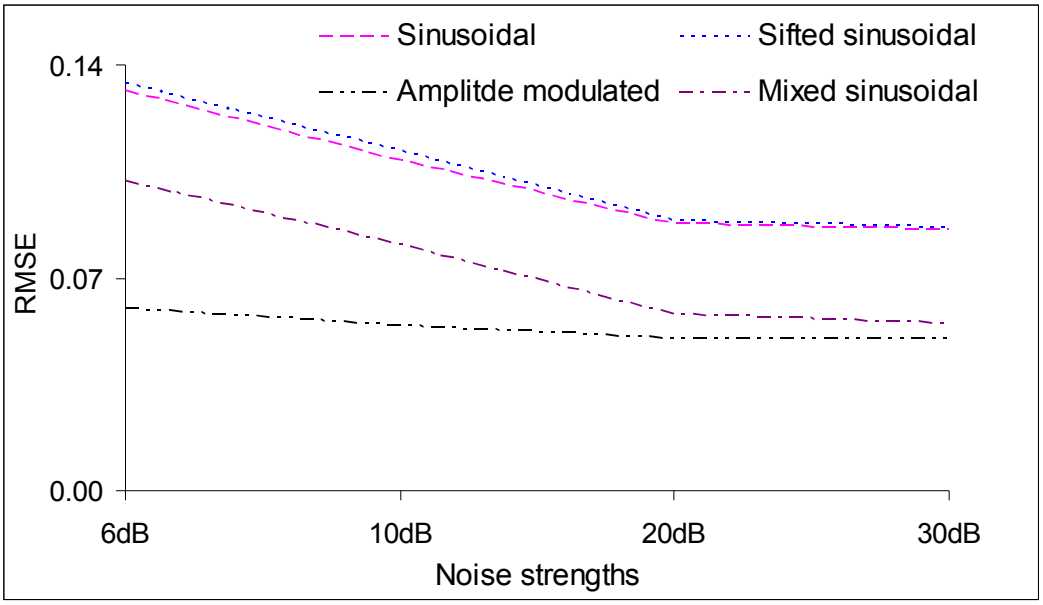



Figure-6.10: Noise strengths vs. RMSE (Calm wave)

From the above table (also from Figure-6.10) it can be seen that as the signal-to-noise ratio (SNR) increases the performance of the filter improves or a small decrease (in the case of amplitude modulated signal) in magnitudes of the RMSE. This small decrease may be because of the insufficient number of neurons in the network and/or weight adjustments in the filter. Investigation was carried out whether or not higher values of SNR cause the increase in RMSE. It is found that this is not the case as shown in Figure-6.11 with the noise strength of SNR 50dB. The signal is tracked with a RMSE of 0.01353.

It can also be observed from the table (Table-6.2, as well as Table-6.2a – 6.2b in Appendix-B) that the RMSE varies from signal to signal even for the same noise strengths. This is because of changes in pattern (or orientation) of the signals. For example, in the amplitude modulated signal (i.e., for signal equation (6.5)) the RMSE is comparably lower than others. The reason for lower RMSE is that of the magnitude of errors at the tail ends which is small (see Figure-6.2) and so the RMSE.

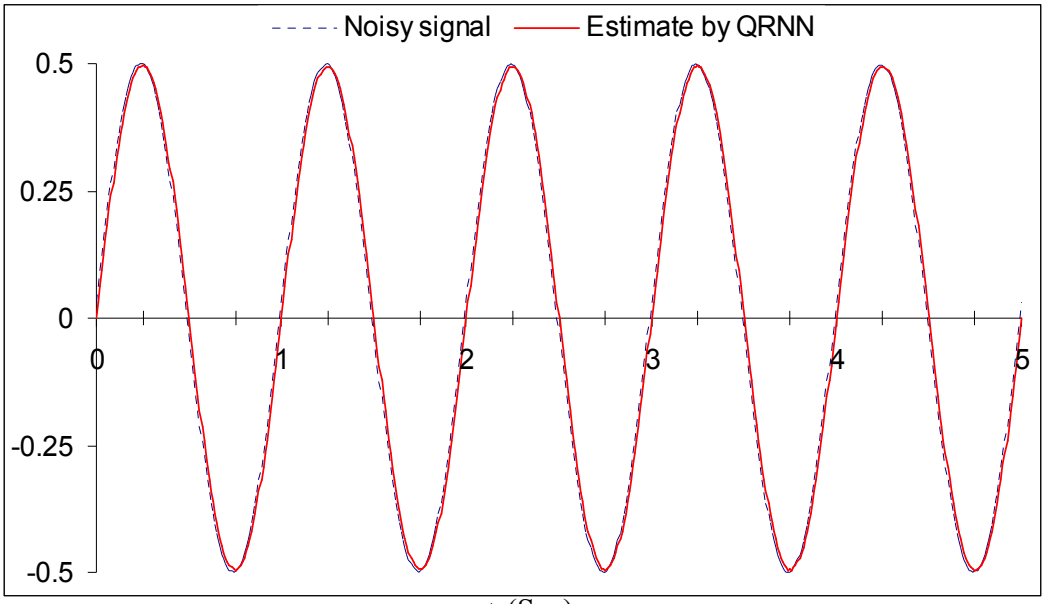



Figure-6.11: Noisy and estimated sinusoidal signals (SNR: 50dB)

$L = 20, N = 200, \Delta x = 0.1, \sigma_0 = 0.6, \xi = 1, \eta = 0.5,$
$m = 0.5, \hbar = 1, \kappa = 1, \Delta t = 0.01.$

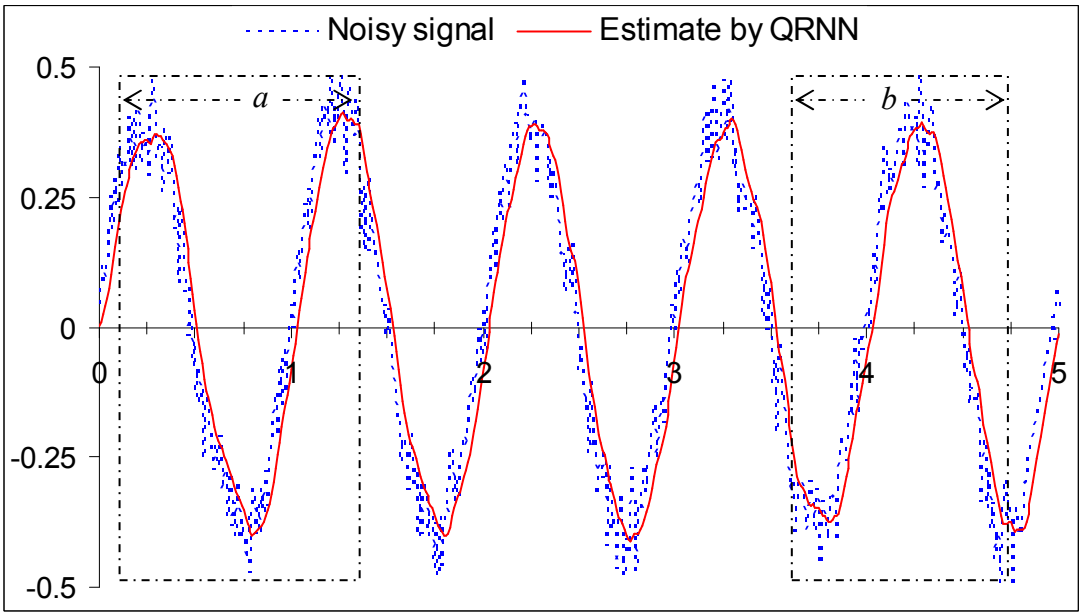



Figure-6.12: Noisy and estimated sinusoidal signals (SNR: 10dB)

$L = 20, N = 200, \Delta x = 0.1, \sigma_0 = 0.6, \xi = 1, \eta = 0.5,$
$m = 0.5, \hbar = 1, \kappa = 1, \Delta t = 0.01.$

To have a clearer view of the consolidated results shown in Table-6.2, signal which was displayed in Figure-6.4 is shown again along with its estimated signal in Figure-6.12 for the first five seconds. Error of the signal is shown in Figure-6.13.

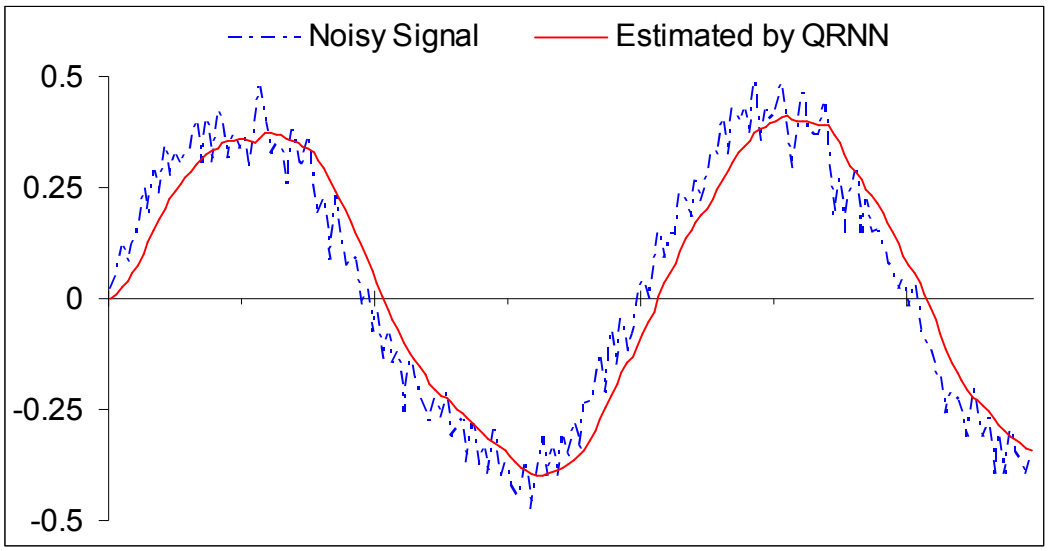



Figure-6.12a: A segment of Figure-6.12 (block-*a*)

It can be seen from the graph that using calm wave the QRNN filter is able to filter out the noise and learnt the entire signal correctly. Two blocks of Figure-6.12 is enlarged and shown in Figure-6.12a (for block-a) and Figure-6.12b (for block-b).

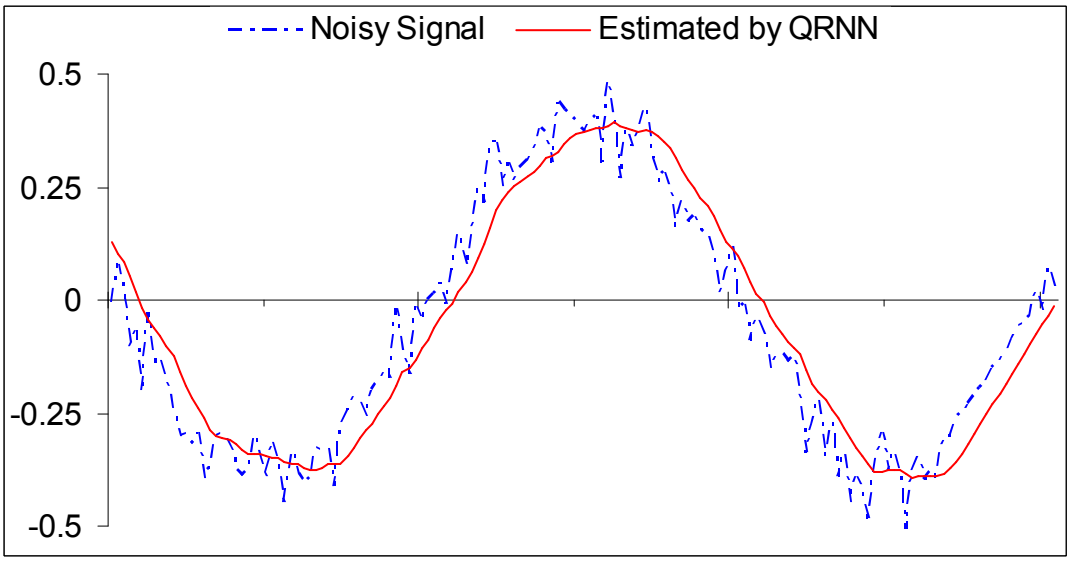



Figure-6.12b: A segment of Figure-6.12 (block-*b*)

It was mentioned earlier that at the outset a wave packet is formed with value zero and is treated as a calm wave at the initial stage. This calm wave is perturbed when the signal arrives as input to the network one after another in an iterative manner and thus a moving (or sliding) window of information is generated (see Section-5.4.2). For each input to the network, a weighted estimate is calculated by using the equation (4.26). The error is calculated (difference between the new measurement and the estimate) which constitutes the potential function. This error is then fed back to the network to update the network parameters which facilitates the learning and the evolution of the *pdf*. Thus presence of this wave in the well helps the evolution of the *pdf* in a way that the network not only stores the past history through the *pdf* as it evolves over time but also through the calm waves and the network weights. It is to be noted that the calm wave is simply a moving window of information that sweeps over in picking up the dynamics of the signal.

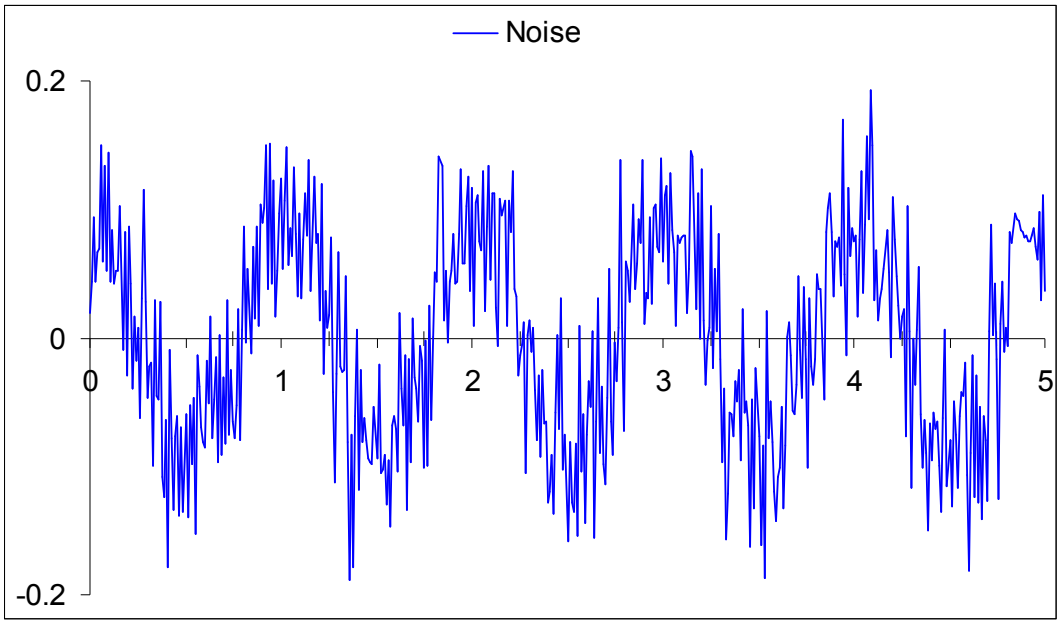



Figure-6.13: Time varying error with calm wave (SNR: 10dB)

An important observation for the calm wave is the apparent robust performance of the QRNN filter. This, in fact, indicates that with the calm wave the QRNN filter is independent of the amount of noise associated with the signal. To show this, further tests are performed on a signal with noise strength of SNR 0dB. Performance of the filter is shown in Figure-6.14. This signal is tracked with a RMSE of 0.17904.

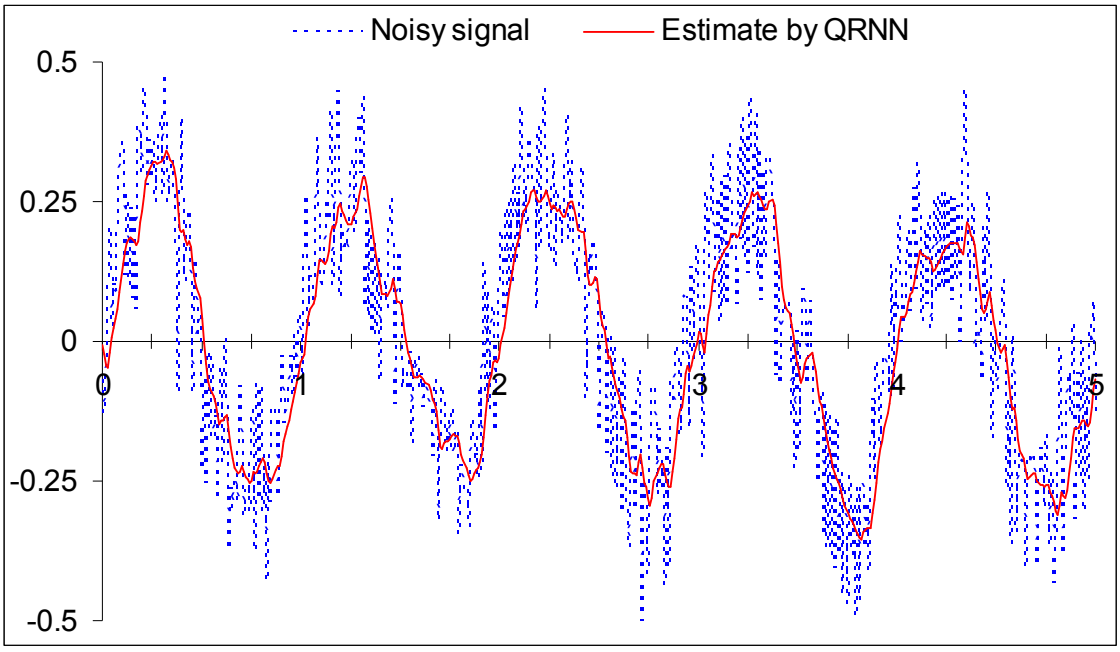



Figure-6.14: Noisy and estimated sinusoidal signals (SNR: 0dB)
L = 20, N = 200, $\Delta x = 0.1$, $\sigma_0 = 0.6$, $\xi = 1$, $\eta = 0.5$,
m = 0.5, $\hbar = 1$, $\kappa = 1$, $\Delta t = 0.01$.

It is to be mentioned here that the number of neurons in the network is related to the length of the well as well as the length of the grid point (see equation 5.2). Therefore, there are two different ways the number of neurons in the network can be created. One fixing the length of the well (i.e., $L$) changing the length of the grid (i.e., $\Delta x$), secondly, fixing the length of the grid (i.e., $\Delta x$) but changing the length of the well (i.e., $L$). In all of the above experiments with calm wave, the length of the well is selected to be $L = 20$ and length of the grid point $\Delta x = 0.1$. It has been observed that taking values

in this way performs better than if the length of the well would have been taken to be $L = 2$ with length of the grid point as $\Delta x = 0.01$ (i.e., number of neurons N = 200). This is shown in Figure-6.15. The signal is tracked with a RMSE of 0.31042. Although the reason for this poor performance is not clearly understood, it should be noted that, the Hamiltonian, where double gradient of the wave is computed in a small interval of space, would be the source of the problem. Evaluating double derivatives numerically always introduce additional approximations, which get amplified. This means that if length of the grid is increased then performance would improve which has already been confirmed in Figure-6.12.

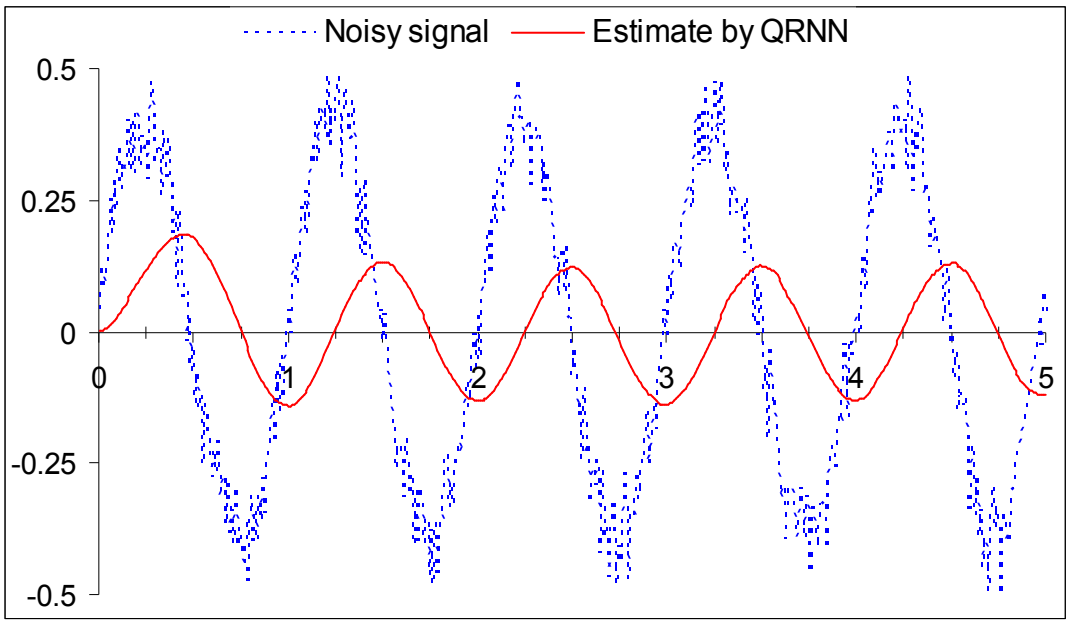



Figure-6.15: Effects of discretisation (sinusoidal signals, SNR: 10dB)
L = 2, N = 200, $\Delta x = 0.01$, $\sigma_0 = 0.6$, $\xi = 1$, $\eta = 0.5$,
m = 0.5, $\hbar = 1$, $\kappa = 1$, $\Delta t = 0.01$.

It is to be mentioned that in all above experiments under calm wave the value of $\sigma$ is reset whenever it goes below a certain limit (e.g., $\sigma < 0.02$). A simulation is run for the sinusoidal signal with sampling frequency of 100 samples per cycle and the noise strengths of 10dB without resetting this value. The result is shown in Figure-6.16. The

signal is tracked with a RMSE or 0.32461. It can be seen from the graph that the filter overshoots because of lower values of $\sigma$. This implies that resetting the values of $\sigma$ can become essential, and indeed critical.

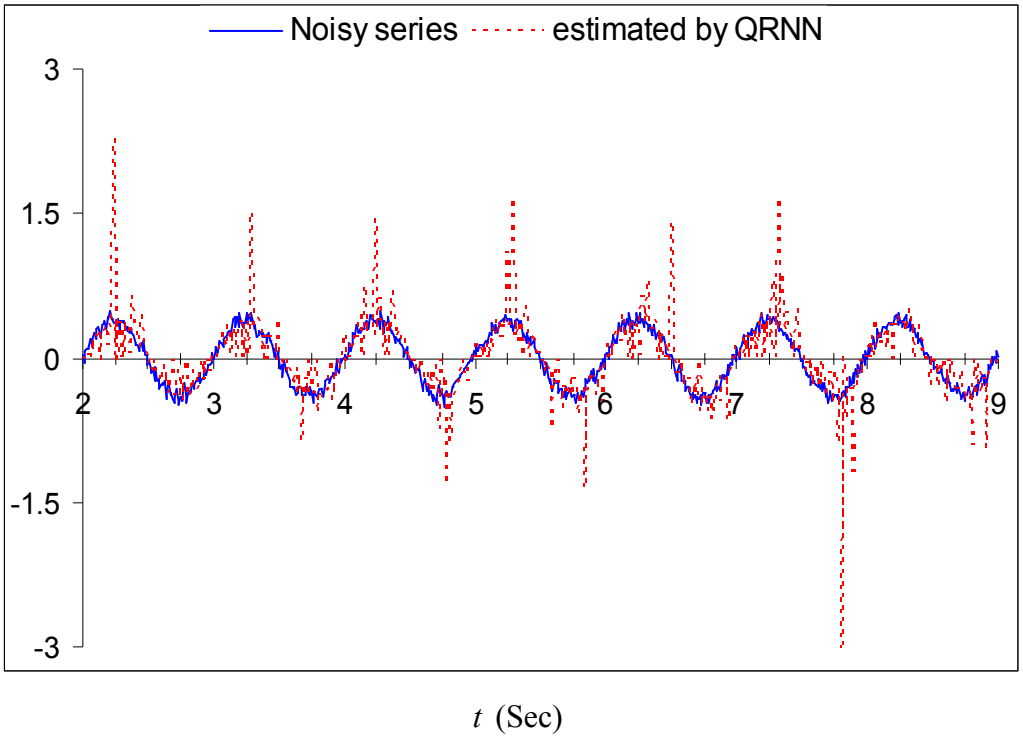



Figure-6.16: Effects of wave spread (sinusoidal signals, SNR: 10dB)
L = 20, N = 200, $\Delta x = 0.1$, $\xi = 1$, $\eta = 0.5$,
m = 0.5, $\hbar = 1$, $\kappa = 1$, $\Delta t = 0.01$.

## 6.4.2 Experiment-2:

This experiment is designed to investigate the performance of the QRNN filter due to change in values for the second set (that is discretisation parameters and hence the number of neuron in the neural parameter, see Section-5.3) keeping values for the first set unchanged (that is keeping values of the parameters same as in the first experiment). Change in the length of the well keeping $\Delta x = 0.01$ will in turn change the number of neurons in the network and it would be observed how this affect the QRNN filter and so the evolution of the *pdf*. The same set of signals (i.e., sampling frequency of 100

samples per cycle) will be used as in the first experiment with the value for the time interval $\Delta t = 0.01$. Wave packet is generated in the well using both normal and clam waves.

## I. Normal wave

In this test, the wave packet is generated in the well using a normal wave. Since the length of the well will be changed, therefore, it will also be required to change the initial spread of the wave proportionately. Values for the parameters of the three different sets are taken as follows:

a) ***SWE parameters***: The universal constant $\hbar = 1$, the mass of the quantum object $m = \frac{1}{2}$,

b) ***Discretisation parameters***: The length of the well is varied on a range of $1 \le L \le 4$, the initial spread of the wave $0.3 \le \sigma_0 \le 0.7$, $\Delta x = 0.01$, $\Delta t = 0.01$, and

c) ***Neural parameters***: Number of neurons (which also represents the grid points for the Crank-Nicolson scheme) $100 \le N \le 400$, the weight $w(k,t)$ vector is again filled in with random numbers drawn from the Gaussian distribution with zero mean and standard deviation of one, the learning parameter $\eta = 0.5$, and potential field excitation $\xi = 1$.

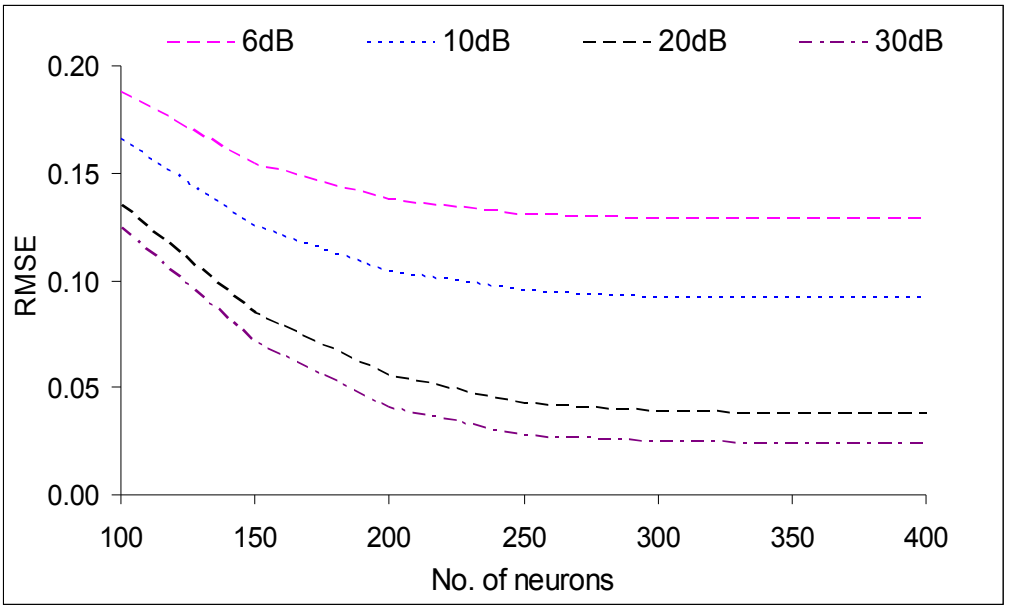



Figure-6.17: Number of neurons versus RMSE (normal wave) for the sinusoidal signal.

With these values of the parameters, the QRNN filter is trained on the sinusoidal signal (see equation 6.3) with four different noise strengths changing the number of neurons over a range of $100 \leq N \leq 400$. Figure-6.17 shows the graph on the number of neurons versus the RMSE. This results along with the results for the other sampling frequencies are presented in Appendix-B (with Table-6.17a: for 10 samples per cycle, Table-6.17b: for 100 samples per cycle, and Table-6.17c: for 1000 samples per cycle). It can be seen from these graphs that as the number of neurons in the network increases the RMSE decreases consequently the performance of the filter improves. It is found that a network with 200 neurons gives an optimal performance of the filter (that is, a small RMSE). However, although more neurons in the network perform better but they cause a large Hamiltonian which in turn slows down the solution of the SWE. It is to be mentioned here that for the sampling frequency of 10 samples per cycle the network did not perform as expected. This is because of the small number of samples available for the given time duration (10 seconds with 100 samples only). [This in turn an indication of the numerical strategy required for the implementation of the filter].

## II. Calm wave

Investigation is also carried out by introducing the calm wave in the well while changing the number of neurons (on a range of $20 \le N \le 80$ adjusting the spread, $\sigma$, of the wave appropriately) in the network taking values for the other parameters same as in the normal wave. First simulation is carried out for the sinusoidal signal with a sampling frequency of 100 samples per cycle on a well of lengths $2 \le L \le 8$ and $\Delta x = 0.1$. Figure-6.18 shows the performance of the filter. Tables for this graph and tables for other sampling frequencies are presented in the Appendix-B (with Table-6.18a: for 10 samples per cycle, Table-6.18b: for 100 samples per cycle, and Table-6.18c: for 1000 samples per cycle).

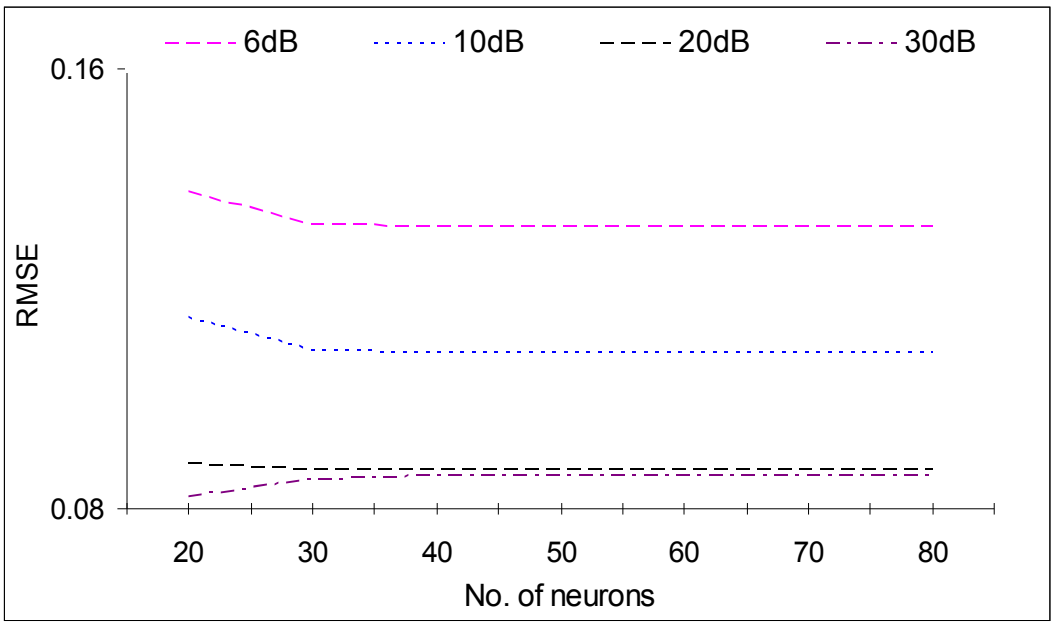



Figure-6.18: Number of neurons versus RMSE (calm wave)
(SR: 100 samples per cycle)
$2 \le L \le 8,\ 20 \le N \le 80,\ \Delta x = 0.1,\ \sigma = 0.6,\ \xi = 1,$
$\eta = 0.5,\ \mathrm{m} = 0.5,\ \hbar = 1,\ \kappa = 1,\ \Delta t = 0.01.$

It can be seen from the graph that as the number of neurons in the network increases (with varying lengths of the well) the performance of the filter remains steady for

respective noise strengths. One particular figure for mixed sinusoidal signal with noise strength of SNR 20dB is shown in Figure-6.19. The signal is tracked with a RMSE of 0.02301.

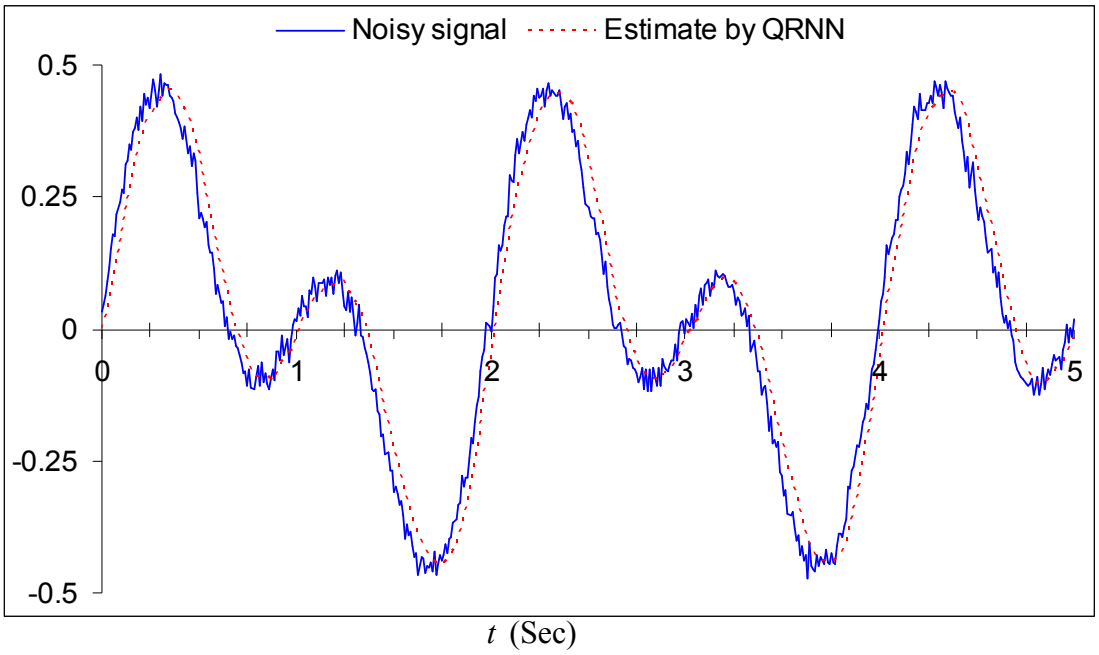



Figure-6.19: Noisy and estimated mixed sinusoidal signals (SNR: 20dB)

$L = 3,\ N = 30,\ \Delta x = 0.1,\ \sigma_0 = 0.6,\ \xi = 1,\ \eta = 0.5,$
$m = 0.5,\ \hbar = 1,\ \kappa = 1,\ \Delta t = 0.01.$

However, one important observation from the graph (see Figure-6.18) is that the RMSE remains steady for the noise strength of SNR 30dB which is supposed to be lowered compare to the noise strength of 10dB and 20dB. Further investigation to this problem shows that if the sampling frequency increases then this problem resolves. This is confirmed in Figure-6.20 for various noise strengths with sampling frequency of 1000 samples per cycle.

Compare to the normal wave, the major advantage of using the calm wave is that the network can learn with fewer neurons (in this case only 30 neurons) and the RMSE remains stable as the number of neurons increases. The only disadvantage is that the network cannot perform with a fewer samples expectedly and remains anomalous (see

Figure-6.21). This is, in fact, the general case for neural network where more samples help the network learn better.

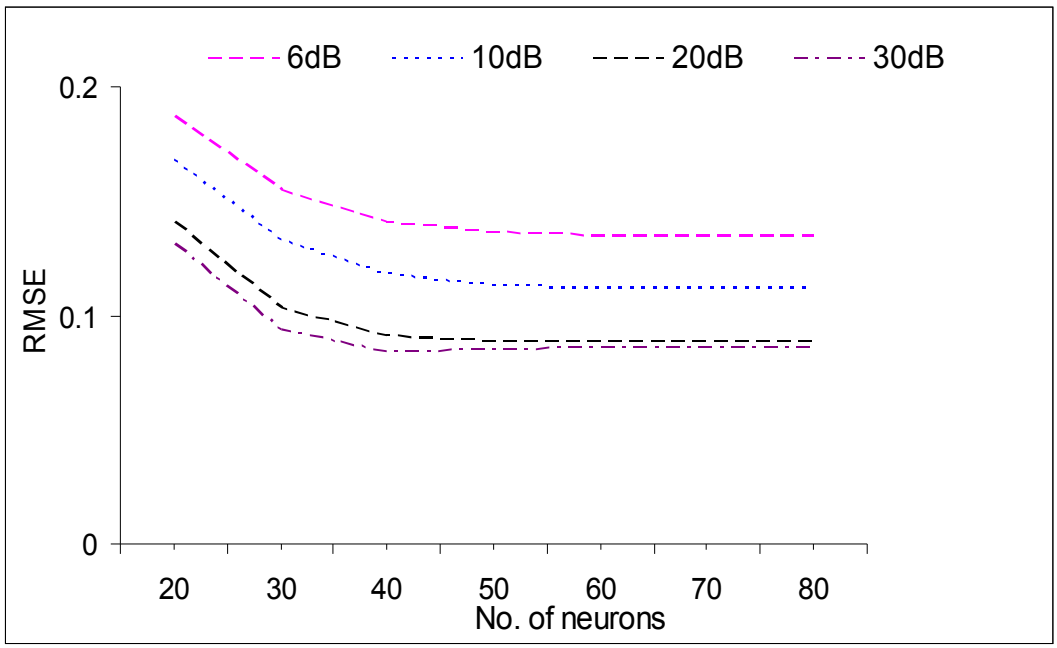



Figure-6.20: Effects of increasing sampling rate (1000 samples per cycle)

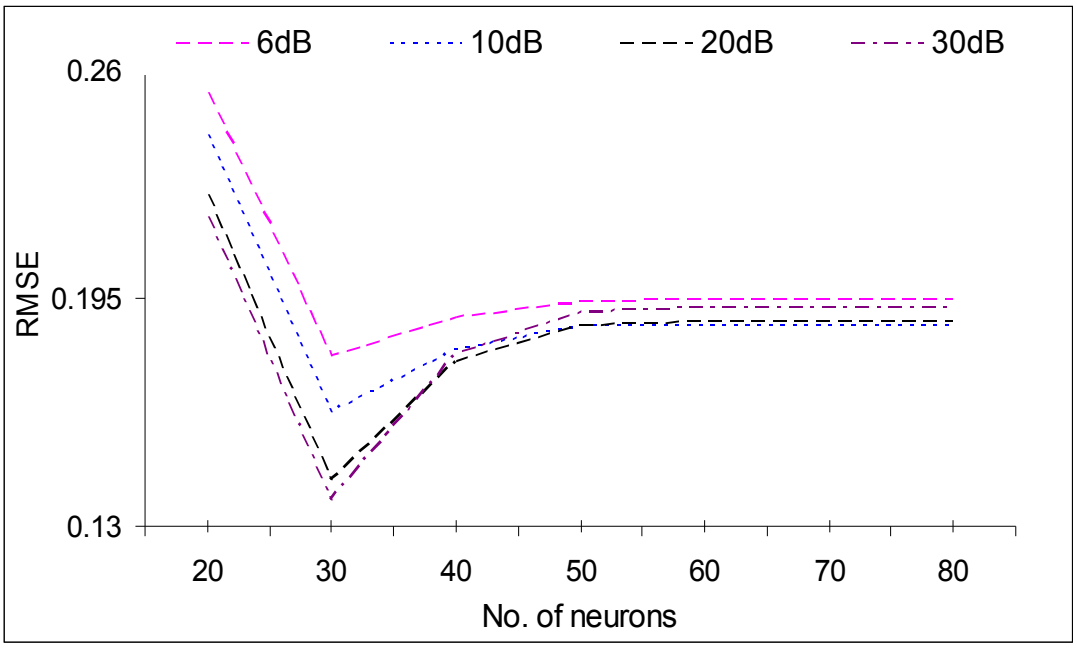



Figure-6.21: Effects of small sampling rate (10 samples per cycle)

## 6.5 Conclusion

This chapter has demonstrated the implementation of the QRNN filter using two different experiments. These tests were designed in a way that they cover methods discussed in Chapter-4 and Chapter-5. Simulations are performed considering various aspects of the filter. Numerical solution of SWE is implemented using Crank-Nicolson scheme. This scheme ensures that there is no rounding error introduced in implementing the filter and thus the numerical solution remains stable. Values of the parameters are selected according to the specification discussed in Chapter-5 and impacts of changing values of these parameters have been verified.

In each experiment it has been shown that the filter is able to capture the *pdf* information using both normal and calm waves. The density function is conditional in the sense that current state of the system is estimated based on all previous states which are propagated in the network through the errors defined by potential function. This in turn brings the essence of Sigma Algebra of the Martingale theory.

Performance of the filter in either way is comparable in terms of RMSE. *With the calm wave the filter performance is considerably better than the normal wave*. It is observed that if the sampling frequency (that is, the number of samples per cycle) increases then performance of the filter improves in terms of the RMSE. It should be noted that giving a higher sampling rate means more samples per cycle. This then implies that the time interval, $\Delta t$, for integrating the SWE becomes smaller. Indeed, it is well known fact that for nonlinear differential equations, smaller $\Delta t$ increases the accuracy of the numerical procedure. Hence, in this situation, performance of the filter improves. This indicates that performance of the filter is linked to the sampling rate which is obvious from the neural network point of view. The results are stable and converge as expected. In the following chapter application of this filter to some real world situations will be investigated.

# Chapter 7

# Chaotic and Medical signals

## 7.1 Introduction

In Chapter-6, it was shown that the QRNN filter is able to a) capture the dynamics of the underlying signal in a noisy process, and b) evolve a density function with which it is able to estimate the true signal. In this chapter, the QRNN filter is tested on benchmarked chaotic time series such as the Lorenz map and Mackey-Glass series. These series (maps) have far reaching applications in the real world problems. For example, the Lorenz map has been used to measure the concentration of wealth in a population, and Mackey-Glass series has been used to model the white blood cell production in human body (Lorenz 1963; Hénon 1976; Mackey 1977; Flake 1998). Apart from these two series, situations where the noise process is a non-stationary process are also been considered. The QRNN filter is also considered to a practical real world situation where the measurement data consists of a set of blood sugar level of a patient, who has been monitored.

In all of these cases, the QRNN filter is applied and tested by generating both normal and calm waves (see Chapter-5 and Chapter-6). The purpose of these tests is to further illustrate the properties of the QRNN filter on a wide verity of signals to demonstrate its advantages. Most of these signals have been studied extensively in the signal processing literature (Eric 1993; Alex 2000; Wang *et al.* 2005).

These results will show an inherent advantage of the QRNN filter. Often, depending on the applications domain, filters have to be modified or fine tuned by the designer. In this chapter, it can be seen that the QRNN filter requires very little tuning (as shown in Chapter-6) and can be applied to most applications with very little rejigging.

## 7.2 Mackey-Glass series

The first benchmarked series is the Mackey-Glass series (Mackey 1977). This series is a continuous time description for the dynamics of the *white blood cell production in the human body* (Mackey 1977; Flake 1998) and is given by the following equation:

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{7.1}$$

where, $\tau$ is a time delay that controls whether the series has a fixed point, a limit cycle, or exhibits chaotic behaviour.

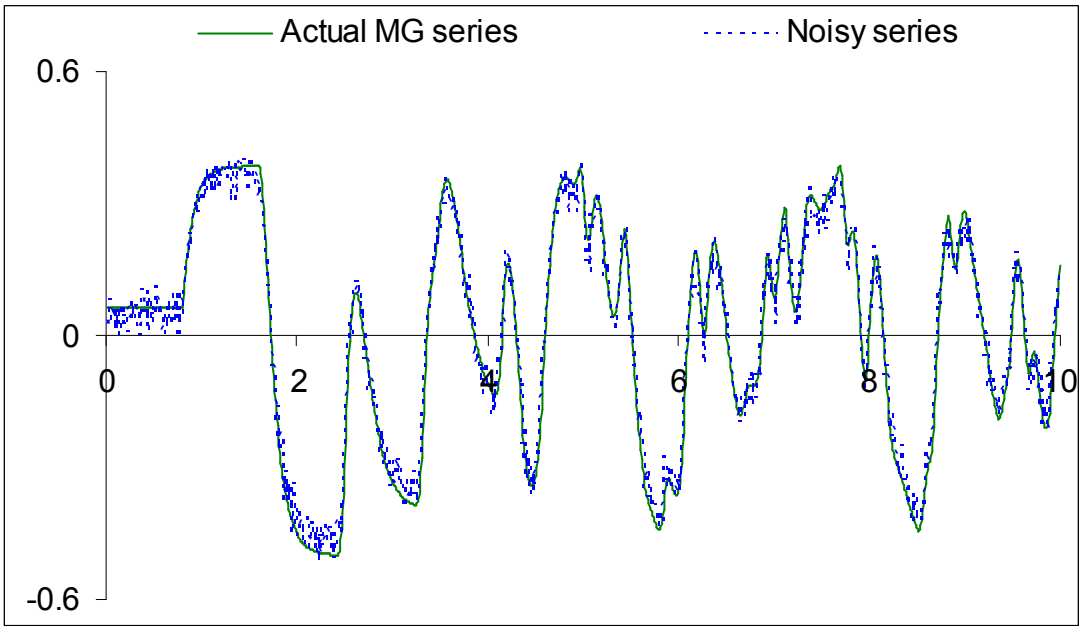



Figure-7.1: Mackey Glass actual and noisy (SNR 20dB) series

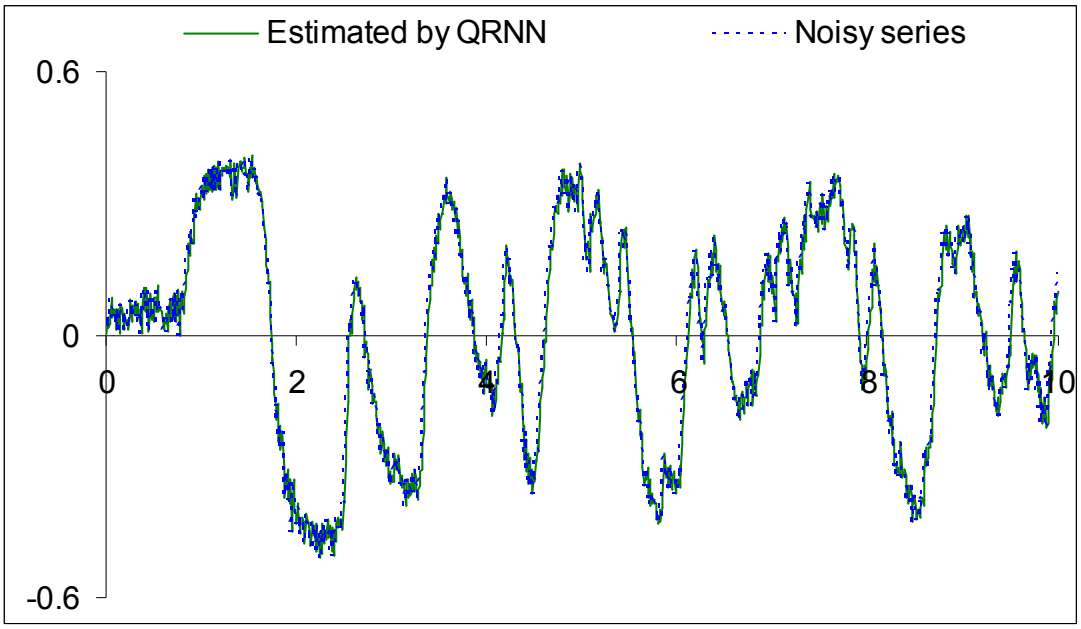



Figure-7.2: Mackey Glass series with normal wave
$L = 5,\ \Delta x = 0.01, N = 500,\ \sigma_0 = 0.6,\ \Delta t = 0.01,\ m = 0.5$
$\xi = 1,\ \eta = 0.5,\ \hbar = 1,\ \kappa = 1.$

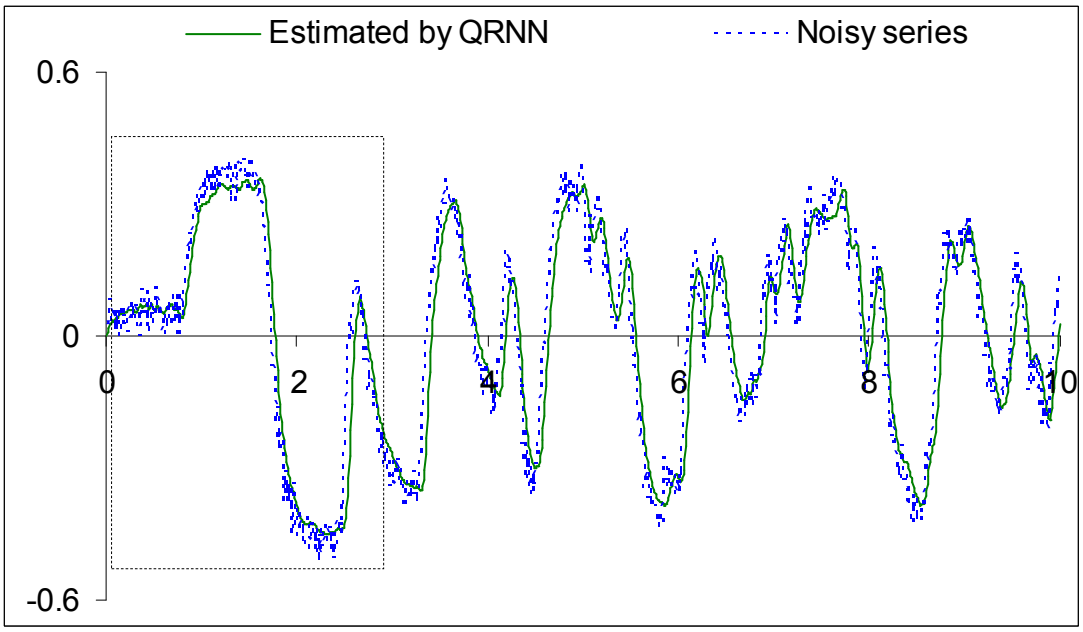



Figure-7.3: Mackey Glass series with calm wave
$L = 5,\ \Delta x = 0.1, N = 50,\ \sigma_0 = 0.6,\ \Delta t = 0.01,\ m = 0.5$
$\xi = 1,\ \eta = 0.5,\ \hbar = 1,\ \kappa = 1$

This benchmark series is used frequently for testing nonlinear predictive models such as in (Eric 1993; Alex 2000). A fourth order Runge-Kutta method is employed to solve the equation numerically. Following the convention in the literature (Mackey 1977), a time delay of $\tau = 80$ is used in generating the signal which is sampled at every 0.01 units of time and a noise strength of SNR 20dB is added to the series. Figure-7.1 shows the actual and noisy signals. The QRNN filter is trained over the series. Figure-7.2 and Figure-7.3 show the graph of the series along with the estimated signals for normal and calm wave respectively. The series is tracked with a RMSE of 0.04059 for normal wave and 0.0727 for calm wave.

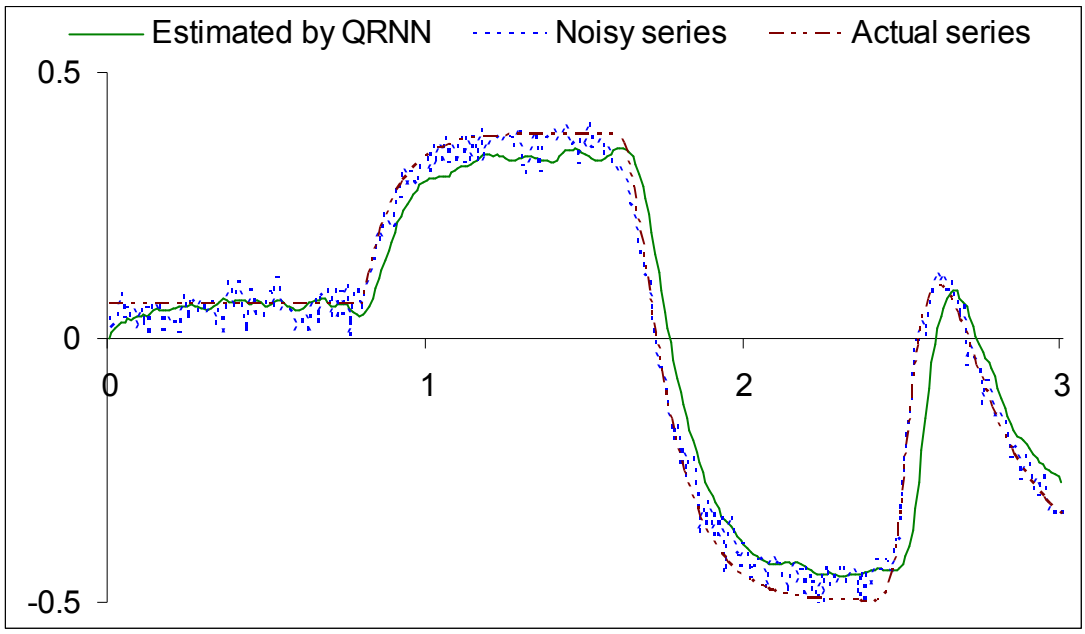



Figure-7.4: A segment of Figure-7.3

It can be observed from Figure-7.2 that the filter could not remove the noise with the normal wave. However, using the calm wave the filter has performed better (Figure-7.3). This is expected, given the manner in which the two methods are implemented (see Chapter-5). A segment of Figure-7.3 is enlarged and is shown in Figure-7.4. It can be

observed from this figure that although using the calm wave the filter performed well in terms of removing noise, there is a small off-set in the amplitude and also a small shift in tracking the signal. The problems of not achieving the actual amplitude probably because of the scaling factor and the shift may be because of the nature of the network and the number of neurons.

## 7.3 Lorenz series

Lorenz series is generated by a set of strongly coupled ordinary nonlinear differential equations which together exhibit chaos and is given by as follows:

$$\dot{x} = \rho(y - x) \tag{7.3}$$

$$\dot{y} = x(r - z) - y \tag{7.4}$$

$$\dot{z} = xy - bz \tag{7.5}$$

where $\dot{x}$ represents derivative with respect to time $t$ and similarly for the other terms, $\rho$, $r$, and $b$ are parameters. It is taken that $x$ is the output of the Lorenz series and is the variable of interest for the QRNN filter. The values of the parameters involved in the series is selected as $\rho = 20$, $r = 45.92$, and $b = 4$ following the convention in the literature (Haykin 2001). The series is sampled at a period of 0.01 seconds and noise strength of SNR 10dB is added to the series. The QRNN filter is trained using both normal and calm waves over the normalized state $x$. Figure-7.5 shows the first 600 samples along with the estimated signal for normal wave while Figure-7.6 shows the graph for the same state $x$ with calm wave. The signal is tracked with a RMSE of 0.0510 for normal wave and 0.0734 for calm wave. It can be seen from the graph (see Figure-7.5) that the estimated signals over fits using the normal wave. Although using calm wave (see Figure-7.6) the filter is able to remove the noise but it fails to achieve the required amplitude of the signals.

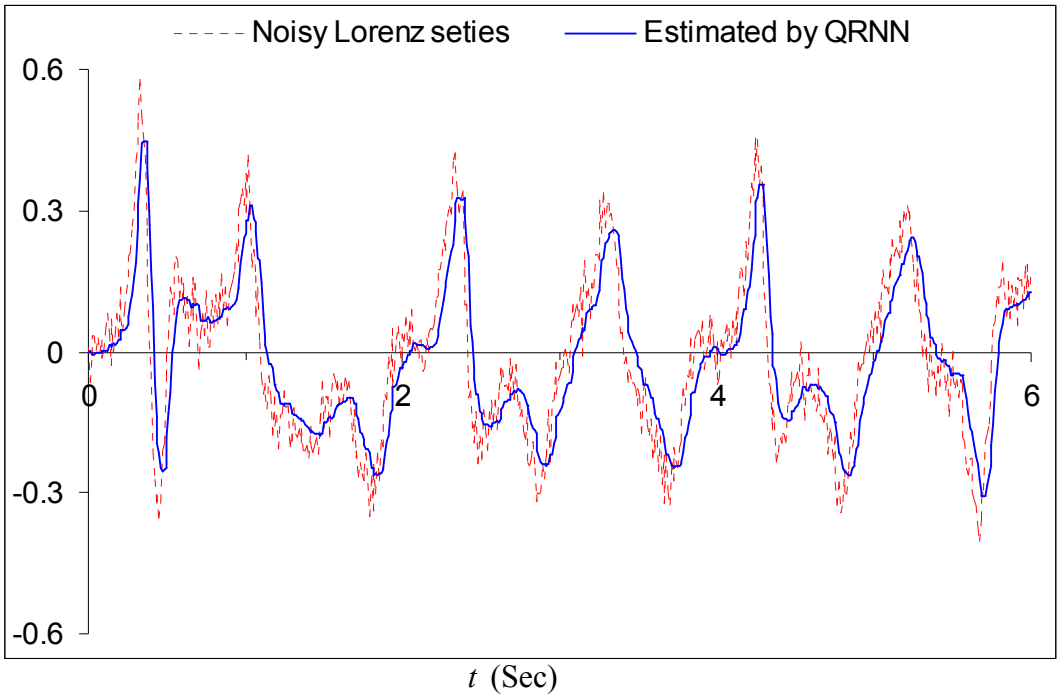



Figure-7.5: State $x$ of the Lorenz series with normal wave
$L = 3,\ \Delta x = 0.01,\ N = 300,\ \sigma_0 = 0.6,\ \Delta t = 0.01,\ m = 0.5$
$\xi = 1,\ \eta = 0.5,\ \hbar = 1,\ \kappa = 1$

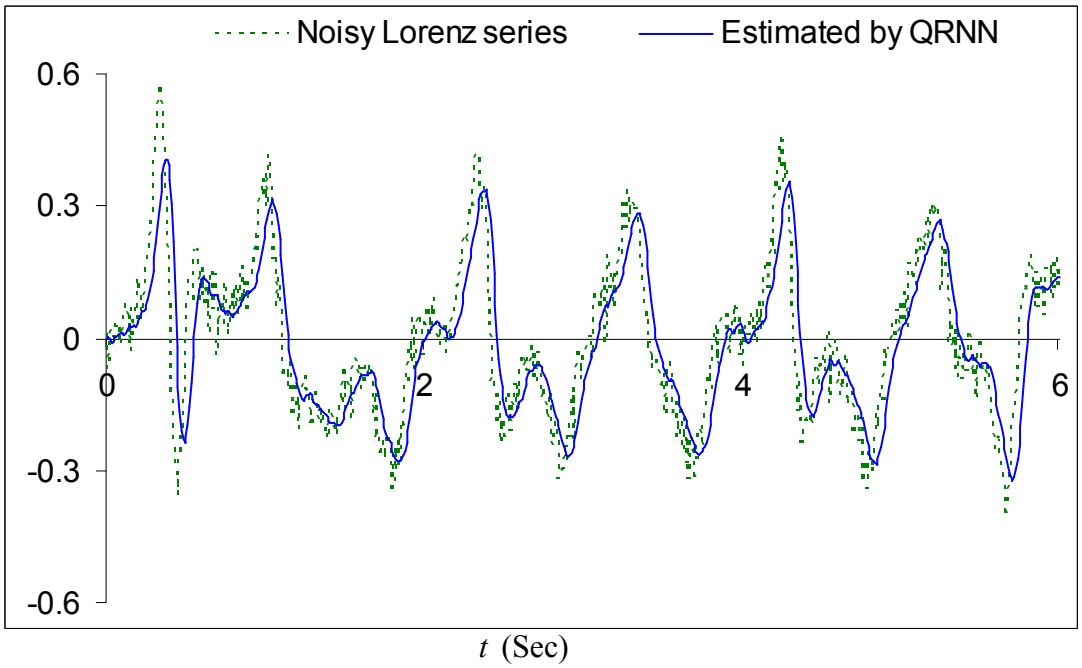



Figure-7.6: State $x$ of the Lorenz series with calm wave
$L = 4,\ \Delta x = 0.1,\ N = 40,\ \sigma_0 = 0.6,\ \Delta t = 0.01,\ m = 0.5$
$\xi = 1,\ \eta = 0.5,\ \hbar = 1,\ \kappa = 1$

## 7.4 Filtering non-stationary signals

So far in this thesis, all previous experiments (Experiments 1–2 presented in Chapter-6) were constructed based on the noise where the noise process were stationary white noise processes (i.e., Gaussian zero mean white noise). White noise refers to a signal whose *value at time t is statistically independent of its value at time (t + 1)*. In this section signal with a non-stationary noise process are considered. A non-stationary white noise sequence is generated by modulating the white noise sequence with a sinusoidal function as described by the amplitude modulated signal (see Chapter-6, equation (6.5)). The amplitude of the function is selected to be 0.2 and the noise process is normalized so that it falls within the range of [-1 1]. A partial graph of the non-stationary noise process is shown in Figure-7.7.

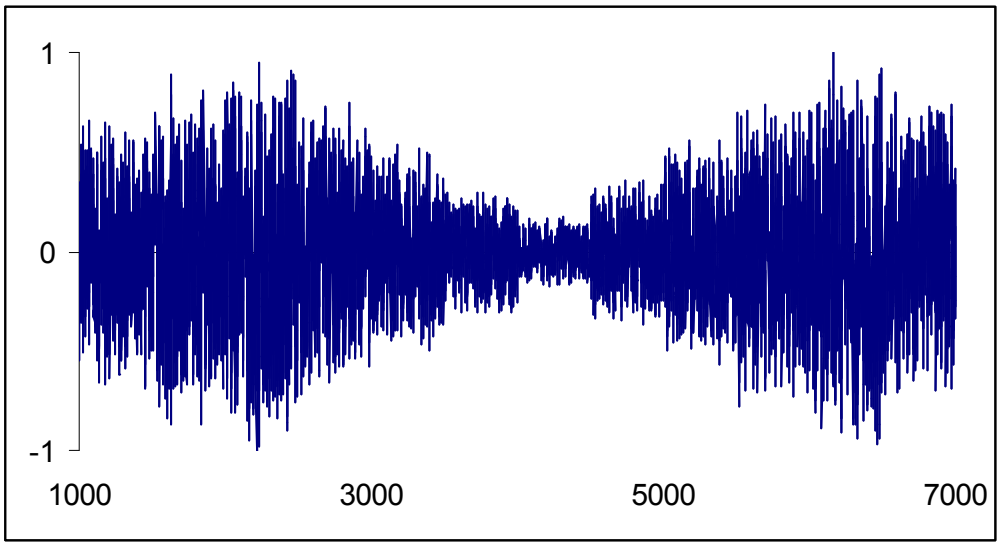



Figure-7.7: Non-stationary white noise of (1000–7000) 6000 samples

The measurement signal is sampled at a rate of 1000 samples per cycle per second using the sinusoidal signal described in Chapter-6 (see equation (6.3)) with non-stationary noise at strength of 10dB. Figure-7.8 shows the non-stationary signals along with the estimated signals with normal wave. The signal is tracked with a RMSE of 0.03215. It

can be seen from this figure that noise strength varies over time (compare to block-*a* and block-*b* as shown in the figure at peak to peak) which is due to the effects of the non-stationary noise process. The filter did not recognize the variations of noise process which is one of the features mentioned in Chapter-1 (see Section-1.3). With normal wave the filter has not been able to filter out the noisy signals (see block-*a*, and block-*b* in Figure-7.9). This is probably because of the higher rate of sampling frequency (in this case 1000 samples/cycle) and the non-stationary noise process.

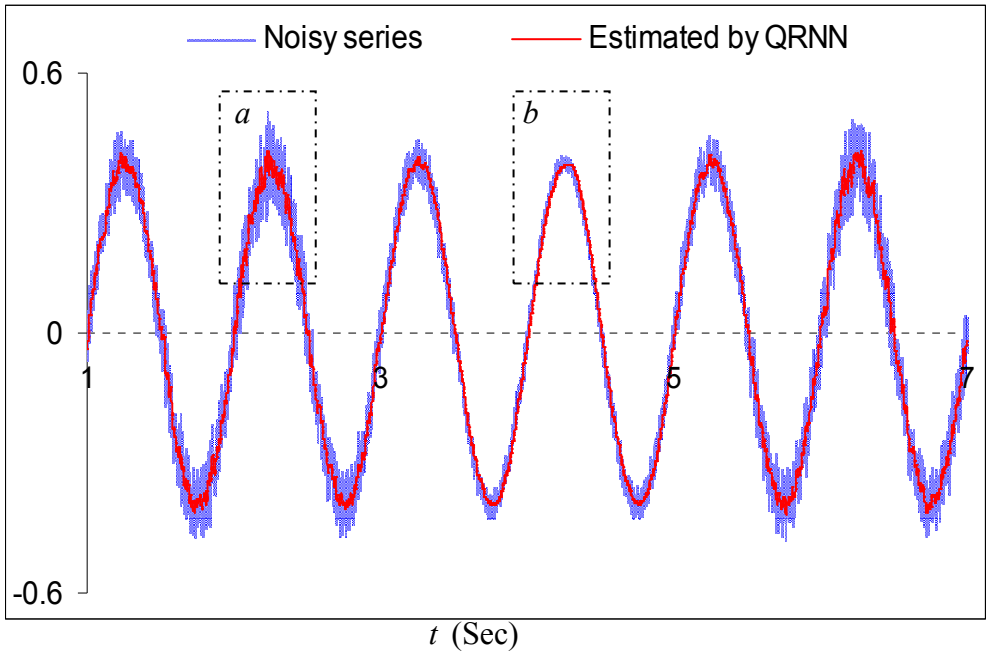



Figure-7.8: Noisy and estimated non-stationary signals with normal wave
$L = 3,\ \Delta x = 0.01, N = 300,\ \sigma_0 = 0.6,\ \Delta t = 0.001,\ m = 0.5$
$\xi = 1,\ \eta = 0.5,\ \hbar = 1,\ \kappa = 1$

The filtered result is shown in Figure-7.11 using calm wave. It can be seen from the figure that QRNN filter is able to retrieve the actual signal with a RMSE of 0.00541. Note that the QRNN filter does not differentiate between stationary and non-stationary noise processes and the two results are similar. In other words, although the effects of non-stationarity can not be seen in the graphs due to normalization but it can be seen from the entire graph that estimate of the true value of the state has not been changed

over time regardless of the non-stationary noise process which in turn indicates that the filter performance has no link to the noise process. To illustrate this, two segments of Figure-7.11 (block-*a* and block-*b*) is enlarged and shown in Figure-7.12 and Figure-7.13 respectively. It can also be observed that if the density of measurement increases (1000 measurements per cycle compare to 100 measurements per cycle) then the performance of the filter improves.

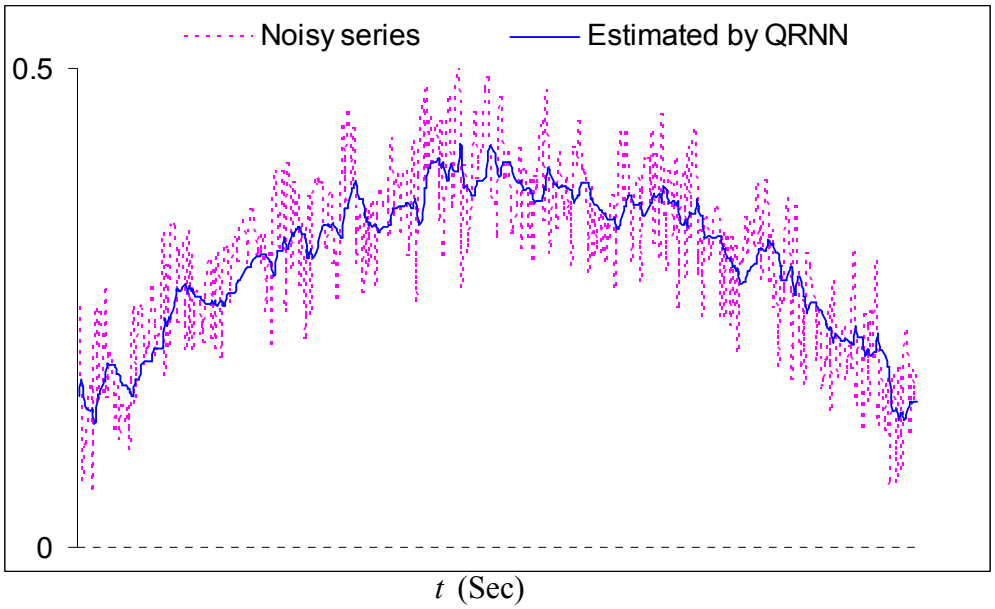



Figure-7.9: A segment (block-*a*) of Figure-7.6

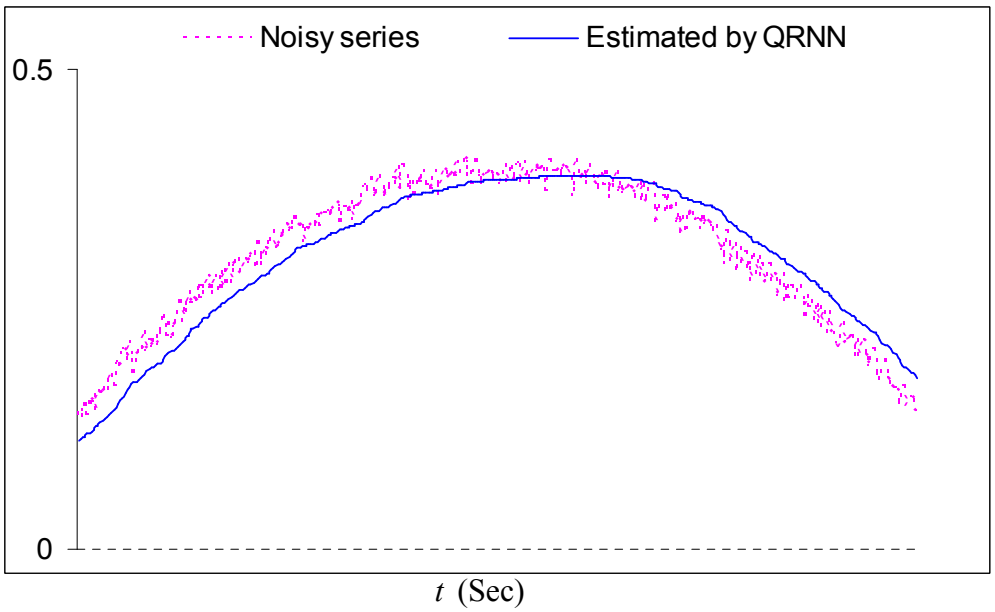



Figure-7.10: A segment (block-*b*) of Figure-7.6

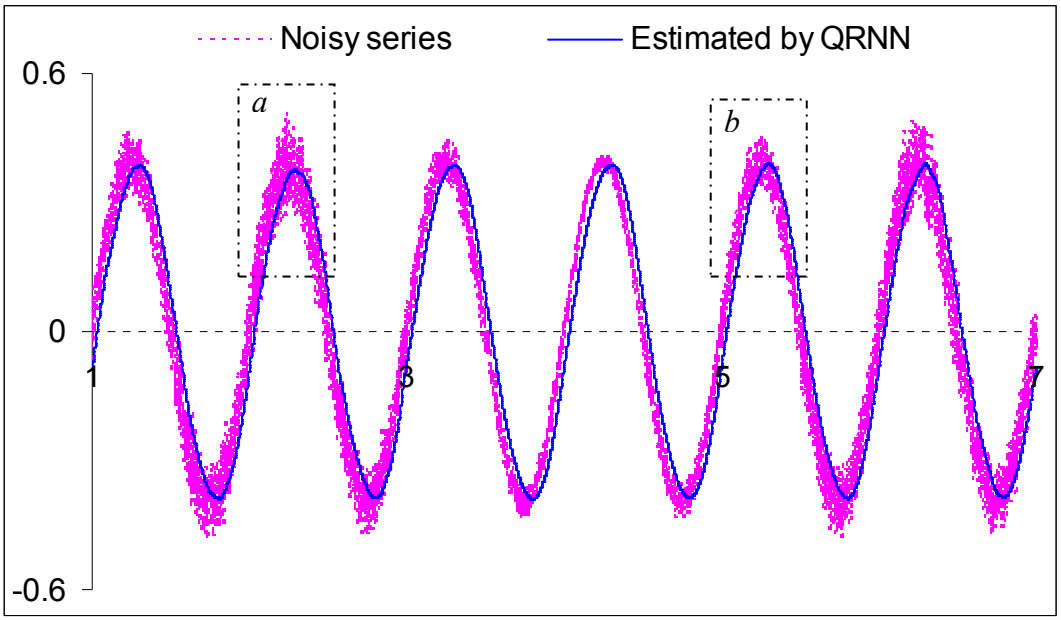



Figure-7.11: Noisy and estimated non-stationary signals with calm wave
$L = 4,\ \Delta x = 0.1, N = 40,\ \sigma_0 = 0.6,\ \Delta t = 0.001,\ m = 0.5$
$\xi = 1,\ \eta = 0.5,\ \hbar = 1,\ \kappa = 1$

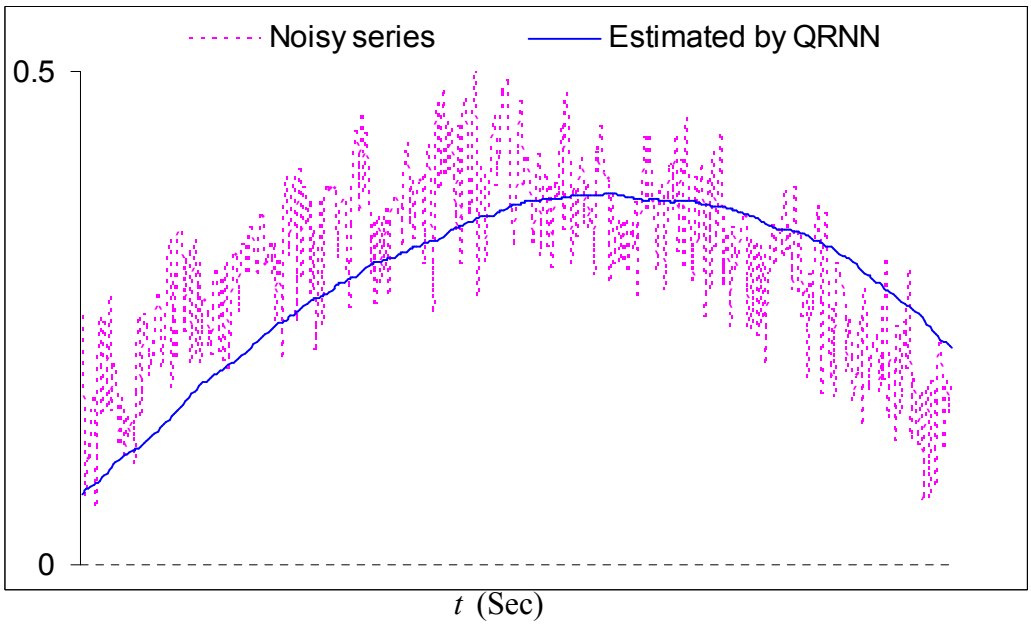



Figure-7.12: A segment (block-*a*) of Figure-7.9

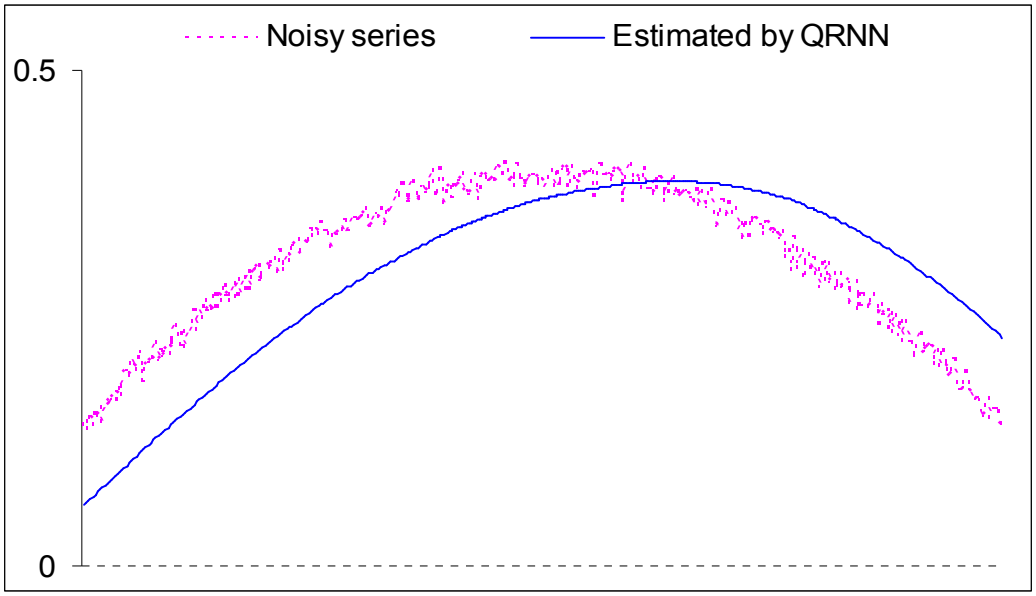



Figure-7.13: A segment (block-*b*) of Figure-7.9

The reason for the better performance of the filter is because double-loop procedure and the fact that the Hamiltonian matrix (see Chapter-4, equation 4.21) contains second order derivatives which are implemented numerically using the forward time centered space (FTCS) scheme. This scheme provides approximations for the second derivative. However, this can be improved by keeping the value of $\Delta t$ small (Press 1992). In this case, the data was generated with such a small $\Delta t$ (i.e., $\Delta t = 0.001$) which can be seen to be small compare to the cycle-time of the signal.

## 7.5 Filtering blood sugar data

A patient's blood sugar level is monitored over a period of time. This is done using a subcutaneous instrument worn around the waist of the patient. This instrument takes measurement at predetermined time intervals. There are 470 data points which is shown Figure-7.14. The problem becomes more acute due to the variation of values in the data set, limited number of data points, lack of information about the collection time interval, and the reliability of the measurement process. Given these complexities, this becomes an ideal test bed for the QRNN filter.

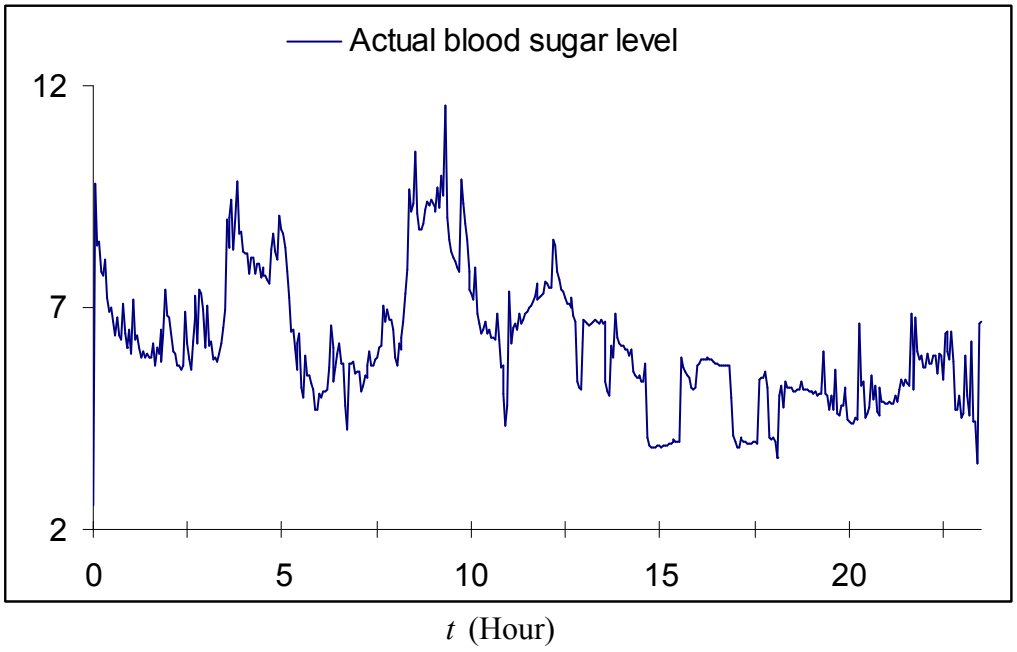



Figure-7.14: Actual data for blood sugar level

The challenge here is to select the values for the parameters given the limited number of measurement points. Most important is to select the values for the spread ($\sigma$) of the wave packet as well as the time interval of integration. These two values are taken to be $\sigma = 0.5,\ \Delta t = 0.05$ and then trying several different values and picking up the one that seems to extract as much of the underlying dynamics of the signal as possible. Although this trail and error method should ideally be carried out more rigorous approach, it produces good results after a few attempts.

The parameters of the SWE are adjusted according to the length of the well. This requires increasing the number of neurons in the network for proper learning process and is a result of the limitations of the data set as described earlier.

With normal wave, the estimated signals along with the normalized signals are shown in Figure-7.15. It can be observed from the graph that the network managed to learn the dynamics and estimate the signal correctly with a RMSE of 0.07214 despite relatively small size of measurements.

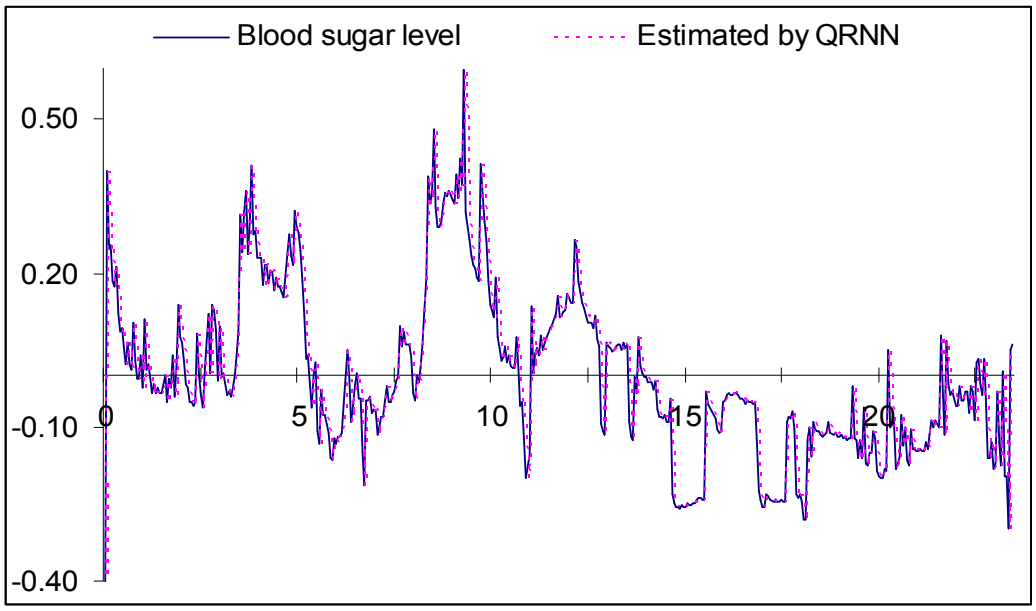



Figure-7.15: Blood sugar level actual and estimated data with normal wave
$L = 3,\ \Delta x = 0.01, N = 300,\ \sigma_0 = 0.6,\ \Delta t = 0.05\text{hour},\ m = 0.5$
$\xi = 1,\ \eta = 0.5,\ \hbar = 1,\ \kappa = 1$

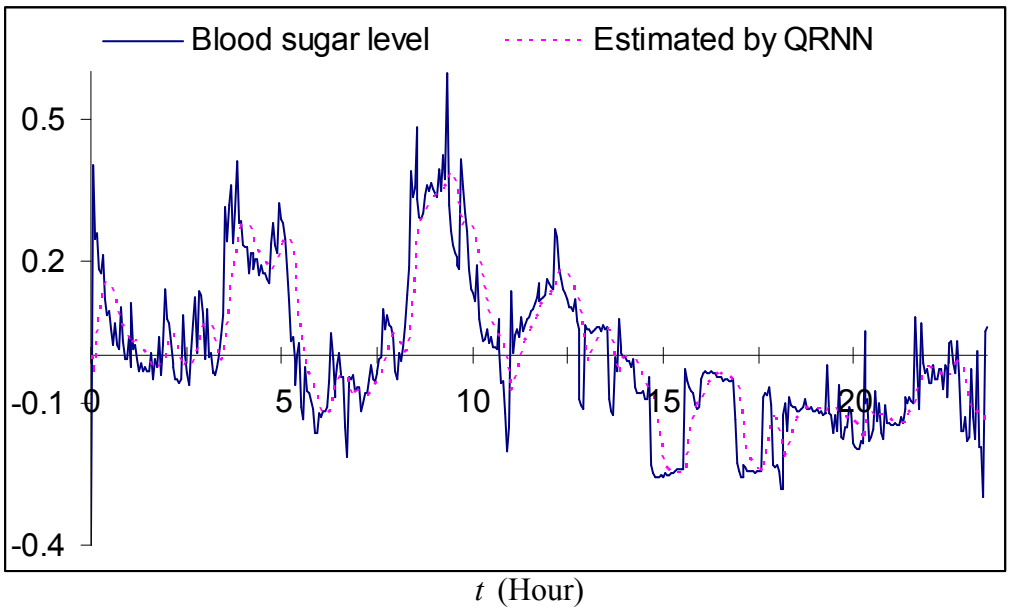



Figure-7.16: Blood sugar level actual and estimated data with calm wave
$L = 3,\ \Delta x = 0.1, N = 30,\ \sigma_0 = 0.6,\ \Delta t = 0.05\text{hour},\ m = 0.5$
$\xi = 1,\ \eta = 0.5,\ \hbar = 1,\ \kappa = 1$

With the calm wave the estimated signals along with the actual signals are shown in Figure-7.16. The signal is tracked with a RMSE of 0.0823. It can be observed from this

figure that the filter has no been able to estimate a good portion of signal using calm wave.

## 7.6 Conclusion

This chapter discusses the issues of application of the QRNN filter in a chaotic series. It has been observed from the results that the QRNN model has the ability to mimic the underlying dynamics of these series (or maps) and thus underscore the potential use of the model in a real world situation. The main issue here is the simplicity of the QRNN filter in terms of network architecture which consists of just one layer neurons. Values of the parameters involving SWE are easy to adjust in handling performance of the network. Mainly, two parameters (spread of the wave, and number of neuron in the network relating the length of the well) play the important part. It has been observed that if the sampling frequency increases then the performance of the filter improves in terms of RMSE.

# Chapter 8

# Overview and future work

## 8.1 Introduction

The aims and objectives of the thesis were outlined in Chapter-1. These are

a. Investigation of the SWE and filter development
b. Numerical procedure for the solution of the SWE
c. Design issues of the filter
d. Training the neural network and the learning process
e. Evolution of the *pdf* under quantum mechanical properties
f. Sensitivity of the parameters involved.

In light of the results presented in this thesis, these objectives are reviewed and discussed in the following sections.

## 8.2 Conclusions

a. It can be seen, from the earlier chapters, that the QRNN filter is based on
    i. the structure of the SWE, and
    ii. the solution of the SWE.

    The original form of the SWE is transformed into a nonlinear form by the structure of the potential function. This transformation allows for the

incorporation of the learning elements associated with the neural networks. Thus, it is possible to obtain a recurrent form which is able to learn and adapt. The potential function, which is composed of the incoming signal to be filtered, defines the force field within which the particles move and also defines the propagated wave function. This wave function is the solution of the SWE, and dictates the expected value of the filtered signal. The expected value is computed from the *pdf* which is the squared-modulus of the wave packets. The nature of the waves and their properties dictate the quality of the filtering (see Chapters-6–7 and later sections).

b. For the implementation of the QRNN filter, there are two issues to consider; a) the nature of the discretisation in both time and space (which are often coupled see Section-4.4), and b) the nature of the initial conditions. In this thesis, it has been shown that selection of an appropriate differencing scheme is required. This is important given that the SWE has to be discretised in both the time domain and the space domain. It is also important that the numerical solution be initiated with a set of conditions which are explicitly related to the discretisation scheme. Role of the initial condition was discussed in Chapter-5. The numerical strategy used is the implicit scheme (which is also known as the Crank-Nicolson scheme). This procedure has allowed the development of the

   - stable solution of the SWE and hence a stable filter
   - placement bounded on the parameters affecting the solution.

   It is to be noted that the scheme will ensure a stable solution if values of the relevant parameters (especially the discretisation step size both in space and time) are kept small. This is one of the critical issues in solving a PDE numerically. Moreover, this will also help maintain the Soliton and the unitary

properties which dictate the manner in which the wave propagates through the well.

c. It has been shown (see Chapter-5) that, there are two different ways the wave packets can be propagated in the well. This will results in either

   i. a normal wave, and

   ii. a calm wave.

   In the case of a normal wave, the wave packet moves in the well forwards and backwards considering immediate past estimate as a centre and its estimated error as a spread. In the case of a calm wave, a moving window of measurement information is generated where the wave packet perturbs from the centre of the well and the input signal is spread through both sides of the well symmetrically. Wave packets are propagated in the well maintaining the Soliton properties. Both of these techniques have been used and the results are tested in this thesis. Advantage of the calm wave over the normal wave is that it requires fewer neurons to learn the incoming signals and its performance is better (see Chapters-6 (Experiment-2), and Chapter–7).

d. Neural networks are trained to learn the behaviour of a system or process. Behaviour of the trained network is stored in the weights associated with the connections of the network. In the case of the QRNN filter, connection weights of the network are initialized with random numbers generated from a Gaussian distribution with mean zero and a standard deviation of one. The input signal excites an array of neurons spatially located along the $x-axis$ after being appropriately weighted. The excitation results in the evolution of a *pdf* and thus

the expected value. This in turn is used to update the weights using the Hebbian learning rule.

e. The QRNN filter is initialized with an initial wave packet. This initial wave packet acts as the initial condition for the solution of the SWE. Initial estimate of the state is taken to be zero. For each input to the network, the SWE transforms the input and yields a wave packet maintaining the Soliton property. Solitons are spatially localized waves travelling with constant speed and shapes. In this research, it is emerged that the transformation of the wave packet can be accomplished in two different ways in the well, namely a *normal wave* and a *calm wave*. These wave packets play critical role in the evolution of the *pdf*. In normal wave, the initial wave is propagated in the well by considering the initial estimate and initial spread as a centre and spread of the wave packet respectively. With the initial wave packet an estimate is made for the input signal and this estimate is used as a centre of the wave for the next input signal. The error is calculated by subtracting the current estimate from the new measurement and absolute value of the error is taken as a spread of the wave to measure the next input signal. After processing the network weights the error is passed through the network for the next state to be estimated.

The other way of propagating the wave in the well is introducing the calm wave. The concept of calm wave is that at the outset a wave packet is created with an initial value of zero and the wave remains calm and so the name. The number of wave particles in the calm wave is the same as the number of neurons in the network. As the wave propagates in the well the calm wave perturbs. The agitation here refers to the situation that when the pre-processed input signal reaches to the network, the SWE drives the dynamics of the input signal and

transforms it into a wave packet. Modulus-squared of this wave packet gives the distribution of the input signal or the *pdf*. The measurement signal receives a weighted estimate and the signal is stored in the calm wave in an iterative manner. The error is computed by subtracting the estimate from the current measurement. This error is then fed back to the network to update the network parameters. It is to be noted that the calm wave is not a wave, but it is a moving window of measurement information. The presence of this wave in the well indicates that the filter has enough knowledge about the signal and hence provides better results.

f. The QRNN filter has a number of parameters. These are – parameters of the SWE itself (mass of the quantum object $m$, the Planck constant $\hbar$), discretisation parameters (length of the well $L$, number of neurons in the network $N$, length of the time interval $\Delta t$, increment in space $\Delta x$), neural parameters (learning rate $\eta$, and potential field excitation $\xi$) and parameters due to initialization of the wave (wave momentum $\kappa$, spread of the wave $\sigma$). Discretisation parameters are related to each other and so the magnitudes of values of these parameters are determined by localizing the wave packet. It is important that the dicretisation step size is kept small since this affects the solution and hence the *pdf*. A large step size in both domains is inappropriate because of numerical reasons and also for poor solutions leading to an undesired approximation to the *pdf*. The SWE parameters appearing in the Hamiltonian matrix are selected in a way that they do not affect the solution space.

## 8.3 Scope for future development

Performance of the filter should be further tested where there is a true randomness in the signal. In this thesis, motion tracking was simplified however, both in terms of measuring the signal and the uncertainties. In this, it would be more realistic if the filter is tested in a real motion tracking situation with many tracking sensors.

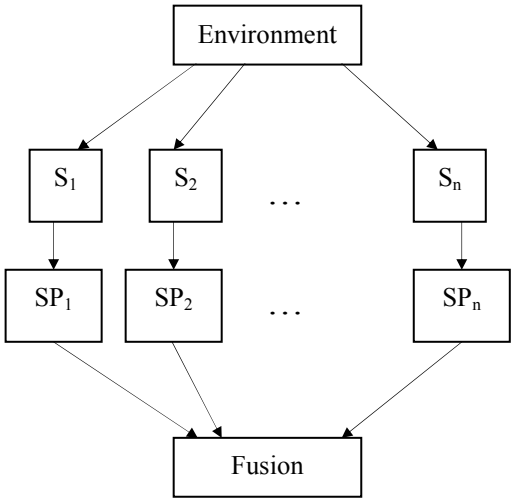



Figure-8.1 Multidimensional sensors fusion

Most modern systems these days consists of a network of sensors where sensors can be the same or different. For example, tracking the motion of a particle in a three-dimensional space using multiple detectors or a set of sensors functioning autonomously communicating with each other or the sensors are acting individually. Both scenarios require different strategies for fusing the data in the first instance. However, filtering strategies could be an extension of the filter discussed in this thesis. Even though this could be a multidimensional problem however, given the structure of the wave function and the initialization of the wave the basic numerical complexity is not heavy but requires some algebraic manipulation. It should be noted that despite such manipulation the behaviour of the wave would not exhibit Soliton behaviour. This would then mean the wave function has to be modified so that so called logarithmic form of the SWE is

obtained. However, in the single dimensional case discussed in this thesis, the result would be the same in both forms.

## 8.4 Conclusion

The QRNN filter is essentially a predictor-error-corrector loop which is similar to the Kalman filters. The difference here is that in the Kalman filters the probability density is assumed to be Gaussian so that it can completely be characterized by mean and covariance matrices. Computations of these quantities are accomplished from one time reference to the next using the Bayesian rule. In QRNN filter, the probability density is not assumed to be Gaussian rather it is given as the modulus-squared of the wave function of the SWE. The QRNN filter is data-driven, generic and can be implemented with or no a *priori* knowledge of the observed system. Solution technique developed using implicit scheme gives stable solution of the SWE. It has been shown that with a simple architecture the QRNN filter is able to capture the *pdf* information of the observed signal and produces desired results. In summary, it can be stated that objectives set out in the first chapter have been achieved.

# References

**Ahamed W. U and Kambhampati C. (2008),** *Stable Quantum Filters with Scattering Phenomena*, International Journal of Automation and Computing, Vol. 5, Issue 2, pp.132–137.

**Alex T. N. (2000),** Doctoral thesis: *Nonlinear Estimation and Modelling of Noisy Time Series by Dual Kalman Filtering Methods*, Oregon Graduate Institute of Science and Technology, USA.

**Anderson Brian D. O., Moore J. B. (1979),** *Optimal Filtering*, Prentice-Hall Information and System Sciences Series, (Ed.: Thomas Kailath), Prentice-Hall, Inc..

**Andrews A. (1968),** *A square root formulation of the Kalman covariance equations*, American Institute of Aeronautics and Astronautics Journal, Vol. 6, pp. 1165– 1166.

**Andrew H. J. (1970),** *Stochastic Processes and Filtering Theory*, Academic Press, Volume 64.

**Arulampalam S.M., Maskell S., Gordon N., and Clapp T. (2002),** *A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking*, IEEE Transactions on Signal Processing, Vol. 50, No. 2, pp. 174–188.

**Bair W. D. (1993),** *Tracking manoeuvring targets with multiple intermittent sensors: Does more data always mean better estimates*?, IEEE AES Vol. 32, pp.450 – 456.

**Bar-Shalom Y., Rong Li X., and Kirubarajan T. (2001),** *Estimation with Applications to Tracking and Navigation, Theory Algorithms and Software*, A Wiley-Interscience Publication, John Wiley & Sons, Inc..

**Behera L., and Indrani K. (2004),** *Adaptive control of robot manipulators using Quantum neural networks*, International Conference on Intelligent Signal Processing and Robotics, ISPR-04, Feb, Allahabad, India.

**Behera L., and Sundaram B. (2004),** *Stochastic filtering and speech enhancement using a recurrent quantum neural network*, Proceedings of International Conference on Intelligent Sensing and Information Processing, pp. 165-170.

**Behera L., Indrani K., and Elitzur A. C. (2005a),** *Chapter 9: Recurrent Quantum neural network and its Applications*, in the Emerging Physics of consciousness, Jack Tuszynski (Ed), Springer Verlag.

**Behera L., Indrani K., Elitzur A. C. (2005b),** *Quantum Brain: A recurrent quantum neural network model to describe eye tracking of moving targets*, Foundations of Physics Letters, Vol. 18, No. 4, 357-370.

**Bialynicki-Birula I., and Mycielski J. (1976)**, *Nonlinear Wave Mechanics*, Annals of Physics, Vol. 100, pp. 63 – 99.

**Boyd R. W. (1992),** *Nonlinear Optics*, Academic Press.

**Bucy R. S. (1970),** *Linear and nonlinear filtering*, Invited paper, Proceedings of the IEEE, Vol. 58, No. 6.

**Bucy R. S. and Joseph P. D. (1968),** *Filtering for stochastic processes with applications to guidance*, Interscience Publishers, A division of John Willey & Sons, New York.

**Capiński M. and Kopp E. (2004),** *Measure, Integral and Probability*, Springer-Verlag London Ltd.

**Carpenter J., Clifford P., Fearnhead P. (1999),** *An Improved Particle Filter for Nonlinear Problems*, IEE-Proceedings, Radar, Sonar and Navigation, Vol. 146, No. 2, pp. 2–7.

**Dawes R. L. (1989a),** *Adaptive Control and Stabilization with the Parametric Avalanche*, Technical Report MRC-ARDEC-89002, Martingale Research Corporation, Allen, TX..

**Dawes R. L. (1989b),** *Quantum Neurodynamics: A new Neural Network Paradigm for Knowledge Representation*, Cognition, and Control, Technical Report MRC-NSF-89003, Martingale Research Corporation, Allen, TX.

**Dawes R. L. (1989c),** *Quantum Neurodynamics and the Parametric Avalanche*, Technical Report MRC-NASA-89004, Martingale Research Corporation, Allen,TX.

**Dawes R. L. (1992),** *Quantum Neurodynamics: Neural Stochastic Filtering with the Schrodinger Equation*, Proc. IJCNN, Vol. I, pp. 133.

**Dawes R. L. (1993),** *Advances in the Theory of Quantum Neurodynamics, Rethinking Neural Networks: Quantum Fields and Biological data*, Editor Karl H. Pribram, Proceeding of the First Appalachian Conference on Behavioral Neurodynamics, Lawrence Erlbaum Associates Publishers, Hillsdale.

**Dorffner G. (1997),** *Neural Networks and a New Artificial Intelligence*, International Thomson Computer Press.

**Eric A. W. (1993)**, Doctoral thesis: *Finite Impulse Response Neural Networks with Applications in Time Series Prediction*, Stanford University.

**Evans G. Blackledge J., and Yardley P. (2000),** *Numerical Methods for Partial Differential Equations* (Springer Undergraduate Mathematics Series, Springer-Verlag London Ltd..

**Feynman R. (1986),** *Quantum mechanical computers*, Foundation of Physics, vol. 12, pp. 507–531.

**Flake G. W. (1998),** *The computational beauty of nature: Computer explorations of fractals, chaos, complex systems, and adaptation*, MIT Press.

**Garcia A L. (1994),** *Numerical Methods for Physics*, 2nd Ed., Prentice Hall.

**Goldberg A., Schey H. M., and Schwartz J. L. (1967),** *Computer-Generated Motion Pictures of One-Dimensional Quantum-Mechanical Transmission and Reflection Phenomena*, American Journal of Physics, Vol. 35, pp. 177-186.

**Haykin S. (1994),** *Neural Networks, A comprehensive Foundation*, 1st edition, Macmillan College Publishing Company Inc., New York.

**Haykin S. (2001),** *Kalman Filtering and Neural Networks*. John Wiley & Sons Inc., New York.

**Hénon M. (1976),** *A two-dimensional Mapping with a Strange Attractor*, Communications in Mathematical Physics, Vol. 50, pp. 67–77.

**Husmeier D. (1999),** *Neural Networks for Conditional Probability Estimation: forecasting beyond points prediction*, Springer-Verlag London Ltd..

**Husmeier D., and Taylor J. G. (1999),** *Predicting conditional probability densities of stationary stochastic time series*, Neural Networks, Vol. 10, No. 3, pp.479–497.

**Julier S.J. and Uhlmann J.K. (1995),** *A new approach to filtering nonlinear systems*, Proceedings of the American Control Conference, pp. 1628 – 1632.

**Julier S.J. and Uhlmann J.K. (1997),** *A New Extension of the Kalman Filter to Nonlinear System*, Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls.

**Kailath T. (1968),** *An innovations approach to least-squares estimation Part II: linear smoothing in additive white noise*, IEEE Transactions on Automatic Control, Vol. 13, Issue 6, pp. 655–660.

**Kailath T. (1970),** *The Innovations Approach to Detection and Estimation Theory*, Proceedings of the IEEE, Vol. 58, No. 5, pp. 680–695.

**Kailath T. (1984),** *State-space modelling: Square root algorithms*, System and Control Encyclopaedia (M. G. Singh, Ed.), Pergamon, Elmsford, New York.

**Kailath T. (1998),** *Detection of Stochastic Processes*, IEEE Transactions on Information Theory, Vol. 44, No. 6, pp. 2230–2259.

**Kalman R. E. (1960),** A New *Approach to Linear Filtering and Prediction Problems*, Transactions of the ASME–Journal of Basic Engineering, 82 (Series D) pp. 35-45.

**Koch C. and Segev I. (1989),** *Methods in Neuronal Modelling: From Synapses to Networks*, Computational Neuroscience, The MIT Press.

**Lee C.-S., Kuo Y.-H., and Yu P.-T. (1997),** *Weighted fuzzy mean filters for image processing*, Fuzzy Sets and System, Vol. 89, pp. 157–180.

**Leung C.-S., and Chan L.-W. (2003),** *Dual extended Kalman filtering in recurrent neural networks*, Neural Networks, Vol. 16, Issue 2, pp. 223-239.

**Lin T.-C. and Yu P.-T. (2004),** *Partition fuzzy median filter based on fuzzy rules for image restoration*, Fuzzy Sets and System, Vol. 147, pp. 75–97.

**Liu J.S., Chen R., Logvinenko T. (2001),** *A theoretical framework for sequential importance sampling and resampling*, Sequential Monte Carlo in Practice, Editors: Doucet A., de Freitas N., and Gordon N.J., Springer-Verlag.

**Lorenz E. N. (1963),** *Deterministic Nonperiodic Flow*, Journal of Atmospheric Sciences, Vol. 20, pp. 130–141.

**MacKay D. J. C. (1992),** *A practical Bayesian framework for back-propagation networks*, Neural Computation, Vol. 4, pp. 448–472.

**Mackey M. C., and Glass L. (1977),** *Oscillation and chaos in physiological control systems*, Science, Vol. 197, pp. 287-289.

**Magnus, N., Poulsen, N. K., Ravn, O. (2000),** *New developments in state estimation for nonlinear systems*, Automatica, Vol. 36, No. 11, pp. 1627–1638.

**Mandic D.P. and Chambers J. A. (2000),** A *normalised real time recurrent learning algorithm*, Signal Processing, Vol. 80, Issue 9, pp. 1909–1916.

**Maybank S. (1996),** *Finite-dimensional Filters*, Philosophical Transaction of the Royal Society of London Series A, Vol. 354, pp.1099–1123.

**Maybeck P. S. (1979),** *Stochastic models, estimation, and control*, Mathematics in Science and Engineering Series, Vol. 141–1, New York: Academic Press.

**Maybeck P. S. (1982),** *Stochastic models, estimation, and control*, Mathematics in Science and Engineering Series, Vol. 141–2, New York: Academic Press.

**McGee L. A. and Schmidt S. F. (1985),** *Discovery of the Kalman Filter as a Practical Tool for Aerospace and Industry*, National Aeronautics and Space Administration, Technical Memorandum 86847.

**Mehra R. K. (1971),** *A Comparison of Several Nonlinear Filters for Reentry Vehicle Tracking*. IEEE Transactions on Automatic Control, Vol. Ac-16, No. 4.

**Merwe R. van der, de Freitas J. F. G., Doucet D., and Wan E. A. (2000),** *The Unscented particle filter*, Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department.

**Mohinder S.G. and Angus P. A. (2001),** *Kalman Filtering: Theory and Practice*, 2nd ed., A Wiley Inter-science publication, John Wiley & Sons, Inc., New York.

**Nave C. R. (2006),** *HyperPhysics*, Department of Physics and Astronomy, Georgia State University, USA, Last visited: February 20, 2009, **URL** Reference: http://hyperphysics.phy-astr.gsu.edu/Hbase/quantum/schrcn.html#c1

**Orderud F. (2005),** *Comparison of Kalman Filter Estimation Approaches for State Space Models with Nonlinear Measurements*, Proceedings of Scandinavian Conference on Simulation and Modeling, SIMS-2005.

**Peleg Y., Pniini R., Zaarur E. (1998),** *Theory and Problems of Quantum Mechanics: Schaum's Outline*, McGraw-Hill.

**Penrose R. (1994),** *Shadows of the Mind*, *A search for the missing science of consciousness*, Oxford University Press, New York.

**Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P. (1992),** *Numerical Recipes in C: the Art of Scientific Computing*, Cambridge University Press.

**Puskorius G. V., and Feldkamp L. A. (1991),** *Decoupled extended Kalman filter training of feed forward layered networks*, Proceedings of the International Joint Conference of Neural Networks, Seattle, WA, Vol. 1, pp. 171–777.

**Rekleitis I. M. (2003),** *Cooperative Localization and Multi-Robot Exploration*, PhD Thesis, School of Computer Science, McGill University, Montreal, Canada.

**Rhodes I. (1971),** *A Tutorial Introduction to Estimation and Filtering*, IEEE Transactions on Automatic Control, Vol. Ac-16, No. 6., pp. 688 – 706.

**Robert J. F. (1971),** *Divergence of Kalman Filter*, IEEE Transactions on Automatic Control, Vol. Ac-16, No. 4.

**Sanjay G. and Zia R. K. P. (2001),** *Quantum Neural Networks*, Journal of Computer and System Science, Vol. 63, Issue 3, pp. 355–383.

**Schiff L. I. (1968),** *Quantum Mechanics*, McGraw-Hill Book Company (3rd edition).

**Schlee F. H., Standish C. J., and Toda N. F. (1967),** *Divergence in Kalman filter*, American Institute of Aeronautics and Astronautics Journal, Vol. 5, pp. 1114–1120.

**Schmidt S. F. (1981),** *The Kalman Filter: Its recognition and development for aerospace applications*, AIAA Journal Guidance, Control and Dynamics, Vol. 4, No. 1, pp. 4–7.

**Seborg D. E., Edgar T. F., and Mellichamp D. A. (1989)**, *Process Dynamics and Control*, John Wiley & Sons, Inc..

**Singhal S. and Wu L. (1989),** *Training multilayer perceptrons with the extended Kalman algorithm,* In Advances in Neural information Processing Systems 1, Morgan Kaufmann Publishers, San Francisco, CA, pp. 133-140.

**Smith G. L. and Schmidt S. F. (1961),** *The application of statistical filter theory to optimal trajectory determination on-board a circumlunar vehicle*, American Astronomical Society Meeting, pp. 61–92.

**Sorenson H. W. (1970),** *Least-square estimation: from Gauss to Kalman*, IEEE Spectrum, pp. 63–68.

**Stratonovich R. L. (1959),** *Optimum nonlinear systems which bring about a separation of a signal with constant parameters with noise*, Radiofizika, Vol. 2 Issue 6, pp. 829–901.

**Sun J. Q., Ma Z. Q., Hua W., and Qin M. Z. (2006),** *New conservation schemes for the nonlinear Schrödinger equation*, Applied Mathematics and Computation, Vol. 177, No. 1, pp. 446 – 451.

**Thornton C. L. (1976),** *Triangular Covariance Factorizations for Kalman Filtering*, PhD Thesis, School of Engineering, University of California at Los Angeles.

**Wan E. A., Merwe R. van der, and Nelson T. A., (2000),** *Dual estimation and the unscented transformation*, editors: S. A. Solla, T. K. Leen, and K. –R. Muller, Advances in Neural Information Processing Systems 12, Cambridge, MA: MIT Press, pp. 666–672.

**Wang L., Chen K., and Ong Y. S., (Eds.) (2005)**, Adaptive Identification of Chaotic Systems and Its Applications in Chaotic Communications, Lecture Notes in Computer Science, Springer-Verlag Berlin/Heidelberg, pp. 332–337.

# Bibliography

**Alexandr A., Ezhov, and Dan Ventura (2000),** *Quantum neural networks*, Department of Mathematics, Troitsk Institute of Innovation and Fusion Research 142092, Troisk, Moscow Region, Russia and Applied Research Laboratory, The Pennsylvania State University, University Park, PA 16802-5018, USA.

**Battin R. H. (1964),** *Astronautical Guidance*, McGraw-Hill, New York, pp. 338-339.

**Behera L., Sundaram B., Singhal G., and Agrawal M.,** *A recurrent quantum neural network model and its application to Stochastic filtering*, TNN#CE476Rev., Department of Electrical Engineering, Indian Institute of Technology, Kanpur, India.

**Behrman E. C., Chandrashekhar V., Wang Z., Belur C. K., Steck J. E., and Skinner S. R. (2003),** *A quantum neural network computes entanglement*, American Physical Society, Annual APS March Meeting.

**Behrman E. C., Nash L. R., Chandrashekhar V., Belur C. K., Steck J. E., and Skinner R. R. (2000),** *Simulations of quantum neural network computes entanglement*, Information Sciences, 128(3-4): 257–269.

**Behrman E. C., Niemel J., Steck J. E., and Skinner S. R. (1996),** *A Quantum Dot Neural Network*, Proceeding of the workshop on Physics and Computation, Boston, pp. 22–29.

**Benioff P. (1982),** *Quantum Mechanical Hamiltonian Models of Turing Machines*, Journal of Statistical Physics, Vol. 29, No. 3, pp 515–546.

**Billings S.A., and Zhu Q.M. (1994),** *A Structure Detection Algorithm For Nonlinear Dynamic Rational Models, International Journal Of Control*, Vol. 59, Issue 6, pp. 1439–1463.

**Bishop C. M. (1995),** *Neural Networks for Pattern Recognition*, Oxford University Press, New York.

**Brigo D., Hanzon B., and LeGland F. (1998),** *A differential geometric approach to nonlinear filtering: the projection filter*, IEEE Transactions on Automatic Control, Volume 43, Issue 2, pp. 247-252.

**Brooks M (Ed) (1999),** *Quantum Computing and Communications*, Springer-Verlag, Berlin/Heidelberg.

**Bullough R. K. (1998),** *Solitons*, Nonlinear Physics for Beginners by Lui Lam, World Scientific Publishing Co. Pte Ltd..

**Chen S., and Billings S.A. (1992),** *Neural Networks for Nonlinear Dynamic System Modeling And Identification*, International Journal of Control, Vol. 56, Issue 2, pp. 319–346.

**Chen S., Billings S.A., and Grant P.M. (1992),** *Recursive Hybrid Algorithm For Nonlinear-System Identification Using Radial Basis Function Networks*, International Journal Of Control, Vol. 55, Issue 5, pp. 1051–1070.

**Chen S., Billings S.A., Cowan C.F.N., and Grant P.M. (1990),** *Nonlinear-Systems Identification Using Radial Basis Functions*, International Journal Of Systems Science, Vol. 21, Issue 12, pp. 2513–2539.

**Chen S., Billings S.A., Cowan C.F.N., and Grant P.M. (1990),** *Practical Identification Of Narmax Models Using Radial Basis Functions*, International Journal Of Control, Vol. 52, Issue 6, pp. 1327–1350.

**Dan Ventura and Tony Martinez,** *Quantum Associative Memory with Exponential Capacity*, Neural Network and Machine Learning Laboratory, Department of Computer Science, Brigham Yong University, Provo, UT 84602.

**Deutsch D. (1985),** *Quantum Theory, the Church-Turing principles and the universal quantum computer*, Proceeding of the Royal Society of London A, Vol. 400, pp. 97–117.

**Deutsch D. (1989),** *Quantum Computational Networks*, Journal Proceedings of the Royal Society of London Series A, Mathematical and Physical Sciences, Vol. 425, No. 1868, pp. 73-90.

**Galbraith I., Ching Y. S., and Abraham E. (1984),** *Tow-dimensional time-dependent quantum-mechanical scattering event*, American Journal of Physics, Vol. 52, No. 1, pp. 60–68.

**Gao F.X.Y., and Snelgrove W.M. (1994),** *Adaptive nonlinear recursive state-space filters*, IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Vol. 41, Issue 11, pp. 760-764.

**Gelb A., Kasper J. F. Jr., Nash R. A. Jr., Price C. F., and Sutherland A. A. Jr. (1984),** *Applied Optimal Estimation*, The M.I.T. Press.

**Goowin G.C., and Sin K.S. (1984),** *Adaptive filtering Prediction and Control*, Information and System Science Series, Editor: Thomas Kailath, Prentice-Hall Inc., Englewood Cliffs, New Jersey.

**Jung, L. (1979),** *Asymptotic Behavior of the Extended Kalman Filter as Parameter Estimator for Linear Systems*, IEEE Transactions on Automatic Control, Vol. 24, No. 1, pp. 36–50.

**Kallianpur G. (1980),** *Stochastic Filtering Theory*. Springer-Verlag New York Inc..

**Montgomery D. C. and Contreras L. E. (1977),** *A Note on Forecasting with Adaptive Filtering*, Operational Research Quarterly, Vol. 28, No. 1, pp. 87-91, Part-1.

**Narayanan A., and Menneer T. (1995)**, *Quantum-inspired Neural Networks*, Technical Report R329, Department of Computer Science, University of Exeter, Exeter, UK.

**Narayanan A., and Menneer T. (2000),** *Quantum artificial neural network architectures and components*, Information Sciences, (128), pp. 231-255.

**Poor H. V. (1998),** An Introduction to Signal Detection and Estimation, Second edition, Springer Texts in Electrical Engineering.

**Pribram K. (1991),** Brain and Perceptron, Holonomy and structure in Figural Processing, Erlbaum, Hillsdale, NJ.

**Robert L. W. (1982),** *Computer graphics for solutions of Time-dependent Schrödinger equations*, American Journal of Physics, Vol. 50, October 1982 pp. 902-906.

**Schouten J.C., Takens F., van-den Bleek Cor.M. (1994),** *Estimation of the dimension of a noisy attractor*, Physical Review E, The American Physical Society, Vol. 50, Issue 3, pp. 1851–1861.

**Wenzel T., Burnham K., Blundell M., and Williams R. (2006),** *Dual extended Kalman filter for vehicle state and parameter estimation*, Vehicle System Dynamics, Vol. 44, Issue 2, pp. 153-171.

**Worden K., Stansby P. K., Tomlinson G. R., and Billings S. A. (1994),** *Identification Of Nonlinear-Wave Forces*, Journal Of Fluids And Structures, Vol. 8, Issue 1, pp. 19–71.

**Zhu Y., and Li X. R. (2003),** *Unified Fusion Rules for Multi-sensor Multi-hypothesis Network Decision Systems*, IEEE Transaction on Systems, Man, and Cybernetics – Part A: Systems and Humans, Vol. 33, No. 4.

# Appendix-A

## I. Normalization of the plane wave

Considering the wave propagating along $x-axis$ we have,

$$\psi(x,t)=\int_{-\infty}^{+\infty}a(\kappa)\exp[i(\kappa x-\omega t)]d\kappa$$

where the wave number is given by

$$a(\kappa)=\frac{A}{\sigma_\kappa\sqrt{2\pi}}\exp\left[-\frac{(\kappa-\kappa_0)^2}{2\sigma_\kappa^2}\right]$$

which has a mean wave momentum $\kappa_0$ and variance $\sigma_\kappa$. When $t=0$, the single dimensional wave function stands at,

$$\psi(x,0)=\frac{A}{\sigma_\kappa\sqrt{2\pi}}\int_{-\infty}^{+\infty}e^{i\kappa x}\times\exp\left[-\frac{(\kappa-\kappa_0)^2}{2\sigma_\kappa^2}\right]d\kappa$$

$$=\frac{Ae^{i\kappa_o x}}{\sigma_\kappa\sqrt{2\pi}}\int_{-\infty}^{+\infty}e^{i\hat{\kappa}x}\times\exp\left[-\frac{\hat{\kappa}^2}{2\sigma_\kappa^2}\right]d\hat{\kappa}\quad\text{where }\hat{\kappa}=\kappa-\kappa_0$$

Putting, $u=\dfrac{\hat{k}}{\sqrt{2}\sigma_k}$ which implies $d\hat{k}=\sqrt{2}\sigma_k du$ gives

$$\psi(x,0)=\frac{Ae^{ik_o x}}{\sigma_k\sqrt{2\pi}}\int_{-\infty}^{+\infty}e^{i\sqrt{2}\sigma_k ux}\times e^{-u^2}du$$

Completing the square and integrating we have,

$$\psi(x,0)=Ae^{ik_o x}\exp\left[-\frac{x^2}{2\sigma_x^2}\right]dx \tag{A1}$$

with $\sigma_x \sigma_\kappa = 1,$ known as uncertainty relation. Using the normalization condition that the wave particle be somewhere at $t = 0$ with the probability one, that is, $\int_{-\infty}^{+\infty} \psi^* \psi dx = 1$, therefore, equation (A1) can be written as

$$A^2 \int_{-\infty}^{+\infty} \exp\left[ -\frac{x^2}{\sigma_x^2} \right] dx = 1$$

which gives

$$A = \frac{1}{\sigma_x \sqrt{\pi}}$$

Thus the fully described normalized single dimensional wave function with mean zero and standard deviation $\sigma_x$ is given by

$$\psi(x,0) = \frac{1}{\sqrt{\sigma_x \sqrt{\pi}}} e^{ik \cdot x} \times e^{-\left( \frac{x^2}{2\sigma_x^2} \right)}$$

# Appendix-B

## Experiment-1

### I. Normal wave

| Signal Equation | Root-mean-square-error (RMSE) | | | |
|---|---|---|---|---|
| | 6 dB | 10 dB | 20dB | 30dB |
| Sinusoidal | 0.23170 | 0.2221 | 0.2156 | 0.2156 |
| Shifted sinusoidal | 0.22528 | 0.2167 | 0.2130 | 0.2146 |
| Amplitude modulated | 0.14754 | 0.1428 | 0.1392 | 0.1385 |
| Mixed sinusoidal | 0.14812 | 0.1382 | 0.1325 | 0.132 |

Table-6.1a: Sampling frequency: 10 samples per cycle ( $\Delta t = 0.1$)

| Signal Equation | Root-mean-square-error (RMSE) | | | |
|---|---|---|---|---|
| | 6 dB | 10 dB | 20dB | 30dB |
| Sinusoidal | 0.13998 | 0.10599 | 0.05539 | 0.03713 |
| Shifted sinusoidal | 0.14103 | 0.10702 | 0.05595 | 0.03733 |
| Amplitude modulated | 0.05930 | 0.04310 | 0.02226 | 0.01592 |
| Mixed sinusoidal | 0.11865 | 0.08936 | 0.04518 | 0.02965 |

Table-6.1b: Sampling frequency: 1000 samples per cycle ( $\Delta t = 0.001$)

For both tables: Length of the well L = 2, number of neuron N = 200, $\Delta x = 0.01$, $\sigma = 0.6$, Potential field excitation $\xi = 1$, learning rate $\eta = 0.5$, m = 0.5, The universal constant $\hbar = 1$, wave momentum $\kappa = 1$.

## II. Calm wave

| Signal Equation | Root-mean-square-error (RMSE) | | | |
|---|---|---|---|---|
| | 6 dB | 10 dB | 20dB | 30dB |
| Sinusoidal | 0.11908 | 0.09392 | 0.06366 | 0.05750 |
| Shifted sinusoidal | 0.12596 | 0.09810 | 0.06420 | 0.05735 |
| Amplitude modulated | 0.05175 | 0.04025 | 0.02984 | 0.02827 |
| Mixed sinusoidal | 0.08399 | 0.06009 | 0.03571 | 0.03131 |

Table-6.2a: Sampling frequency: 10 samples per cycle per second ( $\Delta t = 0.1$)

| Signal Equation | Root-mean-square-error (RMSE) | | | |
|---|---|---|---|---|
| | 6 dB | 10 dB | 20dB | 30dB |
| Sinusoidal | 0.11610 | 0.08274 | 0.03131 | 0.01289 |
| Shifted sinusoidal | 0.11741 | 0.08400 | 0.03204 | 0.01306 |
| Amplitude modulated | 0.04050 | 0.02743 | 0.01031 | 0.00555 |
| Mixed sinusoidal | 0.09485 | 0.06767 | 0.02519 | 0.00920 |

Table-6.2b: Sampling frequency: 1000 samples per cycle per second ( $\Delta t = 0.001$)

For both tables: Length of the well L = 20, number of neuron N = 200, $\Delta x = 0.1$, $\sigma = 0.6$, Potential field excitation $\xi = 1$, learning rate $\eta = 0.5$, m = 0.5, The universal constant $\hbar = 1$, wave momentum $\kappa = 1$.

# Experiment-2

## I. Normal wave

| Noise Strength/ No. of neurons | Root-mean-square-error (RMSE) | | | | | | |
|---|---|---|---|---|---|---|---|
| | 100 | 150 | 200 | 250 | 300 | 350 | 400 |
| 6dB | 0.2455 | 0.2317 | 0.233 | 0.2331 | 0.2331 | 0.2331 | 0.2331 |
| 10dB | 0.2338 | 0.2223 | 0.2258 | 0.2261 | 0.2261 | 0.2261 | 0.2261 |
| 20dB | 0.22 | 0.2161 | 0.2247 | 0.2255 | 0.2255 | 0.2255 | 0.2255 |
| 30dB | 0.2159 | 0.2162 | 0.2274 | 0.2285 | 0.2285 | 0.2285 | 0.2285 |

Table-6.17a: Sinusoidal signal: 10 samples per cycle per second ($\Delta t = 0.1$)

| Noise Strength/ No. of neurons | Root-mean-square-error (RMSE) | | | | | | |
|---|---|---|---|---|---|---|---|
| | 100 | 150 | 200 | 250 | 300 | 350 | 400 |
| 6dB | 0.1877 | 0.154 | 0.1374 | 0.1308 | 0.1288 | 0.1284 | 0.1283 |
| 10dB | 0.1655 | 0.1249 | 0.1037 | 0.0948 | 0.092 | 0.0914 | 0.0913 |
| 20dB | 0.1347 | 0.0844 | 0.0554 | 0.0426 | 0.0387 | 0.0379 | 0.0377 |
| 30dB | 0.1246 | 0.0714 | 0.0401 | 0.0273 | 0.0243 | 0.0239 | 0.0238 |

Table-6.17b: Sinusoidal signal: 100 samples per cycle per second ($\Delta t = 0.01$)

| Noise Strength/ No. of neurons | Root-mean-square-error (RMSE) | | | | | | |
|---|---|---|---|---|---|---|---|
| | 100 | 150 | 200 | 250 | 300 | 350 | 400 |
| 6dB | 0.1903 | 0.1565 | 0.1398 | 0.1332 | 0.1312 | 0.1308 | 0.1308 |
| 10dB | 0.1679 | 0.1272 | 0.1057 | 0.0967 | 0.0939 | 0.0933 | 0.0932 |
| 20dB | 0.1361 | 0.0851 | 0.0548 | 0.0405 | 0.0358 | 0.0347 | 0.0345 |
| 30dB | 0.125 | 0.0705 | 0.0365 | 0.0193 | 0.0132 | 0.0118 | 0.0116 |

Table-6.17c: Sinusoidal signal: 1000 samples per cycle per second ($\Delta t = 0.001$)

Length of the well $1 \le L \le 4$, $\Delta x = 0.01$, $\sigma_0 = 0.6$, Potential field excitation $\xi = 1$, learning rate $\eta = 0.5$, m = 0.5, The universal constant $\hbar = 1$, wave momentum $\kappa = 1$.

## II. Calm wave

| Noise Strength /No. of neurons | Root-mean-square-error (RMSE) | | | | | | |
|---|---|---|---|---|---|---|---|
| | 20 | 30 | 40 | 50 | 60 | 70 | 80 |
| 6dB | 0.254 | 0.1785 | 0.1892 | 0.1939 | 0.1942 | 0.1942 | 0.1942 |
| 10dB | 0.2418 | 0.1623 | 0.1803 | 0.1869 | 0.1872 | 0.1873 | 0.1873 |
| 20dB | 0.2247 | 0.143 | 0.1769 | 0.1873 | 0.1879 | 0.188 | 0.1879 |
| 30dB | 0.2185 | 0.1375 | 0.1789 | 0.1911 | 0.1919 | 0.1919 | 0.1919 |

Table-6.18a: Sinusoidal signal: 10 samples per cycle per second ($\Delta t = 0.1$)

| Noise Strength/ No. of neurons | Root-mean-square-error (RMSE) | | | | | | |
|---|---|---|---|---|---|---|---|
| | 20 | 30 | 40 | 50 | 60 | 70 | 80 |
| 6dB | 0.1376 | 0.1315 | 0.1312 | 0.1312 | 0.1312 | 0.1312 | 0.1312 |
| 10dB | 0.1147 | 0.1086 | 0.1083 | 0.1083 | 0.1083 | 0.1083 | 0.1083 |
| 20dB | 0.0881 | 0.0869 | 0.0871 | 0.0871 | 0.0871 | 0.0871 | 0.0871 |
| 30dB | 0.0821 | 0.0851 | 0.0858 | 0.0858 | 0.0858 | 0.0858 | 0.0858 |

Table-6.18b: Sinusoidal signal: 100 samples per cycle per second ($\Delta t = 0.01$)

| Noise Strength /No. of neurons | Root-mean-square-error (RMSE) | | | | | | |
|---|---|---|---|---|---|---|---|
| | 20 | 30 | 40 | 50 | 60 | 70 | 80 |
| 6dB | 0.1868 | 0.1544 | 0.1406 | 0.136 | 0.1349 | 0.1348 | 0.1347 |
| 10dB | 0.168 | 0.1326 | 0.1178 | 0.1131 | 0.1119 | 0.1118 | 0.1117 |
| 20dB | 0.1408 | 0.1032 | 0.0905 | 0.0884 | 0.0884 | 0.0885 | 0.0885 |
| 30dB | 0.131 | 0.0936 | 0.0838 | 0.0844 | 0.0855 | 0.0859 | 0.086 |

Table-6.18c: Sinusoidal signal: 1000 samples per cycle per second ($\Delta t = 0.001$)

Length of the well $2 \le L \le 8$, $\Delta x = 0.1$, $\sigma_0 = 0.6$, Potential field excitation $\xi = 1$, learning rate $\eta = 0.5$, m = 0.5, The universal constant $\hbar = 1$, wave momentum $\kappa = 1$.