PARAMETER ESTIMATION IN HYBRID DYNAMICAL SYSTEMS WITH APPLICATION TO NEURONAL MODELS

by

Anish Mitra

A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Electrical and Computer Engineering

Committee:

______________________	Dr. Andre Manitius, Dissertation Director
______________________	Dr. Timothy Sauer, Committee Member
______________________	Dr. Janos Gertler, Committee Member
______________________	Dr. Gerald Cook, Committee Member
______________________	Dr. Andre Manitius, Department Chair
______________________	Dr. Kenneth Ball, Dean, The Volgenau School of Engineering
Date: ______________________	Summer Semester 2014 George Mason University Fairfax, VA

Parameter Estimation in Hybrid Dynamical Systems
with application to Neuronal Models

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Anish Mitra
Master of Science
George Mason University, 2011
Bachelor of Engineering
Birla Institute of Technology, 2007

Director: Dr. Andre Manitius, Professor
Department of Electrical and Computer Engineering

Summer Semester 2014
George Mason University
Fairfax, VA

Dedication

I dedicate this dissertation to the memory of my grandfathers, Dr. Nripendra Lal Mitra and Dr. Mahendra Prasad, who have been my inspiration and to my parents, Abhijit and Sujata Mitra, who have always supported me unconditionally.

Acknowledgments

First of all I would like to thank my advisor and dissertation director, Dr. Andre Manitius. He has been present whenever I needed help, for guidance on topics related to my doctoral research as well as graduate school work in general. As an advisor, teacher and Chair of the ECE department he has made sure that my graduate student career has been without any problems. I am grateful to him for his significant role in this dissertation.

The encouragement and insight on neuron models provided by Dr. Tim Sauer was very valuable. It made me believe that I was on the correct path. His inputs played a crucial role in the selection of the topic of my dissertation. I do not think it would have been possible to venture down the path of neuron models without his help.

I thank Dr. Janos Gertler for the beautifully taught courses on Systems Identification and Diagnostics as well as our discussions on these topics. They always motivated me to think and apply these methods to new applications. I believe that the manner of his teaching was extremely helpful in providing me with the fundamentals required to develop algorithms in the areas of parameter estimation for dynamic systems.

Dr. Cook has been like a mentor to me during my time at GMU. He has always been available, for comments on my research, as well as help regarding my teaching responsibilities and the control system courses. The interaction with him has taught me a lot, not just about technical material, but also about professional conduct. I am grateful for his assistance in my doctoral research and graduate courses.

I would also like to acknowledge my colleagues and fellow graduate students for their support. Hossein Ghaffari Nik was my first friend in the Electrical and Computer Engineering department and has been a big help during the courses that we took together. Franz Hamilton has always been available for discussions related to my research and his comments have been particulary helpful whenever I have faced difficult issues.

This dissertation would not have been possible without the role that my entire family has played, not just during my graduate studies, but in teaching and inspiring me to be the person I am today. I would also like to specially mention my uncle and aunt, Arindam and Sunanda Mitra, who have been there to discuss my problems and advise me accordingly.

My wife and best friend, Nalini, has been my biggest critic, reviewer and motivator. She has always been present to help me with my research and put my work ahead of everything else. I thank her for being a constant presence in my life and for all her efforts to make sure that I finished my dissertation in time.

Finally, I owe the success of this dissertation to my parents, Abhijit and Sujata Mitra, in more ways than one. They are the reason that I have continued to believe in myself despite moments of failure and frustration. I thank them for instilling values in me that have been the foundation of the success I have found in life, for never doubting me and what I can achieve while at the same time motivating me to reach my goals.

Table of Contents

List of Tables

List of Figures

Abstract

PARAMETER ESTIMATION IN HYBRID DYNAMICAL SYSTEMS
WITH APPLICATION TO NEURONAL MODELS

Anish Mitra, PhD

George Mason University, 2014

Dissertation Director: Dr. Andre Manitius

Analysis and recreation of brain dynamics has been identified as one of the greatest scientific challenge of this century. Detection of electrical impulses in the brain was the first step towards understanding how it functions. An interconnected network of neurons relay information and communicate with one another through these impulses also referred to as ‘spikes’. Knowledge of the spiking behavior and connectivity in different regions of the brain will help in the diagnosis and treatment of neurological disorders such as epilepsy and Parkinsons disease. There are also efforts to develop intelligent algorithms inspired by the functioning of the brain and build efficient processing and computing units.

Mathematical modeling of neuronal spiking and the different observed phenomenon has helped researchers study the properties of neurons and understand their physiological attributes. Scientists have been able to use these models to develop input-output characteristics. This has enabled us to draw conclusions about brain functionality. Mathematical models can also be used to estimate parameters and neuronal connections that allow the prediction of neuronal spiking activity. Over the last few decades better and more efficient models have been designed. Coupled with the improvement in computing resources, the objective of building a brain simulator is slowly turning into reality. As models are now able to successfully reproduce cognitive tasks, there is the need of algorithms that can use such models to interpret and analyze biological neurons and connectivity in the brain.

The theory of nonlinear dynamical systems plays an important role in reconstructing brain dynamics. There are various dynamic models, linear and nonlinear, that attempt to generate the neuronal spiking patterns observed experimentally. More recently, discontinuous resets have been introduced in otherwise continuous time models to expand the range of the spiking phenomenon that the model can produce. This also reduces the computational complexity of the system allowing for simulation of larger networks.

This doctoral dissertation focuses on two such hybrid neuron models and describes new methods of system identification based on observations of the neuron. Parameter estimation in a single neuron is achieved by designing a novel spike train comparison technique. The optimal parameters of the model are the computed by using the steepest descent method to locate the minima of the squared error between the model and the experimental spikes. Results show that the optimized model is then able to predict the spiking of the biological neuron. This is indicative of not only the accuracy of the model but also the success of the optimization algorithm that enables the automatic fitting of models to experimental data. For networks of neurons, the least squares estimation technique is implemented to identify the connections. This method proves to be robust and accurate in estimation the connectivity even in the presence of significant model error and observation noise. The sensitivity and specificity measures show that the synaptic connections from excitatory as well as inhibitory neurons are identified correctly. The evolution of the neuronal connectivity is also observed by tracking the generation of new synapses and degeneration of existing ones.

Chapter 1: Introduction

The presence of electrical signals in the nervous system was first discovered in 1790 by Luis Galvani [2]. It was almost a 100 years later that the 'neuron theory' was developed and the presence of basic functional units in the nervous system was described.The historical relevance of this theory is summarized in [3]. The twentieth century saw research and development in the measurement technqiues that allowed scientists to record the electrical activity from neurons and neuronal populations. Beginning from the first account of being able to measure electrical discharges in single nerve fibres [4], to patch clamp methods [5], to finally the advancement of multi-unit recording [6], the different types of neural signals and how they are responsible in the functioning of the nervous system are still being discovered. [7] mentions an observation, similar to Moore's Law, that in the past 5 decades the number of simultaneously recorded neurons has doubled every 7 years. Brain-Machine interfaces (BMI) [8] use the electrical activity measured from groups of neurons to interact control devices and processes such as prosthetic limbs [9] and synthetic telepathy [10].

As different types neural signals are being discovered and detected by new experimental techniques, one of the objectives has always been to mathematically model these physiological phenomenon. Over the last 60 years there has been extensive research on nonlinear dynamical systems and their ability to model 'real life' phenomena. The Hodgkin-Huxley model [11] is a bio-plausible model that has equations and parameters relating to physiological variables. To address the issue of computational complexity a number of non bio-plausible models were developed such as the Fitzhugh-Nagumo (FHN) ([12, 13]). [14] compares several dynamical models with respect to biological plausibility and computational efficiency. The Izhikevich model [15] performs well in comparison to other models. Another recent model is the Augmented multiscale adaptive threshold (AugMAT) model

([16, 17]). Gernster and Naud discuss several models in [1] and address the desirables of a good model. According to the results of a competition organized by International Neuroinformatics Coordinating Facility (INCF), it was very tough to tune the parameters of a bio-plausible model. Simple models with fewer parameters are being preferred because they are amenable to mathematical analysis. Another class of models present in literature are those based on stochastic processes such as renewal theory [18] or Markov chain processes [19]. As mentioned previously, the multi-unit recording and observation of groups of neurons has proved the experimental data to build neuronal network models. Measurement of the neural signals that are transmitted from one neuron to another has also assisted in modeling the coupling of the neurons which is a crucial component in network models. Simple neuron models and an increase in computational resources has led to the simulation of a million neurons with more than half a billion connections [20].

1.1 Motivation

Mathematical models have historically been used to simulate physical phenonmenon. Applications of such models range from explaining and analyzing the phenomenon, making predictions to making decisions based on observations. While neuron models, single and network, have been in existence since the 1950s, there is the need of techniques that can analyze such models based on observations from biological neurons. Parameter estimation is one such analysis that can provide more information about neurons and the neural communication that leads to the functioning of the brain and nervous system. Models with correctly identified parameters can be efficient predictors of the biological neuron and can be used to diagnose any changes in neuronal dynamics or to detect abnormalities in the neural signal. Parameters such as the connections between a pair of neurons have physiological significance. Accurate estimation of such parameters will lead to the mapping of the functional network of the brain and help in the study of the dynamics of connectivity.

Parameter estimation in neuronal models remains a developing research field due to the presence of new and improved models and detection of complex neural signals. The implementation of existing methods for models of different classes and types is difficult. One such class of models that has gained considerable attention in the last decade is continuous dynamical systems with a discontinuous reset based on a pre-defined condition. Such systems are also known as hybrid dynamical models [21]. The concept of such models is not new, and has been developed for applications in areas such as power systems and power electronics [22]. However, the use of such systems in modeling neurons is much more recent [23]. They are able to satisfy the requirements of a single neuron model, while being computationally inexpensive at the same time. This makes them useful for analysis as they can be used to construct large networks of data, or to simulate neural activity for a long period of time. However, the hybrid nature of such models presents difficulties when trying to implement existing methods of model analysis. Hence the study of such models, specifically parameter estimation, requires new methods that are able to successfully navigate the discontinuity. The objective of this dissertation is to define, discuss and implement new techniques of parameter identification for neuronal models based on hybrid dynamical systems.

1.2 Contribution

The next chapters of this dissertation are organized as the technical preliminaries, optimization of single neuron models, identification of connections in a neuronal network and discussion of the techniques developed and the results achieved.

1.2.1 Single Neuron Model

Chapter 3 discusses a new technique of fitting a mathematical model to an experimental spike train. The objective is to estimate the optimal parameters of a pre-defined model based on the time of spikes of the biological neuron. The application and relevance of the problem is discussed, before reviewing the existing techniques present in current literature and their disadvantages when applied to the mathematical model in consideration. The criteria for model selection was that it should be a hybrid dynamical system that is able to reproduce the range of spiking behavior observed in a biological neuron. The model chosen for the work related to this chapter is based on previous published results of model fitting so that the accuracy of the proposed technique could be evaluated.

The new technique, consisting of the formulation an error function and the derivation of the gradient based optimization method, is explained in detail. The results are presented based on synthetic data, generated by the model itself, as well as experimental spikes measured from a biological neuron.

1.2.2 Neuronal Network

It is well documented that a large number of interconnected neurons form the nervous system. The nature of the connectivity is of great interest since specifically studying a single neuron does not give us information about population dynamics. The communication between neurons and the transfer of information has been extensively researched and single neuron models have been extended to build simulators of groups of neurons. The model selected in Chapter 4 is based on the large scale hybrid dynamical model of the mammalian thalamocortical systems described in [20]. A systems identification approach is developed to estimate the connectivity of a neuronal network. The chapter discusses the formulation of this novel technique in detail, with emphasis on the derivation based on the model selected.

Results show that the method is accurately able to identify the connections in medium scaled networks of 50 neurons. The method is also used to estimate the changes in the connectivity in the presence of model error and observation noise, demonstrating the robustness of such an approach.

Chapter 2: Technical Background

2.1 Nonlinear dynamic systems

The use of mathematical modeling to describe, analyze and predict physical phenomenon is well documented. The advancement of technology has ensured that we can observe newer and more relevant events and use the observations to generate models. A mathematical model is a simplified version of the real world that employs the tools of mathematics - algebraic equations, probability, statistics, graph theory, etc. One of the different classes of mathematical models available is Dynamical Systems. Developed in the first half of the twentieth century it is used to describe the time dependence of a point in multi-dimensional space. Applications of such models include fluid dynamics, chemical reactions, analysis of cardiac rythms and neuronal spiking as well as description of joint angles in human motion.

In [24] differential equations are used to formally describe dynamic system theory. Typically the independent variable of the equations is time. The basic first order dynamic system contains a linear equation that is used to define the rate of change of the state of the point. Mathematically this can be written as

$$\frac{d\boldsymbol{x}(t)}{dt} = A\boldsymbol{x}(t) \tag{2.1}$$

The variable $\boldsymbol{x}(t)$ represents a vector in n-dimensional space at the time instance t. It is also referred to as the state of the system, or system state. The $n \times n$ matrix A defines the temporal 'dynamics' of the vector $\boldsymbol{x}$. The linear equation is unable to reproduce complex phenomenon such as autonomous oscillations. To achieve this the rate of change is defined using a nonlinear equation.

$$\frac{d\boldsymbol{x}(t)}{dt} = f(\boldsymbol{x}(t)) \tag{2.2}$$

The function $f(.)$ contains nonlinear terms of $\boldsymbol{x}(t)$. The above equation forms the basis of first order nonlinear dynamic models (in continuous time). The exact form of the equation will govern the dynamics of the point. Often the equation is parameterized and different patterns of the same phenomenon (e.g. oscillatory patterns) can be reproduced by changing the parameters of the equation. We illustrate this using a well known example of an oscillatory system.

2.1.1 Van der Pol equation

The Van der Pol Oscillator [25] is a non-conservative oscillator with nonlinear damping. It's dynamics are described by a second order nonlinear differential equation.

$$\frac{d^2y(t)}{dt^2} - \mu[1 - y(t)^2]\frac{dy(t)}{dt} + y(t) = 0 \tag{2.3}$$

This can be redefined as a two dimensional first order nonlinear dynamic system by using the substitutions $x_1(t) = y(t)$ and $x_2(t) = \frac{dy(t)}{dt}$.

$$\begin{aligned} \frac{dx_1(t)}{dt} &= x_2(t) \\ \frac{dx_2(t)}{dt} &= \mu[1 - x_1(t)^2]x_2(t) - x_1(t) \end{aligned} \tag{2.4}$$

As the name suggests, the Van der Pol Oscillator reproduces an oscillating response whose dynamics and frequency depends on the initial conditions and the parameter μ. Fig. 2.1 shows the responses for several parameter values. This particular aspect of nonlinear dynamic systems makes it applicable to phenomenon where the dynamics are non-deterministic but limited to a pre-defined range. The differential equations used in most practical applications do not have a closed-form solution and need to be solved numerically.

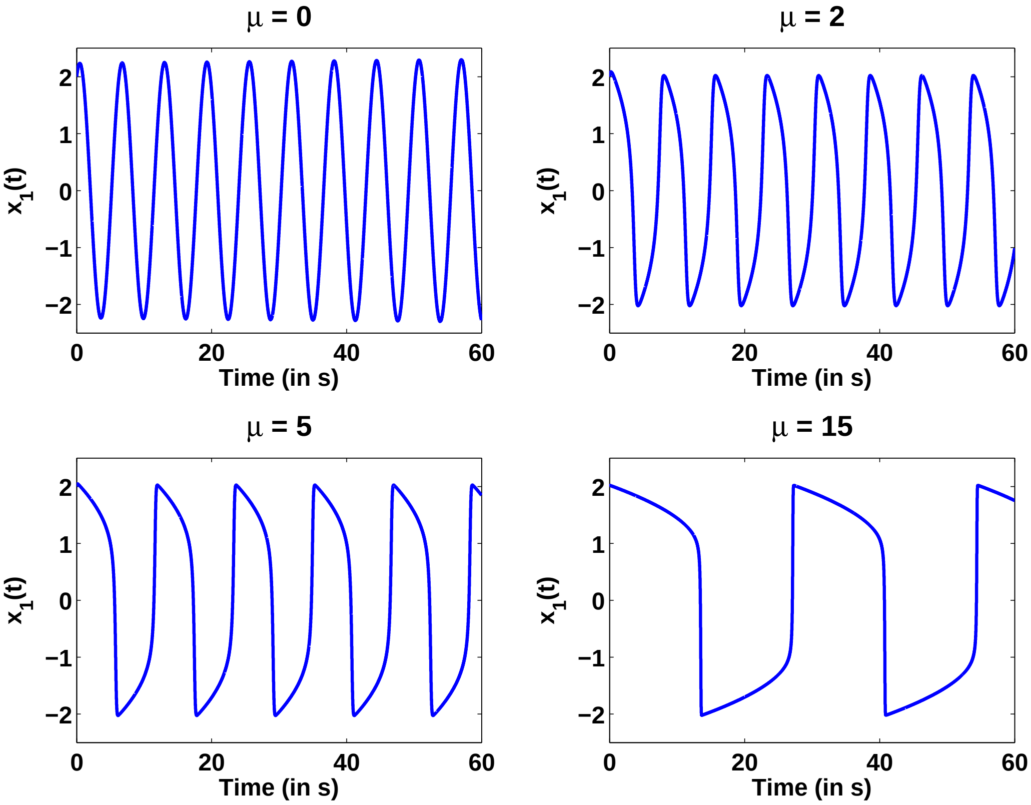


Figure 2.1: Multiple patterns of the Van der Pol oscillator based on value of μ

2.1.2 Discrete time dynamic system

While the dynamical system theory was developed for continuous time mathematical models, they are implemented in discrete time using a defined time-step. Differential equations are replaced by difference equations to describe these models mathematically. For a first order system the value of the system state at instance $k+1$ is calculated based on the value at instance k.

$$\boldsymbol{x}(k+1) = g(\boldsymbol{x}(k)) \tag{2.5}$$

One of the techniques used to convert continuous time equations to discrete difference equations is the explicit forward Euler method [26]. The derivative of the state with respect to time is approximated as

$$x'(t_{k+1}) = \frac{x(t_{k+1}) - x(t_k)}{t_{k+1} - t_k} \tag{2.6}$$

The time increment is kept constant, and equal to T. Using the above equation, the Van der Pol system can be written in discrete time as

$$\begin{aligned} x_1(k+1) =& x_1(k) + Tx_2(k) \\ x_2(k+1) =& x_2(k) + T\{\mu[1 - x_1(k)^2]x_2(k) - x_1(k)\} \end{aligned} \tag{2.7}$$

The quantization error between the discrete and continuous time model depends on the value of T. A smaller value of T will reduce the error and make the discrete model more accurate. However, it will increase the number of iterations required to simulate the model. T should be chosen carefully after analyzing the trade off between the accuracy and the computational cost for the concerned application.

2.2 Kalman filter

The Kalman filter [27], developed in 1959, is an estimator that utilizes the knowledge of the system dynamics to compute an unobserved state based on observations of the system. It has been used in various applications of dynamic systems and has proved to be efficient and accurate in being able to track and predict based on past observations. The algorithm is a combination of two steps - a prediction followed by a correction. It adds process noise and observation noise (typically white Gaussian) to model the uncertainty in system dynamics and measurements respectively. Originally the Kalman filter was developed to be used with linear dynamic systems. The extended Kalman filter (EKF) [28] used linearization techniques on nonlinear dynamic systems that enabled the application of the Kalman filter.

The accuracy of EKF results depends on the linearization error of the system. For systems with a ‘high’ degree of nonlinearity this is often outside of acceptable limits.

2.2.1 Unscented Kalman filter

One of the latest developments in the Kalman filtering literature has been the Unscented Kalman filter (UKF) [29]. It is a nonlinear filter that utilizes the unscented transform [30] to compute the error covariances and the Kalman gain. It generates an ensemble of sample vectors at every time step of the system and uses the non linear equations to propagate all these vectors to the next time step. Assuming a Gaussian distribution, the updated ensemble is used to compute the covariance matrix and the mean. The ensemble for the next time step is generated based on these statistics. While this technique is computationally more expensive than the EKF, it avoids using the linearized equation of the model to update the state covariances and means.

We define the nonlinear dynamic system as

$$\boldsymbol{X}(k+1) = f(\boldsymbol{X}(k)) + \boldsymbol{\nu}(k) \tag{2.8}$$

The observations are a function of the system states

$$\boldsymbol{Y}(k) = h(\boldsymbol{X}(k)) + \boldsymbol{w}(k) \tag{2.9}$$

$\boldsymbol{\nu}(k)$ and $\boldsymbol{w}(k)$ represent the process noise and observation noise.

The augmented state vector is computed by appending to the state vector of the nonlinear dynamic system (Eq. 2.8) the process noise vector, $\boldsymbol{\nu}(k)$.

$$\boldsymbol{X}^a(k) = \begin{bmatrix} \boldsymbol{X}(k) \\ \boldsymbol{\nu}(k) \end{bmatrix} \tag{2.10}$$

The mean and covariance of the state estimate at time step k are assumed to be of the following form

$$\begin{aligned} \hat{\boldsymbol{X}}^a(k|k) &= \begin{bmatrix} \boldsymbol{X}(k) \\ \boldsymbol{0} \end{bmatrix} \\ \boldsymbol{P}^a(k|k) &= \begin{bmatrix} \boldsymbol{P}(k) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_\nu(k) \end{bmatrix} \end{aligned} \tag{2.11}$$

The ensemble or sigma points are created based on the present covariance and mean.

$$\begin{aligned} \boldsymbol{\chi}_0^a =& \hat{\boldsymbol{X}}^a(k|k) \\ \boldsymbol{\chi}_i^a =& \hat{\boldsymbol{X}}^a(k|k) + (\sqrt{n^a \boldsymbol{P}^a(k|k)})_i, \qquad i = 1, 2, .., n^a \\ \boldsymbol{\chi}_i^a =& \hat{\boldsymbol{X}}^a(k|k) - (\sqrt{n^a \boldsymbol{P}^a(k|k)})_i, \qquad i = n^a, n^a + 1.., 2n^a \end{aligned} \tag{2.12}$$

n^a is the dimension of the augmented state vector $\boldsymbol{X}^a$ and $(\sqrt{n^a \boldsymbol{P}^a(k|k)})_i$ is the i^{th} row/column of the matrix square root of $n^a \boldsymbol{P}^a(k|k)$. Details of different computational methods used to generate the ensemble, $\boldsymbol{\chi}$, of $2n^a + 1$ sigma points can be found in [31].

The sigma points are each propagated using the nonlinear dynamics of the system, Eq. 2.8 to generate the ensemble of *apriori* state estimates.

$$\boldsymbol{\chi}_i^a(k+1|k) = f[\boldsymbol{\chi}_i^a(k|k)] \tag{2.13}$$

The *apriori* mean and covariance can be computed as a weighted sum of the sigma points.

[32] discusses the selection of the weights.

$$\begin{aligned}\hat{\boldsymbol{X}}^{a}(k+1|k) =&\sum_{i=0}^{2n^a} W_i \boldsymbol{\chi}_i^a(k+1|k)\\ \boldsymbol{P}^a(k+1|k) =& \sum_{i=0}^{i=2n^a} W_i[\boldsymbol{\chi}_i^a(k+1|k) - \hat{\boldsymbol{X}}^{a}(k+1|k)][\boldsymbol{\chi}_i^a(k+1|k) - \hat{\boldsymbol{X}}^{a}(k+1|k)]^T\end{aligned} \tag{2.14}$$

where

$$W_0 = \frac{\kappa}{\kappa + n^a}, \qquad W_i = \frac{1}{2(\kappa + n^a)} \quad i = 1,..,2n^a$$

W_0 is the weight assigned to the mean and W_i are the weights assigned to other vectors in the emsemble. The ratio of W_0 to W_i depends on κ, a constant. It has been observed that estimation results are more accurate when the mean is weighed higher than other vectors. The output function, Eq. 2.9 is used to compute the predicted output from the ensemble of state estimates.

$$\gamma_i(k+1|k) = h[\boldsymbol{\chi}_i^a(k+1|k)], \qquad i = 0,..,2n^a \tag{2.15}$$

The mean and the innovation covariance are calculated as

$$\begin{aligned}\hat{y}(k+1|k) =& \sum_{i=0}^{2n^a} W_i \gamma_i(k+1|k)\\ P_{yy}(k+1|k) =& R + \sum_{i=0}^{2n^a} W_i[\gamma_i(k+1|k) - \hat{y}(k+1|k)][\gamma_i(k+1|k) - \hat{y}(k+1|k)]^T\end{aligned} \tag{2.16}$$

where R is the variance of the observation noise.

The cross covariance is calculated according to

$$\boldsymbol{P}_{xy}(k+1|k) = \sum_{i=0}^{2n^a} W_i[\boldsymbol{\chi}_i^a(k+1|k) - \hat{\boldsymbol{X}}^{a}(k+1|k)][\gamma_i(k+1|k) - \hat{y}(k+1|k)]^T \tag{2.17}$$

The *aposteriori* states and covariances are then computed using the Kalman gain, $\boldsymbol{K}$.

$$\begin{aligned}
\boldsymbol{K}(k) =& \boldsymbol{P}_{xy}(k+1|k)P_{yy}(k+1|k)^{-1} \\
\hat{\boldsymbol{X}}^{a}(k+1|k+1) =& \hat{\boldsymbol{X}}^{a}(k+1|k) + \boldsymbol{K}(k)[y_{obs}(k) - \hat{y}(k+1|k) \\
\boldsymbol{P}^{a}(k+1|k+1) =& \boldsymbol{P}^{a}(k+1|k) - \boldsymbol{K}(k)\boldsymbol{P}_{xy}(k+1|k)
\end{aligned} \tag{2.18}$$

The *aposteriori* states are used as information that was unavailable from the observations of the system. The UKF can be used in applications where the observations need to be smoothed. In such cases there are no unobservable states, but the measurement noise is very high. It is also used to estimate hidden, or unobservable states based on the measurements.

2.3 Least squares parameter estimation

One of the fundamental techniques for parameter estimation in linear systems is minimizing the prediction errors [33]. The predicted output is a linear combination of the observed variables, with the coefficients being the parameters of the system. The error between the predicted and observed output is computed. The parameter estimate is then computed such that the sum of squared prediction errors over a series of observations is minimal.

The derivation of the least squares algorithm for a multiple input single output dynamic system is shown in the following section.

2.3.1 Multiple input single output dynamic system

The actual discrete time system with M inputs, $\{u_1(k), ..., u_M(k)\}$, and one output, $y(k)$, is described as

$$y(k) + \sum_{i=1}^{n_y} a_i \; y(k-i) = \sum_{m=1}^{M} \sum_{j=0}^{n_m} b_{m,j} \; u_m(k-j) + \nu(k) \tag{2.19}$$

$\nu(k)$ represents the effect of noise and model error. The output can be represented in vector notation.

$$y(k) = \boldsymbol{\varphi}^T(k)\boldsymbol{\theta} + \nu(k) \tag{2.20}$$

where

$$\boldsymbol{\varphi}(k) = [u_1(k), ..., u_1(k-n_1), u_2(k), ..., u_2(k-n_2), ..., u_M(k), ..., u_M(k-n_M),$$
$$-y(k-1), ..., -y(k-n_y)]^T$$

$$\boldsymbol{\theta} = [b_{1,0}, b_{1,1}, ..., b_{1,n_1}, b_{2,0}, ..., b_{2,n_2}, ..., b_{M,0}, ..., b_{M,n_M}, a_1, ..., a_{n_y}]^T$$

The predicted output is based on the estimated parameter vector, $\hat{\boldsymbol{\theta}}$, and the observed regressor, $\boldsymbol{\varphi}(k)$.

$$\hat{y}(k) = \boldsymbol{\varphi}^T(k)\boldsymbol{\theta} \tag{2.21}$$

The sum of the squared prediction errors is computed over N observations.

$$\xi(k,N) = \sum n = 0^{N-1}[\hat{y}(k-n) - y(k-n)]^2 \tag{2.22}$$

The least squares estimate of the parameter vector is derived by minimizing ξ. The gradient of the above function with respect to the parameters is computed as

$$\frac{\partial \xi(k,N)}{\partial \hat{\boldsymbol{\theta}}} = \sum_{n=0}^{N-1} 2\boldsymbol{\varphi}(k-n)[\varphi^T(k-n)\hat{\boldsymbol{\theta}} - y(k-n)] \tag{2.23}$$

and equated to zero at the location of optimimum, $\boldsymbol{\theta}^*$.

$$\sum_{n=0}^{N-1} 2\boldsymbol{\varphi}(k-n)[\varphi^T(k-n)\boldsymbol{\theta}^* - y(k-n)] = 0 \tag{2.24}$$

The optimal estimate of the parameter vector is then derived as

$$\begin{gathered}
\left[\sum_{n=0}^{N-1} \boldsymbol{\varphi}(k-n)\boldsymbol{\varphi}^T(k-n)\right]\boldsymbol{\theta}^* = \sum_{n=0}^{N-1} \boldsymbol{\varphi}(k-n)y(k-n) \\
\boldsymbol{\theta}^* = \left[\sum_{n=0}^{N-1} \boldsymbol{\varphi}(k-n)\boldsymbol{\varphi}^T(k-n)\right]^{-1}\left[\sum_{n=0}^{N-1} \boldsymbol{\varphi}(k-n)y(k-n)\right]
\end{gathered} \tag{2.25}$$

Chapter 3: Optimization of Single Neuron Models

Neurons form the basic building blocks of our brain and central nervous system. Networks of neurons communicate with each other to perform complicated tasks. This communication amongst neurons is through the ‘spiking’of the membrane potential which is between $-70mV$ and $-50mV$ in its ‘inactive’ state and jumps to about $30mV$ when there is a sufficient post synaptic potential generated by other connected neurons. The time between two spikes is called an inter-spike interval (ISI) [34]. The spiking response of a single neuron is used to create a spike train - a string of binary values with a 1 indicating the presence of a spike. The temporal coding scheme [34] is one of methods used to characterize neurons based on spike trains. [35] explains why spike times are meaningful, particularly in the case of fast spiking neurons.

Following the success of these different single neuron models referenced in Chapter 1, the problem of parameter fitting has drawn huge attention in the last decade. Efforts are being made to accurately estimate model parameters from experimental data using innovative methods. For the class of stochastic models, parameter estimation using maximum likelihood (ML) has been explained in [36]. Expectation maximization (EM) techniques and point processes [37] are other methods of estimating stochastic models from data. These probabilistic methods cannot be used for determining dynamic models. However, tracking states of a dynamic system is a well researched problem with very accurate results. The Kalman filter (KF) [27], used since the 1960s, is one of the most successful methods which is able to estimate the state of the system based on observed measurements. Although initially only applicable to linear systems, the extended Kalman filter (EKF) and unscented Kalman filter (UKF) [32] were developed to work with nonlinear systems. In [38] the UKF has been used to track the dynamics of a model of the cerebral cortex. The UKF is also been applied in tracking the parameters of the model based on voltage measurements as described in [39], where the HH model was used to produce the membrane potential of a single biological neuron. A summary of the efforts to use Kalman filters to estimate states of a neuron model can be found in [40].

Another approach to estimate parameters or states is to develop an error function that quantifies the comparison between the experimental and model spike trains and use optimization techniques to find the optimal solution. [41] compares multiple shooting techniques and UKF in estimating the state variable of a Fitzhugh-Nagumo neuron. It is important to construct a proper cost function. If the function is too complex, gradient based methods cannot be used to converge to the optimum. In such cases global optimization techniques, such as particle swarm optimization (PSO) [42] or Nelder-Mead (N-M) method [43] are used. [44] summarizes the efforts in this area and mentions the different methods currently being used. It mentions that in situations where the number of unknown parameters is small even a simple 'exhaustive search' technique is found to be effective as compared to existing methods. This is due to the fact that the models are highly complex and developing a guided search technique is challenging. This suggests that there is still a huge scope of research in this area, mainly to develop automated and efficient techniques with which can fit models to neural spike trains.

A majority of the existing estimation techniques utilize the measured voltage trace to define the error ([44, 45]). In most experimental recordings [46] the entire trajectory of the membrane potential is not available. Only the spike times, or the ISIs, are available. Also, due to the complexity of the dynamical systems, the trajectory of the membrane potential during the 'inactive' state of the neuron model is not always the same. Models are able to synchronize with the experimental ISIs, but not with the entire trace of the membrane potentials. The challenge, therefore, is to estimate parameters using only the ISIs (or the spike times). This is investigated in [47] with measurements from the primary motor cortex area (M1 neurons). The input current and the parameters are adjusted to synchronize the spikes of the model to that of the experimental data. Though the error between the model output and the measured data is very low, there is no guarantee that the model will be a good predictor.

The objective is to accurately fit a model to the measured neuronal spike times. The same input stimulation current is applied to both the biological neuron and the mathematical model. The outputs are compared and using an adaptive algorithm the parameters of the model are adjusted such that the error is minimized. The model is then used to predict the spike times for the biological neuron, as shown in Fig. 3.1. The three major aspects of this problem are (i) the model, (ii) the comparison technique and (iii) the method of optimization.

In this chapter, a new technique to compare spike trains of single neurons is described. Furthermore, the parameters of the model are estimated by formulating a nonlinear optimization problem using the proposed performance function. The solution is found by using the gradient descent method. The method is also successfully applied on experimental data from an L5 neuron and the results are presented. The model is fitted to a 4s experimental spike train and the proposed technique is successful in computing the optimal parameters. It outperforms other automated search techniques such as Nelder-Mead.

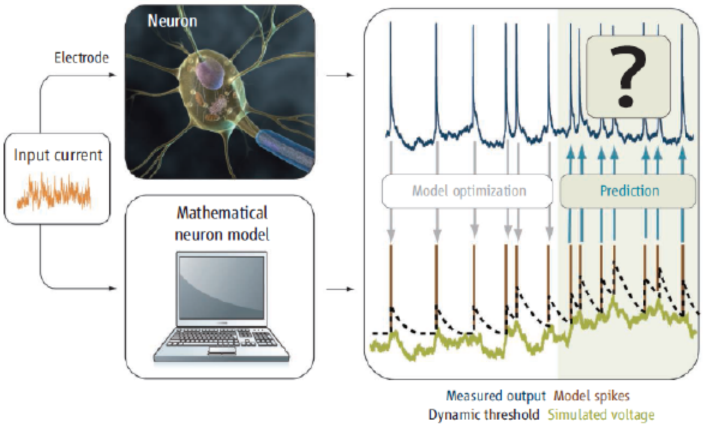


Figure 3.1: Prediction of neuronal spiking using an optimized mathematical model [1]

3.1 The Model

The augmented multi-timescale adaptive threshold model (AugMAT) is chosen to mathematically represent the spike train of a biological neuron. It is a two dimensional nonlinear dynamical system where the first state represents the membrane potential, $V(t)$ (Eq. 3.1), and the other is the adaptive threshold, $\theta(t)$ (Eq. 3.2).

$$\tau_m \frac{dV(t)}{dt} = -V(t) + RI(t) \tag{3.1}$$

$$\begin{aligned} \theta(t) &= \sum_i H(t - t_i) + \beta\ \kappa(t) + \omega \\ H(t) &= \sum_{j=1}^{L} \alpha_j e^{-t/\tau_j} \\ \kappa(t) &= \int_0^t K(s) V'(t-s) ds, \qquad K(s) = s e^{-s/\tau_v} \end{aligned} \tag{3.2}$$

The adaptive threshold represented by Eq. 3.2 is a combination of three terms. The first term, $H(t)$, represents a weighted sum of decaying exponentials. The time constants are represented by the parameters τ_j and each exponential is scaled by α_j. L represents the number of decaying exponentials used. The second term, $\kappa(t)$, integrates the time derivative of $V(t)$ scaled by the kernel function $K(s)$. This term, multiplied by the parameter β, ensures that small perturbations in the input do not result in a spike. The kernel applies maximum scaling at a delay of τ_v. The third term, ω, is a resting potential and it defines the minimum value that the threshold can decay to. If the input, $I(t)$ in nA, is not sufficiently high to raise the voltage to ω then the neuron will never spike.

A neuronal spike is generated if the membrane potential becomes greater than the threshold. The neuron is a hybrid model, that uses a discrete reset to change the value of the threshold at the instance of the spike. This introduces discontinuities in the threshold function. The set of spike times, t_k, is defined as

$$\epsilon(t) = V(t) - \theta(t); \;\; t_k : \epsilon(t_k) = 0 \text{and} \epsilon'(t_k) > 0 \tag{3.3}$$

$\epsilon(t)$ is the difference between the voltage and the threshold and equals to zero with positive slope at each of the instances that belong to the set of spike times.

The selection of the model parameters is described in [17]. The threshold is chosen to be a sum of two exponentials with fixed time constants after simulatons showed that increasing the number of exponentials did not add to the range of model behavior. Also, the time constant, τ_m, and gain, R, of the voltage equation and the kernel function parameter, τ_v, do not vary. As a result of numerical experiments the best values were chosen as

$$\tau_m = 10ms, \;\; R = 50M\Omega, \;\; \tau_v = 5ms, \;\; L = 2, \;\; \tau_1 = 10ms, \;\; \tau_2 = 200ms \tag{3.4}$$

The scaling of the adaptive threshold terms, α_1, α_2, β and ω are defined as the 'varying' parameters of the model. While they are constant for a particular simulation, the values can be changed to achieve different spiking patterns. The range of the parameter space and the corresponding model behavior is reported in [17].

Fig. 3.2 shows a simulation of the model, and the computation of the spike train from the states of the differential equations. The top row is the input current. It is a sum of different exponentials, to reproduce the type of stimulus a biological neuron receives from its dendritic tree. The second row shows $v(t)$ (blue, dashed) and $\theta(t)$ (red, solid) functions. The last row shows the constructed spike train, which is a binary string of 1s and 0s. The parameters chosen for this simulation were

$$\alpha_1 = 180, \alpha_2 = 3, \beta = 0.2, \omega = 15$$

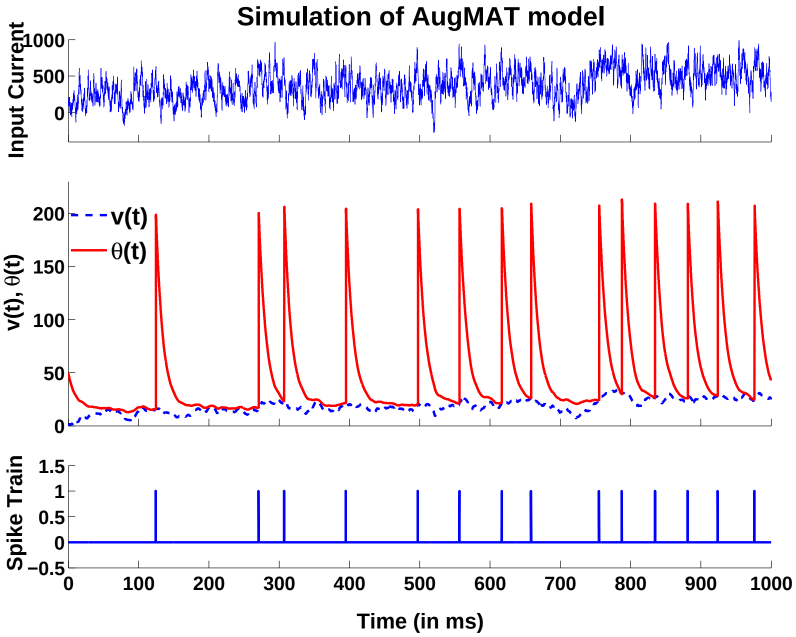


Figure 3.2: Simulation of AugMAT model

This chapter defines a new gradient based optimization technique to optimize the model parameters based on spike times obtained from an experimental spike train. While the model is capable of producing the spiking activity observed in a biological neuron, computing the parameters of the model from the observations is not a trivial task. The method is derived by first defining a suitable error function that represents the difference between the model spike train and the biological neuron spikes. The analytical gradient of the error function with respect to the varying parameters is computed and the steepest descent algorithm is implemented to estimate the optimal parameter vector. The 'fixed' parameters defined in Eq. 3.4 are assumed known. The varying parameters, $\{\alpha_1, \alpha_2, \beta, \omega\}$ along with the initial condition, θ_0, are estimated by the proposed algorithm.

3.1.1 Performance function :

The performance function is the most crucial component of the parameter estimation problem. As mentioned in [44] the function should be relevant to the problem, convex and as smooth as possible. There are several existing methods to compute the error between two spike trains ([44],[48], [49]). However, due to the non-smooth nature the performance of the optimization algorithm is hindered [50]. Kobayashi et al. [16] describe a coincidence factor based on the number of spikes which coincide within an allowable time range, Δ. This has to be maximized to find the optimal parameter set.

$$\Gamma = \frac{N_{coinc} - < N_{coinc} >}{N_{data} + N_{model}} \times \frac{2}{1 - 2\nu\Delta} \tag{3.5}$$

N_{coinc} is the number of spikes which coincide, $< N_{coinc} >$ is the expected number of coincidences for a Poisson spike train of rate ν. N_{data} and N_{model} are the number of spikes in the data and model spike trains respectively. The second component in the equation is a normalizing factor, such that the assessment value is 1 if all the spikes coincide, and 0 if the model performs no better than a poisson spike train. Although this function is representative of the 'goodness of fit' of the model, it is difficult to utilize it for the purpose of model optimization. The complex nature of the function makes it difficult to compute the gradient. Global optimization techniques, such as particle swarm optimization or Nelder-Mead are used to find the optimum. The initialization of the parameter set becomes important and the convergence is not guaranteed for any random starting point.

The properties of a performance function that leads to successful computation of the optimal parameter set using various optimization techniques has been outlined in the literature of applications of optimization. It is well known that it should approximate a convex function as best as it can while at the same time qualitatively represent relation between the parameters and the final objective. It is important then to understand the application for which the optimal parameters need to be computed. The comparison of spike trains must take into consideration the number of spikes in a finite time window as well as the synchronization of these spikes.

To derive the error function, first the time of spikes is extracted from the spike train,

$$t_{spikes} = \begin{bmatrix} t_1 & t_2 & . & . & t_n \end{bmatrix}$$

where $t_1, t_2, ..$ are the time of spikes and n is the total number of spikes in the spike train of length t_f.

A 'staircase' function based on the time of spikes is defined as

$$\psi(t) = \sum_{k=1}^{n} \upsilon(t - t_k) \tag{3.6}$$

where $\upsilon(x)$ is the step function which is equal to 1 for $x > 0$ and 0 otherwise.

The error, ξ, is given by the square of the difference between the area of the staircase function, $\psi(t)$, of each model. $\psi_1(t)$ and $\psi_2(t)$ are the staircase functions of two different spike trains.

$$\xi = \frac{1}{t_f} \int_0^{t_f} [\psi_1(t) - \psi_2(t)]^2 dt \tag{3.7}$$

Fig. 3.3 illustrates the error function that compares two spike trains and has all the required properties to be used as an objective function candidate that is minimized in searching for the optimal parameter set. The top and bottom row represent the two spike trains. The middle row shows the staircase function, $\psi(t)$ for both spike trains - defined in Eq. 3.6. The error between the spike trains is the square of the area between the two lines, defined mathematically in Eq. 3.7.

Fig. 3.4 represents different examples of spike train comparisons, and lists the performance function value for each case. The top row in both (a) and (b) is the observed spike train, in black. The spike trains that are compared to the observed one are shown below, with the numerical value of comparison shown on the left. In example (a), the blue and

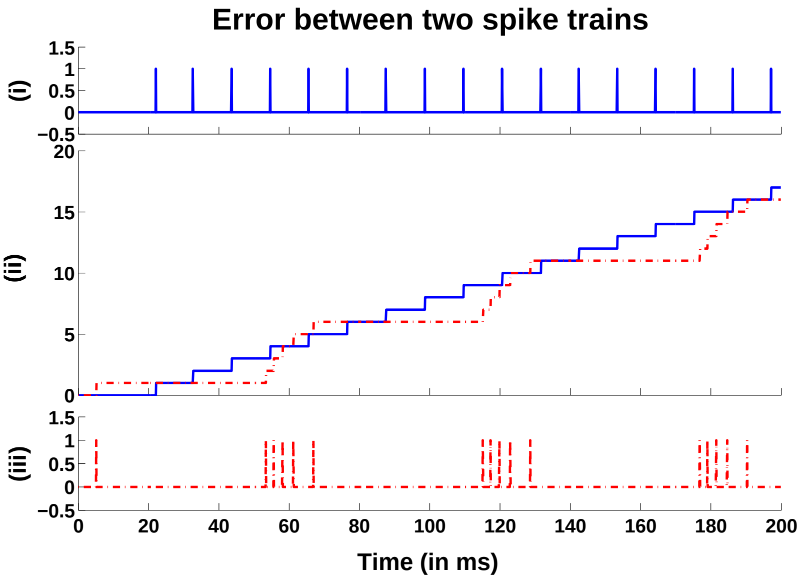
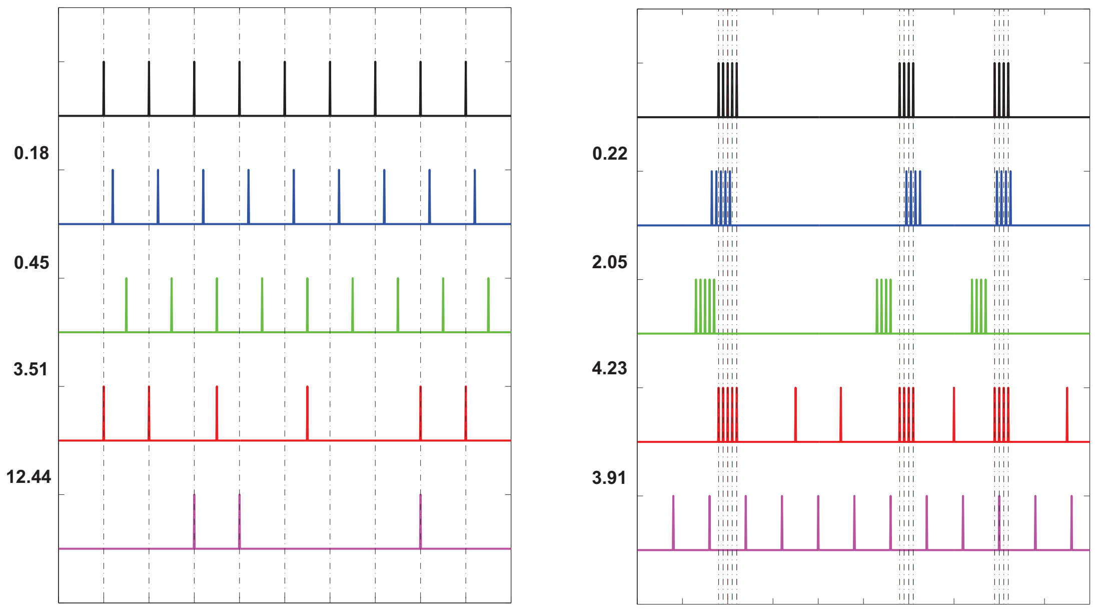


Figure 3.3: Comparison of two neuronal spike trains

green spike trains (rows 2 and 3) are time-shifted versions of the observed spike train. A smaller time-shift results in a smaller error value. The red and pink spike trains (rows 4 and 5) contain a few spikes that coincide exactly with spikes of the observed one. However, the error in these cases should be large due to the fact that they don't have the same number of spikes and the spike patterns do not much. This is captured by the proposed performance function as shown by the values in the figure. Example (b) uses a bursting neuron response as the observed spike train. Similar to the first example, it can be observed that spike trains having the same spike pattern (blue and green) have lower errors even though none of the spike times actually coincide. The numbers for the red and pink spike trains indicate that the comparison of bursting spike neurons is more complicated as there is a bigger trade-off between coincidental spikes and total number of spikes in the finite time window. The extra spikes result in a higher error, as compared to having an equal number of spikes, even though it contains all the spikes of the observed spike train.

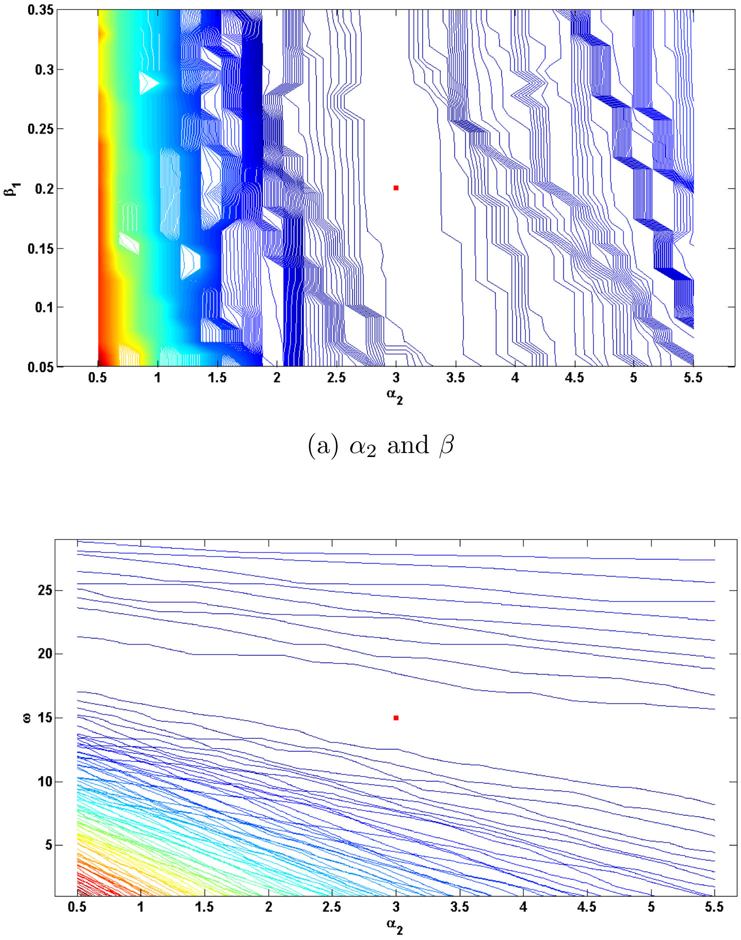


(a) Regular spiking

(b) Bursting

Figure 3.4: Comparison of spike trains based on the proposed error function

It is also necessary to evaluate the convexity of the error function with respect to the parameters of the model. While it is difficult to develop a purely convex function due to nonlinear dynamics, an approximation with a small number of local minimas would result in a fast and accurate convergence of the parameter estimates. Fig. 3.5 plots the surface of the error function with respect to the parameters, taken in pairs. The value of the parameters were considered over the entire range for which different patterns of neuronal spiking could be generated. The error surface is computed as a function of changing parameters and was found to be smooth. It is not a complex function, and hence does not affect the speed of optimization. There is an interesting aspect of the surface, that it has a lot of variation as compared for example to a classical quadratic surface. However, it is not clear that the function is nonsmooth in the sense of not having a derivative. In the next section the derivative with respect to each individual parameter is computed and simulations prove that the computations are accurate.



(a) α_2 and β

(b) α_2 and ω

Figure 3.5: Surface of the proposed error function, plotted as a function of two varying parameters

3.2 Model optimization

There are numerous iterative algorithms available in the literature of numerical optimization to search the minimum of a function (ξ). Gradient based algorithms [51] compute the gradient of the function with respect to the parameter vector, $\mathbf{p}$. The vector is then updated using the gradient. If the update, $\hat{\mathbf{p}}$ is in the opposite direction of the gradient, then by updating the vector iteratively, it converges to a minima. To reach a maxima, the update should be in the direction of the gradient.

$$\hat{\mathbf{p}} = \mathbf{p} - \nu \frac{\partial \xi}{\partial p} \tag{3.8}$$

ν is the step size of the gradient based search. For the problem defined in the previous section, the performance function, ξ, is given by Eq. 3.7 and the parameter vector consists of the variable parameters of the AugMAT model.

$$\mathbf{p} = [\alpha_1, \alpha_2, \beta, \omega, \theta_0]^T$$

Computation of the gradient requires analytical formulae of the derivatives of the performance function with respect to each parameter.

$$\frac{\partial \xi}{\partial p} = \frac{1}{t_f} \int_0^{t_f} 2[\psi(t) - \psi^d(t)] \frac{\partial \psi(t)}{\partial p} dt \tag{3.9}$$

$$\frac{\partial \psi(t)}{\partial p} = \sum_{k=1}^{M} \delta(t - t_k)[-\frac{\partial t_k}{\partial p}] \tag{3.10}$$

In Eq. 3.10, the $\frac{\partial t_k}{\partial p}$ expression represents the change in the k^{th} spike time with respect to the change in parameter p. The perturbations in p changes the value of t_k, but the value of the function $\epsilon(t_k, p)$ (Eq. 3.3) remains zero. A similar computation has been done by Booij and Nguyen in [52]. Applying the total differential identity to the constant contour of the function $\epsilon(t_k, p) = 0$ in the problem described,

$$\begin{aligned} \frac{\partial \epsilon(t_k,p)}{\partial t_k} dt_k + \frac{\partial \epsilon(t_k,p)}{\partial p} dp = 0 \\ \frac{\partial t_k}{\partial p} = \frac{-\dfrac{\partial \epsilon(t_k)}{\partial p}}{\dfrac{\partial \epsilon(t_k,p)}{\partial t_k}} \end{aligned} \tag{3.11}$$

While the derivative of $\epsilon(t_k, p)$ with respect to the spike time, t_k, can be calculated by applying small pertubations in the region of t_k, the derivative with respect to the parameters needs to be calculated analytically.

$$\begin{aligned} \frac{\partial \epsilon(t_k,p)}{\partial t_k} &= \frac{\epsilon(t_k,p) - \epsilon(t_k - \Delta, p)}{\Delta} \\ \frac{\partial \epsilon(t_k,p)}{\partial p} &= \frac{\partial V(t_k,p)}{\partial p} - \frac{\partial \theta(t_k,p)}{\partial p} \end{aligned} \tag{3.12}$$

Gradient computation for parameter α_j

$$\begin{aligned} \frac{d\theta(t_k)}{d\alpha_j} &= \frac{\partial \theta(t_k)}{\partial \alpha_j} + \sum_{i:t_i<t_k} \frac{\partial \theta(t_k)}{\partial t_i} \frac{\partial t_i}{\partial \alpha_j} \\ &= \sum_{i:t_i<t_k} e^{-(t_k-t_i)/\tau_j} + \sum_{i:t_i<t_k} [\sum_{j=1}^{L} \frac{\alpha_j}{\tau_j} e^{-(t_k-t_i)/\tau_j}] \frac{\partial t_i}{\partial \alpha_j} \end{aligned} \tag{3.13}$$

Gradient computation for parameter β

$$\begin{aligned} \frac{d\theta(t_k)}{d\beta} &= \frac{\partial \theta(t_k)}{\partial \beta} + \sum_{i:t_i<t_k} \frac{\partial \theta(t_k)}{\partial t_i} \frac{\partial t_i}{\partial \beta} \\ &= \int_0^{t_f} K(s) V'(t_k - s) ds + \sum_{i:t_i<t_k} [\sum_{j=1}^{L} \frac{\alpha_j}{\tau_j} e^{-(t_k-t_i)/\tau_j}] \frac{\partial t_i}{\partial \beta} \end{aligned} \tag{3.14}$$

Gradient computation for parameter ω

$$\begin{aligned}\frac{d\theta(t_k)}{d\omega} &= \frac{\partial\theta(t_k)}{\partial\omega} + \sum_{i:t_i<t_k} \frac{\partial\theta(t_k)}{\partial t_i}\frac{\partial t_i}{\partial\omega} \\ &= 1 + \sum_{i:t_i<t_k}[\sum_{j=1}^{L}\frac{\alpha_j}{\tau_j}e^{-(t_k-t_i)/\tau_j}]\frac{\partial t_i}{\partial\omega}\end{aligned} \tag{3.15}$$

Gradient computation for parameter θ_0

$$\begin{aligned}\frac{d\theta(t_k)}{d\theta_0} &= \frac{\partial\theta(t_k)}{\partial\theta_0} + \sum_{i:t_i<t_k} \frac{\partial\theta(t_k)}{\partial t_i}\frac{\partial t_i}{\partial\theta_0} \\ &= \frac{\theta_0}{\sum_{j=1}^{L}\alpha_j}\sum_{j=1}^{L}\alpha_j e^{-t_k/\tau_j} + \sum_{i:t_i<t_k}[\sum_{j=1}^{L}\frac{\alpha_j}{\tau_j}e^{-(t_k-t_i)/\tau_j}]\frac{\partial t_i}{\partial\theta_0}\end{aligned} \tag{3.16}$$

Gradient of $V(t)$ with respect to any parameter

$$\frac{dV(t_k)}{dp} = 0 \tag{3.17}$$

Combining Eqs. 3.8-3.11, the gradient of the proposed error function can be computed for each parameter in the vector $\boldsymbol{p}$.

1. Initialize parameter vector at $n = 0$.

$$\hat{\boldsymbol{p}}(0) = \left[\hat{\alpha}_1(0), \hat{\alpha}_2(0), \hat{\beta}(0), \hat{\omega}(0), \hat{\theta}_0(0)\right]^T$$

2. Use equations Eq. 3.9-3.17 to compute the gradient vector.

$$\frac{\partial\xi(\hat{\boldsymbol{p}}(n))}{\partial\hat{\boldsymbol{p}}(n)} = \left[\frac{\partial\xi(\hat{\boldsymbol{p}}(n))}{\partial\hat{\alpha_1}(n)}, \frac{\partial\xi(\hat{\boldsymbol{p}}(n))}{\partial\hat{\alpha}_2(n)}, \frac{\partial\xi(\hat{\boldsymbol{p}}(n))}{\partial\hat{\beta}(n)}, \frac{\partial\xi(\hat{\boldsymbol{p}}(n))}{\partial\hat{\omega}(n)}, \frac{\partial\xi(\hat{\boldsymbol{p}}(n))}{\partial\hat{\theta}_0(n)}\right]^T$$

3. Update the parameter vector. ν is chosen as 10^{-4}.

$$\hat{\boldsymbol{p}}(n+1) = \hat{\boldsymbol{p}}(n) - \nu \frac{\partial \xi(\hat{\boldsymbol{p}}(n))}{\partial \hat{\boldsymbol{p}}(n)}$$

4. Check for termination condition. Otherwise increment iteration. $n = n + 1$.

3.3 Results

3.3.1 Synthetic spike train

The AugMAT model was simulated, with parameters as mentioned in Sec. 3.1. The spike times were extracted and used to estimate parameters of the model from a random starting point. The parameter space was chosen by setting minimum and maximum values.

$$\chi = [\{\alpha_1, \alpha_2, \beta, \omega, \theta_0\} : 100 < \alpha_1 < 220, 0 < \alpha_2 < 8,$$

$$0.1 < \beta < 0.5, 0 < \theta_0 < 100]$$

Fig. 3.8 shows the convergence of the parameter estimates using the gradient descent algorithm (GD) with a two dimensional vector. This allows visual interpretation of the algorithm, and confirm that the analytical gradient computations (derived in Sec. 3.2) are correct. Different combinations of two varying parameters were taken - (i) α_2 and β, (ii) α_2 and ω, (iii) β and ω. The error surface was plotted (contour plot). The red trace shows the convergence of the parameter vector from a random starting point. In each example it can be seen that the parameter vector is updated in the direction of the gradient, and is able to navigate through the error surface and reach the minimum in a finite number of steps. The algorithm is also tested with different initial conditions, and was able to successfully locate the optimal parameter estimate. The irregularity of the surface in the region of the minimum can be attributed to the nonlinear dynamics of the model.

Fig. 3.9 shows the convergence of the parameters and the performance function. It is

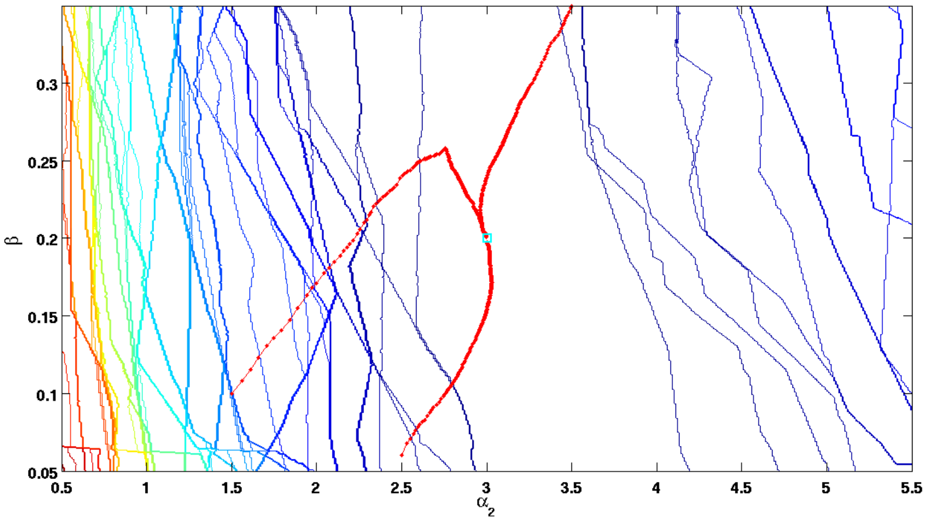


Figure 3.6: Parameter search in the $\alpha_2 - \beta$ plane

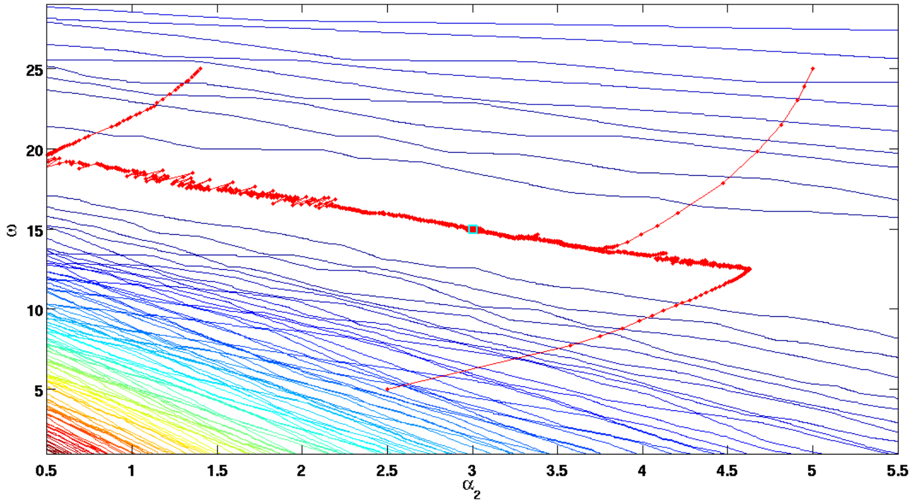


Figure 3.7: Parameter search in the $\alpha_2 - \omega$ plane

observed that the error function is not sensitive to the α_1 parameter. The other parameters converge to their true values, indicated by red dashed lines in 2000 iterations. The mean square error for every iteration is shown in the fourth subplot.

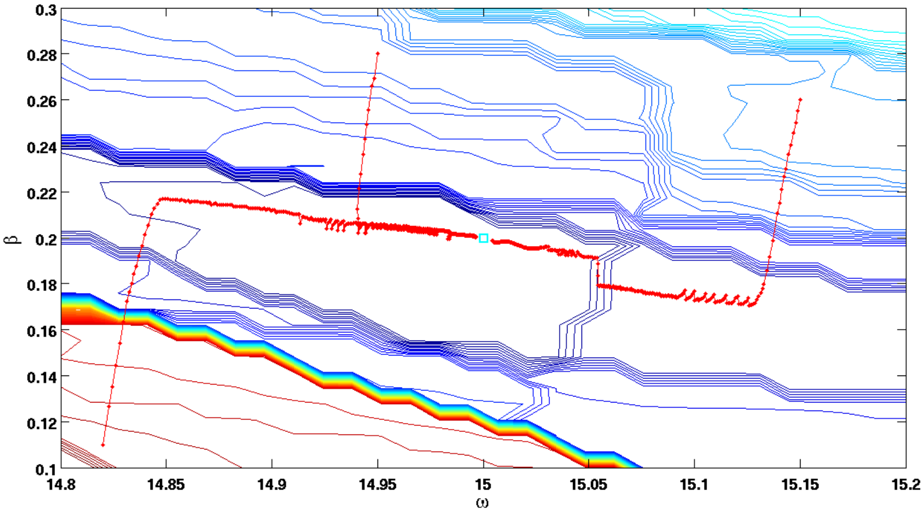


Figure 3.8: Parameter search in the $\omega - \beta$ plane

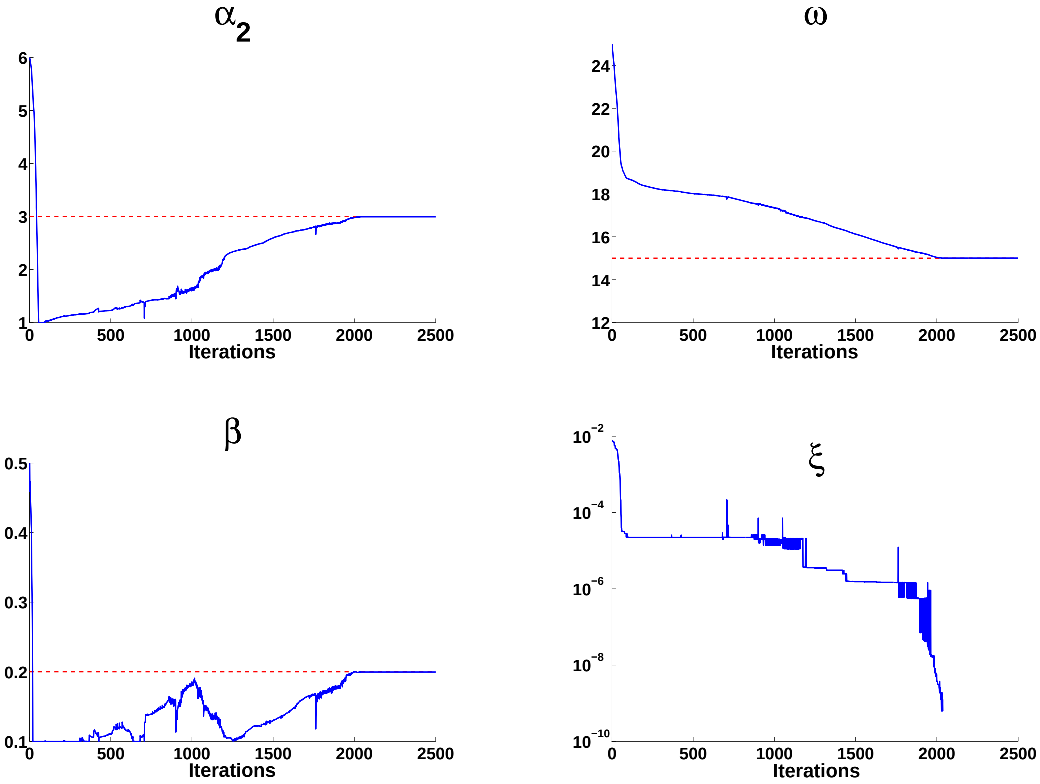


Figure 3.9: Convergence of parameters using th GD method

3.3.2 Experimental spike train

The gradient based parameter search technique is applied to fit the AugMAT model to the spike train data of an L5 neuron, made publicly available by INCF [1]. A single neuron was stimulated and the membrane potential was recorded. There were a total of 13 trials performed and the current and voltage traces were recorded and made publicly available. A 4s time window (17.5s to 21.5s) was chosen to construct the error function and use it for model optimization. Yamauchi et al. [17] have reported the optimal model parameters for the same data set. The coincidence factor computed for the experimental settings mentioned was 0.74. However, the method of optimization was not discussed. Simulations showed that the computation of the optimal parameters depended on the initial conditions. The objective was then to sucessfully converge to the optimal parameter set from random initial conditions. While it can be shown that, theoretically, gradient based techniques will achieve convergence when used on a convex error function, the speed of convergence depends on the nature of the function. Error functions designed for practical problems often have a small gradient in the neighbourhood of the global minimum. This increases the number of iterations needed for the parameters to converge to the optimal values. Non-gradient based methods are able to provide a faster convergence speed, but are more sensitive to the initial values. One such method is the Simplex (or Nelder-Mead) technique developed by J.A. Nelder and R. Mead [43] . This technique is based solely on the evaluation of function values on the vertices of a “simplex” and moving the search in the space of parameters by iteratively constructing new vertices and modifying the simplex. The method has proved to be successful in locating the optimal parameters in applications where calculation of the gradient is not possible [53, 54].

The approach used in this section is to combine the two techniques with a pre-defined switching condition to optimize the neruon model based on experimental spike train observations. The GD method starts from a random point in the parameter space and attempts

[1] http://www.incf.org/community/competitions/spike-time-prediction/2009/challenge-a

to minimize the 'staircase' error function, ξ, defined in Eq. 3.7. This iterates until a predefined switching condition is met. The number of iterations is used as the condition. The parameter estimates of the GD method are then used as the initial estimate in the N-M method which minimizes the coincidence factor. This hybrid approach, outlined in Fig. 3.10 overcomes the limitations of both methods and the final solution is no longer sensitive to the initial conditions.

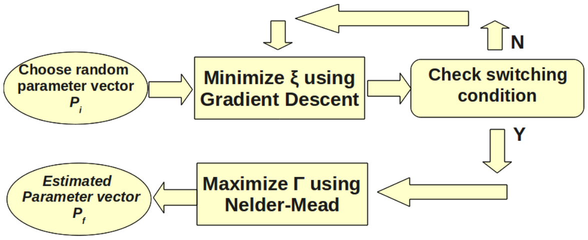

Choose random parameter vector P_i
Minimize ξ using Gradient Descent
Check switching condition
N
Y
Maximize Γ using Nelder-Mead
Estimated Parameter vector P_f

Figure 3.10: Hybrid technique used for optimizing the AugMAT model using experimental spike train data

Fig. 3.11 shows how the gradient method helps to improve the results. In many cases the coincidence factor after the GD method is small, but the final result is much closer to the optimum. This is because even though the GD may not necessarily be able to perfectly align the spikes, it successfully matches the 'staircase' functions bringing the spike trains closer. This allows the N-M method to maximize the coincidence factor and align the spikes.

100 random sets of parameters were generated and used as the initial points for both methods (GD + N-M and N-M). The mean and standard deviation are shown in Fig. 3.12. There is a 20% improvement in the mean. Statistics were computed over the results of these trials : $\mathbf{0.65 \pm 0.09}$ for "GD + N-M" and $\mathbf{0.55 \pm 0.1}$ for "only N-M". 35% of the final results were greater than 0.7 using our hybrid approach. This figure was only 9% while using N-M only.

3.3.3 Prediction of neural spiking activity

In the previous sections the results confirmed that the augmented multi-timescale adaptive threshold model could be optimized based on the spike times of an experimental neuron. While the search for the optimal parameter set typically depends on the initial conditions,

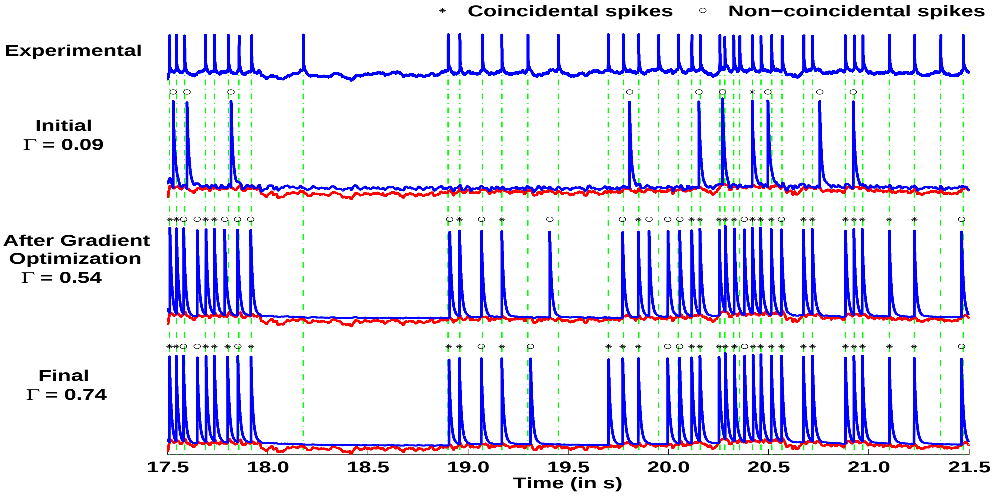


Figure 3.11: Using the hybrib optimization technique to compute the optimal parameter set of the AugMAT model : $[\alpha_1^* = 199.65, \alpha_2^* = 2.53, \beta^* = 0.06, \omega^* = 16.33, \theta_0^* = 32.58]$

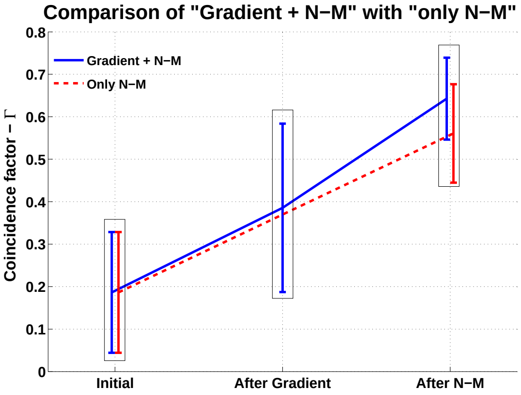


Figure 3.12: The results of the hybrid technique are compared with the results of Nelder-Mead method

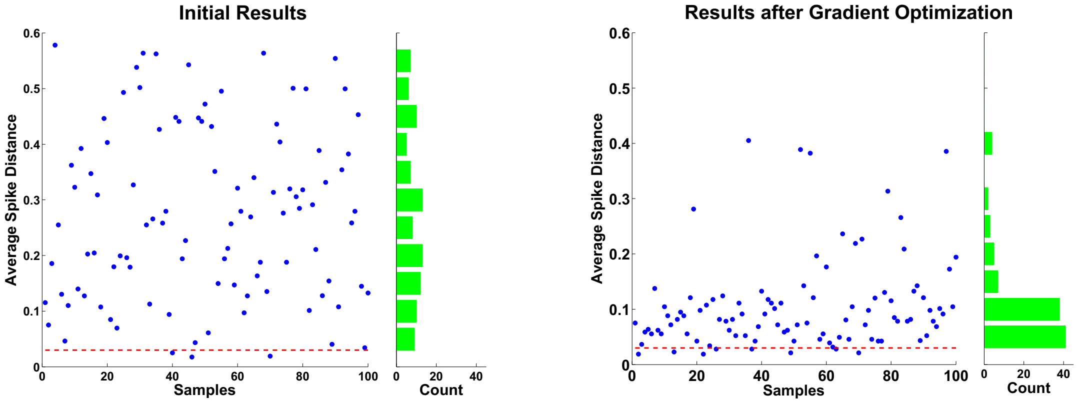
the use of the gradient descent method for minimizing the proposed error function reduces the sensitivity of the search. In this section a small segment of the experimental data [1] is used to compute the parameters of the model. The parameters are then fixed, and the model is simulated with the recorded input current. The spike times of the mathematical neuron are then compared to the spike times of the experimental neuron. Results show that the optimized model is able to successfully predict the response of the biological neuron if the input current is known.

The model was optimized using a 4s spike train where the neuron was stimulated with constant current. The spike times were extracted and used to estimate parameters of the model. The parameter space was chosen by setting minimum and maximum values, taking into consideration the nature of spiking. Different parameter sets from this space were able to generate the different kinds of responses.

$$\begin{aligned}\chi =&[\{\alpha_1, \alpha_2, \beta, \omega, \theta_0\} : 100 < \alpha_1 < 220, 0 < \alpha_2 < 8,\\ &0.1 < \beta < 0.5, 5 < \omega < 15, 0 < \theta_0 < 100]\end{aligned}$$

100 random samples were chosen from the parameter space. Each sample serves as an initial parameter vector for the optimization algorithm. The optimal parameter vector for each starting point was found using the gradient descent method (GD). The number of iterations was set at 20, since it was observed that there was no substantial improvement in the error after 20 iterations. These estimated parameters were then used to simulate the model. The simulated spike train was compared to the experimental spike train using the bi-variate SPIKE-distance [55].

Fig. 3.13 shows the initial and final bi-variate spike distances computed for each sample. (a) shows the initial distribuition of the SPIKE-distance measure (100 samples) and (b) shows the distribution after optimization is achieved. The red dotted line shows the multi-variate distance measure over the 13 different trials of the same experimental neuron. The distribution of the spike distances for the initial population was $\mathbf{0.28} \pm 0.16$. After optimization, the final spike distances were $\mathbf{0.10} \pm 0.08$. 62% of the final predicted spike trains had a SPIKE-distance of less than 0.1. As a benchmark, the multi-variate SPIKE-distance for the 13 trials of the L5 neuron was **0.03**, which is indicative of the variability in the experimental neuron. This indicates that the results achieved are meaningful keeping



(a) Mean SPIKE-distance= 0.28

(b) Mean SPIKE-distance = 0.10

Figure 3.13: Prediction results : Pre-optimization vs Post-optimization

in consideration the inaccuracies of the mathematical model itself.

As a comparison to the proposed optimization technique, the Nelder-Mead (NM) [43] method was also implemented. The same initial population (as in GD) was taken and the model parameters were estimated by minimizing the SPIKE-distance using NM. It was

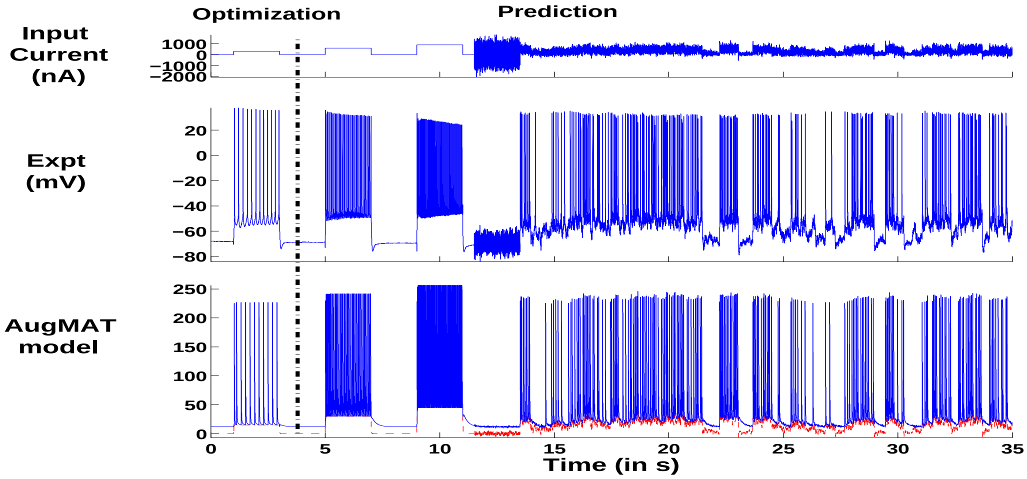


Figure 3.14: Using an optimized model to predict spiking activity of a biological neuron

Table 3.1: Prediction results (SPIKE-distance) for 100 samples

	Initial	**After GD**	**After NM**
Mean	0.28	0.10	0.13
Std. Dev.	0.16	0.08	0.12

observed that the prediction results were $\mathbf{0.13} \pm 0.12$. The GD method also took less computational time as compared to the NM method. Fig. 3.14 shows the spiking activity of the biological neuron and the optimized model. The first 4 s (left of the vertical partition line) is used for model optimization. The model (bottom row) $[\alpha_1 = 183.4, \alpha_2 = 2.53, \beta = 0.087, \omega = 11.93, \theta_0 = 58.2]$ is then simulated using the input current provided (top row) and compared to the experimental data (middle row). The results of the SPIKE-distance between the predicted and actual spike trains for the 100 samples are summarized in Table 3.1.

Chapter 4: Estimation of connectivity in neuronal population models

Mapping the brain and its complex networked structure hsa been one of the most researched topics in the field of computation neuroscience [56]. It continues to be the path towards understanding the communication between different functional regions of the brain. Neurons, the basic building blocks of the brain and central nervous system, relay information to each other via electrical and chemical connections called synapses. The mechanism of the interaction between individual neurons in the network is studied experimentally and has been described in [57] and [58]. As described in the previous chapter, with sufficient excitiation, the exchange of ions in a neuronal cell generates an action potential or a 'spike'. This also results in a post-synaptic potential (PSP) that, depending on the type of pre-synaptic neuron, is responsible for the excitation or inhibition of the post-synaptic neuron. The connectivity between different regions in the brain influences brain functions such as memory, learning, perception and motor control amongst others [59, 60]. There is also evidence of abnormal connectivity in neurological diseases such as schizophrenia [61]. The connection patterns are dynamic and there is a continuous change in the synapses. Functional tasks also alter the brain connectivity and as the brain evolves, new synapses are generated while there is degeneration of the pre-existing ones [62]. This phenomenon has been studied in various experimental settings with functional magnetic resonance imaging (f-MRI) in [63] and [64]. [65] summarizes the importance of connectivity in a neuronal network and mentions how the knowledge of the connections can help in reproducing the dynamics of a brain. Deciphering the connectivity is, therefore, essential as it takes us a step closer towards recreating brain activity. Also, accurate identification of the synapses in the brain and tracking changes in the synaptic connections is crucial in the research of brain diseases such as Parkinsons and epilepsy and in mapping the functional network of the neurons in the brain.

There are a variety of methods in published literature that are used to estimate the functional connectivity in the brain. They can be broadly classified into model based and model free approaches. As the nomenclature suggests, the model based approaches depend on an accurate model of neuronal networks. The first approaches to connectivity estimation, due to the lack of appropriate models, were model free techniques that used statistical measures to compute the influence of one neuron in the network on another. While basic correlation measures produced produced decent estimation of the connectivity structure [66], advanced methods such as the Granger causality index [67] and partial directed coherence [68] provided better results. The Cox method described in [69] is another technique that can be used to compute the synaptic connections in a network based on the spikes of the individual neurons. Recent years have also witnessed an exponential growth of studies related to the application of Graph theory to unravel characteristic features of structural, functional and effective connectivity [70]. In [71], spike train data is used to compute the connections in a discrete acycllic graph. A summary of the model free approaches can be found in [72]. While these methods have been successful and produced positive results in estimating static connections, it is unclear if the time window of membrane potential data used is appropriate for tracking changes in neuronal connectivity. [69] reports that the statistical method used needs spike train data in the otrder of 10^3 spikes. Although it is known that the dynamics of the time varying connections is much slower than the dynamics of the membrane potential, any changes in the synaptic connections within the spike train data used will not be estimated. This disadvantage makes the methods unsuitable to be applied in situations where the generation and degeneration of synapses due to various functional tasks need to be investigated.

Dynamic causal modeling (DCM) [73] is one of the most recent and successful model based approaches to connectivity estimation. It uses Bayesian inference techniques to compute the effective connectivity from neuroimaging data. f-MRIs have also been used in [74] to estimate the functional connectivity in the hippocampal area of the brain. Changes in the endogenous and task-dependent effective connectivity were computed in [75] demonstrating that DCM can be used to track time varying connections. However, this involves the use of expensive equipment and cannot be implemented in real time. Another limitation of the methods described above is that they are not able to estimate asymmetrical connectivity. They incorporate the assumption that the connections in the network are always bi-directional. This is, however, not true in the case of neuronal networks [58].

The focus is, therefore, to develop a model based technique that can use a short window of time-series data to estimate directed connections in a network of neurons as well as track the changes in the connectivity in real time. It is also essential that the method does not use any for of thresholding to compute the connection strengths, as the numerical value of connections is model dependent and cannot be fixed. It is evident that the neuronal network model plays a crucial role in the development of such algorithms. As described previously, dynamic systems have been used extensively to model the spiking patterns of individual neurons. In the last decade the focus has shifted to measurement and modeling a group of neurons due to great acceleration in experimental techniques. Improved hybrid nonlinear dynamic models have been able to successfully reproduce the membrane potential of different neurons at a micro level. Low complexity allows the simulation of large networks as shown in [20]. Post-synaptic potential dynamics are included in the model equations and the individual neurons are coupled together to reproduce the network response observed in experimental data.

[76] have implemented Kalman filters to estimate the parameters of a coupled nonlinear dynamic model from membrane potential observations. This approach meets all the requirements stated above and does not require large amounts of data to estimate the connectivity parameters. However, the convergence of the parameter estimates largely depends on the continuous time dynamics of the model used to simulate the neuronal spiking activity. A large group of neuron models involve discontinuities at the time of spikes and with a growing database of hybrid models [23] and absence of methods in current literature that

can be implemented with such models, it is essential to develop a technique that is able to produce results with the existing discontinuities. While in Chapter 3 the minimzation of the staircase error function using the gradient descent method was successful in estimating the parameters of a single neuron, such an approach would not be appropriate for the problem of connectivity estimation due to the desired properties of real time implementation and tracking changes in neuronal connections.

This chapter presents a new systems identification approach to connectivity estimation in a network of neurons.The problem is represented Fig. 4.1, where membrane potential measurements from individual neurons are used to compute the connections. Hybrid nonlinear dynamic systems are used to model the population of neurons. The recursive least squares algorithm is then applied to compute the synaptic connections from other pre-synaptic neurons in the population. Connectivity estimation is achieved even in the presence of measurement noise, model error and inaccurate assumption of post-synaptic potential dynamics. This allows us to compute the synaptic connections of the neuronal population with $N \approx 70$ neurons in a very small time window of measured electrical activity. The proposed method is also to implemented track changes in the connections making it applicable in instances where the network structure evolves with time. The performance of the method is evaluated by computing the specificity and sensitivity measures, taking into consideration the $N \times N$ connectivity matrix, for randomly generated networks.

4.1 The Model

Network models have always generated significant interest when being used to describe different physical phenomenon. They are considered extremely important as they are able to simulate the intereaction between the different individual elements. Network models exist in different mathematical forms such as statistical network models, artificial neural networks, coupled nonlinear dynamic models and are used in a variety of applications ranging from social network analysis to reconstructing the population dynamics of the neurons in the brain. As discussed previously, neurons form a complex and intricate network inside the

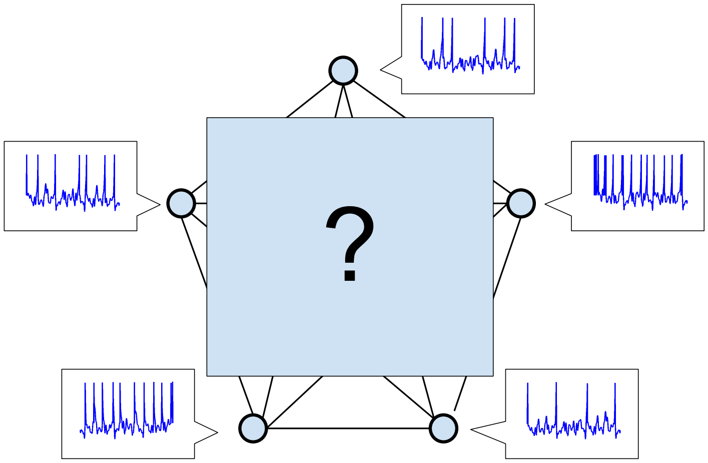


Figure 4.1: Identification of connectivity in a network

brain and central nervous system. Information is relayed through the network via action potentials. Chemical and electrical synapses connect neurons and determining the nature of the connectivity in a population of neurons is vital to understanding the brain.

The Aug-MAT model used in the previous chapter, while successful in modeling a single neuron spike response, was not applicable to network models. While it is capable of reproducing the spike times of a biological neuron, the absence of a variable that represents the membrane potential makes it difficult to extend the model to represent network dynamics by coupling the individual Aug-MAT neurons. In recent literature, Izhikevich et al. [20] describe a hybrid nonlinear dynamic system to simulate a large population of neurons. The single neuron is based on the model explained in [15] and the coupling is mathematically defined based on the conductance values and glial strengths, that act as a scaling factor to the synaptic current. The model has been able to successfully recreate the observed dynamics of different regions in the brain. In this chapter, the large scale population model referenced above has been chosen to reproduce the dynamics of a group of neurons as it is one of the most striking examples of neuronal network modeling in published literature.

This section describes the coupled nonlinear dynamic model for a network of neurons. The model is re-defined as a linear auto-regressive moving average (ARMA) model that enables the use of least squares (LS) estimation methods (Sec. 2.3) to identify the connections.

4.1.1 Single neuron model :

We use a modified version of the Izhikevich neuron, a nonlinear dynamic model that is capable to reproducing the different spiking regimes observed. The model consists of two states and eight parameters. One of the states of the model represents the spiking of the neuron and the other is a lumped variable containing the dynamics of the potassium and sodium ions in the cell. Eq. 4.1 describes the spiking dynamics of the single neuron intracellular membrane potential, $v(t)$ and the recovery variable, $u(t)$. The neuron is stimulated by an external current, $I(t)$. Sufficient stimulus results in the 'spiking' of the neuron, during which the membrane potential reaches the threshold of $30mV$. Eq. 4.2 represents a discrete one-step reset when the membrane potential exceeds the threshold.

$$\begin{aligned} \dot{v}(t) =& p_1 v^2(t) + p_2 v(t) + p_3[u(t) - p_0 I(t)] + p_4 = f_s(v(t), u(t), \mathbf{p}) \\ \dot{u}(t) =& p_5 v(t) + p_6 u(t) = g_s(v(t), u(t), \mathbf{p}) \end{aligned} \tag{4.1}$$

if $v(t) > 30mV$

$$\begin{aligned} v(t) &\to p_7 \\ u(t) &\to u(t) + p_8 \end{aligned} \tag{4.2}$$

$\mathbf{p} = [p_0, p_1, .., p_8]$ is the parameter vector associated with the model. The discrete reset makes it possible for the model to represent the different spiking patterns observed in a biological neuron, specially the bursting phenomenon. However, the presence of the discontinuity makes it difficult to apply existing estimation and tracking techniques to this model. A modified reset condition is described, that preserves the original nature of the model while at the same time makes the model tractable. The dynamic system is split into two different models, one for spiking (Eq. 4.1) and the other for resetting (Eq. 4.3). The switching condition remains the same as before ($v(t) > 30mV$). As the membrane potential exceeds the threshold, the system switches to the reset model for a pre-defined time. At the expiration of this time the system switches back to the spike model.

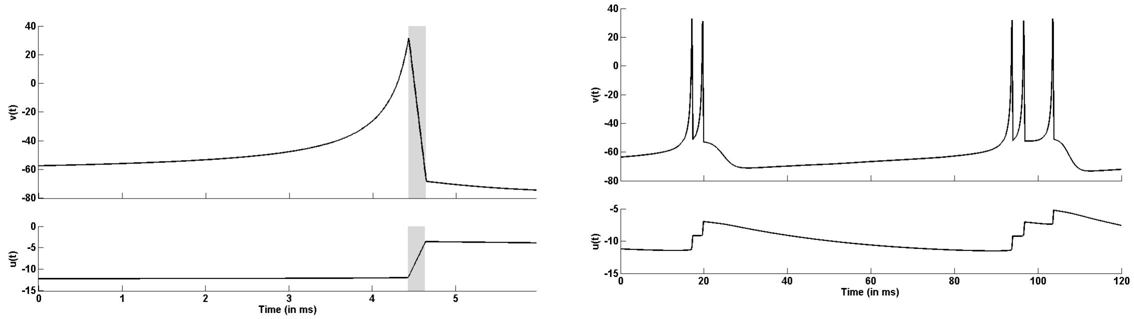
if $v(t) > 30mV$

$$\dot{v}(t) = \frac{-(30 - p_7)}{\tau_R} = f_r(\mathbf{p}); \qquad \dot{u}(t) = \frac{p_8}{\tau_R} = g_r(\mathbf{p}) \tag{4.3}$$

The time taken by the reset model is defined as τ_R. Simulations were performed to test the effect of τ_R on the sensitivity of the response to the parameters. The chosen value of $\tau_R = 0.2ms$ did not affect the spiking patterns generated by the parameter combinations reported in [15]. Fig. 4.2 graphs the membrane potential (top) and recovery variable (bottom) of the modified Izhikevich single neuron model. Fig. 4.2a, on the left, shows the dynamics of the single neuron model. The 'spike' phase and 'reset' phase are represented by the unshaded and shaded regions respectively. The horizontal axis represents a time window of 6 ms. The figure on the right represents a time window of 120 ms and shows the bursting phenomenon of a single neuron. The figure on the right, Fig. 4.2b, represents a time window of 120 ms and shows the bursting phenomenon of a single neuron.

4.1.2 Neuronal network model :

Neurons in a network are connected with each other through synapses. As the pre-synaptic neuron spikes, it provides a stimulus in the form of synaptic current to the post-synaptic neuron. There are millions of synapses that form a complex network of neurons. If the sum of synaptic currents from all the pre-synaptic neurons is sufficient, it results in the spiking of the post-synaptic neuron. The synaptic current also depends on the nature of the particular pre-synaptic neuron. Inhibitory neurons suppress the spiking activity of the



(a) Dynamics of the spike model (unshaded region) and the reset model (shaded region)

(b) ‘Bursting’ of a single neuron. The system switches between the spike model and reset model.

Figure 4.2: Simulations of a single neuron dynamics, implemented with the proposed modified reset

post-synaptic neuron by providing a negative synaptic current while excitatory neurons result in an increase in the spiking activity. Experiments have been conducted to observe the dynamics of the synaptic current. Mathematically they are modeled as weighted sums of decaying exponentials, with a ‘jump’ when the pre-synaptic neuron spikes [].

The single neuron model, described by Eqs. 4.1 and 4.3, is extended to represent the dynamics of a neuronal network. The synaptic current is modeled by a third differential equation. Eqs. 4.4 and 4.5 define the network model. $v_i(t)$, $u_i(t)$ represent the membrane potential and recovery variable of the individual neuron i. The synaptic current is a weighted sum of decaying exponentials with multiple time constants. The synaptic stimulus to the post-synaptic neuron is defined by $I_{i,syn}(t)$ as the weighted sum of the synaptic currents from all the pre-synaptic neurons. The weights, w_{ij} (i : post-synaptic, j : pre-synaptic), are numerically defined as $+1$ for excitatory, -1 for inhibitory and 0 for absence of connection. Fig. 4.3 is a block diagram representation of a single neuron in a network.

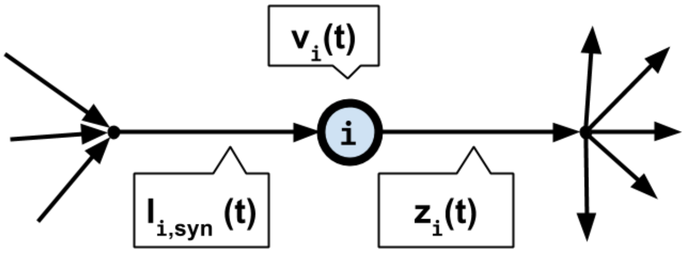


Figure 4.3: Block diagram of the i^{th} neuron in a network.

spiking of neuron i :

$$\begin{aligned}
\dot{v}_i(t) =& f_s(v_i(t), u_i(t), \mathbf{p}_i) + I_{i,syn}(t) \\
\dot{u}_i(t) =& g_s(v_i(t), u_i(t), \mathbf{p}_i) \\
\dot{s}_i(t) =& -\frac{1}{\tau} s_i(t) \\
I_{i,syn}(t) =& \frac{g}{N} \sum_{j=0, j\neq i}^{N} w_{ij} s_j(t)
\end{aligned} \tag{4.4}$$

if $v_i(t) > 30mV$ (reset)

$$\dot{v}_i(t) = \frac{-(30 - p_{i7})}{\tau_R} = f_r(\mathbf{p}_i); \qquad \dot{u}_i(t) = \frac{p_{i8}}{\tau_R} = g_r(\boldsymbol{p}_i); \qquad \dot{s}_i(t) = \frac{1}{\tau_R} \tag{4.5}$$

$\mathbf{p_i} = [p_{i0}, .., p_{i8}]$ represents the parameter vector of the individual neuron i. $I(t)$ is the external input current that stimulates all N neurons of the population. g represents the glial strength that affects the synaptic current, $I_{i_{syn}}(t)$, of the post-synaptic neuron. w_{ij} is the strength of the synaptic connection from neuron i to neuron j. The self-connections, w_{ii} are assumed to be 0. τ is the time constant of the decay of post-synaptic potential. It is empirically chosen as $10ms$, based on the population model described in [20]. τ_R, as described above is the reset time equal to $0.2ms$. The functions f_s, g_s, f_r and g_r are defined in Eqs. 4.1 and 4.3.

Fig. 4.4 shows the dynamics of a single neuron in a neuronal population of twenty neurons. The top row is the membrane potential, middle row is the recovery variable and the bottom row represents the synaptic current that modulates according to the action potentials (spikes) of the neuron. After the neuron spikes, the synaptic current decays exponentially with a fixed time constant. Fig. 4.5 shows the membrane potentials of all neurons in the population. It is observed that the spiking activity is irregular. The spike rate of each neuron depends on the characteristics and type of the neuron, as well as the connectivity to the other excitatory and inhibitory neurons in the network.

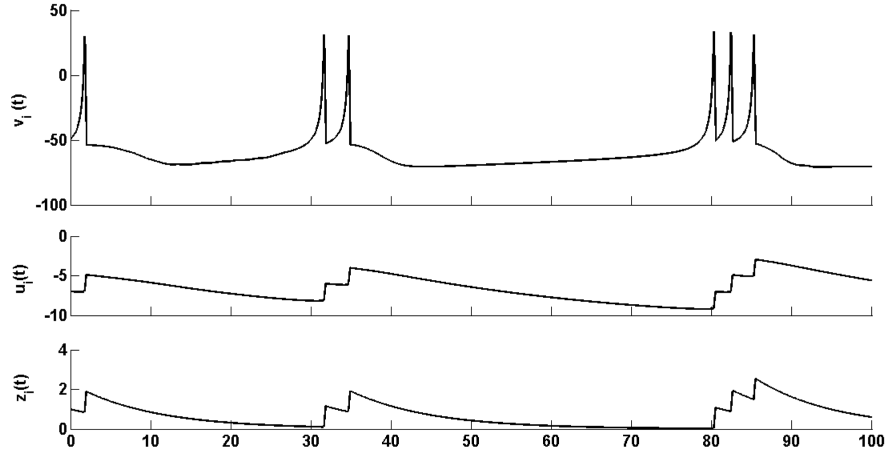


Figure 4.4: Dynamics of a single neuron in a network model

Fig. 4.6 represents the connectivity of the population. The rows and columns represent the

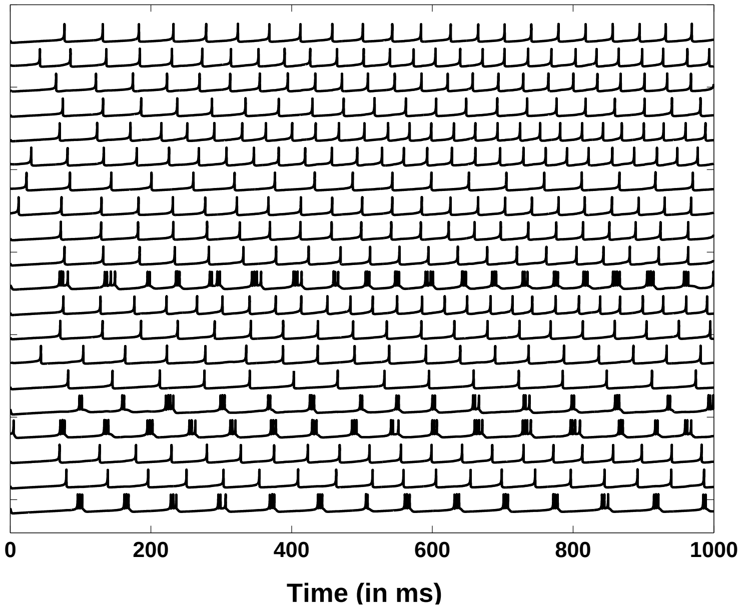


Figure 4.5: Spiking activity in a network of 20 neurons

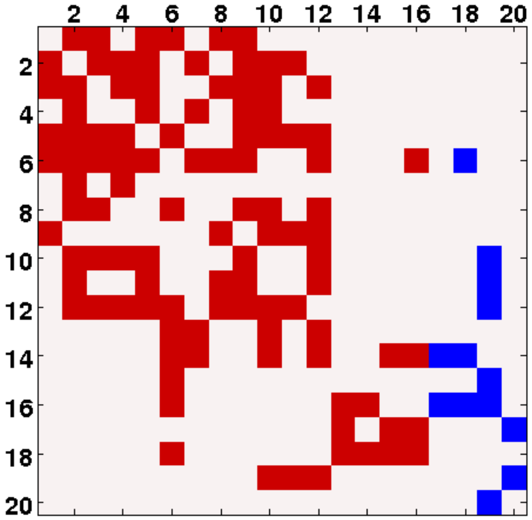


Figure 4.6: Connectivity of a network of 20 neurons

pre-synaptic and post-synaptic neuron numbers respectively. The synaptic strengths from the pre-synaptic neuron i to the post-synaptic neuron j, w_{ij} in Eq. 4.4, is shown by the (i, j) entry in the matrix (i^{th} row and j^{th} column). 80% of the neurons in the population are classified as excitatory and the remaining 20% as inhibitory. In the connectivity matrix used in the model equations, the connections from the pre-synaptic excitatory neurons are quantified as $+1$ (red squares, columns 1 to 16) and -1 is used to indicate the connection from an inhibitory pre-synaptic neuron (blue squares, columns 17 to 20). The absence of a connection between two neurons is represented by a white space in the figure and a weight value of 0 in the model. It can also be seen from the figure that the connectivity matrix is not symmetrical. This is representative of a population of neurons in which the presence of a synapse from neuron i to neuron j does not ensure a connection from neuron j to neuron i. Methods that compute the correlatiion matrix from individual membrane potential measurements are unable to capture this feature of neuronal connectivity.

4.1.3 Linear system (discrete time) :

Computation of unknown parameters in a linear system using a least squares based estimation approach is well established. This method can be extended to nonlinear dynamic models if the following requirments can be met - (a) the model can be redefined such that it is linear in the parameter space and (b) the redefined inputs and outputs are in terms of observable and reconstructed state variables. Fig. 4.7 represents a network of 3 neurons. The connections are represented by the w_{ij} blocks that act as a gain to the post-synaptic potentials, $s_i(k)$. The external input, $I(k)$, stimulates all the neurons of the network.

In the proposed model of a neuronal population, the membrane potentials of the individual neurons are observed and the synaptic currents are reconstructed from the knowledge of the spike times of the neurons. The objective is, therefore, to eliminate the unobserved recovery variable, $u_1(t)$, from the model equations. Eqs. 4.6-4.12 outline the formulation of the nonlinear 'spiking' model of a neuronal population, Eq. 4.4, as a linear auto-regressive moving average (ARMA) system. The derivation is shown for a single neuron ($i = 1$) in a

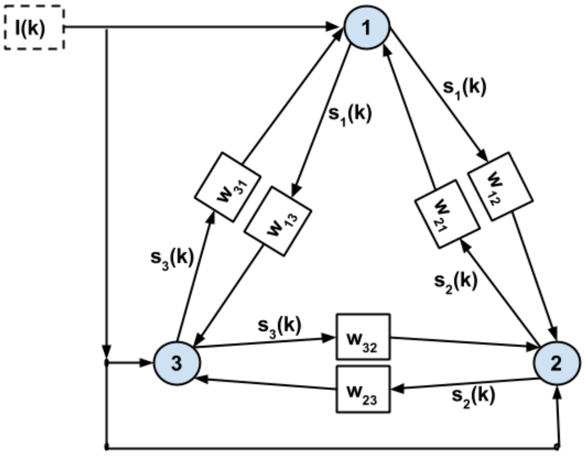


Figure 4.7: Block diagram representation of a neuronal network

network of three neurons ($N = 3$).

The spiking dynamics of neuron 1 in the network is defined by the following differential equations.

$$\dot{v}_1(t) = p_{11}v_1^2(t) + p_{12}v_1(t) + p_{13}[u_1(t) - p_{10}I(t)] + p_{14} + \frac{g}{3}(w_{12}s_2(t) + w_{13}s_3(t)) \tag{4.6a}$$

$$\dot{u}_1(t) = p_{15}v_1(t) + p_{16}u_1(t) \tag{4.6b}$$

$$\dot{s}_1(t) = -\frac{1}{\tau}s_1(t) \tag{4.6c}$$

We use forward eulers method [77] to approximate the first derivative terms. The constant time-step is chosen as $T = 0.01$ ms and the time instances are defined as $t_k = kT$.

$$\dot{x}(t_k) = \frac{x(t_k) - x(t_{k-1})}{t_k - t_{k-1}} \tag{4.7}$$

Eq. 4.6 can be written in discrete time as

$$
\begin{aligned}
v_1(k) =& v_1(k-1) + T\{p_{11}v_1^2(k-1) + p_{12}v_1(k-1) + p_{13}[u_1(k-1) - p_{10}I(k-1)] \\
& + p_{14} + \frac{g}{3}(w_{12}s_2(k-1) + w_{13}s_3(k-1))\} \qquad (4.8a) \\
u_1(k) =& u_1(k-1) + T\{p_{15}v_1(k-1) + p_{16}u_1(k-1)\} \qquad (4.8b) \\
s_1(k) =& s_1(k-1) + T\{-\frac{1}{\tau}s_1(k-1)\} \qquad (4.8c)
\end{aligned}
$$

The $u_1(k-1)$ term in Eq. 4.8a is expanded according to Eq. 4.8b to eliminate the recovery variable, $u_1(k)$, from the equations and express the membrane potential, $v_1(k)$, as an auto-regressive moving average (ARMA) [78] model that contains only observable states. The membrane potential equation is then defined as

$$
\begin{aligned}
v_1(k) =& v_1(k-1) + T\{p_{11}v_1^2(k-1) + p_{12}v_1(k-1) + p_{13}[u_1(k-2) \\
& + T\{p_{15}v_1(k-2) + p_{16}u_1(k-2)\} - p_{10}I(k-1)] + p_{14} \\
& + \frac{g}{3}(w_{12}s_2(k-1) + w_{13}s_3(k-1))\} \qquad (4.9) \\
=& (1 + p_{12}T)v_1(k-1) + p_{15}T^2v_1(k-2) + p_{11}Tv_1^2(k-1) \\
& + (1 + p_{16}T)Tp_{13}u_1(k-2) + p_{14} + \frac{g}{3}(w_{12}s_2(k-1) + w_{13}s_3(k-1))
\end{aligned}
$$

The time delayed recovery variable, $u_1(k-2)$, can be expressed in terms of the membrane potential and synaptic currents. Rearranging the terms of Eq. 4.8a after applying a delay gives us

$$
\begin{aligned}
p_{13}Tu_1(k-2) =& v_1(k-1) - v_1(k-2) - p_{11}Tv_1^2(k-2) - p_{12}Tv_1(k-2) \\
& + p_{13}p_{10}TI(k-2) - p_{14}T - \frac{gT}{3}(w_{12}s_2(k-2) + w_{13}s_3(k-2))
\end{aligned} \qquad (4.10)
$$

Eq. 4.10 is substituted in Eq. 4.9

$$
\begin{aligned}
v_1(k) =& [2 + (p_{12} + p_{16})T]v_1(k-1) + [-1 - (p_{12} + p_{16})T + p_{15}T^2]v_1(k-2) \\
& + [p_{11}T]v_1^2(k-1) + [-p_{11}T]v_1^2(k-2) + [-p_{13}T]I(k-1) \\
& + [p_{13}T + p_{13}p_{16}T^2]I(k-2) + [-p_{14}p_{16}T^2] \\
& + \left[\frac{gT}{3}\right] w_{12}s_2(k-1) + \left[-\frac{gT}{3} - \frac{p_{16}gT^2}{3}\right] w_{12}s_2(k-2) \\
& + \left[\frac{gT}{3}\right] w_{13}s_3(k-1) + \left[-\frac{gT}{3} - \frac{p_{16}gT^2}{3}\right] w_{13}s_3(k-2)
\end{aligned}
\tag{4.11}
$$

The nonlinear differential equation model, described in Eq. 4.6 can now be expressed as an ARMA system, Eq. 4.12. The nonlinear term, $v_1(k)^2$, is defined as an input to the system. Let $vs_1(k) = v_1^2(k-1)$.

$$
A_1(z)v_1(k) = B_1(z)vs_1(k) + C_{1,2}(z)s_2(k-1) + C_{1,3}(z)s_3(k-1) + D_1(z)I(k-1) + d_1 \tag{4.12}
$$

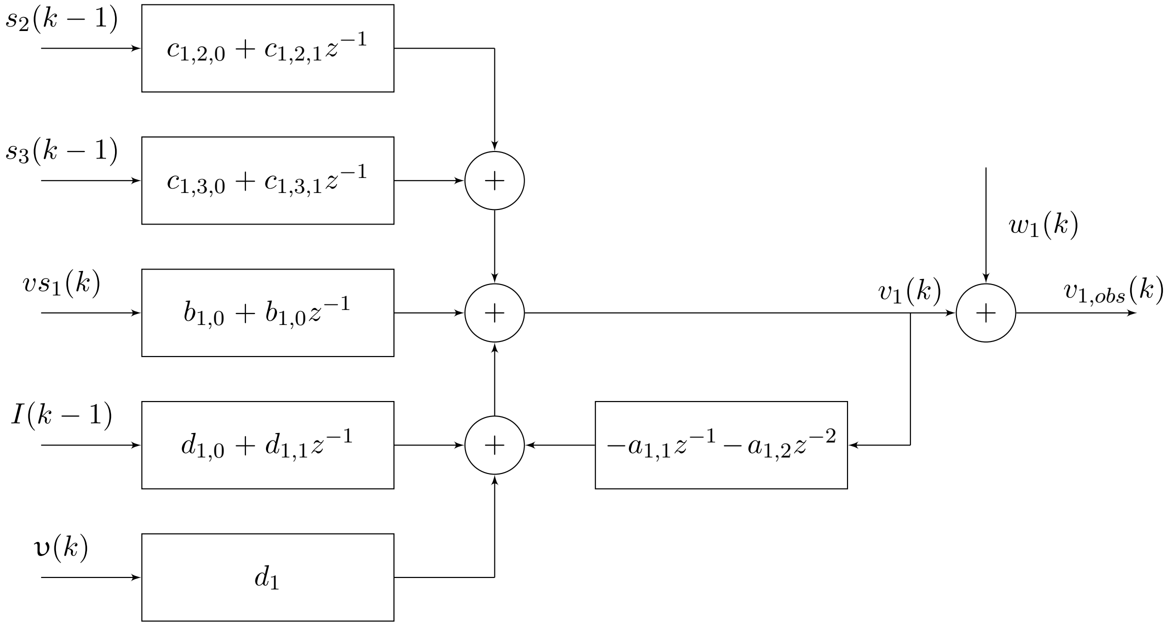
where

$$
A_1(z) = 1 + a_{1,1}z^{-1} + a_{1,2}z^{-2}, \quad B_1(z) = b_{1,0} + b_{1,1}z^{-1},
$$

$$
C_{1,j}(z) = c_{1,j,0} + c_{1,j,1}z^{-1}, \quad D_1(z) = d_{1,0} + d_{1,1}z^{-1}
$$

The following equations represent the relation between the linear system parameters and the nonlinear model parameter set, $\mathbf{p_1} = [p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, w_{12}, w_{13}]^T$. The linear system parameters depend on the discretization time-step, T (0.01 ms). The higher order terms of T are neglected.

$$
\begin{aligned}
&a_{1,1} = -2 - (p_{12} + p_{16})T, \quad a_{1,2} = 1 + (p_{12} + p_{16})T - p_{15}T^2 \approx 1 + (p_{12} + p_{16})T \\
&b_{1,0} = p_{11}T, \quad b_{1,1} = -p_{11}T \\
&c_{1,j,0} = \left[\frac{gT}{3}\right] w_{1j}, \quad c_{1,j,1} = \left[-\frac{gT}{3} - \frac{p_{16}gT^2}{3}\right] w_{1j} \approx \left[-\frac{gT}{3}\right] w_{1j} \\
&d_{1,0} = -p_{13}p_{10}T, \quad d_{1,1} = p_{13}p_{10}T + p_{13}p_{10}p_{16}T^2 \approx p_{13}p_{10}T \\
&d_1 = -p_{14}p_{16}T^2
\end{aligned}
\tag{4.13}
$$



$s_2(k-1)$
$c_{1,2,0} + c_{1,2,1}z^{-1}$
$s_3(k-1)$
$c_{1,3,0} + c_{1,3,1}z^{-1}$
$+$
$w_1(k)$
$vs_1(k)$
$b_{1,0} + b_{1,0}z^{-1}$
$+$
$v_1(k)$
$+$
$v_{1,obs}(k)$
$I(k-1)$
$d_{1,0} + d_{1,1}z^{-1}$
$+$
$-a_{1,1}z^{-1} - a_{1,2}z^{-2}$
$\upsilon(k)$
d_1

Figure 4.8: Block diagram representation of a single neuron in a 3-neuron network

Fig. 4.8 represents the block diagram schematics of the dynamics of a single neuron in a 3-neuron network as a linear ARMA system. $s_2(k-1)$ and $s_3(k-1)$ are the synaptic currents of neurons 2 and 3 respectively, and act as inputs to neuron 1. $vs_1(k)$ is the squared membrane potential (nonlinear term) and is re-defined as an input to the system. The external current stimulus to the network is $I(k)$ and $\upsilon(k)$ is a constant step input to the system. The output of the system is the membrane potential, $v_1(k)$. The measurement noise, represented by $w_1(k)$, is white gaussian. From Eq. 4.13 we can conclude that the $C_j(z), B(z)$ and $D(z)$ approximate first difference filters. The gain of these filters depends on the parameters of the original nonlinear dynamic model (Eq. 4.4).

The three neuron example (Eqs. 4.11-4.13, Fig. 4.8) can be extended to derive the linear ARMA model for every individual neuron, i, in a network of N neurons.

$$A_i(z)v_i(k) = B_i(z)vs_i(k)^2 + \sum_{j=1,j\neq i}^{N} C_{i,j}(z)s_j(k-1) + D_i(z)I(k-1) + d_i \tag{4.14}$$

The system has a total of $(N+2)$ inputs. There are $(N-1)$ inputs from other neurons in the network ($s_j(k)$), 2 external inputs ($I(k)$ and $u(k)$) and the squared membrane potential, $vs_i(k)$, re-defined as an input. The system has $(N+2)$ FIR filters and a total of $(2N+5)$ filter coefficients to be estimated. The measured membrane potential is assumed to be corrupted by zero mean white Gaussian observation noise.

$$v_{i,obs}(k) = v_i(k) + w_i(k) \tag{4.15}$$

4.2 Parameter estimation using model generated test data

Parameter estimation in linear systems is a field that has been extensively researched. There are a variety of existing system identification algorithms that have performed well in tests of accuracy, speed and robustness [79]. However, the performance of such techniques also depend on the dynamics of the system. The linear model described in Sec. 4.1.3 is an ARMA system with multiple inputs and a single output. In this section data generated from the model is used to estimate the parameters by minimizing the mean squared error.

The prediction error least squares (PE-LS) error algorithm is formulated for Eq. 4.14 and its performance is tested for accuracy and robustness. It is important to note that for the typical parameters of the neuron model ($\mathbf{p}_i$ in Sec. 4.1.2) the poles of the linear system, roots of the polynomial $A(z)$, lie outside the unit circle. This makes the system unstable and difficult to implement the Output Error algorithm as discussed in [80]. Also, since the performance of the PE-LS algorithm with pre-filtered data was found to be excellent for the model in question, other unbiased identification algorithms such as the instrumental variable approach were not investigated. The estimated value of the parameters are compared with the known true values and the bias and variance of the estimates in the presence of observation noise is investigated.

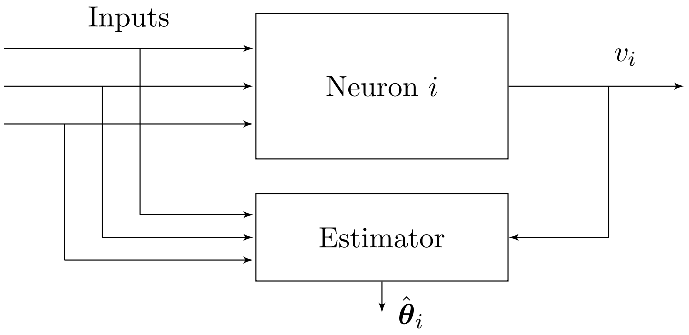


Figure 4.9: Schematic representation of parameter estimation for neuron i

Fig. 4.9 shows the block diagram representation of the parameter estimation. The algorithms used for the estimation are discussed in the following sections.

4.2.1 Prediction error least squares estimation :

The linear ARMA system of a neuronal network (Eq. 4.14) can be written in terms of a parameter vector, $\boldsymbol{\theta}_i$, and a regressor, $\boldsymbol{\varphi}_i(k)$.

$$\begin{aligned}\boldsymbol{\varphi}_i(k) = [&vs_i(k), vs_i(k-1), s_j(k-1), s_j(k-2), I(k-1), I(k-2), u(k),\\ &- v_i(k-1), -v_i(k-2)]^T\end{aligned} \tag{4.16}$$

$$\boldsymbol{\theta}_i = [b_{i,0}, b_{i,1}, c_{i,j,0}, c_{i,j,1}, d_{i,0}, d_{i,1}, d_i, a_{i,1}, a_{i,2}]^T \tag{4.17}$$

where $j \in \{1, 2, \cdots, N\}, j \neq i$ and $vs_i(k) = v_i^2(k-1)$.

$u(k)$ represents the step input.

The membrane potential (output) of the i^{th} neuron in the network can be written as

$$v_i(k) = \boldsymbol{\varphi}_i(k)^T \boldsymbol{\theta}_i \tag{4.18}$$

From Eq. 4.15 the observed membrane potential can be expressed as

$$v_{i,obs}(k) = \boldsymbol{\varphi}_i(k)^T \boldsymbol{\theta}_i + w_i(k) \tag{4.19}$$

The prediction of the output is defined as

$$\hat{v}_i(k) = \boldsymbol{\varphi}_{i,obs}(k)^T \hat{\boldsymbol{\theta}}_i(k) \tag{4.20}$$

where $\hat{\boldsymbol{\theta}}_i$ is the estimate of the parameter vector and $\varphi_{i,obs}(k)$ represents the regressor (Eq. 4.16) constructed from the observed variable, $v_{i,obs}(k)$.

The error function $\epsilon_i(k)$ is defined as the difference between the measured output, $v_{i,obs}(k)$ and the predicted output, $\hat{v}_i(k)$. The mean squared error (MSE) of neuron i is represented by ξ_i.

$$\epsilon_i(k) = v_{i,obs}(k) - \hat{v}_i(k) \tag{4.21a}$$

$$\xi_i = E[\epsilon_i(k)^2] \tag{4.21b}$$

The formula for LS estimate is derived in Sec.2.3.

$$\boldsymbol{Q}_i = E\left[\boldsymbol{\varphi}_i(k)\boldsymbol{\varphi}_i(k)^T\right] \tag{4.22a}$$

$$\boldsymbol{P}_i = E\left[\boldsymbol{\varphi}_i(k)v_{i,obs}(k)\right] \tag{4.22b}$$

$$\hat{\boldsymbol{\theta}}_i = \boldsymbol{Q}_i^{-1}\boldsymbol{P}_i \tag{4.22c}$$

Recursive storage (using inter-spike intervals) -

The PE-LS algorithm is implemented with recursive data collection. The observed membrane potential data in between two consecutive spikes is used to construct the regressor. The estimate obtained is stored. The ‘reset’ of the membrane potential (immediately after the spike) is disregarded, and data is collected over the next inter-spike interval. The first few estimates are ignored to account for the transcience and the remaining estimates obtained for all the ISIs are averaged to calculate the steady state parameter estimates.

Fig. 4.10 represents the trace of membrane potential data that is recorded (blue). k_s is the discrete time instance (integer value) of the s^{th} spike, represented by $\{k_1, k_2, k_3\}$ in the figure. The value of the membrane potential is greater than 30 at these instances, $v_{i,obs}(k_s) > 30$. The time of spike in ms can be computed as $t_s = k_sT$. The reset time, τ, is used to calculate the number of data points that need to be disregarded (red).

The $\boldsymbol{Q}$ and $\boldsymbol{P}$ matrices are computed recursively over each segment of inter-spike interval (ISI) data. The linear system described in Eq. 4.14 defines the dynamics of the spiking model

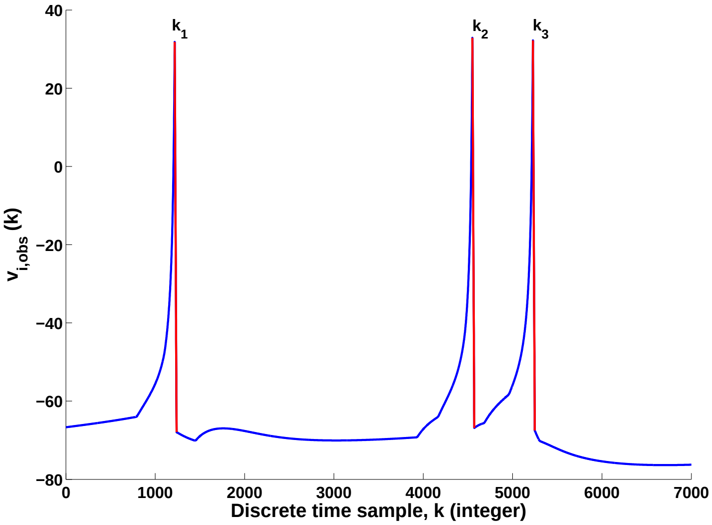


Figure 4.10: Construction of regressor from observed membrane potential data.

illustrated in Eq. 4.4 and shown by the blue trace in the above figure. The regressor and output vectors are constructed as

$$\begin{aligned}\boldsymbol{\varphi}_i(k) =&[vs_{i,obs}(k), vs_{i,obs}(k-1), s_j(k-1), s_j(k-2),\\ &I(k-1), I(k-2), \upsilon(k), -v_{i,obs}(k-1), -v_{i,obs}(k-2)]^T \qquad (4.23a)\\ y_i(k) =&v_{i,obs}(k) \qquad (4.23b)\end{aligned}$$

The recursive computation of the $\boldsymbol{Q}$ and $\boldsymbol{P}$ matrices are shown in Eq. 4.25. As each new segment of data is available a 'forgetting factor', λ, is implemented to reduce the influence of past data on the parameter estimate. The parameter vector is updated at each spike. This ensures that the algorithm adapts to changes in the model parameters. Also, perturbations in the observed variable do not have a permanent effect on the parameter estimates. The disadvantage of adding the 'forgetting factor' is that the algorithm is more sensitive to noise. The trade-off is discussed in Sec. 4.2.4 and the value of λ is chosen after repeated simulation exercises.

The matrices for the current ISI are computed as

$$\boldsymbol{q}_{i,s} = \sum_{k=k_{s-1}+\frac{\tau}{T}}^{k_s} \boldsymbol{\varphi}_i(k)\boldsymbol{\varphi}_i(k)^T \tag{4.24a}$$

$$\boldsymbol{p}_{i,s} = \sum_{k=k_{s-1}+\frac{\tau}{T}}^{k_s} \boldsymbol{\varphi}_i(k)y_i(k) \tag{4.24b}$$

The current matrices are then added to the older matrices (from past ISI data) with exponentially decreasing weight values.

$$\boldsymbol{Q}_{i,s} = \boldsymbol{q}_{i,s} + \lambda\boldsymbol{Q}_{i,s-1} \tag{4.25a}$$

$$\boldsymbol{P}_{i,s} = \boldsymbol{p}_{i,s} + \lambda\boldsymbol{P}_{i,s-1} \tag{4.25b}$$

The parameter estimate at every spike is computed as

$$\hat{\boldsymbol{\theta}}_{i,s} = \boldsymbol{Q}_{i,s}^{-1}\boldsymbol{P}_{i,s} \tag{4.26}$$

The inputs to the linear system need to satisfy the persistence excitation conditions to ensure that the matrix $Q_{i,s}$ is invertible. The truncation of the regressor elements at each spike time results in a finite signal that prevents the Q matrix from being ill-conditioned and approaching singularity. Also, the nonlinear relation between the inputs of the linear ARMA system (Fig. 4.8) results in the absence of a linear dependance amongst the elements of the collected ISI data. Computing the condition number (ratio of maximum eigenvalue to the minimum eigenvalue) of the Q matrix for different simulations showed that the persistence excitation conditions had been met and the inverse matrix was computed accurately.

4.2.2 3-neuron example (noise free measurements, fixed connections) :

The system identification algorithm described in the previous section is used to compute the linear system parameters of a network of 3 neurons (Eq. 4.12). The algorithm is implemented with a forgetting factor of 1. The observed membrane potential is not corrupted by noise. The input stimulus to the neurons was a sinusoidal current,

$$I(k) = 5\sin(4\pi k) + 3\sin(10\pi k + \pi/3)$$

Convergence of linear ARMA system parameter estimates -

Fig. 4.11 shows the convergence of the parameters of the linear ARMA system (Eq. 4.12) corresponding to a neuron in the network. The membrane potential of the neuron is recorded and the estimates are updated at each spike time. The final values of the estimated parameter vector are compared in Table 4.1. In the absence of observation noise all parameters are estimated accurately, indicating that the data is sufficiently conditioned for implementation of LS techniques. The condition number of $\boldsymbol{Q}$ was found to be of the order of 10^{10}.

Table 4.1: Parameter estimates of the linear ARMA system (Neuron 1 in a 3-neuron network) with noise free measurements

	True Value	**Recursive PE-LS**
$a_{1,1}$	-2.0498	-2.0498
$a_{1,2}$	1.0498	1.0498
$b_{1,0}$	0.0004	0.0004
$b_{1,1}$	-0.0004	-0.0004
$c_{1,2,0}$	0.0333	0.0333
$c_{1,2,1}$	-0.0333	-0.0333
$c_{1,3,0}$	0.0333	0.0333
$c_{1,3,1}$	-0.0333	-0.0333
$d_{1,0}$	-0.01	-0.01
$d_{1,1}$	0.01	0.01
d_1	0.0003	0.0003

Fig. 4.11 shows the comparison of the recursive storage LS algorithm estimates with the true

parameter values. The figures represent a 200 ms time window. The parameter estimate at each spike (computed using the ISI data) is denoted by a ○. The true value of the parameter is indicated by a horizontal line. It is concluded that the algorithm performs well in terms of accuracy and is able to achieve convergence in one ISI.

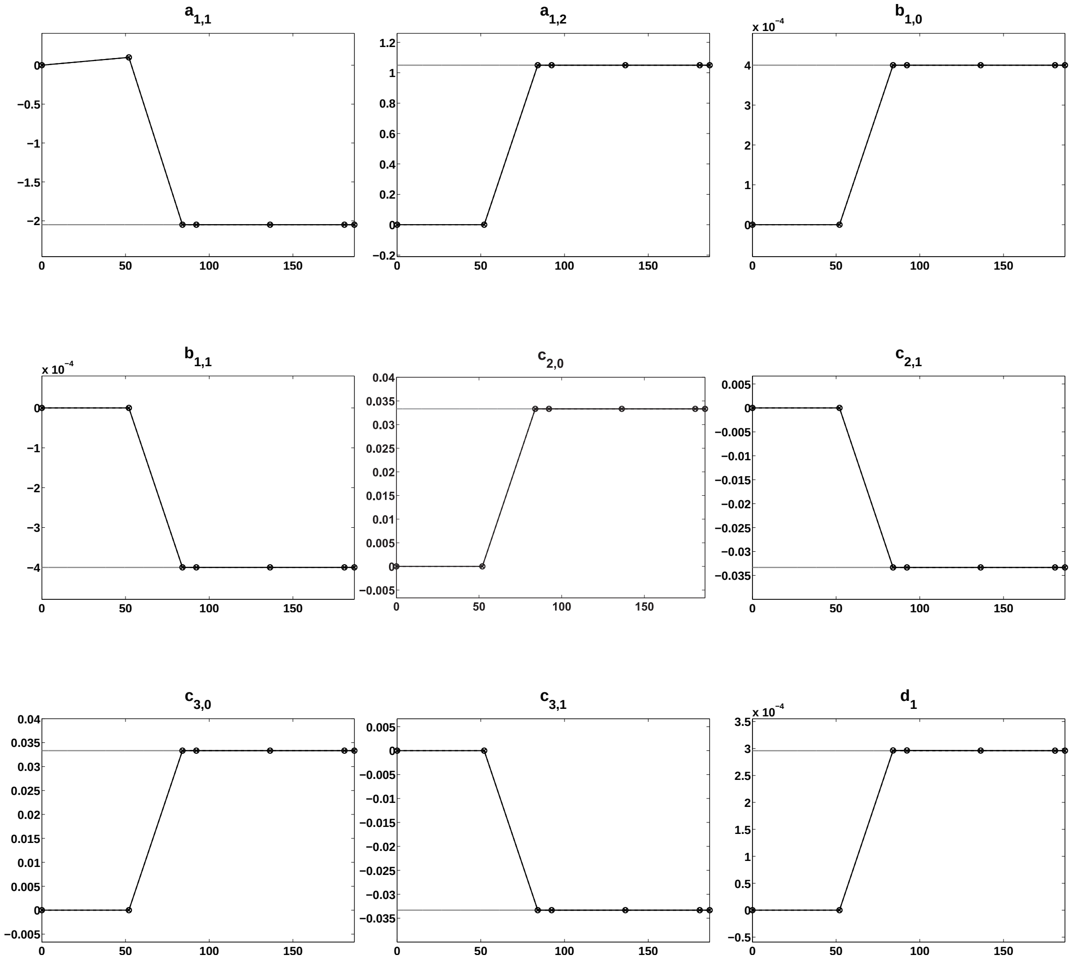


Figure 4.11: Convergence of all nine parameters of the 3-neuron linear regression model with noise free measurements

Transformation to nonlinear model parameter space -

In Table 4.1 and Fig. 4.11 presented results of the convergence of the linear system parameters are presented. The objective is, however, to estimate the parameters of the original nonlinear dynamic model. Eq. 4.13 defines the relation between the linear model parameter vector, $\boldsymbol{\theta}_i$ and the nonlinear model parameter vector, $\boldsymbol{p}_i$. Selected nonlinear model parameters are calculated based on the respective equations.

$$\hat{p}_{i1} = \frac{\hat{b}_{i0} - \hat{b}_{i1}}{T} \tag{4.27}$$

$$\hat{p}_{i3}\hat{p}_{i0} = \hat{p}_{i3}^2 = \frac{\hat{d}_{i1} - \hat{d}_{i0}}{T} \tag{4.28}$$

$$\hat{w_{ij}} = \frac{N}{gT}(\hat{c}_{i,j,0} - \hat{c_{i,j,1}}) \tag{4.29}$$

It is observed that the remaining nonlinear model parmaeters form an overdetermined set of nonlinear equations. To solve for p_{i2}, p_{i4}, p_{i5} and p_{i6} based on the estimate of $\boldsymbol{\theta}_i$, we use the nonlinear least squares algorithm to minimize the weighted quadratic function shown in Eq. 4.30.

$$\{\hat{p}_{i2}, \hat{p}_{i4}, \hat{p}_{i5}, \hat{p}_{6}\} = \arg \min_{\hat{\boldsymbol{p}}_i} \quad \frac{1}{2}\, \mathbf{f}(\hat{\boldsymbol{\theta}}_i, \hat{\boldsymbol{p}}_i)^T \, \boldsymbol{W} \, \mathbf{f}(\hat{\boldsymbol{\theta}}_i, \hat{\boldsymbol{p}}_i) \tag{4.30}$$

where the function vector $\mathbf{f}(.)$ represents the equality constraint for the nonlinear optimization problem (for a 3-neuron network).

$$\mathbf{f}_3(\hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{p}}_1) = \begin{bmatrix} a_{1,1} + 2 + (p_{12} + p_{16})T \\ a_{1,2} - 1 - (p_{12} + p_{16}) + p_{i5}T^2 \\ d_1 + p_{14}p_{16}T^2 \end{bmatrix} \tag{4.31}$$

The matrix W is a diagonal matrix consisting of the weights assigned to each individual equality condition. The weights are chosen such that the contribution of each equality condition is of the same order.

The estimated parameters of the nonlinear dynamic model (Eq. 4.4, Neuron 1) are reported in Table 4.2 and are compared to the true values. The results show that $\{p_{13}, p_{14}, p_{15}\}$ are not estimated accurately due to non-unique solutions to the set of equations defined by $\mathbf{f_3}(.) = 0$. The parameters w_{12} and w_{13} are of particular importance. They quantify the connection from pre-synaptic neurons 2 and 3 to the post-synaptic neuron 1 and are estimated correctly.

Table 4.2: Parameter estimation results for the 1^{st} neuron in a network of 3 neurons

	True Value	**Estimated Value**
p_{11}	0.040	0.040
p_{12}	5.000	5.000
p_{13}	-1.000	-1.000
p_{14}	140.0	9.7
p_{15}	0.004	0.03
p_{16}	-0.020	-0.024
w_{12}	1.000	1.000
w_{13}	-1.000	-1.000

4.2.3 3-neuron example (measurement noise added, fixed connections) :

In the previous section the linear model parameters were estimated in the absence of noise in the observation. It is essential to analyze the performance of the algorithm when the observation is corrupted by noise. Observation noise (white, gaussian) is added to the membrane potential, $v(t)$. The output and the regressor is constructed from the observed membrane potential (Eq. 4.15) and is used to compute the estimate of the parameter vector. PE-LS algorithms (batch and recursive storage) provide biased estimates in the presence of observation noise. To reduce the bias, we apply low pass filters to the input and output measurements. The filtered measurements are then used to construct the regressor and compute the LS estimate of the parameter vector. The filter reduces the bias in the LS estimate by improving the signal-to-noise power. The schematic is shown in Fig. 4.12.

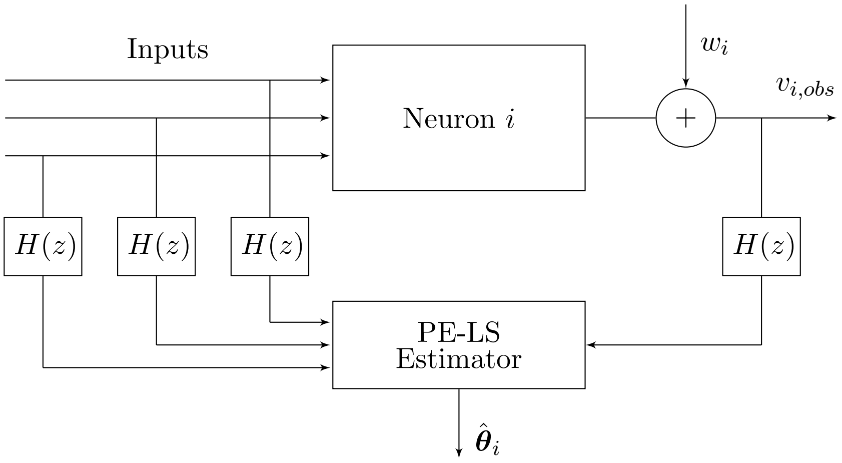


Figure 4.12: Schematic representation of prediction error least squares estimation in presence of observation noise for neuron i

A second order system is chosen to pre-filter the inputs and outputs.

$$H(z) = \frac{1}{(1 - 0.99z^{-1})^2} \tag{4.32}$$

Convergence of linear model parameter estimates -

Fig. 4.13 shows the convergence of mean square error for a 3-neuron model. The top two rows represent the PSP from pre-synaptic neurons 2 and 3 that act as inputs to the post-synaptic neuron 1 system. The observed and true membrane potential is plotted in the third row. The ratio of the observation noise to signal variance is 0.1. The fourth row shows the pre-filtered data where the noise to signal variance is reduced to 0.001. It is observed that the mean square error converges close to the minimum value in two to three ISIs. The convergence of the parameter estimates takes longer and is shown in Fig. 4.14. The recursive storage PE-LS algorithm was used (black line). The circles represent the parameter estimate at each spike, using the ISI data from the previous to the current spike. The grey line indicates the true value of the parameter. The variance of noise was 10% of the signal variance before pre-filtering. The low-pass filter reduced the ratio to 2%.

Table 4.3 summarizes the results of the PE-LS algorithm for increasing noise levels. The

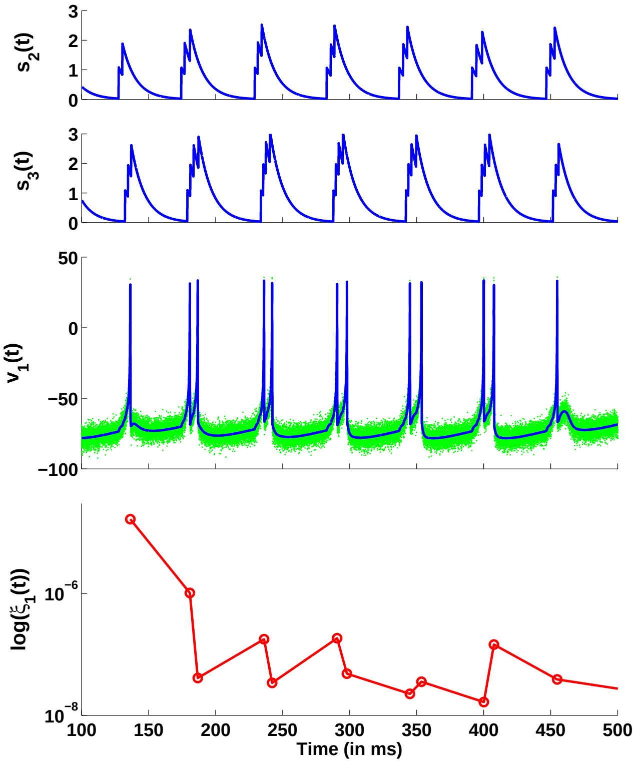


Figure 4.13: Convergence of the mean square error for a post-synaptic neuron in a 3-neuron network

ratio of noise to signal variance after pre-filtering is reported.

Table 4.3: Parameter estimates of the linear regression model (Neuron 1 in a 3-neuron network) with white gaussian noise added to the measurements. $\frac{\sigma_w^2}{\sigma_v^2} = 0.1$

	True Value	**Recursive PE-LS**
$a_{1,1}$	-2.0498	-2.0428
$a_{1,2}$	1.0498	1.0428
$b_{1,0}$	0.0004	0.0003
$b_{1,1}$	-0.0004	-0.0003
$c_{1,2,0}$	0.0333	0.0275
$c_{1,2,1}$	-0.0333	-0.0275
$c_{1,3,0}$	0.0333	0.0283
$c_{1,3,1}$	-0.0333	-0.0283
d_1	0.0003	0.0008

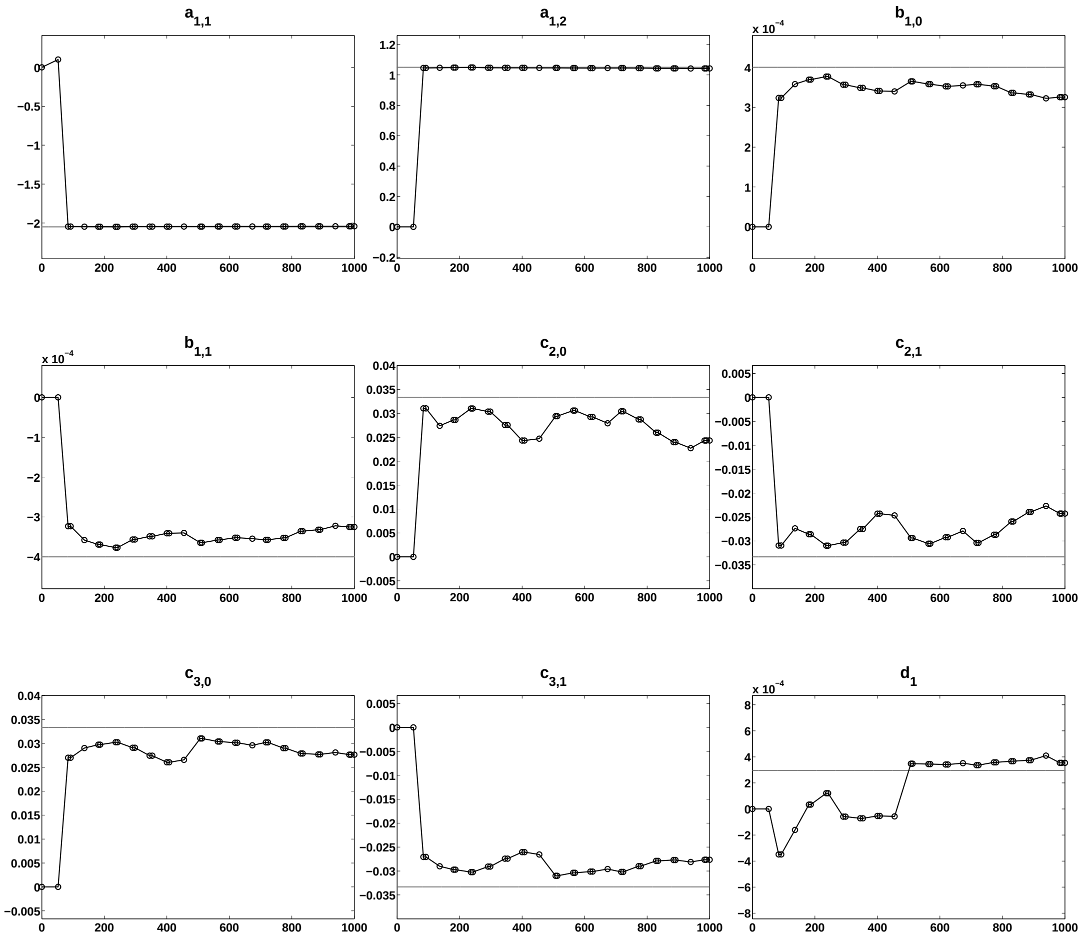


Figure 4.14: Convergence of all nine parameters of Neuron 3 in a 3-neuron linear regression model with noisy measurements.

Transformation to nonlinear model parameter space -

Eqs. 4.27-4.31 are solved using the linear model parameter estimates obtained in Table 4.3. The results presented in Table 4.2 show that all parameters are not computed correctly. This is again due to the presence of local minimas in the non-convex objective function defined in Eq. 4.30. However, the parameters of interest - the connection strengths (w_{12} and w_{13}) - are estimated with reasonable accuracy.

Table 4.4: Parameter estimates of an individual neuron in a network of 3 neurons. PE-LS estimates were used to compute the nonlinear model parameters.

	True Value	**Estimated Value**
p_{31}	0.040	0.033
p_{32}	5.000	4.674
p_{33}	-1.000	0.013
p_{34}	140.0	155.8
p_{35}	0.004	0.18
p_{36}	-0.020	-0.003
w_{31}	1.000	0.825
w_{32}	1.000	0.849

4.2.4 3-neuron example (time-varying connections) :

In the previous section the recursive storage PE-LS algorithm produced biased estimates for linear model parameters when the observations were corrupted by white noise. The level of bias was reduced by pre-filtering the data generated from a 3-neuron network model (Eqs. 4.4,4.5).The strengths of the synaptic connections in the network were fixed. This is however not true in actual neuronal networks where the nature of the synapses can vary with time and its dynamics are much slower than the dynamics of the neuronal spiking. In this section the PE-LS algorithm is implemented with a forgetting factor. λ to track the time varying connections between neurons in the network. The choice of λ is discussed with the help of multiple simulations with varying noise levels and perturbation. The dynamics of the connections do not represent the changing synaptic connections of an actual network of neurons. The purpose here is purely to analyze the least squares algorithm for the model defined in Sec. 4.1.

Fig. 4.15 shows the tracking of time-varying connections in a noise-free environment. There was no noise added to the observed membrane potentials. The true value of the connections is shown by the black trace. The blue, red and green traces represent the estimates with forgetting factors as 0.5, 0.7 and 0.9 respectively. The changes in the connections are a combination of step changes as well as gradual sinusoids. The PE-LS recursive algorithm is successfully able to track the changing synaptic strengths over a 10s time window. It is observed that, as expected, a lower forgetting factor ($\lambda = 0.5$) is more suitable for tracking changes in the connection parameters.

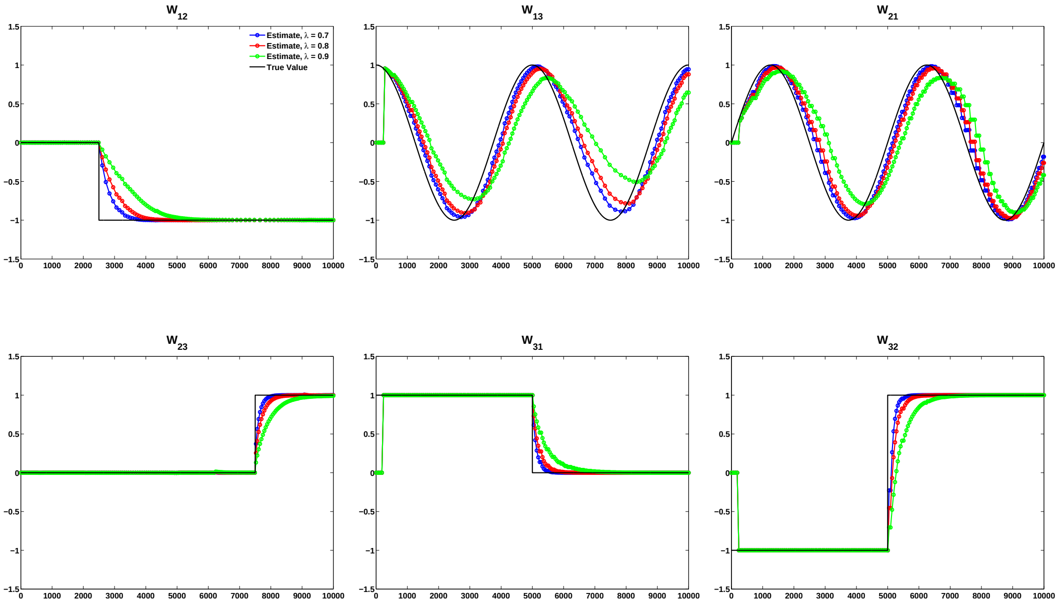


Figure 4.15: Tracking of connections in a 3-neuron network

Addition of measurement noise to the observed membrane potentials of the individual neurons introduces a bias and variance in the estimate. The effect of the forgetting factor on the estimates is shown in Figs. 4.16,4.17. While a smaller forgetting factor enables the algorithm to adapt more quickly to the change in connections, it also makes it more sensitive to perturbations. The results clearly show that while $\lambda = 0.9$ is much slower in converging to the actual value, there is no siginificant improvement of $\lambda = 0.5$ over $\lambda = 0.7$. The variance in the estimates for $\lambda = 0.5$ is very high and it is difficult to justify the convergence of the parameters to steady state values. The results are compared for difference noise levels and the forgetting factor is chosen as $\lambda = 0.7$.

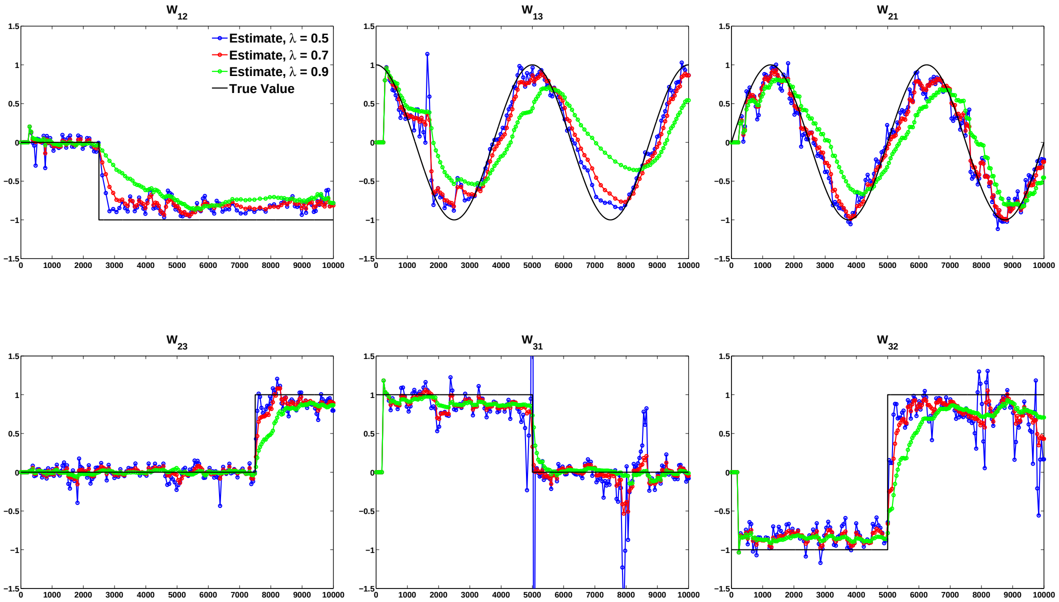


Figure 4.16: Tracking of connections in a 3-neuron network. Ratio of noise to signal variance (after pre-filtering) = 0.002

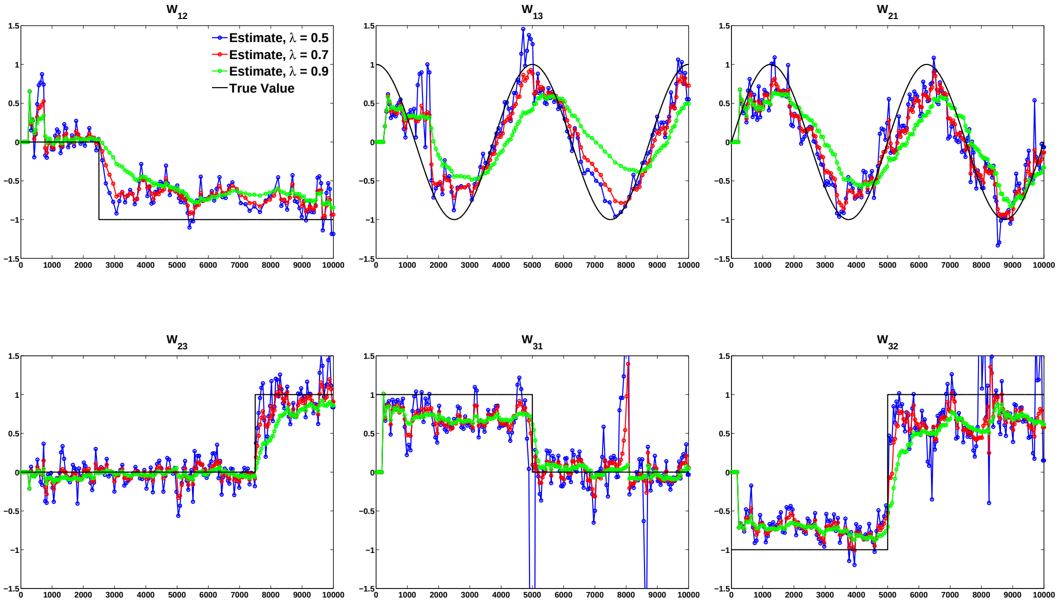


Figure 4.17: Tracking of connections in a 3-neuron network. Ratio of noise to signal variance (after pre-filtering) = 0.01

4.2.5 Results and statistical testing :

Successful estimation of a 3-neuron network was shown in the previous sections. The method is extended to larger networks and the effect of the number of neurons on the accuracy is investigated in this section. The number of linear model parameters (N_θ) increases linearly with increase in network size (N), as shown in Eq. 4.14.

$$N_\theta = 2N + 3 \tag{4.33}$$

Figs. 4.18 and 4.19 show the convergence of connection estimates for networks of $N = 20$ and $N = 30$. Few selected post-synaptic neurons are represented from each population. For each neuron, the red trace represents an estimate for connection from an excitatory pre-synaptic neuron, blue for a connection from an inhibitory pre-synaptic neuron and green for the absence of a synaptic connection. The linear model parameters of each neuron are estimated using the recursive storage PE-LS technique and the connection values are computed using Eq. 4.29. White noise was added to the membrane potential observations and the noise to signal variance was 0.05. The figures show that convergence time is short and steady state value is achieved in 500 ms. This typically corresponds to 20-30 ISIs of the neuron. The plots show that while there is a degradation in the convergence time with increase in the network size, steady state estimation is achieved in reasonable time.

Fig. 4.20 illustrates the absolute error, $\epsilon_{i,j} = |w_{i,j} - \hat{w}_{i,j}|$, between the estimated and actual connections matrix for networks of different sizes. 10% white noise was added to the observed membrane potentials for each simulation. The trend of the absolute error in connectivity estimates with increasing network size can be observed. The number of connections identified incorrectly increases, as represented by the darker colored squares. However, the error percentage remains within acceptable limits for systems with large number of parameters (60-100). The matrices (a),(b) and (c) are all plotted to the same color scale. The effect of noise on the estimates is shown in Fig. 4.21. The error bars for absolute

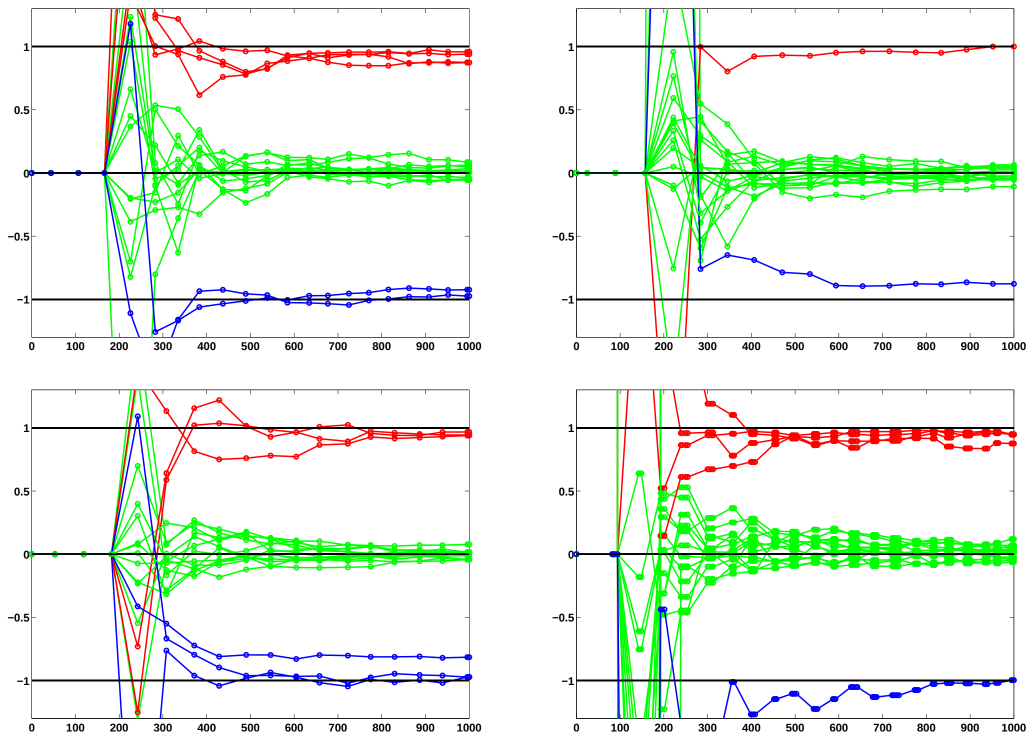


Figure 4.18: Convergence of connection parameters in a network of 20 neurons

errors is shown for networks ranging from $N = 10$ to $N = 50$ neurons. The noise to signal variance is represented on the horizontal axis. It is observed that the error in the estimates increases with increase in network size. The increase in the number of linear model parameters results in a larger variance in the final estimates of all connections in the network. The mean, however, remains within satisfactory limits for large networks and high noise to signal variance. Table 4.5a reports the maximum error for each case while the percentage of connections estimated with an error less than 20% is shown in Table 4.5b. We conclude that while the accuracy of the estimates reduces with increase in network size, the PE-LS technique is able to correctly identify a high percentage of connections (allowing for a small error).

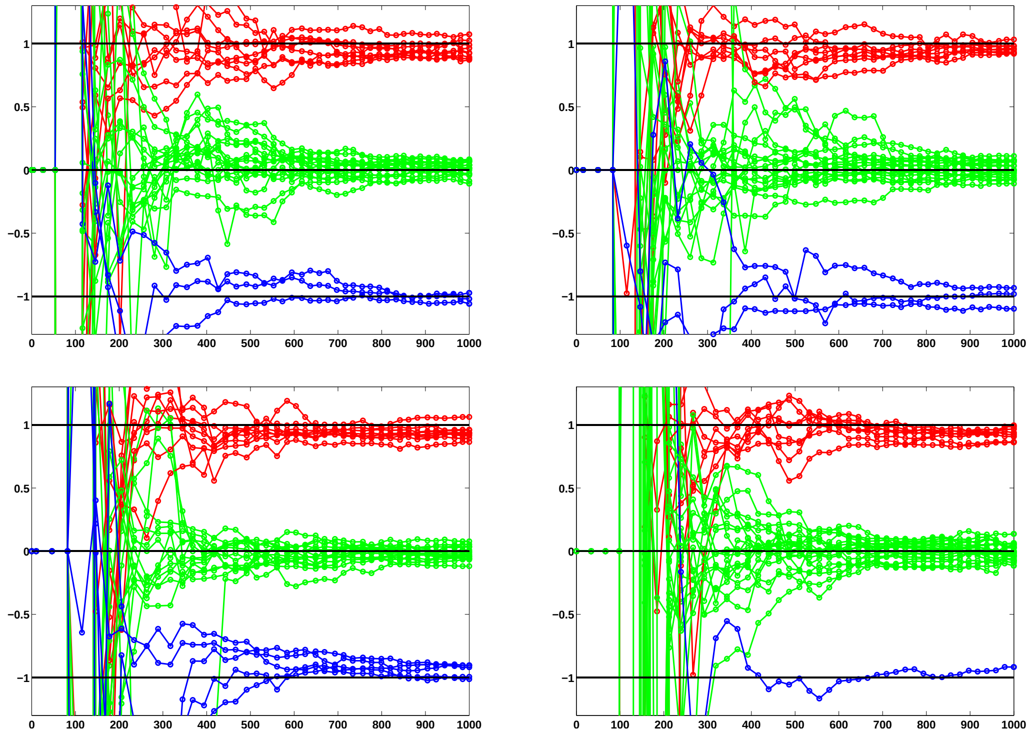
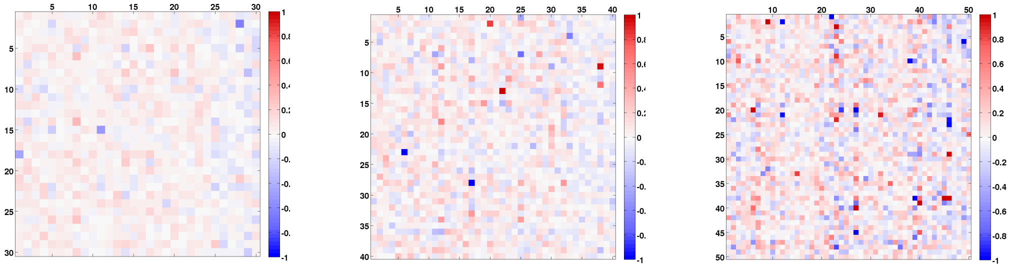


Figure 4.19: Convergence of connection parameters in a network of 30 neurons



(a) $N = 30$

(b) $N = 40$

(c) $N = 50$

Figure 4.20: Error in connectivity estimation for networks of different sizes. Ratio of noise to signal variance for all networks was 0.001 after pre-filtering.

4.3 Identification of connectivity in a different neuronal model

In the previous section the least squares system identification approach was tested on model generated data. Following the results that showed accurate estimation of the connection

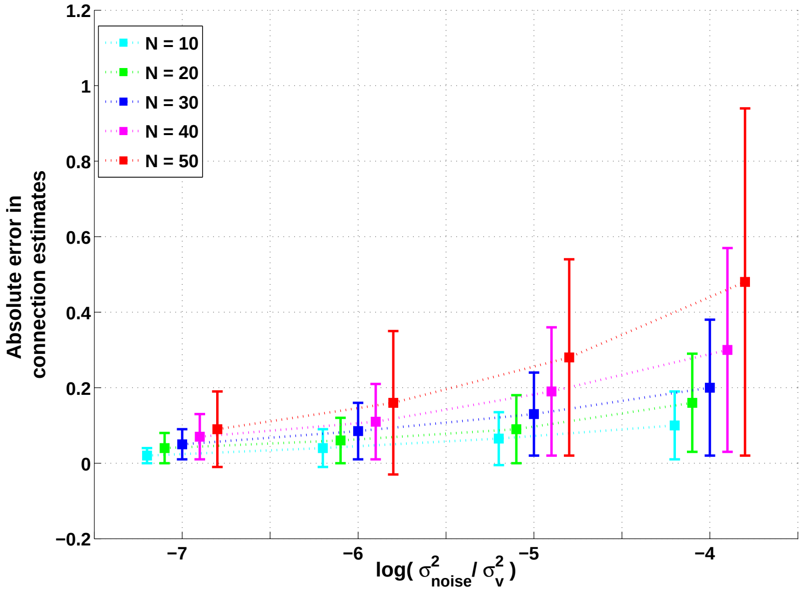


Figure 4.21: Error bars for connection estimates of different networks and varying noise levels

Table 4.5: Statistics for connectivity estimates with varying network sizes and noise levels

(a) Maximum Error

$\frac{\sigma^2_{noise}}{\sigma^2_v} \rightarrow$	10^{-6}	10^{-5}	10^{-4}	10^{-3}
$N = 10$	0.10	0.32	0.40	0.44
$N = 20$	0.40	0.51	0.62	0.67
$N = 30$	0.47	0.75	0.95	1.70
$N = 40$	0.84	1.12	1.41	2.10
$N = 50$	1.21	2.16	3.18	4.02

(b) Percentage of connection estimates with $\epsilon < 0.2$.

$\frac{\sigma^2_{noise}}{\sigma^2_v} \rightarrow$	10^{-6}	10^{-5}	10^{-4}	10^{-3}
$N = 10$	100	99	98	85
$N = 20$	99	97	87	75
$N = 30$	99	90	77	62
$N = 40$	96	84	69	50
$N = 50$	89	75	49	35

parameters for large networks, it is important to test the robustness of the proposed technique in the presence of unknown membrane potential and post-synaptic potential dynamics. There are multiple neuronal models present in existing literature that, while capable of producing the required spiking behavior, have significantly different dynamics to the model considered in Eq. 4.4. This creates a model error while constructing the linear ARMA system, Eq. 4.14, using the membrane potential observations. As a result, the least square estimate of the parameter vector is not reliable and the connectivty estimates are not accurate. The difference in dynamics between the mathematical model and the biological neuron is only bridged in bio-plausible models such as the Hodgkin-Huxley (HH). However, as discussed in Chapter 3, the use of non bio-plausible models to construct network simulations is more relevant because of low computational cost. It is therefore essential that a hybrid dynamical system such as the Izhikevich model is used to identify the connections of the network when the membrane potential dynamics are unknown.

Experimental evidence [35, 81] suggests that the charateristics of a neuron is captured in the spiking patterns. Extending this idea, a two step approach is applied to estimate the connectivity of the neuronal population. The first step involves synchronization of the spikes of the model with the spikes in the membrane potential observations. The proposed system identification technique is then applied on the synchronized model data to compute the parameters of every individual neuron as well as the synaptic connections from all pre-synaptic neurons. In this section the Hindmarsh-Rose [82] and Hodgkin-Huxley [11, 83] models are used to generate membrane potentials in an interconnected neuronal network (50 neurons with 40 % connectivity). The pre-synaptic neurons are classified as excitatory (80 %) and inhibitory (20 %). The results of the estimated connectivity matrix for random networks and networks with directed connections are reported. As the majority of the connections in the network are identified correctly, it can be suggested, with a certain degree of confidence, that the proposed method can be applied to data obtained from biological neurons.

4.3.1 Spike synchronization using ensemble Kalman filter :

State and parameter dynamics -

An ensemble Kalman filter is implemented to estimate the state variables and parameters of an individual neuron. The discretized states of the nonlinear dynamic model can be derived from Eq. 4.4.

$$\boldsymbol{x}_i(k+1) = g(x_i(k), \boldsymbol{p}_i(k), I_{i,syn}(k), I(k)) + \boldsymbol{\nu}_x(k) \tag{4.34}$$

The state vector, $\boldsymbol{x}_i(k) = [v_i(k), u_i(k)]^T$, represents the membrane potential and recovery variable of the Izhikevich model. $\boldsymbol{\nu}_x(k)$ is the process noise that is assumed to be white gaussian with zero mean. The external input to the neuron, $I(k)$, is assumed to be known and observable.

The parameter vector, $\boldsymbol{p}_i(k)$, and synaptic input, $I_{i,syn}(k)$, are assumed to have much slower dynamics than the state variables.

$$\begin{aligned} \boldsymbol{p}_i(k+1) =& \boldsymbol{p}_i(k) + \boldsymbol{\nu}_p(k) \\ I_{i,syn}(k+1) =& I_{i,syn}(k) + \nu_I(k) \end{aligned} \tag{4.35}$$

The vector combining the states and parameters, $\boldsymbol{X}$, and its covariance is defined as

$$\begin{aligned} \boldsymbol{X} &= \left[\boldsymbol{x}_i^T, \boldsymbol{p}_i^T, I_{i,syn}\right]^T \\ \boldsymbol{P}(k) &= \begin{bmatrix} cov\{\boldsymbol{x}_i(k)\} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & cov\{\boldsymbol{p}_i(k)\} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & var\{I_{i,syn}(k)\} \end{bmatrix} \end{aligned} \tag{4.36}$$

The output of the Kalman filter is a weighted function of the membrane potential. Successful synchronization of the model with the observations is achieved by using a nonlinear kernel such that the output is weighted more in the neighborhood of the spikes.

$$y_i(k) = h(\boldsymbol{x}_i(k)) = (v_i(k) + 100)e^{\frac{v_i(k)-30}{\tau}} + \nu_y(k) \tag{4.37}$$

where $\tau = 50ms$.

$\boldsymbol{\nu}(k) = \left[\boldsymbol{\nu}_x(k)^T, \boldsymbol{\nu}_p(k)^T, \nu_I(k)\right]^T$ is the process noise vector. The individual variances are tuned to provide an accurate estimate of the membrane potential, $v_i(k)$. The covariance of the process noise vector is represented by $\boldsymbol{Q}_\nu$.

$$\boldsymbol{Q}_\nu(k) = \begin{bmatrix} cov\{\boldsymbol{\nu}_x(k)\} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & cov\{\boldsymbol{\nu}_p(k)\} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & var\{\nu_I(k)\} \end{bmatrix} \tag{4.38}$$

As the connectivity of the network is unknown, the dynamics for the synaptic current, $I_{i,syn}(t)$ cannot be used. The sum of the external current and synaptic input is treated as a single variable in the model dynamics within the Kalman filter framework.

The observed output used by the Kalman filter to provide the correction to the parameter and state estimates is generated by applying the nonlinear kernel function from Eq. 4.37 to the observed membrane potential.

$$y_{i,obs}(k) = h(v_{i,obs}(k)) \tag{4.39}$$

Covariance and state propagation -

The Unscented Kalman filter [32], described in Sec. 2.2, is implemented to estimate the membrane potential and synchronize the spikes of the model and observed data. The ensemble is created using the matrix square root of the state covariance. The algorithm applies the unscented transform to generate the predicted values of each ensemble members and computes the weighted mean and covariance. Weighing the mean greater than the other ensemble members has shown to produce better results [30].

The *aposteriori* states and covariances are then computed using the Kalman gain, $\boldsymbol{K}$.

$$\begin{aligned}\boldsymbol{K}(k) =& \boldsymbol{P}_{xy}(k+1|k)P_{yy}(k+1|k)^{-1} \\ \hat{\boldsymbol{X}}^{a}(k+1|k+1) =& \hat{\boldsymbol{X}}^{a}(k+1|k) + \boldsymbol{K}(k)[y_{obs}(k) - \hat{y}(k+1|k) \\ \boldsymbol{P}^{a}(k+1|k+1) =& \boldsymbol{P}^{a}(k+1|k) - \boldsymbol{K}(k)\boldsymbol{P}_{xy}(k+1|k)\end{aligned} \tag{4.40}$$

Comparison of observed membrane potential and filter estimate -

Fig. 4.22 shows the results of spike synchronization using the ensemble Kalman filter. The membrane potential is shown in the top left subplot, output of nonlinear kernel in the bottom left and parameter estimates on the right. The estimates are shown in red, observations in green and the true value in blue. The parameters converge to minimize the error between the observed output, Eq. 4.39 and the filter output, Eq. 4.37 (shown in the bottom left subplot). The comparison of the membrane potential estimate (red) and its true value (blue) is shown in the top left subplot. The green trace represents the observations of the membrane potential corrupted by white noise with zero mean. It can be seen that the spikes of the filter estimate is synchronized with the observations. However, as shown in Fig. 4.23, there remains a significant difference in the dynamics of the membrane potential of the two models. The unshaded region represents the spike model, Eq. 4.4, and the shaded region represents the reset model, Eq. 4.5. The spike model trace of the membrane potential is used to identify the parameters of the neuron model.

4.3.2 Estimating the post-synaptic potential dynamics

Fig. 4.24 shows the membrane potential and post-synaptic potential of the individual neuron model described in Sec. 4.1.2. Empirical analysis of the dynamics, treating the membrane

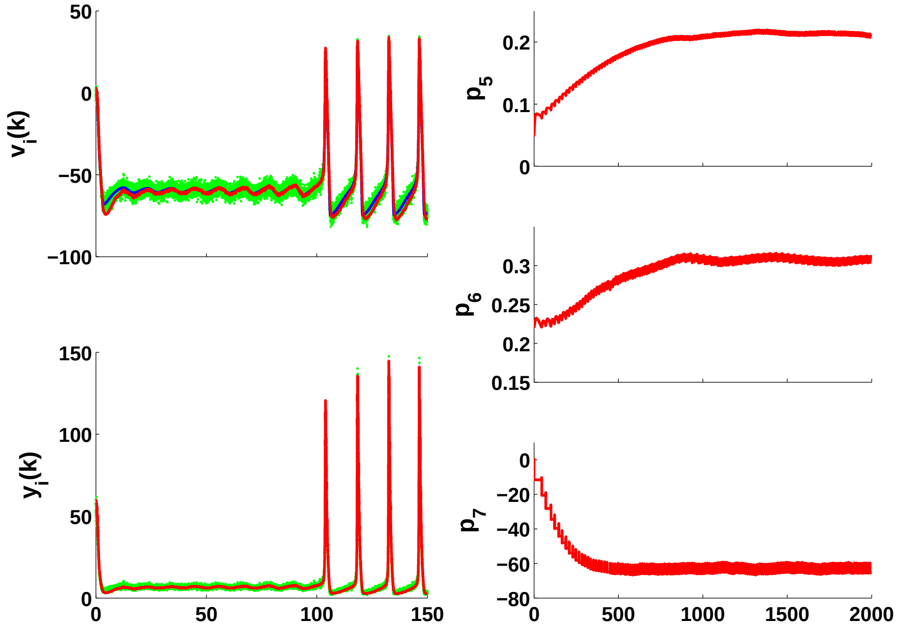


Figure 4.22: Estimation results of the ensemble Kalman filtering.

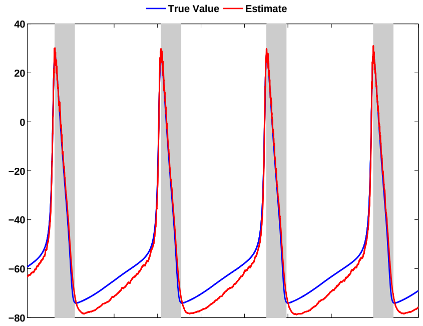


Figure 4.23: Comparison of observed and estimated membrane potentials.

potential as an input and the PSP as an output, showed that the response can be approximated with a first order system. The PSP, $s_i(k)$, (bottom row) is generated as a first order response of the spike train (middle row). The membrane potential of the neuron is plotted in the first row.

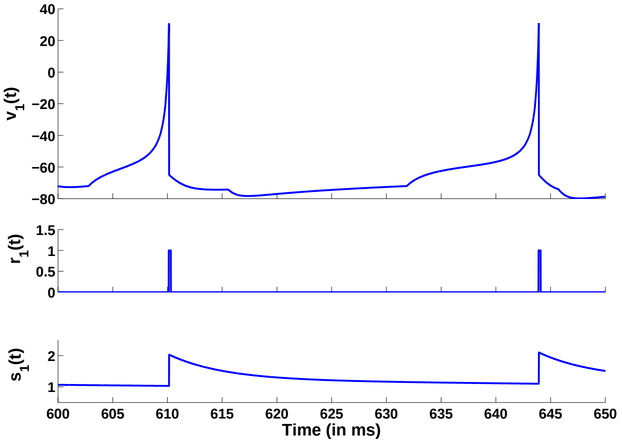


Figure 4.24: Representing the pos-synaptic potential as an output of the spike train pulse.

The spike train pulse, $r_i(k)$, shown in the middle row is an artificially generated rectangular pulse signal that is high when the neuron spikes and low otherwise. The pulse width is a pre-defined parameter equal to the reset time of the neuron model, τ_R, described in Eq. 4.3. The input-output relationship of $r_i(k)$ and $s_i(k)$ can be represented by a gain, σ_i and filter coefficient, η_i.

$$s_i(k) = \frac{\sigma_i}{1-\eta_i z^{-1}} r_i(k), \;\; 0 < \eta_i < 1 \tag{4.41}$$

Substituting the above equation in the ARMA system representation described in Sec. 4.1.3 we get

$$\begin{aligned} A_i(z)v_i(k) =& B_i(z)vs_i(k)^2 \\ &+ \sum_{j=1, j\neq i}^{N} C_{i,j}(z)\frac{\sigma_j}{1-\eta_j z^{-1}} r_j(k-1) \\ &+ D_i(z)I(k-1) + d_i \end{aligned} \tag{4.42}$$

The regressor used for least squares estimation is then defined as

$$\begin{aligned}\boldsymbol{\varphi}_i(k) = [&vs_i(k), vs_i(k-1), vs_i(k-2),\\ &r_j(k-1), r_j(k-2),\\ &I(k-1), I(k-2), I(k-3),\\ &u(k), -v_i(k-1), -v_i(k-2), -v_i(k-3)]^T\end{aligned}$$

The synaptic weight w_{ij}, can be computed from the elements in the estimated parameter vector corresponding to $r_j(k-1)$ and $r_j(k-2)$ in the regressor $\boldsymbol{\varphi}_i$.

4.3.3 Clustering of connectivity estimates

The presence of model error and a non-observable input (PSP) to the neuron introduces a scaling factor in the connection estimates. The final value of the weights cannot be decoupled from the gain of the PSP, σ_j. However, the scaling factor remains constant for connections from all the pre-synaptic neurons to the specific post-synaptic neuron. To ensure that the connection estimates of different post-synaptic neurons in the network are in the same range, the steady state values are normalized and clustered using the k-means algorithm [84] in MATLAB (`kmeans`). The steady state values of the estimates are computed by averaging the data points over the past 30 ISIs. They are then classified into three clusters with centers at 1, -1 and 0 signifying an excitatory pre-synaptic neuron, inhibitory pre-synaptic neuron and no synaptic connection respectively. The clustered connectivity matrix is then compared to the true connectivity and the results are quantified using sensitivity and specificity measures.

$$\begin{aligned}sensitivity =& \frac{TP}{TP+FN}\\ specificity =& \frac{TN}{FP+TN}\end{aligned} \tag{4.43}$$

where $\{TP, FN, FP, TN\}$ denote the number of true positives, false negatives, false positives and true negatives respectively. In numerical terms, a true positive is a $+1$ or -1 connection being identified correctly and a false positive is a 0 connection being identified as a $+1$ or -1. Similarily true negatives and false negatives relate to the 0 connection, the absence of a synapse.

The next section includes results of connectivity estimation for a network of (a) Hindmarsh-Rose (HR) neurons and (b) Hodgkin-Huxley (HH) neurons. The equations for both nonlinear dynamic systems are included in the appendix. The dynamics of both models are, however, significantly different when compared to the dynamics of the Izhikevich neuron. Results show that inspite of the presence of model error and observation noise, the identification algorithm is able to correctly estimate the connectivity of the network.

4.3.4 Results :

The identification of the connectivity in a network of neurons discussed in this chapter deals with the implementation of the prediction error least squares algorithm to estimate the linear model parameters. The derivation of the linear ARMA model from the nonlinear model equations is crucial to the convergence of the parameter estimates. After the formulation of the regressor and the output, Eqs. 4.8-4.14, the proposed identification algorithm involves the following steps.

1. Synchronization of the model membrane potential with the observed membrane potential using the Unscented Kalman filter (Sec. 4.3.1).

2. Pre-filter the ARMA model output and regressor data (Eq. 4.32).

3. Use the recursive storage PE-LS method (Eqs. 4.24-4.26) to generate the linear model parameter estimates.

4. Convert the linear model parameter estimate to the nonlinear neuron model parameter estimates, specifically the connection parameters, based on the derived relation (Eqs. 4.27-4.31)

5. Normalize and cluster the connection estimates (Sec. 4.3.3).

Data is generated using HR and HH neurons for networks of different sizes. The post-synaptic potential is modeled as a first order response to the spike pulses. Each network had 40 % connectivity. 80 % of the neurons were classified as excitatory and the remaining 20 % as inhibitory. White noise was added to the observed membrane potentials of all individual neurons in the network. The ratio of noise to signal variance was maintained at 0.1. The results were based on 100 simulations with randomly generated connections for each simulation. The convergence of the connection estimates (for four selected post-synaptic neurons) is shown in Figs. 4.25 and 4.27. The ellipses represent the three clusters - red for excitatory, blue for inhibitory and green for no connection). A mismatch in the color of the trace and the ellipse in which it converges indicates that the particular connection was not identified correctly. It was observed that convergence of the parameters was achieved in less than 1000 ms which corresponded to 20-40 ISIs depending on the nature of the particular post-synaptic neuron. The estimated connectivity matrix is then compared to the true connectivity (Figs. 4.26,4.28) for different network sizes ($N = 20, 30, 40$). All images are plotted on the same color scale and the sensitivity and specificity calculations are reported in Tables 4.6 and 4.7. Results do not indicate a significant deterioration with increase in network size. The model equations used to generate data are included in the appendix.

Table 4.6: Statistical results for neuronal network data generated from HR model

	$N = 10$	$N = 20$	$N = 30$	$N = 40$	$N = 50$
Sensitivity	0.99 ± 0.01	0.97 ± 0.03	0.93 ± 0.03	0.90 ± 0.05	0.88 ± 0.05
Specificity	0.99 ± 0.01	0.98 ± 0.02	0.96 ± 0.03	0.94 ± 0.04	0.91 ± 0.04

Table 4.7: Statistical results for neuronal network data generated from HH model

	$N = 10$	$N = 20$	$N = 30$	$N = 40$	$N = 50$
Sensitivity	0.99 ± 0.01	0.99 ± 0.02	0.95 ± 0.04	0.93 ± 0.05	0.91 ± 0.05
Specificity	0.99 ± 0.01	0.97 ± 0.03	0.96 ± 0.03	0.94 ± 0.04	0.93 ± 0.05

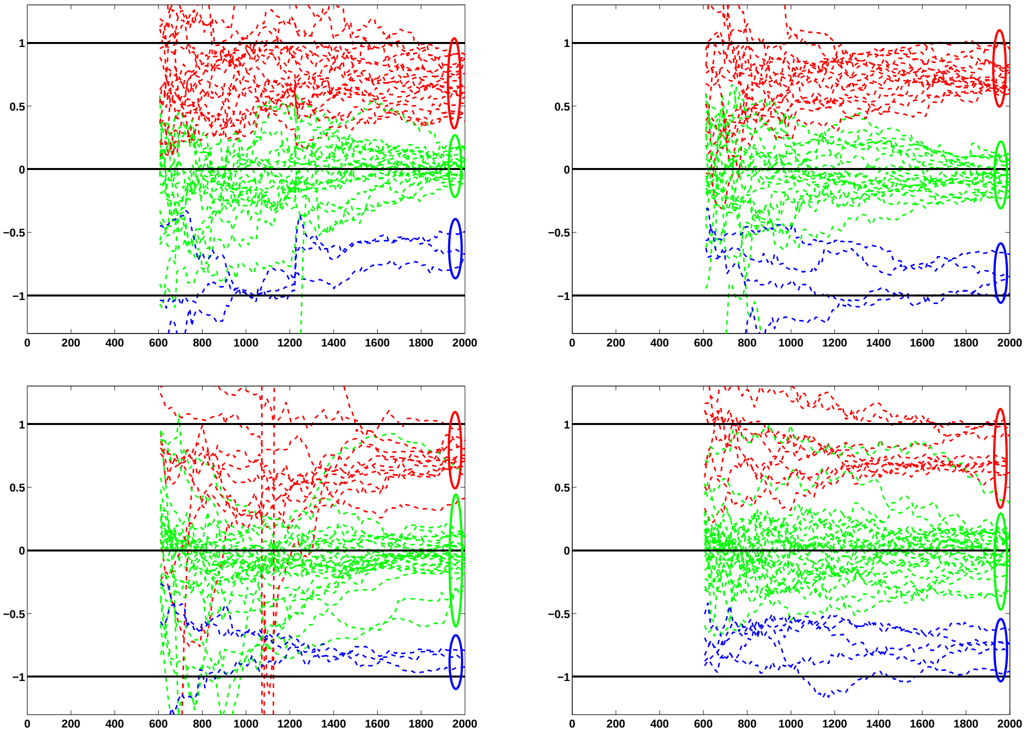
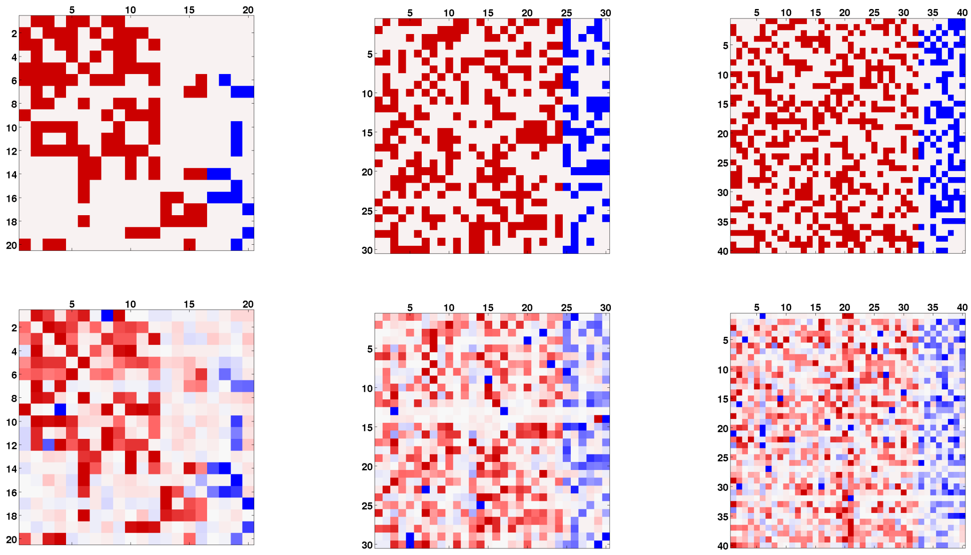


Figure 4.25: Convergence of connection parameters in a network of 40 HR neurons



(a) $N = 20$

(b) $N = 30$

(c) $N = 40$

Figure 4.26: Connectivity estimation for Hindmarsh-Rose networks of different sizes.

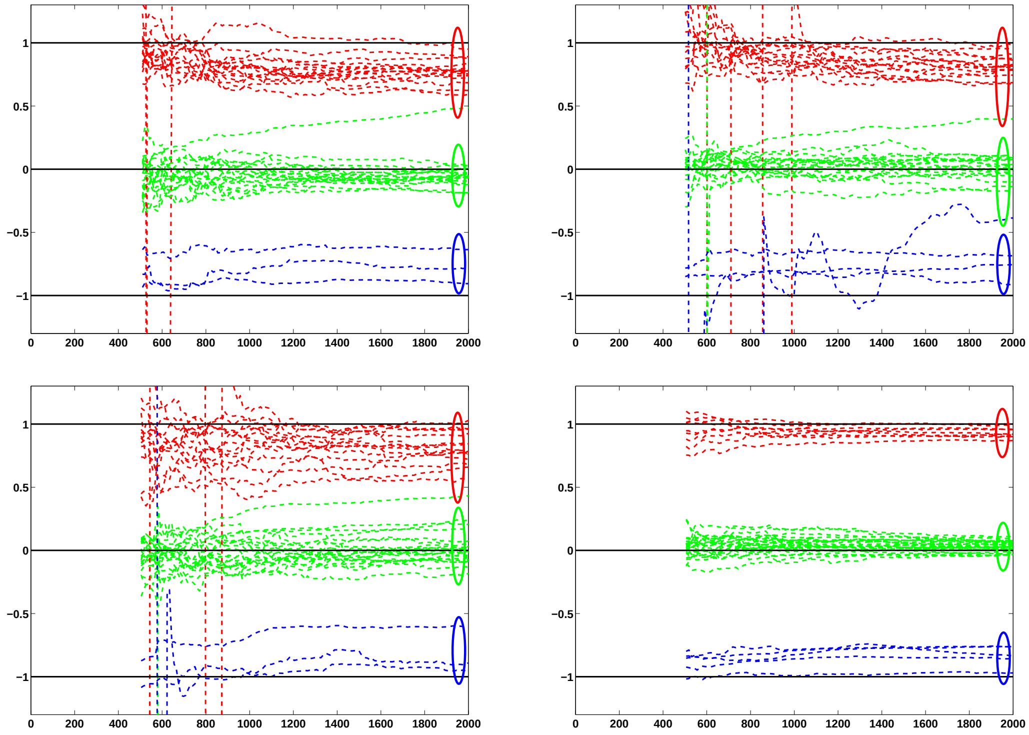
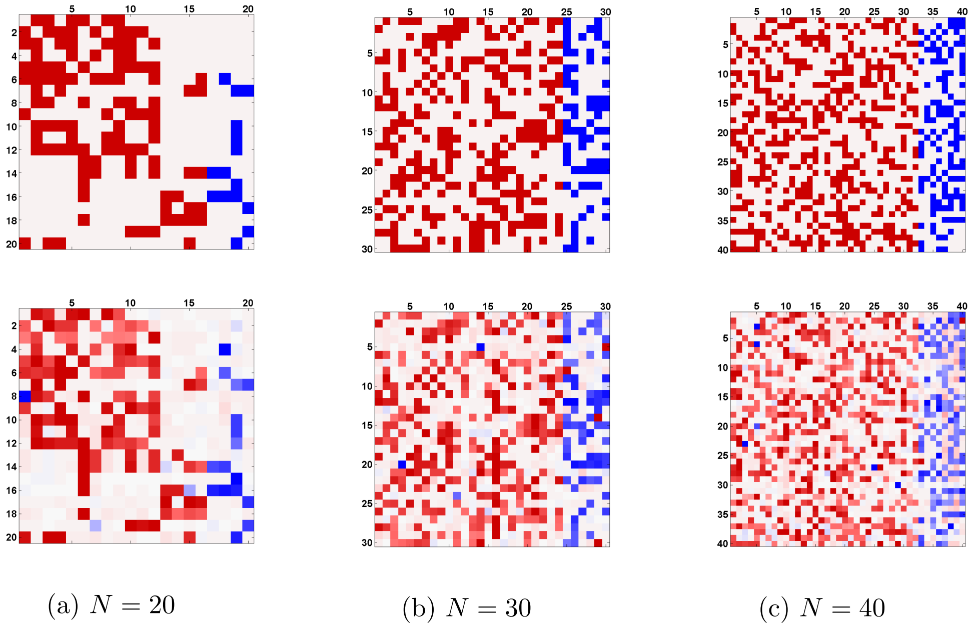


Figure 4.27: Convergence of connection parameters in a network of 40 HH neurons



(a) $N = 20$

(b) $N = 30$

(c) $N = 40$

Figure 4.28: Connectivity estimation for Hodgkin-Huxley networks of different sizes.

4.4 Tracking changes in large neuronal networks

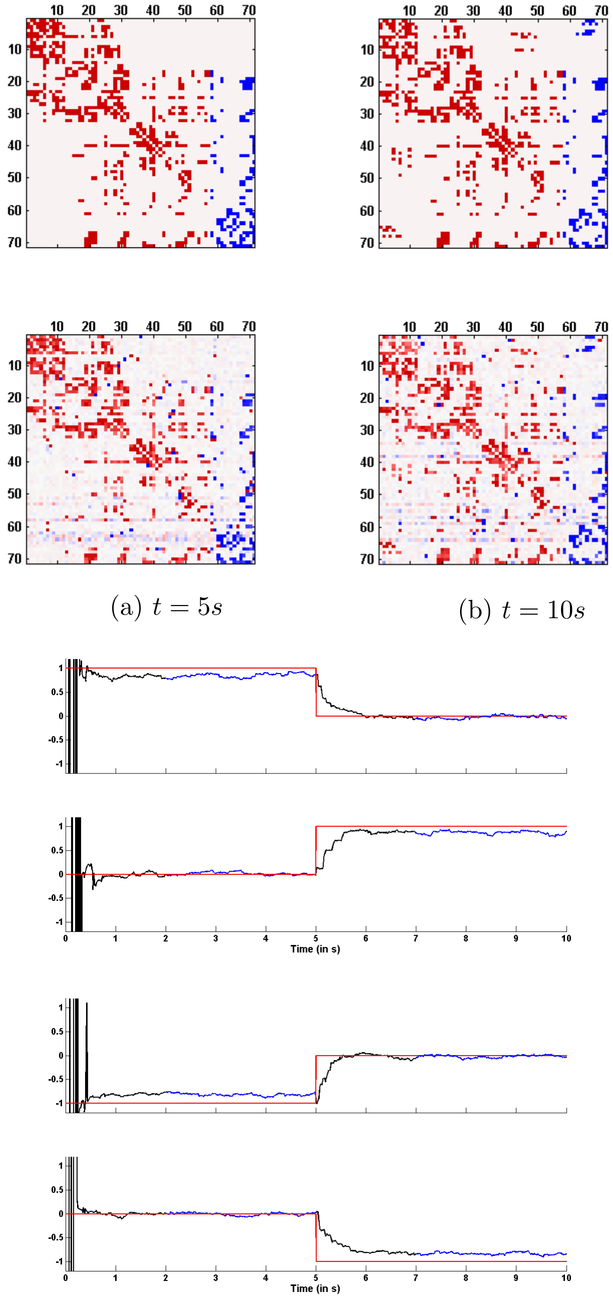
The proposed connectivity estimation technique performed excellently on small networks, even in the presence of a model error and incorrect dynamics of the post-synpatic potential. The challenge is, however, to estimate the connectivity in large networks (50-100 neurons) and also to be able to track the changes in the synaptic strengths. We generate membrane potentials from a network of 71 HH and HR neurons. The network (40% connectivity) is simulated for 10s. Approximately 10% of the connections, chosen at random, are modified at 5s. The excitatory and inhibitory neurons do not change nature. The changes in the connections signify the development of new connections and the degeneration of synapses in the brain. We estimate the connections, and present results of the estimation at 5s and 10s comparing it with the respective connectivity matrices at those times. We also report the specificity and sensitivity measures in Table 4.8.

Table 4.8: Specificity and sensitivity measures for a 71-neuron network.

	Sensitivity	**Specificity**
$t = 5s$	0.90	0.92
$t = 10s$	0.88	0.91

Fig. 4.29 shows the results of connectivity estimation. The parameters were averaged over a 3s window and normalized. The images on the first row represent the actual connections, and the estimated connections are shown on the second row (red for +1 and blue for -1). The color scale of all images are the same. The connections are estimated at two time instances, 5s and 10s. The estimated connections are shown in the second row. The subplots at the bottom represents the convergence of four of the connection parameters. The red trace is the actual connection with a step change in value at $t = 5s$. The blue/black trace is the estimation. The blue part of the trace are the data points that are averaged to compute the steady state value of the connection. It can be observed that the identification algorithm is able to accurately capture the changes in the connectivity. The transient time,

after the step change in the actual connection, is approximately 1s.



(a) $t = 5s$

(b) $t = 10s$

(c) Estimating time varying connections

Figure 4.29: Connectivity estimation in a network of 71 neurons with time-varying connections.

Chapter 5: Discussion

Over the last two decades there has been a lot of research on developing neuron models. The focus has primarily been in building simple and efficient mathematical models that can simulate the membrane potential recordings of a biological neuron. While the complex nonlinear dynamics of the neuron can be achieved by using higher order differential equations, it is difficult to generate the response of a network of neurons due to computational limitations. Researchers have been able to use a discontinuous reset to achieve the range of responses in a two dimensional nonlinear state space system. These hybrid systems have been used to simulate large populations of neurons. However, due to the presence of the reset, it is not possible to use estimation techniques to compute the parameters and states of the models. There is the need to develop new and improvised methods that are able estimate the parameters for this class of neuron models. This dissertation presents parameter identification techniques for single neuron models and neuronal networks.

5.1 Single Neuron Models

The objective was to optimize a model for neural spiking activity using the ISIs of a single neuron. Results with synthetic data showed that we were able to estimate the parameters of the model with sufficiently high accuracy. Since existing error functions for spike trains do not satisfy the requirements of convexity and smoothness, a new function is designed based on the time of spikes. The analytical gradient of this function is computed with respect to the parameters being estimated. The gradient descent algorithm was used to navigate the parameter space and find the optimal parameter vector.

The model was also optimized to fit the publicly available INCF experimental data. The gradient descent technique was combined with the Nelder-Mead method. The results showed a notable improvement over using Nelder-Mead optimization technique only.

The optimization technique used has not been applied to bio-plausible models. The method developed in this chapter can be extended to different nonlinear dynamic models, such as the Hodgkin-Huxley model. Another limitation of the method is the length of the spike train. The spike train data has been chosen in a way that allows comparison of the results with the efforts of other researchers in the field. The model can be optimized for longer spike trains that will enable the formulation of better predictors of neural spiking activity. The trade off to be considered is that the computational cost of the algorithm increases proportionally with an increase in the length of the spike train.

5.2 Neuronal Networks

Synaptic connections in the brain are responsible for the transfer of information from one neuron to the other and the communication between the different functional regions. There is experimental evidence of changing connectivity in the brain with new synapses continuously being formed as well as decay of existing ones. While a strong correlation exists between the transformation in the patterns and the response of the brain, there is a lack of methods that can estimate the connectivity in real time. Dynamic systems have successfully simulated large populations of neurons and are moving towards building a complete brain model. It is therefore important to use such models and develop techniques that can identify the synaptic connections in a network from the membrane potential observations of the individual neurons.

Linearly coupled Izhikevich neurons were used to simulate the dynamics of a neuronal population. The nonlinear model was mathematically re-defined as an auto regressive moving average system that enabled the implementation of the least squares algorithm for system identification. The performance of this method was tested for data generated from the same model to ensure that there is convergence and to report the effect of observation noise and network size on the accuracy of parameter identification. Data was then generated from different models of neuronal networks to demonstrate the robustness of the method in the presence of model error. Sensitivity and specificity results for networks of different sizes are reported. While an increase in the number of neurons effects in fewer percentage of correctly identified connections, the degradation is not significant and the results remains within acceptable limits even for networks of 50 neurons. The method is also able to track changes in the connectivity as it utilizes recursive storage of the data implemented with a forgetting factor. This enables it to estimate the dyamics of time-varying connections and provide more information about the effects of changing connectivity.

While the method is successful in computing the connections of moderate size networks (50-100 neurons), it is important to extend this to larger networks (≈ 1000 neurons). Also, the technique can be adapted to be used with other nonlinear dynamic models with linear coupling, such as extracellular models of groups of neurons. This would enable us to estimate the synaptic strengths in *in vitro* recordings from neuronal tissue.

Appendix A: An Appendix

A.1 Hindmarsh Rose model

The dynamics of a single neuron is described according the equations below. The excitation to the neuron is the sum of the external current, $I(t)$, and the synaptic current which is a weighted sum of the post-synaptic potentials.

$$\begin{aligned} \dot{x}_i(t) =& y_i(t) - a_i x_i^3(t) + b_i x_i^2(t) - z_i(t) + I(t) + \frac{g}{N}\sum_{j=1}^{N} w_{ij} s_j(t) \\ \dot{y}_i(t) =& c_i - d_i x_i^2(t) - y_i(t) \\ \dot{z}_i(t) =& \epsilon_i[\gamma_i(x_i(t) - x_{i0}) - z_i(t)] \end{aligned} \tag{A.1}$$

The state $x_i(t)$ represents the membrane potential of the single neuron, i. It is scaled based on the range of the experimental measurements of a biological neuron. This range is also in agreement with the range represented by the Izhikevich model.

A second order differential equation is used to represent the dynamics of the post-synaptic potential.

$$\ddot{s}_i(t) + \left(\frac{1}{\tau_1} + \frac{1}{\tau_2}\right)\dot{s}_i(t) + \frac{1}{\tau_1\tau_2} s_i(t) = \dot{p}_i(t) + \alpha\left(\frac{1}{\tau_1} - \frac{1}{\tau_2}\right)p_i(t) + p_i(t) \tag{A.2}$$

where

$$p_i(t) = \frac{1.5}{1 + ke^{-100x_i(t)}}$$

The parameters used for the single neuron dynamics are as follows :

$$a_i = 0.5; \quad b_i = 5; \quad c_i = -8; \quad d_i = 5; \quad \gamma_i = 1; \quad \epsilon_i = 0.001; \quad x_{i0} = -1;$$

The parameters for the PSP dynamics are randomly chosen from a pre-defined set. The ranges for each parameter is given below :

$$\tau_1 \in [5, 15]; \quad \tau_2 \in [300, 500]; \quad \alpha \in [0.7, 0.9]; \quad g \in [0.5, 1.5]; \quad k = 9$$

A.2 Hodgkin Huxley model

A bio-plausible model is used to simulate the dynamics of a single neuron in the network, as discussed in [76]. The post-synaptic potential dynamics remain the same as described in the previous section, Eq. A.2.

$$\begin{aligned} \dot{v}_i(t) =& -g_{Na}m^3(t)h(t)[v_i(t) - E_{Na}] - g_K n^4(t)[v_i(t) - E_K] - g_L[v_i(t) - E_L] + \\ & I(t) + \frac{g}{N}\sum_{j=1}^{N} w_{ij}s_j(t) \\ \dot{m}_i(t) =& a_m(v_i(t))[1 - m_i(t)] - b_m(v_i(t))[m_i(t)] \\ \dot{h}_i(t) =& a_h(v_i(t))[1 - h_i(t)] - b_h(v_i(t))[h_i(t)] \\ \dot{n}_i(t) =& a_n(v_i(t))[1 - n_i(t)] - b_n(v_i(t))[n_i(t)] \end{aligned} \tag{A.3}$$

The model functions are :

$$\begin{aligned}
a_m(v_i(t)) =& \frac{0.1(v_i(t)+40)}{1-e^{-(v_i(t)+40)/10}} \\
b_m(v_i(t)) =& 4e^{-(v_i(t)+65)/18} \\
a_h(v_i(t)) =& 0.07e^{-(v_i(t)+35)/20} \\
b_h(v_i(t)) =& \frac{1}{1+e^{-(v_i(t)+35)/10}} \\
a_n(v_i(t)) =& \frac{0.01(v_i(t)+55)}{1-e^{-(v_i(t)+55)/10}} \\
b_n(v_i(t)) =& 0.125e^{-(v_i(t)+65)/80}
\end{aligned} \tag{A.4}$$

Bibliography

Bibliography

[1] W. Gerstner and R. Naud, "How good are neuron models?" *Science*, vol. 326, no. 5951, pp. 379–380, 2009.

[2] M. Piccolino, "Luigi galvani and animal electricity: two centuries after the foundation of electrophysiology," *Trends in neurosciences*, vol. 20, no. 10, pp. 443–448, 1997.

[3] F. López-Muñoz, J. Boya, and C. Alamo, "Neuron theory, the cornerstone of neuroscience, on the centenary of the nobel prize award to santiago ramón y cajal," *Brain research bulletin*, vol. 70, no. 4, pp. 391–405, 2006.

[4] E. D. Adrian, "The basis of sensation." 1928.

[5] O. Hamill, A. Marty, E. Neher, B. Sakmann, and F. Sigworth, "Improved patch-clamp techniques for high-resolution current recording from cells and cell-free membrane patches," *Pflügers Archiv*, vol. 391, no. 2, pp. 85–100, 1981.

[6] M. Taketani and M. Baudry, "Advances in network electrophysiology," *US: Springer*, 2006.

[7] I. H. Stevenson and K. P. Kording, "How advances in neural recording affect data analysis," *Nature neuroscience*, vol. 14, no. 2, pp. 139–142, 2011.

[8] M. A. Lebedev and M. A. Nicolelis, "Brain–machine interfaces: past, present and future," *TRENDS in Neurosciences*, vol. 29, no. 9, pp. 536–546, 2006.

[9] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz, "Cortical control of a prosthetic arm for self-feeding," *Nature*, vol. 453, no. 7198, pp. 1098–1101, 2008.

[10] K. Brigham and B. V. Kumar, "Subject identification from electroencephalogram (eeg) signals during imagined speech," in *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on.* IEEE, 2010, pp. 1–8.

[11] A. L. Hodgkin and A. F. Huxley, "Currents carried by sodium and potassium ions through the membrane of the giant axon of loligo," *The Journal of physiology*, vol. 116, no. 4, p. 449, 1952.

[12] R. FitzHugh, "Impulses and physiological states in theoretical models of nerve membrane," *Biophysical journal*, vol. 1, no. 6, pp. 445–466, 1961.

[13] J. Nagumo, S. Arimoto, and S. Yoshizawa, "An active pulse transmission line simulating nerve axon," *Proceedings of the IRE*, vol. 50, no. 10, pp. 2061–2070, 1962.

[14] E. M. Izhikevich, “Which model to use for cortical spiking neurons?” *IEEE transactions on neural networks*, vol. 15, no. 5, pp. 1063–1070, 2004.

[15] E. M. Izhikevich *et al.*, “Simple model of spiking neurons,” *IEEE Transactions on neural networks*, vol. 14, no. 6, pp. 1569–1572, 2003.

[16] R. Kobayashi, Y. Tsubo, and S. Shinomoto, “Made-to-order spiking neuron model equipped with a multi-timescale adaptive threshold,” *Frontiers in computational neuroscience*, vol. 3, 2009.

[17] S. Yamauchi, H. Kim, and S. Shinomoto, “Elemental spiking neuron model for reproducing diverse firing patterns and predicting precise firing times,” *Frontiers in computational neuroscience*, vol. 5, 2011.

[18] R. Barbieri, M. C. Quirk, L. M. Frank, M. A. Wilson, and E. N. Brown, “Construction and analysis of non-poisson stimulus-response models of neural spiking activity,” *Journal of neuroscience methods*, vol. 105, no. 1, pp. 25–37, 2001.

[19] J. Tapson, C. Jin, A. van Schaik, and R. Etienne-Cummings, “A first-order nonhomogeneous markov model for the response of spiking neurons stimulated by small phase-continuous signals,” *Neural computation*, vol. 21, no. 6, pp. 1554–1588, 2009.

[20] E. M. Izhikevich and G. M. Edelman, “Large-scale model of mammalian thalamocortical systems,” *Proceedings of the national academy of sciences*, vol. 105, no. 9, pp. 3593–3598, 2008.

[21] P. Antsaklis, *Hybrid systems II.* Springer, 1995, vol. 2.

[22] M. Senesky, G. Eirea, and T. J. Koo, “Hybrid modelling and control of power electronics,” in *Hybrid Systems: Computation and Control.* Springer, 2003, pp. 450–465.

[23] E. M. Izhikevich, “Hybrid spiking models,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1930, pp. 5061–5070, 2010.

[24] M. W. Hirsch, S. Smale, and R. L. Devaney, *Differential equations, dynamical systems, and an introduction to chaos.* Academic press, 2004, vol. 60.

[25] B. Van der Pol, “The nonlinear theory of electric oscillations,” *Radio Engineers, Proceedings of the Institute of*, vol. 22, no. 9, pp. 1051–1086, 1934.

[26] A. Iserles, *A first course in the numerical analysis of differential equations.* Cambridge University Press, 2009, no. 44.

[27] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[28] Y. Song and J. W. Grizzle, “The extended kalman filter as a local asymptotic observer for nonlinear discrete-time systems,” in *American Control Conference, 1992.* IEEE, 1992, pp. 3365–3369.

[29] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter," *Kalman filtering and neural networks*, pp. 221–280, 2001.

[30] S. J. Julier, "The scaled unscented transformation," in *American Control Conference, 2002. Proceedings of the 2002*, vol. 6. IEEE, 2002, pp. 4555–4559.

[31] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[32] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000.* IEEE, 2000, pp. 153–158.

[33] K. J. Astrom and P. Eykhoff, "System identification - a survey," *Automatica*, vol. 7, no. 2, pp. 123–162, 1971.

[34] P. Dayan and L. F. Abbott, *Theoretical neuroscience.* Cambridge, MA: MIT Press, 2001.

[35] R. VanRullen, R. Guyonneau, and S. J. Thorpe, "Spike times make sense," *Trends in neurosciences*, vol. 28, no. 1, pp. 1–4, 2005.

[36] L. Paninski, J. W. Pillow, and E. P. Simoncelli, "Maximum likelihood estimation of a stochastic integrate-and-fire neural encoding model," *Neural computation*, vol. 16, no. 12, pp. 2533–2561, 2004.

[37] A. Smith and E. Brown, "Estimating a state-space model from point process observations," *Neural Computation*, vol. 15, no. 5, pp. 965–991, 2003.

[38] S. J. Schiff and T. Sauer, "Kalman filter control of a model of spatiotemporal cortical dynamics," *BMC Neuroscience*, vol. 9, no. Suppl 1, p. O1, 2008.

[39] G. Ullah and S. J. Schiff, "Tracking and control of neuronal hodgkin-huxley dynamics," *Physical Review E*, vol. 79, no. 4, p. 040901, 2009.

[40] S. J. Schiff, "Kalman meets neuron: the emerging intersection of control theory with neuroscience," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE.* IEEE, 2009, pp. 3318–3321.

[41] H. U. Voss, J. Timmer, and J. Kurths, "Nonlinear dynamical system identification from uncertain and indirect measurements," *International Journal of Bifurcation and Chaos*, vol. 14, no. 06, pp. 1905–1933, 2004.

[42] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the sixth international symposium on micro machine and human science*, vol. 1. New York, NY, 1995, pp. 39–43.

[43] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The computer journal*, vol. 7, no. 4, pp. 308–313, 1965.

[44] W. Van Geit, E. De Schutter, and P. Achard, "Automated neuron model optimization techniques: a review," *Biological cybernetics*, vol. 99, no. 4-5, pp. 241–251, 2008.

[45] A. Pavlov, E. Steur, and N. van de Wouw, "Controlled synchronization via nonlinear integral coupling," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on.* IEEE, 2009, pp. 5263–5268.

[46] K. Kitamura, B. Judkewitz, M. Kano, W. Denk, and M. Häusser, "Targeted patch-clamp recordings and single-cell electroporation of unlabeled neurons in vivo," *Nature methods*, vol. 5, no. 1, pp. 61–67, 2007.

[47] G. Kumar, V. Aggarwal, N. V. Thakor, M. H. Schieber, and M. V. Kothare, "Optimal parameter estimation of the izhikevich single neuron model using experimental inter-spike interval (isi) data," in *American Control Conference (ACC), 2010.* IEEE, 2010, pp. 3586–3591.

[48] N. Keren, N. Peled, and A. Korngreen, "Constraining compartmental models using multiple voltage recordings and genetic algorithms," *Journal of neurophysiology*, vol. 94, no. 6, pp. 3730–3742, 2005.

[49] M. C. Vanier and J. M. Bower, "A comparative survey of automated parameter-search methods for compartmental neural models," *Journal of computational neuroscience*, vol. 7, no. 2, pp. 149–171, 1999.

[50] P. Achard and E. De Schutter, "Complex parameter landscape for a complex neuron model," *PLoS Computational Biology*, vol. 2, no. 7, p. e94, 2006.

[51] I. Griva, S. G. Nash, and A. Sofer, *Linear and nonlinear optimization.* Siam, 2009.

[52] O. Booij *et al.*, "A gradient descent rule for spiking neurons emitting multiple spikes," *Information Processing Letters*, vol. 95, no. 6, pp. 552–558, 2005.

[53] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the nelder–mead simplex method in low dimensions," *SIAM Journal on optimization*, vol. 9, no. 1, pp. 112–147, 1998.

[54] R. Shamir, "The efficiency of the simplex method: a survey," *Management Science*, vol. 33, no. 3, pp. 301–334, 1987.

[55] T. Kreuz, D. Chicharro, M. Greschner, and R. G. Andrzejak, "Time-resolved and time-scale adaptive measures of spike train synchrony," *Journal of neuroscience methods*, vol. 195, no. 1, pp. 92–106, 2011.

[56] K. J. Friston, "Functional and effective connectivity: a review," *Brain connectivity*, vol. 1, no. 1, pp. 13–36, 2011.

[57] W. D. Knowles and P. A. Schwartzkroin, "Local circuit synaptic interactions in hippocampal brain slices," *The Journal of Neuroscience*, vol. 1, no. 3, pp. 318–322, 1981.

[58] D. Purves, G. Augustine, D. Fitzpatrick, W. Hall, A. LaMantia, J. McNamara, and L. White, "Neuroscience, 2008," *De Boeck, Sinauer, Sunderland, Mass.*

[59] C. Büchel, J. Coull, and K. Friston, "The predictive value of changes in effective connectivity for human learning," *Science*, vol. 283, no. 5407, pp. 1538–1541, 1999.

[60] S. Marom and G. Shahaf, "Development, learning and memory in large random networks of cortical neurons: lessons beyond anatomy," *Quarterly reviews of biophysics*, vol. 35, no. 01, pp. 63–87, 2002.

[61] A. Meyer-Lindenberg, J.-B. Poline, P. D. Kohn, J. L. Holt, M. F. Egan, D. R. Weinberger, and K. F. Berman, "Evidence for abnormal cortical functional connectivity during working memory in schizophrenia," *American Journal of Psychiatry*, vol. 158, no. 11, pp. 1809–1817, 2001.

[62] S. Ruediger, C. Vittori, E. Bednarek, C. Genoud, P. Strata, B. Sacchetti, and P. Caroni, "Learning-related feedforward inhibitory connectivity growth required for memory precision," *Nature*, vol. 473, no. 7348, pp. 514–518, 2011.

[63] L. Wang, Y. Zang, Y. He, M. Liang, X. Zhang, L. Tian, T. Wu, T. Jiang, and K. Li, "Changes in hippocampal connectivity in the early stages of alzheimer's disease: evidence from resting state fmri," *Neuroimage*, vol. 31, no. 2, pp. 496–504, 2006.

[64] P. R. Karunanayaka, S. K. Holland, V. J. Schmithorst, A. Solodkin, E. E. Chen, J. P. Szaflarski, and E. Plante, "Age-related connectivity changes in fmri data from children listening to stories," *Neuroimage*, vol. 34, no. 1, pp. 349–360, 2007.

[65] O. Sporns, G. Tononi, and G. M. Edelman, "Connectivity and complexity: the relationship between neuroanatomy and brain dynamics," *Neural Networks*, vol. 13, no. 8, pp. 909–922, 2000.

[66] A. Zalesky, A. Fornito, and E. Bullmore, "On the use of correlation as a measure of network connectivity," *Neuroimage*, vol. 60, no. 4, pp. 2096–2106, 2012.

[67] K. Friston, R. Moran, and A. K. Seth, "Analysing connectivity with granger causality and dynamic causal modelling," *Current opinion in neurobiology*, vol. 23, no. 2, pp. 172–178, 2013.

[68] L. Leistritz, B. Pester, A. Doering, K. Schiecke, F. Babiloni, L. Astolfi, and H. Witte, "Time-variant partial directed coherence for analysing connectivity: a methodological study," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1997, p. 20110616, 2013.

[69] T. Berry, F. Hamilton, N. Peixoto, and T. Sauer, "Detecting connectivity changes in neuronal networks," *Journal of neuroscience methods*, vol. 209, no. 2, pp. 388–397, 2012.

[70] E. Bullmore and O. Sporns, "Complex brain networks: graph theoretical analysis of structural and functional systems," *Nature Reviews Neuroscience*, vol. 10, no. 3, pp. 186–198, 2009.

[71] C. Diekman, K. Dasgupta, V. Nair, and K. Unnikrishnan, "Discovering functional neuronal connectivity from serial patterns in spike train data," *Neural computation*, pp. 1–35, 2014.

[72] E. W. Lang, A. Tomé, I. R. Keck, J. Górriz-Sáez, and C. Puntonet, "Brain connectivity analysis: a short survey," *Computational intelligence and neuroscience*, vol. 2012, p. 8, 2012.

[73] A. Marreiros, S. Kiebel, and K. Friston, "Dynamic causal modelling for fmri: a two-state model," *Neuroimage*, vol. 39, no. 1, pp. 269–278, 2008.

[74] S. Rombouts, C. Stam, J. Kuijer, P. Scheltens, and F. Barkhof, "Identifying confounds to increase specificity during a "no task condition'': Evidence for hippocampal connectivity using fmri," *Neuroimage*, vol. 20, no. 2, pp. 1236–1245, 2003.

[75] C. Grefkes, D. A. Nowak, L. E. Wang, M. Dafotakis, S. B. Eickhoff, and G. R. Fink, "Modulating cortical connectivity in stroke patients by rtms assessed with fmri and dynamic causal modeling," *Neuroimage*, vol. 50, no. 1, pp. 233–242, 2010.

[76] F. Hamilton, T. Berry, N. Peixoto, and T. Sauer, "Real-time tracking of neuronal network structure using data assimilation," *Physical Review E*, vol. 88, no. 5, p. 052715, 2013.

[77] F. B. Hildebrand, *Introduction to numerical analysis.* Courier Dover Publications, 1987.

[78] G. E. Box, G. M. Jenkins, and G. C. Reinsel, *Time series analysis: forecasting and control.* John Wiley & Sons, 2013.

[79] L. Ljung, *System identification.* Springer, 1998.

[80] U. Forssell and L. Ljung, "Identification of unstable systems using output error and box-jenkins model structures," *Automatic Control, IEEE Transactions on*, vol. 45, no. 1, pp. 137–141, 2000.

[81] M. R. DeWeese, M. Wehr, and A. M. Zador, "Binary spiking in auditory cortex," *The Journal of neuroscience*, vol. 23, no. 21, pp. 7940–7949, 2003.

[82] J. Hindmarsh and R. Rose, "A model of neuronal bursting using three coupled first order differential equations," *Proceedings of the Royal society of London. Series B. Biological sciences*, vol. 221, no. 1222, pp. 87–102, 1984.

[83] A. L. Hodgkin and A. F. Huxley, "The dual effect of membrane potential on sodium conductance in the giant axon of loligo," *The Journal of physiology*, vol. 116, no. 4, pp. 497–506, 1952.

[84] G. Gan, C. Ma, and J. Wu, *Data clustering: theory, algorithms, and applications.* Siam, 2007, vol. 20.

Curriculum Vitae

Anish Mitra received his Bachelors in Engineering from Birla Institute of Technology, Ranchi, India in 2007. He then proceeded to pursue graduate studies in Electrical and Computer Engineering at George Mason University, obtaining his Masters in Electrical Engineering in 2011.

At GMU, Anish has been involved with multiple projects in the field of control systems which involve the real time control of physical systems. As a graduate teaching assistant he has taught multiple courses and labs in the areas of signals and systems, control systems and random processes. He has been actively involved with undergraduate senior design groups, and designing experiments for advanced technical labs. He has maintained an excellent academic record throughout the course of his graduate studies and been a multiple time recepient of the Volgenau School Doctoral Fellowship.

His primary area of research is parameter estimation in dynamical systems, and for his doctoral dissertation he has focused on neuronal models. Other areas of interest include mathematical modeling of physical systems, nonlinear optimization and optimal control.