UNIVERSITY OF CALIFORNIA, SAN DIEGO

Use of Data Assimilation to Determine Features of Neuron Structure and Connectivity

A Dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Physics with a specialization in Computational Science

by

Michael J. Eldridge

Committee in charge:

Professor Henry D.I. Abarbanel, Chair
Professor Gert Cauwenberghs
Professor George Fuller
Professor Julius Kuti
Professor Stefan Tanaka

2016

The Dissertation of Michael J. Eldridge is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2016

TABLE OF CONTENTS

Signature Page iii

Table of Contents iv

List of Figures vii

List of Tables viii

Acknowledgements ix

Vita x

Abstract of the Dissertation xi

Chapter 1 Data Assimilation 1
1.1 General problem 3
1.1.1 Formal Description 4
1.1.2 Constraints 5
1.2 Overview of Techniques 7
1.2.1 Numerical Integration 7
1.2.2 Overview of Optimization Techniques 7
1.3 Twin Experiments 8
1.4 Software 10
1.5 Path Integral Formulation 10
1.5.1 Conditional Probability 11
1.5.2 Conditional Mutual Information 12
1.5.3 Markov Processes 12
1.5.4 Derivation of Path Integral 13
1.5.5 Action 16
1.5.6 Transition Probabilities 17
1.5.7 Computational Form of the Action 18
1.5.8 Approximating the Path Integral 18
1.6 Simulated Annealing 19

Chapter 2 Nonlinear Dynamics 20
2.1 Phase Space 21
2.2 Chaos 23
2.2.1 Lyapunov Exponents 25
2.3 Synchronization 27
2.3.1 Conditional Lyapunov Exponents 30

Chapter 3 Time Delayed Newton's Method . . . 31
3.1 Introduction . . . 31
3.2 Time Delay Embedding . . . 33
3.3 Synchronization Error . . . 35
3.4 Formulation As Error Estimation . . . 36
3.5 Gauss-Newton . . . 38
3.6 Variational Equation . . . 40
3.7 Calculating the Pseudoinverse . . . 42
3.8 Description of Algorithm . . . 44
3.9 State Estimation in Lorenz 63 . . . 45
3.10 Exploring the Cost Function for Lorenz 96 . . . 49
3.11 State & Parameter Estimation in Rössler's System . . . 52

Chapter 4 Neurobiology Background . . . 57
4.1 Gross Neuroanatomy . . . 57
4.2 Ion Flow . . . 59
4.2.1 Nernst-Plank . . . 60
4.2.2 GHK Equation . . . 61
4.2.3 Numerical stability of GHK . . . 62
4.2.4 Ohm's Law . . . 62
4.2.5 Energy Barrier Model . . . 63
4.2.6 Gates . . . 65
4.3 Parallel Conductance Model . . . 67
4.3.1 Hodgkin-Huxley Model . . . 68
4.4 Synaptic Transmission . . . 69
4.4.1 Synaptic Models . . . 72
4.5 Experiments . . . 73
4.5.1 Electrophysiology . . . 73
4.5.2 Synaptic Measurements . . . 74

Chapter 5 HVC Single Neuron Models . . . 77
5.1 Introduction . . . 77
5.2 HVC Background . . . 78
5.3 Assimilation Procedure . . . 80
5.4 A4 Model . . . 81
5.4.1 Units . . . 85
5.4.2 A4 Model Optimization Estimating Conductances . 85
5.5 B4 Model . . . 91
5.5.1 B4 Model Optimization Results Estimating All Parameters . . . 92
5.5.2 B4 Model Optimization Estimating Conductances . 96

Chapter 6 Use of Assimilation on Synaptic Plasticity Models 98
6.1 Overview of Plasticity Phenomena 99
6.1.1 Discussion of Model Features 102
6.1.2 Phenomenological Models 104
6.2 Plasticity Model 104
6.2.1 Verification of Model 106
6.2.2 Plasticity Twin Experiments 107
6.3 Unified Model 111
6.3.1 Synaptic Model 111
6.3.2 Unified Model Twin Experiments 114

LIST OF FIGURES

Figure 2.1: Simulation of NaKL neuron 21
Figure 2.2: NaKL limit cycle 24

Figure 3.1: Use of time-delay Newton's method on the Lorenz 63 system . . . 47
Figure 3.2: Synchronization error in Lorenz 63 48
Figure 3.3: Synchronization error for various D_M in Lorenz 96 50
Figure 3.4: 2D projection of cost function for Lorenz 96 51
Figure 3.5: Prediction of the 4D Rössler system 55
Figure 3.6: Estimated parameters for the 4D Rössler system 56

Figure 4.1: Depiction of gross neuroanatomy 58
Figure 4.2: Illustration of the parts of a synapse 69
Figure 4.3: Synaptic currents produced by model synapse 75

Figure 5.1: Predictions of A4 model assimilated on experimental data 88
Figure 5.2: Histograms of maximal conductance estimates for A4 model . . . 89
Figure 5.3: Maximal conductances plotted in 2D conductance space 90
Figure 5.4: Predictions of B4 model assimilated on experimental data 94

Figure 6.1: Predictions of plasticity rule given by eqn. 6.6 108
Figure 6.2: Simulation of the plasticity rule given by eqn. 6.3 110
Figure 6.3: Behavior of combined synapse model 115
Figure 6.4: Results of combined model with fixed postsynaptic voltage 118

LIST OF TABLES

Table 5.1: A4 model ion currents 86
Table 5.2: A4 model gating dynamics 86
Table 5.3: A4 model optimization results 87
Table 5.4: B4 model ion currents 92
Table 5.5: B4 model optimization results 93

Table 6.1: Parameters used in simulation of model given in 6.6. 106
Table 6.2: Parameter estimates for synapse model 111
Table 6.3: Parameter estimates in combined model 116
Table 6.4: Parameter estimates for fixed postsynaptic voltage 118

ACKNOWLEDGEMENTS

My sincere thanks to my family, friends, and colleagues for your support. Your feedback and encouragement helped see me through this endeavor. I'd also like to extend thanks to the Margoliash lab and especially Arij Daou for providing excellent data and taking the time to explain his experiments in detail.

VITA

2010	B. A. in Physics; Religion, Bowdoin College, Brunswick, ME
2013	M. S. in Physics, University of California, San Diego, San Diego, CA
2015	UCSD Summer Graduate Teaching Scholar
2016	Ph. D. in Physics with a specialization in Computational Science, University of California, San Diego, San Diego, CA

PUBLICATIONS

Ye, J., Rey, D., Kadakia, N., Eldridge, M., Morone, U., Rozdeba, P., Abarbanel, Henry DI, and Quinn, John C "Systematic variational method for statistical nonlinear state and parameter estimation" *Physical Review E* 92(5) (2015)

Schumann-Bischoff, J., Parlitz, U., Abarbanel, Henry DI, Kostuk, M., Rey, D., Eldridge, M., and Luther, S."Basin structure of optimization based state and parameter estimation" *Chaos: An Interdisciplinary Journal of Nonlinear Science* 25(5) (2015)

Rey, D., Eldridge, M., Kostuk, M., Abarbanel, Henry DI, Schumann-Bischoff, J., and Parlitz, U. "Accurate state and parameter estimation in nonlinear systems with sparse observations" *Physics Letters A*, 378(11) p869-873. (2014)

Rey, D., Eldridge, M., Morone, U., Abarbanel, Henry DI, Parlitz, U., and Schumann-Bischoff, J. "Using waveform information in nonlinear data assimilation" *Physical Review E* 90(6) (2014)

ABSTRACT OF THE DISSERTATION

Use of Data Assimilation to Determine Features of Neuron Structure and Connectivity

by

Michael J. Eldridge

Doctor of Philosophy in Physics with a specialization in Computational Science

University of California, San Diego, 2016

Professor Henry D.I. Abarbanel, Chair

Neurons and the synaptic connections between them underlie the computational power of the brain. We present numerical models of neural behavior and show how to tune these models based on experimental evidence. Though the basic principles behind the creation and propagation of action potentials are understood, it is experimentally feasible to measure only a small number of the quantities that go into our models, substantially increasing the difficulty of making accurate predictions. Additionally, because biologically motivated models are very often nonlinear, we will focus on tools and techniques which do not require linearity.

We present novel methods of using time series of measurements to determine the features of nonlinear systems and predict their future behavior. We show how time-delayed coordinates can substitute for additional measurements and provide us with a better estimation of the state and parameters of the underlying system. A general expression for our objective function as a path integral is derived from probabilistic considerations and methods for evaluating the expression are discussed.

We demonstrate how the techniques developed can be used to determine properties of a biophysical system from a realistic set of limited measurements. We examine experimental electrophysiological recordings of zebra finch neurons and use them to hone the predictive powers of our models for single cells. Then, moving beyond the single cell level, we demonstrate how our approach can be used to determine changes in network connectivity due to synaptic plasticity in ways that direct experiment cannot.

Chapter 1

Data Assimilation

In order to test a given hypothesis, a scientist may construct a mathematical model of a system. Using measurements, the scientist creates a specific instance of that model, developing a version of the mathematical model which better describes a particular set of observations. Data assimilation is the process by which observations are incorporated into such a model.

For our purposes, the term numerical model refers to a set of differential equations consisting of *state variables*, which change in time, and *parameters* which have fixed values. If the vector of state variables is $\mathbf{x}$, and the set of parameters used in the model is $\mathbf{p}$, then the equations describing the numerical model can be written,

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{p}). \tag{1.1}$$

We will focus on dynamical models, or those which evolve in time. Typically, the systems we will consider are autonomous, meaning that time does not appear explicitly in the dynamical equations. From the perspective of the underlying physics we focus

exclusively on classical, as opposed to quantum mechanical, systems. Additionally, the underlying dynamics of the models we consider are assumed to be deterministic, though we will allow for the introduction of random additive noise in the measurement of the underlying system.

Typically, only a subset of the states and parameters in the model can be measured directly and are referred to as *observed* states and parameters. In some cases, the number of observations available may be orders of magnitude less than the number of states required to fully describe the model, as may be the case in weather prediction models (Lorenc and Payne [40]). More generally, we may measure only some function of the states and parameters in the model. This function is called the measurement function and produces a time series of observations $\mathbf{Y}$, which can be written $\mathbf{Y} = \mathbf{h}(\mathbf{x}, \mathbf{p})$.

Further complicating things, the models themselves are typically only an approximation to the underlying physics of the system, an adage which tends to be especially true when dealing with biophysical systems. Typically the systems we consider are either too large for a first principles description to be practical, or a precise description of the underlying biophysics is unknown. In practice, most mathematical models of biological systems try to balance an approximate description of the underlying processes with the goal of creating a simple model that accurately describes empirical results.

Data assimilation also allows us to test the accuracy of these models. Based on the measurements made data assimilation provides estimates for the unknown states and parameters of the model. The assimilated model can then be used to make predictions which can be measured against other data sets to test the model's accuracy.

In this way data assimilation can aid in validating or rejecting various models based on experimental data.

1.1 General problem

The problem of data assimilation is closely related to problems of optimization (Gill et al. [20]) and optimal control theory (Kirk [30]). To test whether or not the model hypothesized to describe a data set is accurate, it is necessary to formulate a function which can be used to quantify any discrepancies between data and model. The function which describes how well the model and data agree is called the *cost function* or *objective function*. This function defines what it means for one estimate to be better than another. The goal of data assimilation is then to minimize the cost function through numerical optimization, thereby achieving the best fit of the model to the data.

The process of iterative optimization can be thought of as consisting of two distinct steps. In the first step, a vector of estimated states and parameters is chosen. Different optimization routines are defined by the methods employed to choose this vector. In the second step, the cost function is evaluated to determine whether or not the estimated vector is an improvement. By repeating these steps, one can iteratively search for the minimum of the cost function, which corresponds to the optimal estimate for the states and parameters.

In all the examples given here, we treat both states and parameters in our models as being bounded above and below. These bounds can be *ad hoc* or motivated from the underlying mathematical, physical, or biological description of the system. For example, when employing models of cells, we may know from experimental observation

that concentrations of some chemical above or below some values will kill the cell and therefore choose to bound the concentration to within the survivable range. Typically optimization problems become more difficult as the bounds are relaxed, meaning that the more restricted our search, the greater the chance for success.

1.1.1 Formal Description

Formally, we can describe the optimization procedure as a search over a space of dimension N for the optimal point called $\mathbf{w}^*$. Suppose we wish to determine the initial conditions of a dynamical system along with its parameters or the set $\mathbf{w}^* = \{\mathbf{x}(0), \mathbf{p}\}$. If the vector of initial conditions $\mathbf{x}$ has dimension D and there are P parameters, then the dimension of the search space is $N = D + P$. For many of the examples considered here, we look for the complete state of the system at N_T different points in time using *collocation* methods, which search for the state at all times simultaneously. In this case the dimension of the search space is $N = D \cdot N_T + P$. It is typically desirable to split time into small discrete steps in order to approximate continuous differential equations by finite discrete mappings. However, the smaller the time steps, the larger the number of them N_T required to cover any interval. In other words, the higher the temporal resolution we employ, the more states we must estimate, and the more difficult the optimization problem becomes.

Calculus provides us with a mathematical description of what it means to be a minimum through so-called *optimality conditions*. A minimum point of a one dimensional function has a zero first derivative and positive concavity. Generalized to

a multidimensional function, these conditions can be written

$$\left.\frac{d\phi}{d\mathbf{w}}\right|_{\mathbf{w}^*} = 0$$
$$\left.\frac{d^2\phi}{d^2\mathbf{w}}\right|_{\mathbf{w}^*} > 0,$$

and correspond to a vanishing Jacobian and a positive definite Hessian. When dealing with non-linear functions, there is no guarantee that these conditions will give us the lowest of all possible minima, called the *global minimum*. Indeed, any *local minimum* will satisfy the optimality conditions. As we will see, the failure of the optimality conditions to define a global minimum greatly complicates the optimization process.

1.1.2 Constraints

The problem in which we need only find a solution which satisfies the optimality conditions is called the unconstrained optimization problem. In addition to the optimality conditions, we may wish to restrict the search space through equations of constraint. By imposing constraints we obtain a modified optimization problem, the goal of which is to minimize $\phi(\mathbf{w})$ subject to a set of equations $\mathbf{r}(\mathbf{w}) = 0$ called constraints. One common way of incorporating the equations of constraint is through the method of Lagrange multipliers (see for example Nearing et al. [47]).

For linear systems, we can easily answer the question of what effect constraints have on the uniqueness of the solution (i.e. is the system over-determined, under-determined, or uniquely determined). However, for nonlinear systems of equations this question can not be easily answered and poses one of the chief difficulties of nonlinear optimization. Compounding this, we have seen that the optimality conditions alone are

not sufficient to determine if we have found the global minimum. Thus, in nonlinear optimization problems not only are we unable determine if any point is the global minimum, we cannot even be sure such a point (or points) exists.

So far, we have discussed constrained and unconstrained optimization problems. A third possibility called *weak constraint* exists between these limiting cases. In the weakly constrained form, the equations of constraint are added into the cost function so that there is some additional penalty associated with violating the constraints. To illustrate this, we consider a modified cost function $\psi(\mathbf{w})$, in which we have introduced the equations of constraint along with a weighting parameter α, such that

$$\psi(\mathbf{w}) = \phi(\mathbf{w}) + \frac{(1-\alpha)}{\alpha} r(\mathbf{w}). \tag{1.2}$$

Letting $\alpha \to 1$, the penalty associated with the constraints vanishes, and we recover the unconstrained optimization problem. Conversely, in the limit $\alpha \to 0$, the penalty associated with violating the constraints approaches infinity, and we recover the constrained optimization problem.

The formulation given in eqn. (1.2) is by no means the only way of enforcing weak constraints. Indeed, approaches to enforcing constraints in an active area of research and will be visited again in the discussion of simulated annealing.

1.2 Overview of Techniques

1.2.1 Numerical Integration

In order to integrate the dynamical equations computationally, we must first determine a discrete approximation to the continuous dynamics. First, we discretize time into N_T steps of size dt. We then employ a forward mapping $\mathbf{F}$, which moves the state vector forward in time by an amount dt. In other word, $\mathbf{F}$ is the discrete mapping described by

$$\mathbf{x}(t + dt) = \mathbf{F} \circ \mathbf{x}(t), \tag{1.3}$$

and can be applied repeatedly N_T times in order to yield the state at end of the time interval. We denote this as

$$\mathbf{x}(t + N_T \cdot dt) = \mathbf{F}^{N_T}\mathbf{x}(t). \tag{1.4}$$

There are many schemes for choosing the function $\mathbf{F}$, which actually does the numerical integration.

1.2.2 Overview of Optimization Techniques

No single technique yet exists which can find the global minimum of a nonlinear system of equations. Indeed, some make the argument that such a technique is highly unlikely to even exist (see for example Press et al. [52]). There exist a wide variety of approaches which each have various strengths and weaknesses. Here, we briefly examine several of the wide categories of techniques.

With rare exceptions, all known methods of nonlinear optimization are iterative. In the broadest sense, the strategy of optimization methods is to move in the direction of decrease for the cost function. Determining the direction of decrease requires computing derivatives, typically using finite difference techniques. Methods for optimization are typically categorized by whether they require evaluating the gradient and/or Hessian. While calculating the Hessian typically improves a search, this evaluation is computationally expensive and so in many implementations the Hessian is only approximated.

Most of the techniques considered here fall under the umbrella of sequential Bayesian estimation (see for example Särkkä [56]). Bayesian estimation provides a general probabilistic framework for estimating the distribution of unknown states and parameters given some set of measurements. Subclasses of Bayesian estimation methods, such as *filters* and *smoothers* are defined based on what subset of measurements are used. Many of the approaches, including filters (Judd and Stemler [28]) and estimators (Cessna et al. [10]), are designed for the large scale assimilation problems associated with weather forecasting.

1.3 Twin Experiments

The ultimate goal of data assimilation is to use numerical techniques to determine attributes of some unknown system which produces experimental data. However when using experimental data, if the assimilation procedure fails to accurately capture the behavior of the data, we are left guess as to the causes of failure. It could be that the underlying model we have chosen to describe the data is incorrect and cannot adequately explain the observations for *any* values of states and parameters. A second

possibility is that the model is correct, but that the data assimilation procedure itself has failed. It is therefore desirable to test out the assimilation procedures developed in cases where we know the underlying model which produced the data. The procedures we use to test the assimilation procedures themselves are called *twin experiments* and are examined below.

In a twin experiment artificial data are produced using a dynamical model of our choosing. Noise may be added to the data produced, and the data are then assimilated to an identical system called the model system. In this way, we can eliminate the possibility of model error and also control the amount of measurement noise recorded. Because we know the full details of the data system, we can compare estimations for states and parameters that were not directly observed as one check on the accuracy of the assimilation procedure. This process is broken into following steps:

1. Choose a dynamical model to investigate, specified by a set of differential equations $\mathbf{f}$

2. Select a set initial conditions and parameters $\{\mathbf{y}(0), \mathbf{q}\}$ to represent the data system

3. Integrate the data system, $\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}(0), \mathbf{q})$ forward from the initial data conditions to produce a data trajectory $\mathbf{Y}$

4. Create a *twin* model, $\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}(0), \mathbf{p})$, obeying the same dynamics as the data system, but with undetermined states and parameters.

5. Using some assimilation technique, produce estimates for the states and parameters of the data system $\mathbf{x}^* = \{\mathbf{x}^*(0), \mathbf{p}^*\}$

6. Set the model's state and parameters equal to the estimated values, $\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}^*(0), \mathbf{p}^*)$, and integrate the dynamics beyond the assimilation window

7. Compare the model trajectory to the data trajectory via the cost function

1.4 Software

In the examples presented here, the optimization problem is handled using IPOPT, an interior point solver (Wächter [65]). IPOPT is available as open source code under the Eclipse Public License and can be found at (https://projects.coin-or.org/Ipopt).

To translate dynamical equations into the C++ source code required by IPOPT, we employ a front end which parses text files containing the equations. The front end uses python scripts developed through the work of Brian Toth, Chris Knowlton, and Jingxin Ye (Ye et al. [69]). Numerical integration to produce twin data is handled by the *odeint* libraries in python. The code used to produce the results and figures in this document is available upon request.

1.5 Path Integral Formulation

When using real data, the measurements will necessarily be noisy and we rarely know whether the model chosen is an accurate description of the underlying system. Because of these complications, it is desirable to develop a statistical description of the data assimilation procedure. In particular, we would like the probability distribution for the states and parameters appearing in the model. If we have a distribution for the parameters and initial states (we can assume a uniform distribution for example),

then there are two factors which shape our distribution as we go forward in time. First, we need to account for how the dynamics of the system change the distribution at each time step. Second, we must include the impact of measurements on the distribution. Formally, this process is described by the recursive Bayesian estimation (see for example Särkkä [56]). Before delving into the detailed form of our approach, we will review the necessary basic probability theory.

Throughout this section, we will assume that the time interval $[0, T]$ has been split into discrete time steps $t_0, t_1, ..., t_N$, each of length Δt. We write the distribution for the state of the system at each time as a D dimensional vector $\mathbf{x}_i$ and the state of the system up until some time t_m as $\mathbf{X}_m = \{\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_m\}$. We denote the measurements as L dimensional vectors $\mathbf{y}_j$, where $L \leq D$ and the set of measurements up until time t_m as $\mathbf{Y}_m = \{\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_m\}$. The case of measurements at all times can be trivially generalized to measurements at an arbitrary subset of times by letting some $\mathbf{y}_j = 0$. Additionally, we note that the set of measurements is *separable*, allowing us to write $\mathbf{Y}_m = \{\mathbf{Y}_{m-1}, \mathbf{y}_m\}$.

1.5.1 Conditional Probability

Conditional probability is defined as the probability that one event has occurred given knowledge of a second prior event. We will call the event in question event A and the prior event B. The probability that both A and B occur is then computed by multiplying the probability of B occurring by the probability that A occurs, given that B has occurred, which we write as

$$P(A, B) = P(A|B) * P(B). \tag{1.5}$$

In this context, we can rephrase the estimation problem to ask what is the distribution of final states given knowledge of the measurements, or using the definition given in eqn. (1.5), what is $P(\mathbf{x}_N|\mathbf{Y}_N)$.

1.5.2 Conditional Mutual Information

Conditional mutual information (CMI) provides a useful way to think about the estimation process. We can think of the measurements as transferring new information to the model estimate in the sense described in Fano and Hawkins [15]. The conditional mutual information can be written in terms of the conditional probability as

$$CMI(\mathbf{x}_m, \mathbf{y}_m|\mathbf{Y}_{m-1}) = log\left[\frac{P(\mathbf{x}_m, \mathbf{y}_m|\mathbf{Y}_{m-1})}{P(\mathbf{x}_m|\mathbf{Y}_{m-1})P(\mathbf{y}_m|\mathbf{Y}_{m-1})}\right], \tag{1.6}$$

and if the logarithm is taken in base 2, the resulting quantity has units of bits. CMI thus provides a quantitative measure of the information transferred to the model system by each measurement.

1.5.3 Markov Processes

It is assumed that the discretized dynamics of the equations we consider obey the Markov property, which is to say that the state of the system at a time t_{m+1} is completely determined by the state of the system at time step t_m. This is the case for the ordinary differential equations considered here, but would not be the case for delay differential equations, for example.

The Markov assumption allows us to greatly simplify expressions for the joint

probability distribution by recognizing

$$P(\mathbf{x}_m|\mathbf{x}_{m-1}, \mathbf{x}_{m-2}, ..., \mathbf{x}_0, \mathbf{Y}_m) = P(\mathbf{x}_m|\mathbf{x}_{m-1}), \tag{1.7}$$

where the dependence on anything other than the previous state, $\mathbf{x}_{m-1}$, has been dropped in the last term.

1.5.4 Derivation of Path Integral

Once measurements are introduced to inform the model used in assimilation, what we seek is the final probability distribution of states and parameters conditioned on the time series of measurements. We can write the distribution of state and parameters at time t_m as $P(\mathbf{x}_m|\mathbf{Y}_m)$. The basic idea is to obtain an expression for the distribution which depends only on the previous time, and then apply this technique recursively. Following this procedure ultimately leads an expression for the model state at the final time.

We begin by using an identity which gives $P(\mathbf{x}_m|\mathbf{Y}_m)$ in terms of only probabilities which are conditioned on the previous $m-1$ measurements. By twice applying

a factor of unity we find,

$$P(\mathbf{x}_m|\mathbf{Y}_m) = \frac{P(\mathbf{x}_m, \mathbf{Y}_m)}{P(\mathbf{Y}_m)} \tag{1.8}$$

$$= \frac{P(\mathbf{x}_m, \mathbf{y}_m, \mathbf{Y}_{m-1})}{P(\mathbf{y}_m, \mathbf{Y}_{m-1})} \tag{1.9}$$

$$= \frac{P(\mathbf{x}_m, \mathbf{y}_m|\mathbf{Y}_{m-1})P(\mathbf{Y}_{m-1})}{P(\mathbf{y}_m|\mathbf{Y}_{m-1})P(\mathbf{Y}_{m-1})} \tag{1.10}$$

$$= \frac{P(\mathbf{x}_m, \mathbf{y}_m|\mathbf{Y}_{m-1})}{P(\mathbf{y}_m|\mathbf{Y}_{m-1})} \tag{1.11}$$

$$= \frac{P(\mathbf{x}_m, \mathbf{y}_m|\mathbf{Y}_{m-1})}{P(\mathbf{y}_m|\mathbf{Y}_{m-1})} \frac{P(\mathbf{x}_m|\mathbf{Y}_{m-1})}{P(\mathbf{x}_m|\mathbf{Y}_{m-1})} \tag{1.12}$$

In going from the first line to the second, we have used separability, while in going from the second to the third, we have used the definition of conditional probability given in eqn. (1.5).

What we would like to have is the final probability distribution given in terms of the distribution of initial states. In other words, while the identity appearing in eqn. (1.8), steps us back in time by one measurement (connecting $\mathbf{Y}_{m-1}$ and $\mathbf{Y}_m$), we now need an equation that steps us back by one state (connecting $\mathbf{x}_m$ to $\mathbf{x}_{m-1}$). We use the Markov property, along with the Chapman-Kolmogorov equation (see for example Van Kampen [64]) to connect the distribution of states at different points in time. In the form needed here, the Chapman-Kolmogorov equation reads

$$P(\mathbf{x}_m|\mathbf{Y}_{m-1}) = \int d^D x_{m-1} P(\mathbf{x}_m|\mathbf{x}_{m-1}, \mathbf{Y}_{m-1}) P(\mathbf{x}_{m-1}|\mathbf{Y}_{m-1}) \tag{1.13}$$

$$= \int d^D x_{m-1} P(\mathbf{x}_m|\mathbf{x}_{m-1}) P(\mathbf{x}_{m-1}|\mathbf{Y}_{m-1}), \tag{1.14}$$

where the last equality comes from application of the Markov property. Repeating this process allows us to connect distribution at the initial time to the time in question.

Now we can expand using eqn. (1.13) to expand the expression derived in eqn. (1.8). This yields

$$P(\mathbf{x}_m|\mathbf{Y}_m) = \frac{P(\mathbf{x}_m, \mathbf{y}_m|\mathbf{Y}_{m-1})}{P(\mathbf{y}_m|\mathbf{Y}_{m-1})} \frac{P(\mathbf{x}_m|\mathbf{Y}_{m-1})}{P(\mathbf{x}_m|\mathbf{Y}_{m-1})}$$
$$= \frac{P(\mathbf{x}_m, \mathbf{y}_m|\mathbf{Y}_{m-1})}{P(\mathbf{y}_m|\mathbf{Y}_{m-1})P(\mathbf{x}_m|\mathbf{Y}_{m-1})} \int d^D x_{m-1} P(\mathbf{x}_m|\mathbf{x}_{m-1}) P(\mathbf{x}_{m-1}|\mathbf{Y}_{m-1}),$$

Next, we apply the same formula to the term $P(\mathbf{x}_{m-1}|\mathbf{Y}_{m-1})$ appearing inside the integral. This yields

$$P(\mathbf{x}_m|\mathbf{Y}_m) = \frac{P(\mathbf{x}_m, \mathbf{y}_m|\mathbf{Y}_{m-1})}{P(\mathbf{y}_m|\mathbf{Y}_{m-1})P(\mathbf{x}_m|\mathbf{Y}_{m-1})} \int d^D x_{m-1} P(\mathbf{x}_m|\mathbf{x}_{m-1})$$
$$\times \frac{P(\mathbf{x}_{m-1}, \mathbf{y}_{m-1}|\mathbf{Y}_{m-2})}{P(\mathbf{y}_{m-1}|\mathbf{Y}_{m-2})P(\mathbf{x}_{m-1}|\mathbf{Y}_{m-2})} \int d^D x_{m-2} P(\mathbf{x}_{m-1}|\mathbf{x}_{m-2}) P(\mathbf{x}_{m-2}|\mathbf{Y}_{m-2}),$$

which allows us to go from $\mathbf{x}_{m-2}$ to $\mathbf{x}_{m-1}$ to $\mathbf{x}_m$. Applying this formula recursively N_T times, we eventually arrive at the term $P(\mathbf{x}_0|\mathbf{Y}_0) = P(\mathbf{x}_0|\mathbf{y}_0) = P(\mathbf{x}_0)$, since no measurement is made at $\mathbf{y}_0$. The full formula resulting from the repeated application of the Chapman-Kolmogorov equation is given by

$$P(\mathbf{x}_m|\mathbf{Y}_m) = \int \prod_{m=1}^{N_T} d^D x_{m-1} \frac{P(\mathbf{x}_m, \mathbf{y}_m|\mathbf{Y}_{m-1})}{P(\mathbf{y}_m|\mathbf{Y}_{m-1})P(\mathbf{x}_m|\mathbf{Y}_{m-1})} P(\mathbf{x}_m|\mathbf{x}_{m-1}) P(\mathbf{x}_0) \quad (1.15)$$

which has the interpretation of a path integral. Alternatively, we can express the path integral in terms of the conditional mutual information,

$$P(\mathbf{x}_m|\mathbf{Y}_m) = \int \prod_{m=1}^{N_T} d^D x_{m-1} exp(\sum_{n=0}^{m} CMI(\mathbf{x}_n, \mathbf{y}_n|\mathbf{Y}_{n-1}) + \log(P(\mathbf{x}_n|\mathbf{x}_{n-1})) + \log(P(\mathbf{x}_0))) \quad (1.16)$$

In either case, we are left with a high dimensional integral, which cannot be easily

evaluated in general. Techniques for approximating the integral are discussed further on.

1.5.5 Action

From comparison to other well studied path integral problems in physics (Feynman and Hibbs [16], Zinn-Justin [70]), we can define a quantity called the action, based on the definition

$$P(\mathbf{x}_m|\mathbf{Y}_m) = \int d\mathbf{X} \exp(-A_0(\mathbf{X}_m, \mathbf{Y}_m)), \tag{1.17}$$

which leads to the formulation of the action according to

$$A_0(\mathbf{X}, \mathbf{Y}_m) = -\sum_{n=0}^{m} CMI(\mathbf{x}_n, \mathbf{y}_n|\mathbf{Y}_{n-1}) - \sum_{n=0}^{m-1} \log(P(\mathbf{x}_{n+1}|\mathbf{x}_n)) - \log(P(\mathbf{x}_0)). \tag{1.18}$$

The first term represents the information transferred between the measurements and model state and the second represents the transition probabilities from the state at one time to the state at the next.

In analogy to statistical physics, we can compute expectation values of quantities of interest via the formula

$$E[G(\mathbf{X})|\mathbf{Y}] = \frac{\int d\mathbf{X} G(\mathbf{X}) exp[-A_0(\mathbf{X}, \mathbf{Y}_m)]}{\int d\mathbf{X} exp[-A_0(\mathbf{X}, \mathbf{Y}_m)]} \tag{1.19}$$

From inspection of eqn. (1.19), we can see that the term $-\log(P(\mathbf{x}_0))$ appearing in eqn. (1.18) will cancel when computing expectation values and can be dropped from the *effective action*.

1.5.6 Transition Probabilities

So far, we have glossed over the explicit form for the transition probabilities.We treat the transitions differently depending on whether or not we account for the presence of noise, and consider the noise-free case first. Assuming the correctness of our model dynamics, then the transition probability is given by a delta function through the equation

$$P(\mathbf{x}_{m+1}|\mathbf{x}_m) = \delta^D(\mathbf{x}_{m+1} - \mathbf{F} \cdot \mathbf{x}_m). \tag{1.20}$$

In the case of model noise, it is not immediately clear how we should broaden the distribution. However, in the limit that the noise goes to zero, we should be able to recover the delta function. If the model errors are given by a Gaussian, we then have

$$P(\mathbf{x}_{m+1}|\mathbf{x}_m) = exp\left[-\sum_{a,b=1}^{D}(F_a(\mathbf{x}_m) - \mathbf{x}_{m+1})\frac{[R(m)]_{ab}}{2}(F_b(\mathbf{x}_m) - \mathbf{x}_{m+1})\right], \tag{1.21}$$

where $[R(m)]_{ab}$ is the inverse of the covariance matrix specifying the model noise at each time point.

In practice $\mathbf{F} \circ \mathbf{x_m}$, the discretized forward mapping is, given by some numerical rule. In selecting $\mathbf{F}$, we typically choose a function which depends only on the end points of the interval, such as the trapezoidal rule.

1.5.7 Computational Form of the Action

We now consider the impact of additive noise in the measurements. Taking the noise to be independent in the measurements and Gaussian distributed,

$$CMI(\mathbf{x}_m, \mathbf{y}_m|\mathbf{Y}_{m-1}) = -\sum_{l,k=1}^{L}(h_l(\mathbf{x}_m) - \mathbf{y}_{lm})\frac{[R_f]_{lk}}{2}(h_k(\mathbf{x}_m) - \mathbf{y}_{km}) \tag{1.22}$$

Combining equations 1.22 and 1.21 results in an expression for the action,

$$A(\mathbf{x}_m, \mathbf{y}_m) = \frac{R_F}{2N_T}\sum_{t=0}^{N_T-1}\sum_{i=1}^{M}(h_i(\mathbf{x}(t)) - y_i(t))^2 + \frac{R_M}{2N_T}\sum_{t=0}^{N_T-2}\sum_{i=1}^{N_x}(x_i(t+1) - F_i(\mathbf{x}(t)))^2. \tag{1.23}$$

1.5.8 Approximating the Path Integral

The path integral appearing in eqn. (1.19) is high dimensional and therefore computationally expensive to evaluate directly. To approximate the path integral we consider two methods: Laplace's method and Monte-Carlo techniques. We will discuss Laplace's method in some detail and refer the reader to Kostuk [32] for a discussion of how Monte Carlo techniques can be implemented to evaluate the path integral.

Laplace's method is a well established technique for evaluating integrals of exponentials (see for example Orszag and Bender [48]). By requiring a stationary path, we get additional criteria that the Jacobian vanishes,

$$\frac{\partial A_0(\mathbf{X})}{\partial \mathbf{X}}|_{X=X^q} = 0 \tag{1.24}$$

and that the Hessian,

$$\frac{\partial^2 A_0(\mathbf{X})}{\partial^2 \mathbf{X}}|_{X=X^q} \tag{1.25}$$

is positive definite. Non-extremum paths give contributions to the integral which vanish exponentially.

One advantage of this formulation is that Laplace's method also gives us a way to compute higher order corrections if we desire higher accuracy. In geophysical literature, finding the extremum path is called 4DVar. The general case given by eqn. (1.23) is weakly constrained and is referred to as weak4Dvar. Note that if we consider the limit $R_f \to \infty$, we recover the strongly constrained case.

1.6 Simulated Annealing

There exists a class of heuristic optimization algorithms which perform a process called *simulated annealing*. The driving idea behind these algorithms is to start with a distribution that matches the measurements and slowly impose constraints. This has the physical analog of the eponymous annealing process, in temperature is changed to adjust the effective height of a potential well.

To accomplish this process computationally, we parameterize $R_F = \alpha^\beta$ (α and β are sometimes referred to as *hyperparameters*). We begin with some small value for R_F and step through values of β to some desired maximum. As this happens, the constraints are slowly imposed on the model. The idea behind the method is to start with paths in the vicinity of the data points, and slowly enforce the dynamics in order to obtain a path that is both close to the data, and obeys the equations of motion. We will employ this technique in the subsequent chapters.

Chapter 2

Nonlinear Dynamics

Nonlinear dynamical systems are both qualitatively and quantitatively different from those described by linear equations. Nonlinear dynamical systems provide a general framework that allows us to produce more accurate approximations to the behavior of the underlying system, but at the expense of depriving us of the many tools which depend on linearity.

Many of the tools developed to visualize and intuitively explain the results of nonlinear dynamics are intended for low dimensional spaces. As the spaces we typically work in are very high dimensional these tools can sometimes gives us intuition, but rarely hold the same explanatory power as in lower dimensional space.

Throughout this section, we will use the Hodgkin-Huxley model as any example of a nonlinear system. A full description of the equations and their biophysical motivation can be found in Chapter 4.

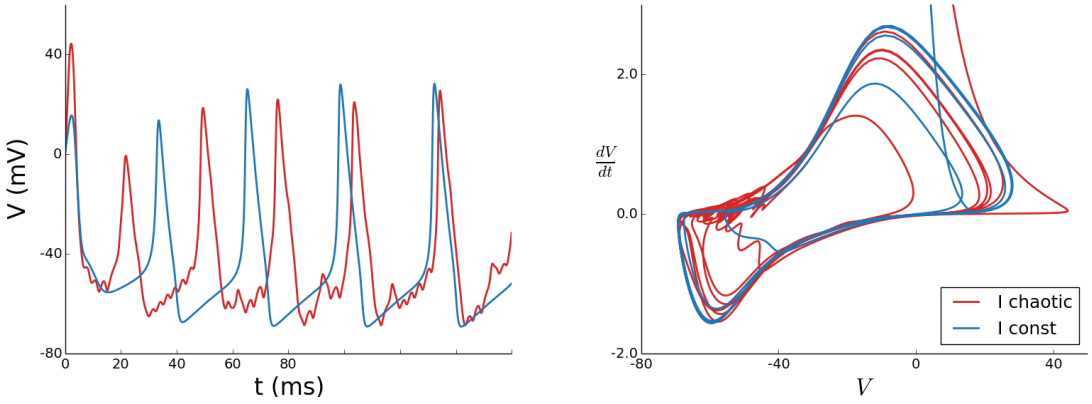

Figure 2.1: Simulation of an NaKL neuron. The left panel shows the voltage time series produced by the model, and the right panel shows the same trajectory in the $\dot{V}$ vs. V projection of the phase space.

2.1 Phase Space

For a nonlinear system defined by the dynamical equation $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{p})$, after integrating forward from some initial condition, we obtain a time series representing the motion of the system through state space. The dynamics of the system are fully determined by specifying $\mathbf{x}$ and $\dot{\mathbf{x}}$, the coordinates of the system in phase space. We can plot trajectories in phase space by plotting $\dot{\mathbf{x}}$ vs. $\mathbf{x}$. For more than a 1-dimensional system, these plots will be cross-sections of a higher dimensional phase space, and may not accurately capture the structure of the underlying dynamics.

When trajectories are plotted in this way, it becomes possible to define graphically several important concepts in nonlinear dynamics. The first of these concepts is called a *fixed point*, and is defined as the solution $\mathbf{x_f}$ to $f(\mathbf{x_f}, \mathbf{p}) = 0$. If a system reaches a fixed point, because the time derivative vanishes, the system remains at the point for all future time. For nonlinear systems, we cannot say in general where such points occur, how many there are, or even whether such points exist, but the concept is still useful in classifying the behavior of nonlinear systems. A related concept is

that of the *nullcline*, which is defined as the line in phase space along which a single coordinate x_i satisfies $f(\mathbf{x_i}, \mathbf{p}) = 0$. From the definition of the nullcline, one may draw the conclusion that fixed points of the system exist at the intersection of nullclines.

Once we have found the fixed points of the system (if any exist), we begin to classify the system based on its stability. Stability refers to the behavior of trajectories which begin in the vicinity of a fixed point. When trajectories which begin near a fixed point tend to converge to that point, we say such a point is *attracting*. Conversely, when trajectories which begin near a fixed point tend to diverge from that point, we say such a point is *repelling*. Trajectories which begin and end on the same fixed point are referred to as *homoclinic*, while trajectories which begin and end on different fixed points are called *heteroclinic*.

For the Hodgkin-Huxley model shown in Figure 2.1, the same simulation was run twice, but with different forcing currents. The model used to produce the blue trace was injected with a constant value of current ($I_{inj} = 0.3\,nA$) while the red trace resulted from a model injected with a current in the shape of a chaotic waveform generated from the Lorenz 63 system. In the phase space diagram, it can be seen that the trajectory of the model injected with a chaotic waveform explores much more of the phase space.

Many nonlinear systems tend to explore a relatively limited subset of phase space. We characterize this tendency for the system to remain localized in a region through the concept of an attractor. Borrowing from Strogatz [60], we can define an attractor as an invariant closed set A for which any trajectory starting in A will stay in A for all time. Additionally, for some set of initial conditions outside of A, the trajectories will converge to A as $t \to \infty$. The set of all starting points for which

trajectories will converge to A is referred to as the *basin of attraction*. These criteria fit with the intuitive definition of an attractor as a region of phase space which tends to attract nearby trajectories.

From examining Figure 2.1, we can see that both trajectories (which begin from initial condition $\mathbf{x}(t_0) = 0$) quickly converge to the same region of phase space. Once they reach this region, they repeatedly trace out the shape of the limit cycle for the system. In general we should not expect that a lower dimensional projection of trajectories in the phase plane will capture the structure of the attractor, but the Hodgkin-Huxley model is an exception to this. Similarly, in Figure 2.2 we start trajectories at various different places in n-V phase plane. As can be seen in the figure, regardless of the initial condition, the trajectories converge to the same limit cycle.

So far we have not discussed the dependence of the dynamics on the set of parameters $\mathbf{p}$. While the behavior of the systems we consider will clearly depend on the specific values we choose $\mathbf{p}$, tools such as *bifurcation diagrams*, which are designed to show the dependence of a nonlinear system on a particular parameter are not typically useful because of the high dimensions of the systems we consider.

2.2 Chaos

To explain some of the potential difficulties of dealing with nonlinear systems, it is useful to introduce the concept of chaos. Chaotic phenomena span the chasm between repetitive and random behavior. The trajectories of chaotic systems are often highly structured, yet not periodic. They are deterministic and yet their behavior cannot be predicted for arbitrarily long times, no matter how good the finite precision of our measurements. They may tend to occupy particular regions of phase space, but

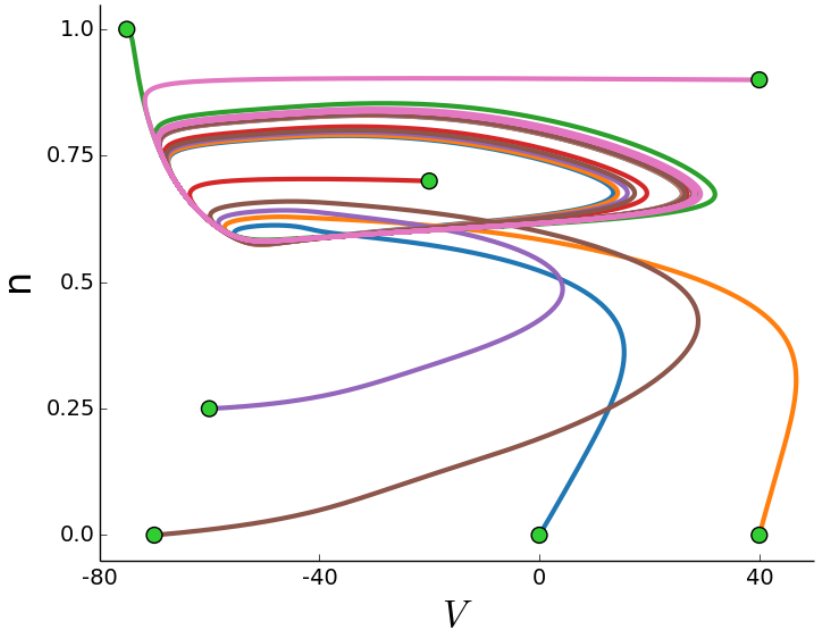

Figure 2.2: NaKL models started from different regions of state space all converge onto the same limit cycle.

do not settle down to fixed points or limit cycles.

Chaos is loosely defined as a sensitive dependence on initial conditions. More precisely, what we mean is that two trajectories which begin very close to one another diverge exponentially in time. For our eventual goal of data assimilation chaos has both positive and negative implications. The good news is that even if we are slightly off in our estimates of the state of a chaotic system, if we wait long enough the divergence in trajectories will become apparent (i.e. bad estimates won't give answers which look good). Conversely, even if our estimate for the state of the system is close to the true solution we may end up with a trajectory which looks very different from the data.

From the Poincare-Benedixson theorem, we know that chaotic trajectories are not possible for a two-dimensional system. In order to observe chaos, we require a

system with $n \geq 3$. This requirement provides a necessary, but by no means sufficient condition for a system to exhibit chaos. To directly measure whether or not a particular dynamical system exhibits chaos, we need a more precise definition provided by the concept of Lyapunov exponents.

2.2.1 Lyapunov Exponents

To quantify the behavior of chaotic systems, we consider the divergence of two trajectories which start with separation $\delta\mathbf{x}(t_0)$. We wish to know the separation of these two points at a later time $T = t_0 + \tau$. If the two trajectories diverge exponentially, we can write the separation at the later time as $\delta\mathbf{x}(T) = \delta\mathbf{x}(t_0)e^{\lambda_1 \tau}$. Here λ_1 is called a Lyapunov exponent.

From the perspective of data assimilation, this exponent can be used to obtain an estimate for the time horizon beyond which our predictions will fail. We now interpret $\delta\mathbf{x}(t_0)$ as the error, or limit in precision, of our estimation of the state $\mathbf{x}$ at time t_0. If our prediction for the state of the system becomes intolerably bad when it is off by some error $\delta\mathbf{x}$, then the prediction may be considered valid for a time of length $t \approx \frac{1}{\lambda_1} \ln \frac{\Delta\mathbf{x}}{\delta\mathbf{x}(t_0)}$. The logarithmic dependence on $\delta\mathbf{x}(t_0)$ means that we need exponentially more precise measurements to predict accurately for another length of time $\frac{1}{\lambda_1}$.

While this discussion provides some intuition into the definition of a Lyapunov exponent, it glosses over the subtlety that there are actually D unique Lyapunov exponents. The impact of these exponents on the system can be imagined by considering the evolution of an infinitesimal D-dimensional sphere of radius $\delta x(t_0)$, representing perturbed initial conditions. During its evolution, the sphere will become distorted

into an ellipsoid with D different principal axes. The D different Lyapunov exponents $(\lambda_1, ..., \lambda_D)$ will yield perturbations described by

$$\delta x_i(T) \approx \delta x_i(t_0) e^{\lambda_i \tau} \tag{2.1}$$

In order to actually compute Lyapunov exponents, it is useful to consider a definition of the exponents stemming from Oseledet's theorem (Oseledec [49]). For numerical calculation, we first break a continuous time interval $[0, T]$ into N_T discrete chunks of time dt. We use a discrete forward mapping $\mathbf{F}$ to connect these according to $x_i(t + dt) = F_i \circ (\mathbf{x}, \mathbf{p})$. Next, we consider the linearized dynamics local to each point in space, which can be approximated by the Jacobian of the mapping given by

$$J_{ij} = \frac{\partial x_i(t + dt)}{\partial x_j(t)}|_x \tag{2.2}$$

By repeatedly applying the Jacobian, we can get the transformation of the system over the entire time interval. This equation can be written,

$$\mathbf{x}(T) = \mathbf{J}^{N_T}(\mathbf{x}(t_0)) = \mathbf{J}|_{\mathbf{x}(t_{N_T-1})} \cdot \mathbf{J}|_{\mathbf{x}(t_{N_T-2})} \cdots \mathbf{J}|_{\mathbf{x}(t_1)} \cdot \mathbf{J}|_{\mathbf{x}(t_0)} \cdot \mathbf{x}(t_0) \tag{2.3}$$

We then take the long time limit of the product of this repeated mapping with its transpose, which ensures the eigenvalues are real. Expressing this matrix in terms of the product given in eqn. (2.3), we can write

$$\mathbf{OSL}(\mathbf{x}(t_0)) = \lim_{N_T \to \infty} \left[(\mathbf{J}^{N_T})^T (\mathbf{J}^{N_T}) \right]^{\frac{N_T}{2}} \tag{2.4}$$

The eigenvalues of this matrix are the Lyapunov exponents of the system. A practical

method for using QR decomposition to compute the Lyapunov exponents is described in Abarbanel [1] and Sandri [55].

2.3 Synchronization

Synchronization is one of the more interesting discoveries in nonlinear dynamics (see for example Pecora and Carroll [50]). The discovery of synchronization can arguably be traced to Christiaan Huygen's 17th century discovery that clocks connected to a common beam will synchronize, but has much more general applications. In particular, nonlinear and even chaotic nonlinear systems can synchronize when coupled together.

We will now explore how synchronization can be used in numerical data assimilation. The general ideas behind this concept have been developed over some time (see for example So et al. [57] and Sorrentino and Ott [58]). Consider a data system whose dynamics are specified by a D-dimensional set of equations,

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}, \mathbf{g}), \tag{2.5}$$

where $\mathbf{g}$ is the P-dimensional set of parameters. If we could measure all of the components of $\mathbf{y}$ and $\mathbf{g}$, the problem of state estimation would be trivial. Therefore, we will assume that we can measure only the first L components (note that this case can be generalized to the set of any L components by simply reordering the equations for the data system). The idea behind synchronization is that we will use the time series of measurements to drive our estimation towards the data in much the same way that a spring force drives a mass towards its equilibrium position.

To represent our estimation we introduce a twin system called the model system. The model system has the same dynamics as the data system, but not necessarily the same state or parameters. We then augment the model dynamics by adding a coupling term along the L measured dimensions. The strength of the coupling in each component is given by γ_i where $i = 1, ..., L$. We can write the set of D differential equations giving the state of the model as

$$\begin{aligned}
\frac{dx_1}{dt} &= f_1(\mathbf{x}, \mathbf{p}) + \gamma_1(y_1 - x_1) \\
&\vdots \\
\frac{dx_L}{dt} &= f_L(\mathbf{x}, \mathbf{p}) + \gamma_L(y_L - x_L) \\
\frac{dx_{L+1}}{dt} &= f_{L+1}(\mathbf{x}, \mathbf{p}) \\
&\vdots \\
\frac{dx_D}{dt} &= f_D(\mathbf{x}, \mathbf{p}),
\end{aligned} \tag{2.6}$$

where we have only included coupling terms for the components which can be measured. Sometimes γ is referred to as the control parameter, a term commonly used when this technique is applied to problems of control theory.

The coupling term is *not* part of the dynamics describing the data system, will tend to drive the model toward the measured components of the data. *A priori* this is no great accomplishment, because the best estimates of the measured components can be determined directly from using the measurements at a given time. What makes this result remarkable is that the D-L *unmeasured* components will also synchronize, allowing us to also estimate the unmeasured components of the state.

There are several limitations to this technique. First, in the form given in eqn.

(2.6), there is no way to estimate the parameters. Second, if the underlying dynamics or dimensionality of the model and data systems differ, we cannot account directly for this discrepancy. Third, by introducing a control parameter, we alter the dynamics of the system. The larger the value of the control parameter, the faster model is driven towards data, but the more erroneous the dynamics. This last complication can be addressed by adding a penalty term for a large control into the cost function. Adding a term such as γ^2 to the cost function is referred to as regularization, and prevents search algorithms from simply maximizing the coupling term to drive model and data together, preventing overfitting of the data.

To measure whether or not synchronization has occurred it is useful to define the synchronization error (SE). The synchronization error between two time series is a running average of the squared distance between the two given by

$$SE(\mathbf{X}, \mathbf{Y}) = \frac{1}{N_T + 1} \sum_{t=0}^{N_T} \sum_{i=0}^{D} (x_i(t) - y_i(t))^2. \tag{2.7}$$

Note that we can only compute this quantity when we can directly measure all components. Techniques for approximating the synchronization error in the case of limited measurements will be explored in the next chapter. Nevertheless, the concept of SE is theoretically useful especially when we consider the long time limit. For functionally equivalent systems (systems which have the same dynamical equations and parameters) then as $N_T \to \infty$ we will have $SE \to 0$. However, for systems with different parameters we will have $SE \to c$ where $c > 0$ even as $N_T \to \infty$. A discussion of SE in relation to critical phenomena and the necessary conditions for synchronization to occur can be found in Kostuk [32].

2.3.1 Conditional Lyapunov Exponents

It is desirable to know under what conditions two systems will synchronize. We can offer a criterion for synchronization by considering properties of the coupled system described by eqn. (2.6). We can characterize the behavior of the coupled system by calculation of its Lyapunov exponents called conditional Lyapunov exponents (CLEs). If all of the CLEs are negative, the two systems will eventually converge.

Chapter 3

Time Delayed Newton's Method

3.1 Introduction

In this section, we consider an approach to estimate the initial state of a system given a time series of measurements. This estimation problem is distinct from those discussed previously in that we will seek only the estimate of the state at the initial time t_0 and do not explicitly solve for the path at every time.

For the sake of simplicity, we will consider only cases in which a single component of the data can be measured (yielding a 1-D time series of measurements). Additionally, we will primarily consider the case of noise free measurements. Both of theses constraints can be generalized.

Imagine that we are observing a D-dimensional system whose state is given by $y_i(t)$ where $i = 1, ..., D$. We will assume our measurements have some finite time resolution dt and can be recorded only during the interval $t = [t_0, T]$. Of the D states, we will assume that we can measure only the first, $y_1(t)$. In terms of a measurement function, we can define $h(\mathbf{y}(t)) = y_1(t)$. Geometrically, we can think of think of the

measurement operator as projecting the D-dimensional path taken by the system down onto the y_1-axis.

We will call the true state of the data system at the initial time $\mathbf{y}^*(t_0)$. Our goal is to determine $\mathbf{y}^*(t_0)$ from the time series of available measurements and our estimate can be checked through a model system described by $\mathbf{F}(\mathbf{x}(t), t) = \dot{\mathbf{x}}(t)$. Since we can directly measure $y_1(t_0)$, we can immediately use this measurement to set $x_1(t_0) = y_1(t_0)$ and all that remains is to find those $x_i(t_0)$ for which $i \neq 0$. We can define the difference between our estimate and the true state $\mathbf{\Delta x}(t_0) = \mathbf{y}^*(t_0) - \mathbf{x}(t_0)$, and since we can directly measure the first component, in the case of a noise-free measurement $\Delta x_1(t_0) = y_i^*(t_0) - x_1(t_0) = 0$. In general, we can think of the problem as trying to drive all components of $\mathbf{\Delta x}(t_0) = 0$. Under this condition, the initial state of the model system is identical to the initial state of the data system and we have achieved an arbitrarily accurate estimate.

The broad idea of the estimation algorithm described below is to use information passed from the components $y_i(t)$ $(i \neq 1)$ to $y_1(t)$ by means of the dynamics. Using numerical integration allows us to look forward in time and see any divergence of our model from the data along the measured $x_1(t)$ trajectory. At the initial time, because we can measure $y_1^*(t_0)$ directly our model and the observed data coincide. However, when we integrate forward in time the other components of $y_i^*(t_0)$ start to influence $y_1(t)$ and we can begin to see the effects of any errors in our estimate. We will use the separation between $y_1(t)$ and $x_1(t)$ to inform our estimate of the initial state.

3.2 Time Delay Embedding

The well established concept of time delay embedding (Abarbanel [1]) allows us to extract additional information from a scalar time series. The general idea is to choose temporally separated points the time series to serve as independent coordinates of a proxy or embedding space. The scalar time series can then be unfolded or unprojected onto the embedding space in a way that preserves system topology if the number of time delayed coordinates is large enough. The embedding process depends on two parameters, τ and D_M. The time delay, τ, is the temporal separation between points used as independent coordinates in the embedding space. The time delay should be sufficiently long that points separated by a time τ can be considered independent. If the dynamics under consideration were linear, we could search for a zero of the autocorrelation function as a way to choose a time delay, but for nonlinear systems this approach fails. There are many ways to determine an appropriate time delay (Fraser and Swinney [17]).

As with the choice of an appropriate time delay, there exist multiple proscriptions for choosing an appropriate embedding dimension. Takens' theorem (Takens [61]) shows that choosing an embedding dimension of $2D_A + 1$, where D_A is the dimension of the attractor described by the system of differential equations, is sufficient to fully unproject the scalar time series. Other techniques, such as the method of false nearest neighbors (Kennel et al. [29]), and an approach based on functional continuity (Pecora et al. [51]), provide means for determining a necessary embedding dimension that is potentially of lower dimension than the upper limit provided in Takens' theorem. Twin experiments also allow for the additional option of determining the embedding dimension experimentally.

Once we have chosen our time delay and embedding dimension, we can construct vectors in embedding space. However,we should note that the use of time delay embedding here is not to fully unfold a projection, but rather to introduce additional independent measurements that can be used to extract more information about a data system. Assuming the temporal resolution of our data is Δt and the series of data points is of length T,we can write the time series of the first component of the data system, starting at time t_0, as

$$y_1(t) = \{y_1(t_0), y_1(t_0 + \Delta t), y_1(t_0 + 2\Delta t), ..., y_1(t_0 + T)\}.$$

Since we measure only the first component of the system, this time series is the sum total of what we have to use in estimation. From $y_1(t)$, we choose a time series consisting a of a number $D_M - 1$ of *time-advanced coordinates*. The set of D_M time-advanced coordinates can be obtained from the time series $y_1(t)$ by choosing samples along the measured trajectory separated by a time $\tau = K * \Delta t$, where K is some integer. We can write the set of time advanced coordinates as

$$S_j(y_1(t)) = \{y_1(t_0), y_1(t_0 + \tau), y_1(t_0 + 2\tau), ..., y_1(t_0 + (D_M - 1)\tau)\},$$

where j $= 1, ..., D_M - 1$. In theory, K can range from a lower bound of 0, in the case no time-advanced coordinates, to an upper value of $K \leq T/(D_M * \tau))$. Note that the subscript j labels the components of S, which in turn consists entirely of the $i = 1$ state.

The method we use for choosing K is somewhat heuristic, though there is a case to be made against choosing values too small or too large. In the short time limit

$K = 1$ and we data points at the same frequency as the data is recorded. Depending on the system and same frequency Δt, the system may not have time to vary substantially between data points and so the coordinates will not be independent. In the case of noise, if the underlying dynamics have not had enough time to move the system substantially, we will struggle to discern the signal from the noise. On the other extreme, using too long a time delay will limit the number of proxy space coordinates we can choose. Additionally, for chaotic systems errors will tend to grow in time and so using too long of a delay may cause instabilities.

3.3 Synchronization Error

Synchronization error provides us with a way to measure the progress of our algorithm. Synchronization error (in state space) for a D-dimensional state variable can be defined by the equation

$$SE_{state}(t) = \frac{1}{D}\sum_{i=1}^{D}(y_i(t) - x_i(t))^2, \tag{3.1}$$

which provides a running measure of the separation between the model estimation for the state and the data state. While the definition of synchronization error provided in equation (3.1) gives a precise measure of the proximity of model and data systems, it is an unusable measure when we cannot directly measure each $y_i(t_0)$. Indeed, a system in which we could directly measure $y_i(t_0)$ for all i would be fully observable and require no estimation.

We can construct a version of synchronization error using only measured quantities by including only those components we can measure directly. For the case

in which we measure only the first component, we thus define

$$SE_1(t) = (y_1(t) - x_1(t))^2, \tag{3.2}$$

where summing over the components is no longer necessary as the measurement function has been assumed to pick out only the first component of the state vector. However, this quantity provides no information on the unmeasured components and therefore is insufficient to be used to estimate the complete state of the data system.

To do better we will construct the synchronization error in embedding space, where all components are known by construction, and then map the synchronization error back to state space to provide an approximation of the error in all components. We define the synchronization error in embedding space by,

$$SE_s(t) = \frac{1}{D_M}\sum_{j=1}^{D_M}(\Delta S_j(t))^2 = \frac{1}{D_M}\sum_{j=1}^{D_M}(S_j(y_1(t)) - S_j(x_1(t)))^2. \tag{3.3}$$

This formulation will serve as our primary means of searching for the data state.

3.4 Formulation As Error Estimation

Seen in another light, estimating the state of the data system is equivalent to determining the error in the model's initial condition. If we have an accurate estimate of the error, we can simply change the coordinates of the model's initial condition to eliminate it. The complication comes from estimating the error along those directions for which we lack measurements, but equipped with the tool of time-delay coordinates, we can derive an expression to approximate this error. Expanding the notation used

in describing the model state, we will call the n^{th} guess of the model state $x_i^n(t_0)$. The superscript is here introduced to keep track of the iteration of our estimation procedure. For example, $x_i^0(t_0)$ refers to the our first guess at the data state at time t_0. If we can somehow obtain information about the unmeasured states of the model, we may be able to improve upon our initial guess with a second estimation $x_i^1(t_0)$ and continue this process iteratively.

We will call the separation in vector space between any given guess and the data state $\Delta x_i^n(t)$. Thus, by definition we have $\Delta x_i^n(t_0) = y_i^*(t_0) - x_i^n(t_0)$. Further we will define $\mathbf{S}$ to be the function that generates the a time-delay vector given an initial set of coordinates $\mathbf{x}^n(t_0)$. Computationally, this requires integrating the dynamics of the model forward from $\mathbf{x}^n(t_0)$, and then picking out the time-delayed coordinates. The set of measurements available in the data system is thus given by $\mathbf{S}(\mathbf{y}^*(t_0))$, while the trajectory produced by any estimate for the initial state is given by $\mathbf{S}(\mathbf{x}^n(t_0))$.

We would like to use local information to improve our guess. If we are close to the true solution, it is reasonable to use a linearized version of $\mathbf{S}$ to improve our estimate. Using the equivalence of $\mathbf{S}(\mathbf{y}^*(t_0)) = \mathbf{S}(\mathbf{x}^n(t_0) + \Delta\mathbf{x}^n(t_0))$, we can Taylor expand the right hand side. This yields,

$$S_j(y_i^*(t_0)) \approx S_j(x_i^n(t_0)) + \left.\frac{\partial S_j}{\partial x_i}\right|_{t_0} \cdot \Delta x_i^n(t_0) \tag{3.4}$$

Substituting into equation 3.4, we find,

$$\Delta x_i^n \approx \left[\left.\frac{\partial S_j}{\partial x_i}\right|_{t_0}\right]^{-1} \cdot \left[S_j(y_i^*(t_0)) - S_j(x_i^n(t_0))\right], \tag{3.5}$$

suggesting that a good next guess would then be

$$x_i^{n+1}(t_0) = x_i^n(t_0) + \Delta x_i^n(t_0) \tag{3.6}$$

The expression given in equation 3.6 gives us yet another way to think about synchronization error. Using,

$$\Delta x_i \approx \frac{\partial S_j}{\partial x_i} \Delta S_j, \tag{3.7}$$

we can define an estimated state space error

$$SE_{est}(t) = \frac{1}{D} \sum_{i=1}^{D} (\Delta x_i)^2. \tag{3.8}$$

The value of SE_{est} tells us how large a step should be taken. This last definition represents how close the algorithm "thinks" the model is to the true solution and when it goes to zero the algorithm has converged on a solution.

Thus, from equations 3.5 and 3.6, we can create an iterative algorithm to seek out $y_i^*(t_0)$. A discussion of the actual implementation of this algorithm is given in the next section. The computationally tricky step in the estimation described above comes from determining $\frac{\partial \mathbf{S}}{\partial \mathbf{x}}$ and $\frac{\partial \mathbf{x}}{\partial \mathbf{S}}$ and warrants substantial further discussion.

3.5 Gauss-Newton

We can connect the algorithm shown here to a broad class of techniques based on the Gauss-Newton method. There are many well established non-linear solvers which draw on Gauss-Newton, so making the connection opens up a trove of existing experience. We pause briefly to introduce and define the Gauss-Newton method.

Gauss-Newton is used to solve the non-linear least squares problem. Consider a vector of m functions $\mathbf{r} = r_1, ..r_m$ each of which depends on D variables $\mathbf{x} = x_1, ..., x_D$. The least squares cost function in then defined according to

$$C(\mathbf{x}) = \sum_{i=1}^{m} r_i^2(\mathbf{x}). \tag{3.9}$$

Starting from an inital guess $\mathbf{x}^0$, the algorithm proceeds iteratively, with the next guess determined according to

$$\mathbf{x}^{n+1} = \mathbf{x}^n - (\mathbf{J_r}^T\mathbf{J_r})^{-1}\mathbf{J_r}^T\mathbf{r}(\mathbf{x}^n), \tag{3.10}$$

where J_r is the Jacobian matrix defined by

$$\mathbf{J_r} = \frac{\partial r_i(\mathbf{x}^{(n)})}{\partial x_j}. \tag{3.11}$$

We will now show a way of using time-delays to estimate state and parameters by applying Gauss-Newton to the cost function given in equation (3.3). We begin by noting that the combination $(\mathbf{J_r}^T\mathbf{J_r}^{-1})\mathbf{J_r}^T$ is a generalized inverse, which is not necessarily unique for rectangular matrices. The details of calculating this quantity will be discussed below. If we choose $\mathbf{r}(\mathbf{x}(t_0)) = \mathbf{S}(\mathbf{x}(t_0) - \mathbf{S}(\mathbf{y}^*(t_0)) = \mathbf{\Delta S}$, then the

formula for the next Gauss-Newton step becomes

$$\begin{aligned}\mathbf{x}^{(n+1)} &= \mathbf{x}^{(n)} - (\mathbf{J_r}^T\mathbf{J_r}^{-1})\mathbf{J_r}^T\mathbf{\Delta S} \\ &= \mathbf{x}^{(n)} - \left[\left(\frac{\partial \mathbf{s}}{\partial \mathbf{x}}\right)^T\left(\frac{\partial \mathbf{s}}{\partial \mathbf{x}}\right)\right]^{-1}\left(\frac{\partial \mathbf{s}}{\partial \mathbf{x}}\right)^T\mathbf{\Delta S} \\ &= \mathbf{x}^{(n)} - \mathbf{R}^{-1}\mathbf{\Delta S}\end{aligned}$$

where the second equality follows from noting $J_t = \frac{\partial \mathbf{r}}{\partial \mathbf{x}} = \frac{\partial \mathbf{s}}{\partial \mathbf{x}}$.

As a last remark, Gauss-Newton can in general be derived from linearly approximating the vector of functions $\mathbf{r}$. In light of this, it is clear that the derivation given in terms of error estimation is a specific example of this.

3.6 Variational Equation

Variational calculus provides us with the tools to construct a local mapping between embedding and state space coordinates. The variational equation describes the relationship between two sets of coordinates as a function of time. Because we are using embedding coordinates, the variational equation describes the impact of a infinitesimal perturbation to a state space coordinate at a later time. If the Jacobian is defined according to

$$J_{ab}(t) = \frac{\partial S_a(x(t))}{\partial x_b(t)}, \tag{3.12}$$

then the variational equation describing the time evolution of that Jacobian reads,

$$\frac{dJ_{ab}(t)}{dt} = \sum_{c=1}^{D} \frac{\partial F_a(x(t))}{\partial x_c(t)} J_{cb}(t). \tag{3.13}$$

Using the variational equation, we can integrate $J_{ab}(t)$ forward in time. In practice, integration of the variational equation and of the dynamics is performed simultaneously, using the same numerical routine. Initially state and embedding coordinates are identical ($S_1(t_0) = x_1(t_0)$) which implies that the appropriate initial condition to use in equation 3.13 is $J_{ab}(t_0) = I$, the identity matrix.

The method used to integrate the variational equation is based on the procedure outlined in Sandri [55]. Because the matrix grows in time, to prevent overflow we use QR decomposition at each step. QR decomposition factors a matrix into the product of an orthogonal matrix, Q, and an upper triangular matrix R. The Q portion of this decomposition is propagated along to the next time step at which point it is factored again. The decomposition procedure thus allows us to rescale our matrix as we integrate, circumventing overflow issues. As a side note, the R portion can be used to easily compute the Lyapunov exponents.

Some subtlety arises in extracting the appropriate coordinate mapping from the variational equation. Specifically, we need the matrix

$$R_{ab}(t) = \frac{\partial S_a(x(t))}{\partial x_b(t_0)}, \tag{3.14}$$

so that we can measure the impact of a perturbation in the state space coordinate at a fixed initial time. Assuming that we are only able to measure the $x_1(t)$ component of the data, we form $R_{ab}(t)$ from the rows of the Jacobian matrix as defined in equation

(3.12) according to the formula,

$$R|_t = \begin{pmatrix} J_{11}(t) & \cdots & J_{1D}(t) \\ J_{11}(t+\tau) & \cdots & J_{1D}(t+\tau) \\ \vdots & \vdots & \vdots \\ J_{11}(t+(D_M-1)\tau) & \cdots & J_{1D}\big(t+(D_M-1)\tau\big) \end{pmatrix}. \tag{3.15}$$

The resulting matrix $R_{ab}(t)$ tells us the local relationship between an embedding coordinate and an infinitesimal state space perturbation at t_0.

3.7 Calculating the Pseudoinverse

So far, we have overlooked the thorny issue of how to compute the quantity $\left[\frac{\partial \mathbf{s}}{\partial \mathbf{x}}\right]^{-1}$, which is the bulk of the computational work required and thus warrants a more in depth discussion. In order to find this quantity, first the variational equation is integrated forward and used to produce $\frac{\partial \mathbf{s}}{\partial \mathbf{x}}$, which is then inverted numerically to find the Moore-Penrose pseudoinverse. Because $\frac{\partial \mathbf{s}}{\partial \mathbf{x}}$ is a $D_M \times D$ matrix, it is rectangular in general and therefore does not necessarily have a unique inverse. This ill-posed problem can be solved through a process known as Tikhonov regularization (Tikhonov et al. [63]), the general idea of which is to provide an additional criterion by which to choose *which* pseudoinverse we obtain.

The simplest choice for constructing the pseudoinverse involves direct inversion to obtain the matrix product $\mathbf{M}^+ = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T$, but results in numerical instability. Instead it is useful to use singular value decomposition to invert the matrix, a good discussion of which can be found in Press et al. [52].

Singular value decomposition is a generalization of the eigenvalue decomposition

for square matrices and states that for an $m \times n$ matrix $\mathbf{M}$, there exists a factorization

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{3.16}$$

where the diagonal entries of $\mathbf{\Sigma}$ are the singular values, $\mathbf{U}$ is an $m \times m$ unitary matrix, and $\mathbf{V}$ is an $n \times n$ unitary matrix. Furthermore, the columns of $\mathbf{U}$ corresponding to non-zero σ_i span the range of $\mathbf{M}$, and the columns of $\mathbf{V}$ corresponding to non-zero singular values span the nullspace of $\mathbf{M}$. This factorization is useful, amongst other reasons because $\mathbf{U}$ and $\mathbf{V}$ do not affect the length of vectors, or the angle between vectors. Any numerical instabilities can be identified in $\mathbf{\Sigma}$. The SVD is unique up to permutations and sign exchanges of the singular values and typically algorithms order the singular values such that $\sigma_1 > \sigma_2 ... > \sigma_{r_{max}}$, where $r_{max} = min(m, n)$. One measure of the instability of $\mathbf{M}$ is the condition number defined by the ratio $\frac{\sigma_1}{\sigma_n}$, where σ_i refers to the i-th largest singular value. The larger the condition number, the more ill conditioned a matrix is, with a singular matrix having an infinite condition number.

Once $\mathbf{M}$ has been factored in this way, it is straightforward to show that the inverse of $\mathbf{M}$ is

$$\mathbf{M}^{-1} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T, \tag{3.17}$$

where the inverse of $\mathbf{\Sigma}$ is computed by taxing the inverse of each nonzero element along the diagonal, leaving the zeros in place.

The numerical ambiguity associated with this procedure comes in determining which of the singular values are so small as to be dominated by roundoff error. Typically, this is accomplished by specifying a cutoff value below which the singular values are simply replaced by zeros. Determining the number of singular values used

in the inversion determines the rank of the pseudoinverse.

The pseudoinverse that comes from using SVD gives the least length solution in the following sense. If we suppose the matrix $\mathbf{M}$ satisfies the equation $\mathbf{M} \cdot \mathbf{x} = \mathbf{b}$, we can see from considering $\mathbf{U}$ and $\mathbf{V}$ that the solution $\mathbf{x}$ will not be unique. We can add any vector in the null space to x without changing the result, so more restricted solution is necessary for uniqueness. In particular, if the solution $\mathbf{x} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T$ corresponds to the least length, minimum $|\mathbf{x}|^2$, solution. In the case that $\mathbf{b}$ is not in the range of $\mathbf{M}$, the singular value decomposition finds the value of $\mathbf{x}$, which minimizes $|\mathbf{M} \cdot \mathbf{x} - \mathbf{b}|$. In this sense, SVD gives the best approximation to $\mathbf{b}$ available.

3.8 Description of Algorithm

With SVD as a tool to construct the pseudoinverse, we are now equipped to outline the actual estimation algorithm. As discussed, the broad idea in using the above machinations is that even if we start with some reasonably bad guess for the state of the system, we can repeatedly improve our estimate by using equation 3.5 to calculate the error of our current guess and stepping in such a direction as to reduce this error. It is assumed that we have some preferred means of obtaining a time delay τ and selecting and appropriate embedding dimension D_M. Additionally, it is assumed that we have some measured time series $y_1(t)$ sampled with a resolution Δt. With these stipulations, the algorithm then proceeds as follows:

1. Guess some values for the state of the model, $x_i^n(0)$

2. Use the dynamical equations, $F_i(x_i^n(t), t) = \dot{x_i^n}(t)$, to integrate forward in time

to obtain a time series $x_i^n(t)$

3. Construct the vectors $S(x_1^n(t))$ and $S(y_1(t))$ by choosing time delayed coordinates from the time series and use these to form $\Delta S_j^n(t) = S_j(y_1(t)) - S_j(x_1^n(t))$

4. Estimate the state space error, Δx_i^n, using
$\Delta x_i^n \approx \left[\left.\frac{\partial S_j}{\partial x_i}\right|_{t_0}\right]^{-1} \cdot [S_j(y_i^*(t_0)) - S_j(x_i^n(t_0))]$

5. Update the estimate according to the rule $x_i^{n+1}(t_0) = x_i^n(t_0) + \Delta x_i^n$

6. Repeat the above procedure replacing $x_i^n(0)$ with the improved guess, $x_i^{n+1}(0)$

In order to get better convergence, it is useful to put some bounds on the search space. This prevents the search from exploring regions far away from the attractor and prevents some forms of numerical instability.

3.9 State Estimation in Lorenz 63

There are subtleties and complications in each of these steps, which will be illustrated in through the examples below. Using the Lorenz 63 system (Lorenz [41]) as a test of the algorithm described above, we perform the following twin experiment. The data system is described by the Lorenz 63 equations,

$$\begin{aligned} \dot{x_1} &= \sigma(x_2 - x_1) \\ \dot{x_2} &= -x_1 x_3 + r x_1 - x_2 \\ \dot{x_3} &= x_1 x_2 - b x_3 \end{aligned} \tag{3.18}$$

where the parameters chosen are $\rho = 28$, $\sigma = 10$, and $\beta = 8/3$. Starting from the initial condition $\mathbf{y}^*(t_0) = [-8.2, -14.3, 15.0]$, the system is integrated forward in time

with a time step $\Delta t = 0.01$, to produce time series for $\mathbf{y}(t)$. From these time series, we select only $\mathbf{S}(y_1(t))$ to use as measurements.

Next, we choose a time-delay and embedding dimension. Ad hoc, we select $\tau = 0.03$ ($K = 3$) and $D_M = 5$, which means that in each step requires at least $T = D_M * \tau = 0.15 = 15\Delta t$ of data.

Proceeding to the first step of the algorithm, we make a guess for $\mathbf{x}^0(t_0)$. Without any measurements of the $i = 2, 3$ components of $\mathbf{y}^*(t_0)$, we choose points randomly from the dynamical range explored by these components of the system. The starting guess for each coordinate of the condition $\mathbf{x}(t_0)$ was drawn from a uniform distribution centered on $\mathbf{y}^*(t_0)$ and spanning the typical range of that coordinate. Here we have used $x_2 \in [-80, 80]$ and $x_3 \in [0, 80]$. In the examples given below, we consider the case of a randomly drawn model position $\mathbf{x}(t_0) = [y_1^*(t_0), -80.4, 96.9]$.

Now, starting from $x_1^0(t_0)$, the model is integrated forward in time to $T = 0.15$. As expected, the model trajectory quickly diverges from the measured trajectory, due to errors in our estimations of the unmeasured components. This divergence between trajectories is shown clearly by the $n = 0$ lines in Figure 3.1.

Once we have integrated the model to obtain the predicted trajectory, we choose D_M points from both the model and data trajectories to form the time-delay vectors $\mathbf{S}(x_1^n(t))$ and $\mathbf{S}(y_1(t))$. The points along the path used to form the time delay vectors are indicated by the large dots in the plot. After integration of the variational equation, the pseudoinverse was constructed via singular value decomposition, implemented in python using the scipy.linalg.pinv function. Combining the time-delay vectors and the pseudoinverse, the guess is improved, and subsequent iterations are shown as well.

As can be seen in Figure 3.1, the result of the improving estimate for the initial

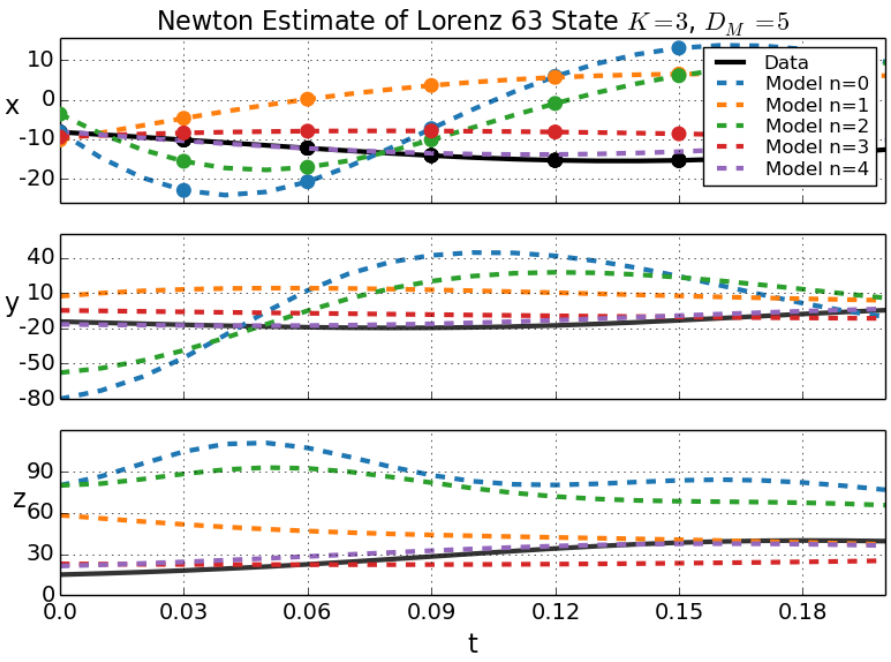

Figure 3.1: Illustration of use of the time delayed Newton's method applied to the Lorenz 63 system for $n = 5$ iteration. In this example, only the x-component of the data system has been measured. The prediction can be seen to improve with each subsequent iteration of the algorithm. The large dots along the trajectories indicate points used as time-delay coordinates.

condition is evident in the prediction generated from that estimate. As the estimate for $x_i(t_0)$ improves with each iteration of the algorithm, the prediction generated by the model more closely aligns with the data. One exception to the trend of improvement is estimation of $x_1(t_0)$, which because it starts at the correct value, can only get worse. Though it moves away from the correct value for a few iterations, it eventually returns to agreement with the measured value. This is shown by the trajectories corresponding to the higher values of n aligning better with the data. After around $n = 8$ iterations, the model converges to $y_i^*(t_0)$ to within machine precision. Once convergence has passed this threshold, the model and data become numerically identical and make indistinguishable predictions.

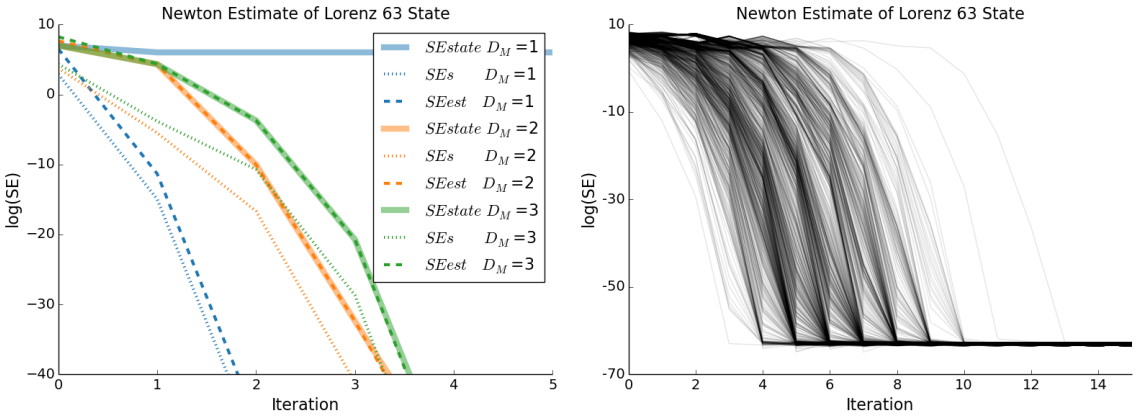

Figure 3.2: Left: Synchronization error in the estimate of $x_i(t_0)$. Note that SE_s and SE_{est} are quantities that are observable. The quantity SE_{state} serves as a reference to illustrate that the shape of the measurable synchronization error approximates that of the true synchronization error. Right: Embedding space synchronization error for 1000 randomly chosen initial conditions. Of the 1000 initial conditions chosen, all were able to converge to the correct value within 15 iterations.

We consider next, the various definitions of synchronization error as laid out in equations 3.1,3.3, and 3.7. There are two questions we can ask of the synchronization error. First, if the model converges to the data, does the synchronization error correctly go to zero. Second, if the model and data fail to converge, does the synchronization error indicate this failure. For the case $D_M = 1$, both the embedding space synchronization error and the estimated synchronization error fail to accurately represent the true state space error. The algorithm still succeeds in the sense that it minimizes the embedding space synchronization error, but the embedding space error is a poor proxy for the true state space error. In higher embeddings dimensions, the embedding space error accurately indicates when the state space error goes to zero. Additionally, the estimated error given by equation 3.7 gives the correct value for the state space error.

3.10 Exploring the Cost Function for Lorenz 96

To better understand how the use of time-delays improves the searching process, it is useful to look directly at the cost function minimized by our algorithm. As explained above, the estimation procedure can be thought of as attempting to minimize the synchronization error in embedding space. To do this, we consider the 5 dimensional Lorenz 96 model (Lorenz [42]),described by the differential equations

$$\frac{dx_a(t)}{dt} = x_{a-1}(t)(x_{a+1}(t) - x_{a-2}(t)) - x_a(t) + f \tag{3.19}$$

with $a = 1, 2, ..., D$, $x_{-1}(t) = x_{D-1}(t), x_1(t) = x_D(t)$, and $x_{D+1}(t) = x_1(t)$; $D = 5$. We choose the forcing parameter to be $f = 8.17$ and generate data by integrating forward with a time step $\Delta t = 0.01$. From this data we select $y_1(t)$ to serve as our measured time series.

For the numerical experiments shown here, the four unobserved initial model conditions were chosen at random from a uniform distribution centered around $\mathbf{y}(t_0)$ and with side length 1 before being integrated forward for $1 \times 10^5 \Delta t$ to ensure a point somewhere on the attractor. The data generated for use below comes from the randomly chosen initial data state

$$\mathbf{y}^*(t_0) = [9.21, -1.20, -0.98, -0.44, 0.42] \tag{3.20}$$

,and the fixed initial data state

$$\mathbf{x}^0(t_0) = [9.21, 1.90, 2.51, 0.70, 0.81]. \tag{3.21}$$

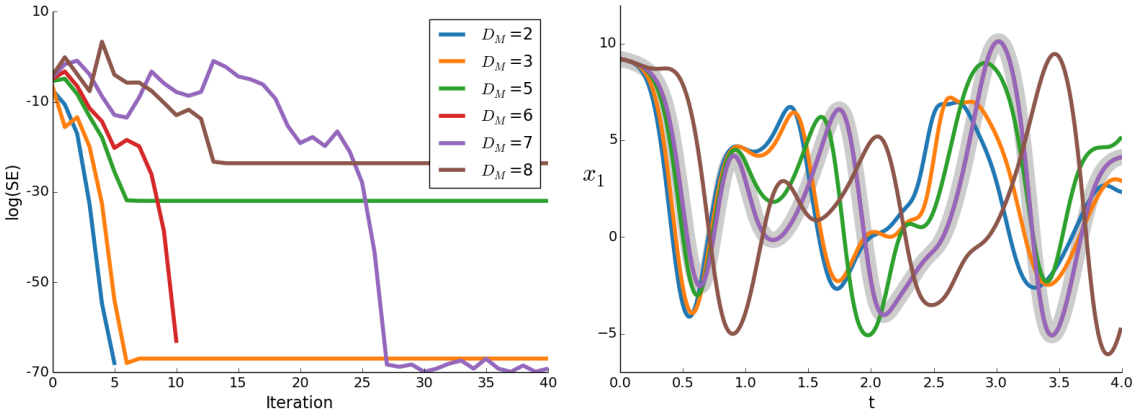

Figure 3.3: Left: Plot of SE_s for select values of D_M. Note that for smaller values of $D_M = 2, 3$, the drop in synchronization error does not result in good prediction. Right: Prediction after 40 iterations using the time-delay Newton method. The broad line indicates the data trajectory $y_1(t)$.

First, we can look how well the algorithm performs under a range of values for D_M and $K = 1$. In the cases of low embedding dimension $D_M = 2, 3$, the synchronization error falls off (indeed the sudden end of the line for $D_M = 2$ indicates that the synchronization errors is numerically indistinguishable from zero), but does not correlate with a good prediction. This is because for small embedding dimension, the algorithm only has access to a short time series of data. Within this small window, the behavior of $y_1(t)$ is not significantly influenced by the initial values of the other components and therefore the synchronization error can be driven to very small number even for incorrect values of $y_i(t_0)$, $i = 2, 3, 4, 5$.

By the time we have reached $D_M = 5$, the embedding space error correctly indicates whether or not the model has found the correct solution. In the cases $D_M = 6, 7$ the extremely small value of synchronization error results in a prediction that is indistinguishable from $y_1(t)$ in the window shown. By contrast, in the cases $D_M = 5, 8$, the synchronization error never drops quite as low and the predictions diverge from the data sooner.

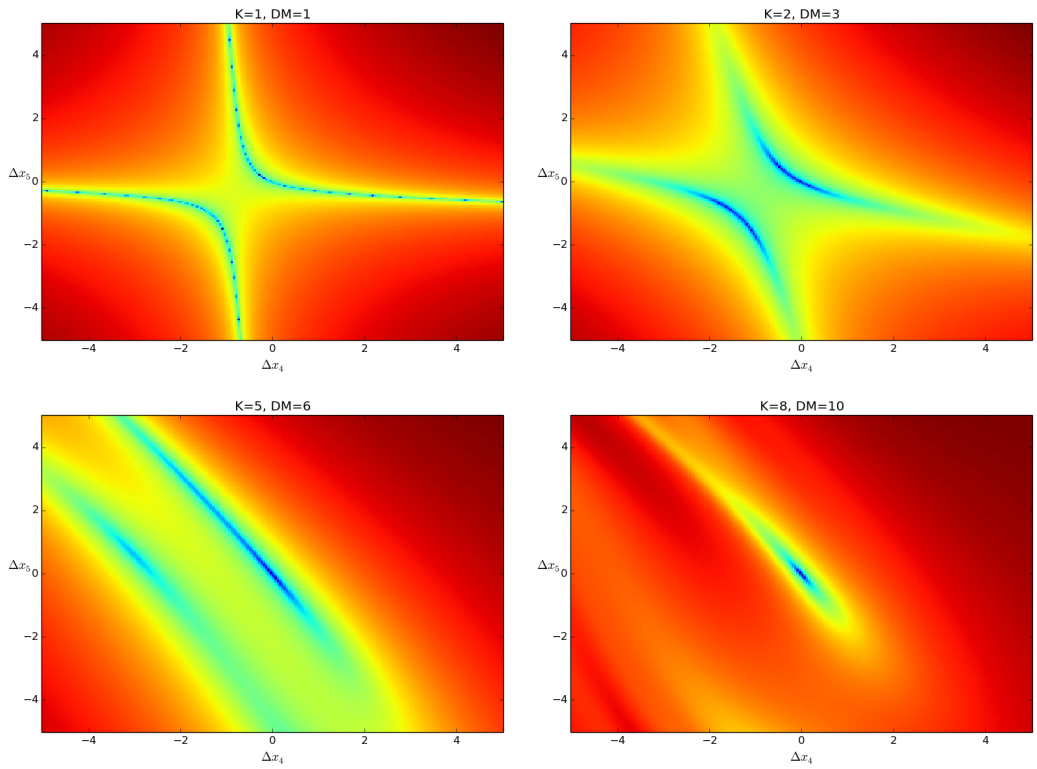

Figure 3.4: Plot of the cost function given in equation 3.3, with x_1,x_1, and x_1 fixed at their correct values. As additional time delays are added, local minima disappear and the surface smooths out.

To explore the surface described by this function, we will look at the cost function for the Lorenz 96 system. Since we cannot easily visualize the full 5-dimensional surface, we will restrict ourselves to a function of two independent variables by fixing the values of the model variables $x_i(t_0) = y_i^*(t_0)$ for $i = 1, 2, 3$.

To get an idea of what the search space looks like, we choose some range of x_4 and x_5-coordinates and evaluate the cost function at every pair of points within the range. To generate the plots shown in Figure 3.4, we discretized the region from $[y_4^* - 4, y_4^* + 4]$ into 200 distinct points and the region from $[y_5^* - 5, y_5^* + 5]$ in to 200 pieces. For each combination $(x_4(t_0), x_5(t_0))$, we integrate the model forward in time to determine the value of $\mathbf{\Delta S}$ and then calculate the value of the cost function as

given in equation 3.3. The logarithm of the value of the cost function is then plotted with blue indicating lower values and red indicating higher values. By design, the global minimum of the cost function is at the origin of the x_4x_5-plane.

The plots in Figure 3.4 show the plot projected into the x_4x_5-plane, and illustrate the disappearance of local minima with increasing embedding dimension. The minimal points, shown as blue areas in Figure 3.4, congregate about the global minimum as the embedding dimension is increased. Additionally, the plots show that the cost function more closely resembles the true the state space separation, defined according to $\frac{1}{2}(\Delta x_4^2 + \Delta x_5^2)$, which has circular contours in the x_4x_5-plane. As the time-delay and embedding dimension are increased, the surface smooths out making it easier to find the global minimum.

The smoothing effect seen in Figure 3.4 can be partly understood as the effect of averaging in calculating the synchronization error in equation 3.3. If the different components of S_j vary independently of one another (the proscription for choosing the time-delay), then the separate ΔS_j terms in the embedding space synchronization error should average one another out.

3.11 State & Parameter Estimation in Rössler's System

Using this same technique, it is also possible to estimate the parameters in a system which, by assumption, cannot be measured directly. Our strategy is to promote the parameters to the status of state variables and apply the same techniques as described above. By definition, parameters do not change in time and therefore

obey the trivial equations of motion $\dot{\mathbf{p}} = 0$. To incorporate this information into our algorithm, the matrix used in the variational equation is now augmented to include the terms

$$R_{ab}(t) = \frac{\partial s_a(x(t))}{\partial p_b(t_0)}, \tag{3.22}$$

where $p_b(0)$ refers to the initial estimate for some model parameter. Using the augmented variational equation, we can see that in a time delay scheme, parameter estimation in determined by

$$p_i^{n+1} = p_i^n + \Delta p_i^n = p_i^n(t_0) + \frac{\partial p_i}{\partial S_j} \Delta S_j^n, \tag{3.23}$$

in precisely the same way as for regular state variables.

Rössler's system (Rossler [53]) serves as a fruitful example to illustrate the application of time delay synchronization to problems of nonlinear state and parameter estimation. The Rössler attractor is described by the equations

$$\frac{d\mathbf{x}}{dt} = \begin{cases} x_1'(t) = -x_2(t) - x_3(t) \\ x_2'(t) = x_1(t) + p_1 x_2(t) + x_4(t) \\ x_3'(t) = p_2 + x_1(t) x_3(t) \\ x_4'(t) = p_3 x_3(t) + p_4 x_4(t) \end{cases}, \tag{3.24}$$

where we have chosen $a = 0.25$, $b = 3$, $c = -0.5$, and $d = 0.05$ as the parameters. It is assumed that $y_1(t)$ is observable, and that all other variables and parameters can not be measured directly. Rössler's system is said to be hyperchaotic, meaning that there are multiple positive Lyapunov exponents (two in this case) associated with the dynamics of the system. Thus, even small imprecisions in estimations for state

and parameters will quickly magnify into erroneous predictions as the system evolves in time. For this system, the time-delayed Newton's method can estimate state and parameters even when the model initial conditions are far from the true values.

For the numerical experiments shown here, the three unobserved initial model conditions were chosen at random from a uniform distribution that spans the region of state space visited by the system where $x_i(t_0) \in [x_i^{min}, x_i^{max}]$. The data generated for use below comes from the randomly chosen initial state $x_i(t_0) = [-20, -18.6, 25.7, 122.4]$ and the fixed initial data state $y_i^*(t_0) = [-20, 0, 0, 15]$. Additionally, the initial estimates for the model parameters were set at half their correct values so that $[p_1^0, p_2^0, p_3^0, p_4^0] = [0.125, 1.5, -0.25, 0.025]$.

In the embedding process, the time delay between observations was $\tau = 0.05 = 5\Delta t$ and the embedding dimension fixed at $D_M = 20$. The state and parameter estimation algorithm thus used 20 data points from a time series of total length $T = 100\Delta t$ to perform the estimation. After the state and parameter estimation process , we fix the model parameters at their estimated values, and integrate both data and model forward for another $3 \times 10^4 \Delta t$, beginning the model at the estimated initial condition.

Figure 3.6 shows prediction for the observed component of both model and data systems. During the first $2 \times 10^4 \Delta t$ of prediction, the discrepancies between data and model cannot be discerned visually.

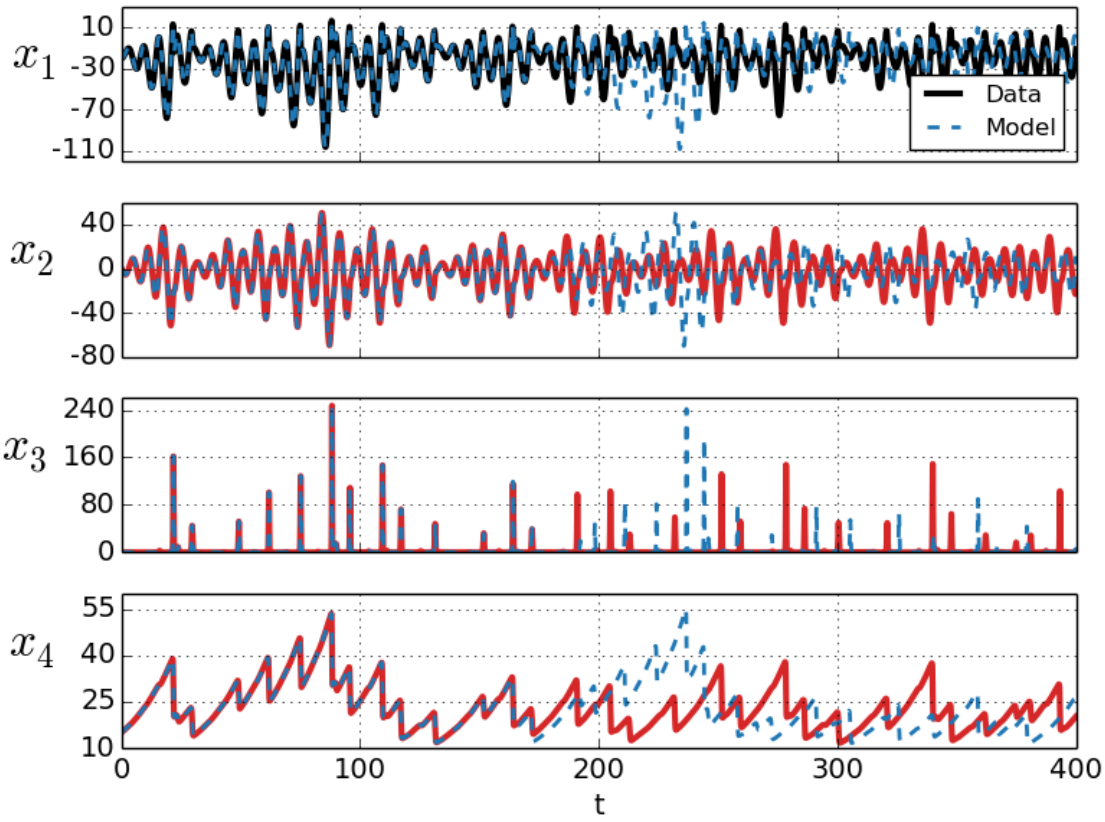

Figure 3.5: After synchronization of the model component $x_1(t)$ with the values of the equivalent data components, the model is used to predict. Discrepancies between the predictions of the model and data system take more than $2 \times 10^4 \Delta t$ to become apparent.

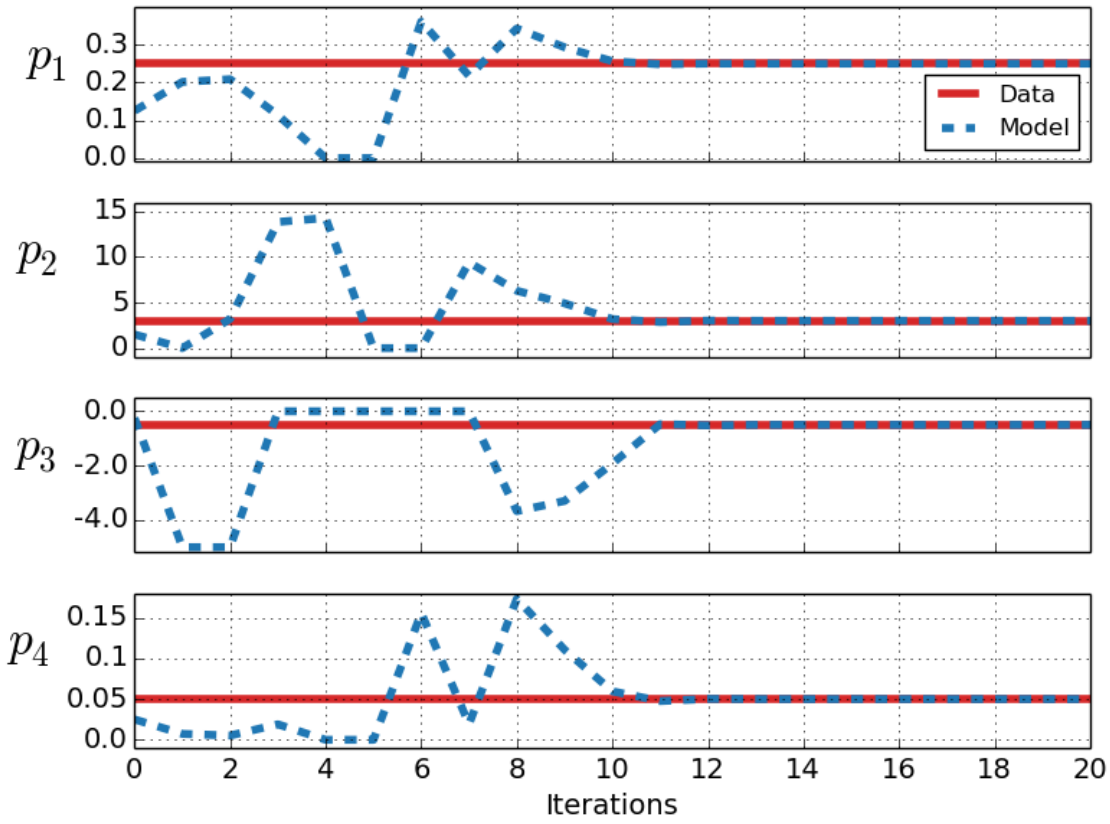

Figure 3.6: Estimated parameters for the 4D Rössler system. The correct values appear as red lines and the estimate for that parameter is shown as blue line plotted against the number of iterations.

Chapter 4

Neurobiology Background

The goal of this chapter is to provide sufficient background information to be able to explain the details of the dynamical neuron models to come. We begin by describing the function and gross anatomy of neurons. Next, we explore the processes by which ions flow into and out of neurons and derive equations to describe this behavior. Discussion of ion channels allows us to define a general class of dynamical neuron models which we will includes all neuron models discussed in this and other chapters. Lastly, we provide a broad overview of the connections between neurons and develop basic models to describe the transmission of signals between neurons.

4.1 Gross Neuroanatomy

Neurons are specialized cells that communicate and process information by means of both chemical and electrical signals, transmitted from one neuron to the next by connections called synapses. There is extraordinary diversity among neurons, but for our purposes we can ignore much of the detail of the sophisticated structures that

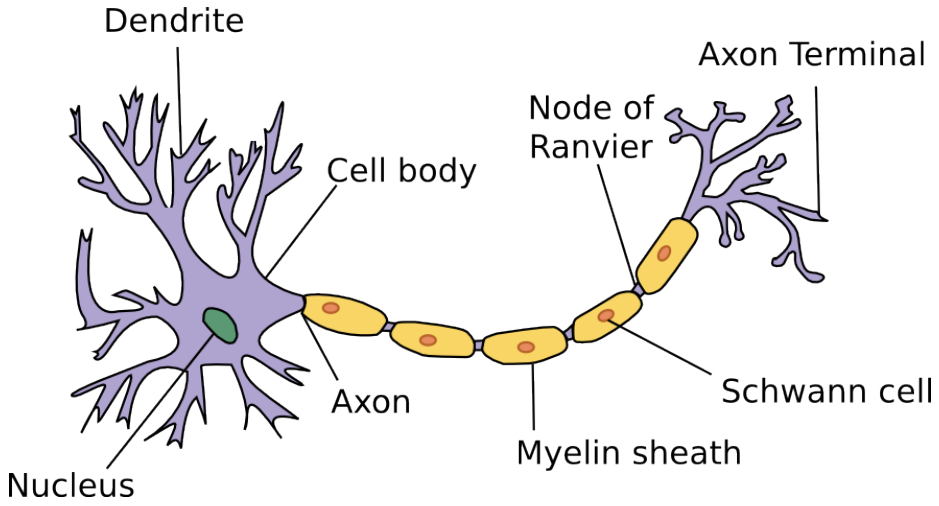

Figure 4.1: Depiction of gross neuroanatomy, illustrating the soma, dendrites, and axon. Reproduced under CC BY-SA 3.0 via Wikimedia Commons from wikipedia.org

makes up real neurons, focusing instead on the functional impact biological findings have on our models. A detailed discussion of many of the features not considered in detail here can be found in Johnston et al. [27] and Hille et al. [24].

Neurons transmit electrical signals in the form of pulses called action potentials. Action potentials move both within and between them and are impacted both by intracellular chemical changes and the morphological properties of the cell. While the morphology of most neurons is complex, from a functional standpoint we will consider the basic components of a neuron to include the dendrites, soma, and axon. The dendrites are long tree-like structures which receive incoming synaptic signals and pass them along to the soma. The soma is typically the bulk of the neuron itself and may engage in some amount of signal processing. Lastly, the axon is a branching structure which may connect in a synapse in order to transmit a signal to another neuron. The models used here are single-compartment neuron models, meaning that all activity within the neuron is treated as though it occurs at a single point.

The nerve cell is separated from the extracellular fluid by a phospholipid bilayer called the cell membrane. Electrical signals are carried within and between neurons by means of the dissociated ions K^+, Na^+, Cl^-, and Ca^{2+}. When these ions flow across the membrane of a neuron, they cause changes in the potential across the membrane. Embedded in the plasma membrane surrounding the cells are proteins which typically select for a single ion species and can move that ion against the concentration gradient. These proteins function as pumps, moving ions in the opposite direction of diffusive forces. The pumps themselves are typically activated in one of two ways. First, they can be voltage gated, meaning that when the potential reaches a certain level the protein undergoes some conformal change, causing ions to pour in or out. Second, the protein can be ligand gated, meaning that it requires a particular signalling molecule, called a ligand, to open it.

The electrical activity within and between neurons arises because of the complicated interplay between protein channels, the work of diffusive forces, and the synaptic inputs from other neurons. Our goal will be to try to disentangle and model the effects of the various contributions to electrical activity, typically using measurements of the voltage alone.

4.2 Ion Flow

In order to build up to models describing the propagation of action potentials, we begin by investigating the flow of ions in mathematical detail. An ion current will either depolarize a cell (drive it toward a higher potential) or hyperpolarize the cell (drive it toward a more negative potential), depending on ion species and direction of current flow. While the basic equation governing ion flow is relatively easy to derive,

in order to obtain analytic results it is necessary to make simplifying assumptions.

4.2.1 Nernst-Plank

There are three large categories of ion motion which contribute to the distribution of ions and thus affect the membrane potential. First, there is thermodynamic diffusion of ions, causing flow down the concentration gradient. Second, there is drift caused by an electric potential gradient (i.e. ion motion caused by an electric field). Lastly, there is the active transport of ions through protein pores.

The flux of particles, represented by the variable J $(mol/(cm^2 \cdot s))$, moving under the influence of drift and diffusion can be described by the equation

$$J = J_{diff} + J_{drift} = -D\nabla[C] - \mu z[C]\nabla V, \tag{4.1}$$

where $[C]$ is the concentration (mol/cm^3), μ is the mobility $(cm^2/(V \cdot s))$and z is the valence of the ion (dimensionless). The constant D, the diffusion coefficient can be related to the mobility through the Einstein relation, $D = \frac{kT}{q}\mu$, where k is Boltzmann's constant, T is the absolute temperature $(^\circ K)$, and q is the charge of the molecule (C). The description given in equation 4.1 is called the Nernst-Planck equation and can be solved analytically for the value of the potential which results in no net current flow. Under these circumstances, it can be found that

$$E_i = V_{in} - V_{out} = \frac{RT}{zF} \ln\left(\frac{[C_{out}]}{[C_{in}]}\right), \tag{4.2}$$

where the number E_i is referred to as the Nernst or reversal potential and indicated the voltage at which a particular species of ion will come to equilibrium. Thought of

another way, if the neuron is held at the potential E_i, the i^{th} ion species will cease to flow and will no longer contribute to the transmembrane current.

4.2.2 GHK Equation

Under simplifying assumptions, there is an analytic solution to the Nernst-Plank equation called the GHK equation, which is necessary for the treatment of calcium. The GHK equation provides a description of ion movement based on 3 assumptions: 1) Ion movement with membrane obeys the Nernst-Planck equation, 2) Ions do not interact with one another, 3) the potential gradient is constant. Solving the Nernst-Planck equation under these assumptions results in the equation derived in Goldman [21],

$$I = PzF\xi \left(\frac{[C]_{in} - [C]_{out} e^{-\xi}}{1 - e^{-\xi}} \right), \tag{4.3}$$

where $\xi = \frac{zVF}{RT}$. Here F is Faraday's constant $(96,480 C/mol)$, and R is the gas constant $(1.98 cal/(^\circ K \cdot mol)$. The value of P is determined by the expression $P = \frac{\beta u^* RT}{lF}$, where β is the dimensionless water membrane partition coefficient for the ion, l is the width of the membrane, and u^* is the molar mobility of the ion within the membrane $(cm^2/(V \cdot s \cdot mol))$. Note that the GHK equation gives the membrane current as a nonlinear function of the voltage.

For the models considered, there are two reasons why calcium must be treated differently from other ions. First, differences between calcium concentrations inside and outside the cell are typically much greater than for other ion species. Typical interior concentrations for calcium range from 5-100 nM (Kostuk [32]), while exterior concentrations are commonly in the mM range. The low interior concentrations of

calcium mean that there is no real reversal potential, as there are insufficient calcium ions to carry a substantive outward current. This in turn implies that the caclium current is better represented by the GHK equation than by the Nernst equation.

Second, the timescale of calcium dynamics is typically much longer than that of the other ion species. This can lead to chaos as shown in Ye et al. [68], and other complications.

4.2.3 Numerical stability of GHK

The GHK equation 4.3 discussed above presents numerical stability problems in the limit $V \to 0$. While the exact expression remains finite in this limit, numerically it is necessary to expand using L'Hopital's rule to obtain an accurate value near this limit. The evaluation of functions is often the most time-consuming step of the optimization procedure, and therefore it is necessary to truncate the expansion at a small number of terms.

The particular expansion used here is

$$J_{Ca} = V_T \frac{g_{out} - g_{in} exp(V/V_T)}{\sum_{k=0}^{k>25} \frac{(V/V_T)^k}{(k+1)!}} \tag{4.4}$$

$$= V_T \frac{g_{out} - g_{in} exp(V/V_T)}{1 + \frac{V}{2V_T}\left(1 + \frac{V}{3V_T}\left(1 + \frac{V}{4V_T}\left(1 + ...\left(1 + \frac{V}{25V_T}\right)\right)\right)\right)}, \tag{4.5}$$

and is taken from Meliza et al. [44].

4.2.4 Ohm's Law

When the ratio $\frac{C_{out}}{C_{in}}$ is approximately unity, the Nernst equation can be used to approximate the current flux using Ohm's law,

$$I_s = g_s \cdot X(V) \cdot (E_s - V), \tag{4.6}$$

where E_s is the reversal potential given in 4.2. Here g_s is the maximal conductance with units of siemens (S). The unitless function $X(V)$ gives the probability that the channels associated with the current are open and thus ranges between zero and one. If $X(V) = 1$, equation describes a linear membrane. A neuron whose membrane conductance (represented by the product $g_s \cdot X(V)$) changes with voltage is said to experience rectification. For linear and non-linear membranes alike the net driving force is then $(E_s - V)$. The validity of this equation, a linearization of the current-voltage relations is empirical and requires testing in each new application (Hille et al. [24]).

The Nernst and GHK equations accounted only for drift and diffusion. To include active transport phenomena in our model, we introduced the function $X(V)$, which models the pumping behavior due to proteins embedded in the plasma membrane actively move ions from one side to the other. The motion of ions through active transport is a more complex, often nonlinear, phenomena to be discussed in greater detail.

4.2.5 Energy Barrier Model

Recall that the opening and closing of the proteins is caused by conformal changes in the protein which allow specific ions to flow into or out of the cell. Whether caused by voltage or the presence of a ligand, these changes occur stochastically on the level of individual channels. However, when all channels are considered as an ensemble, the averaged behavior can be described smoothly. In order to derive an

approximate expression for this opening and closing, we will consider an energy barrier model in which undergoing a conformal change corresponds to crossing a single energy barrier. The energy barrier model treats rate coefficients in chemical reactions as thermodynamic quantities. In order for the reaction to occur, the reactants must overcome some energy barrier (e.g. by moving from one side of a membrane to the other).

The law of mass action states that the flux of a chemical reactant is proportional to the concentration of the reactant, with constant of proportionality called the rate coefficient k_i. For an ion flux across a single barrier, we can write

$$J_{in} = k_1 \beta [C]_{out} \tag{4.7}$$

$$J_{out} = k_2 \beta [C]_{in}, \tag{4.8}$$

where β is called the partition coefficient of water-membrane for the ion. If we look at conditions of thermodynamic equilibrium, then we can write the rate coefficients in terms of the free energy of activation and Boltzmann's constant as

$$k_1 = A e^{-\Delta G_0 / RT} = k_0 \tag{4.9}$$

$$k_2 = A e^{-\Delta G_0 / RT} = k_0, \tag{4.10}$$

where ΔG_0 is the free energy of activation.

In this scenario, the rates are symmetric, but if an external electric field is applied, the barrier is shifted by a factor $\delta z F V$. δ gives the fractional influence of

V on ΔG_0, meaning that it can go from an extreme of $\delta = 1$ where the barrier has shifted to the outside margin of the membrane to a minimum of $\delta = 0$ on the inside margin. In general any amount of shift δ will result in an asymmetric situation where

$$k_1(V) = Ae^{-(\Delta G_0 + (1-\delta)zFV)/RT} = k_0 e^{(1-\delta)zFV)/RT} \tag{4.11}$$

$$k_2(V) = Ae^{-(\Delta G_0 - \delta zFV)/RT} = k_0 e^{\delta zFV/RT}. \tag{4.12}$$

Under these circumstances we can write the current flow as

$$I = zF(J_{out} - J_{in}) = zF\beta k_0 \left[[C]_{in} e^{\delta zFV/RT} - [C]_{out} e^{-(1-\delta)zFV)/RT}\right] \tag{4.13}$$

With this model, the I-V relations depend on the ratio of concentrations and whether they are varying. We can then define $\alpha(V) = Ae^{-(\Delta G_0 - \delta zFV)/RT}$ to be the opening rate of channels and $\beta(V) = Ae^{-(\Delta G_0 + (1-\delta)zFV)/RT}$ to be the closing rate of channels.

4.2.6 Gates

Many experimentally observed currents have nonlinear membrane properties and so it is necessary to develop a model to deal with them. While there exist multiple options, we will focus on the gate model proposed by Hodgkin and Huxley (Hodgkin and Huxley [25]). This model has the advantage of allowing for both voltage dependent and time dependent nonlinearities.

The ion channels which create passive membrane currents, or *leak currents*,

are typically not ion specific and can be well represented by 4.6 by setting $X(V) = 1$. To represent active channels, we turn to the gate model to describe the opening and closing of these active channels.

The gating model proposes that ions can flow through aqueous pores which can be blocked by charges or gating particles. Only when the gate is open can there be a net ion flow either into or out of the cell. When the gate is open, the ion will flow down the concentration gradient. The behavior of the gating particle is described by the single energy barrier model, where $\beta(V,t)$ is the rate of closing and $\alpha(V,t)$ is the rate of opening. It is also assumed that the reaction between open and closed states is first-order.

The time dependence can then be determined from the reaction equation. The rate of change in the probability that the particle is in an open state y is proportional to the number of particles in a closed state times the opening rate, minus the number of particles in an open state times the closing rate. Mathematically, we have

$$\frac{dy}{dt} = \alpha(V,t)(1-y) - \beta(V,t)y \tag{4.14}$$

This equation can be generalized to the case where P independent particles are are involved in gating a channel. In this case, because the particles are independent the probability of the channel opening is simply the product of the individual probabilities or

$$Y(t) = [y(t)]^P. \tag{4.15}$$

4.3 Parallel Conductance Model

Using Kirchoff's Laws and an equivalent circuit representation of a neuron, one can derive a differential equation for the change in membrane potential. To do this, we first recognize that the total current flowing across a membrane is due to two sources. First, there are the ion channels which can cause a current to flow into or out of the cell. Second, from treating the cell membrane as a capacitor we get a capacitive current due to charges building up on either side of the cell walls.

In what is known as the parallel conductance model, we treat the different ion channels as resistors which add in parallel. This makes biological sense, as the various ion channels which stud the cell wall are effectively all wired to one potential outside the cell (whatever the potential of the extracellular medium is), and another within the cell, which defines parallel connectivity. From looking at the ionic currents due to these parallel sources, we conclude that the total ionic current is simply the sum of the currents due to the individual species,

$$I_{tot} = \sum_s I_s, \tag{4.16}$$

where I_s is the current to one species, and in general can be a nonlinear function of both the potential and time. Using this pedagogy, the problem of devising a neuron model is thus reduced to selecting the appropriate currents and adding them together.

The change in potential across the cell wall can then be written as a function of time by using the differential equation

$$C_m \frac{dV}{dt} = \sum_s I_s(V,t), \tag{4.17}$$

where C_m is the capacitance of the cell.

4.3.1 Hodgkin-Huxley Model

The Hodgkin-Huxley model will appear in numerous examples and is worth exploring in some detail. The model is a parallel conductance single-compartment model for the membrane potential of a squid giant axon and consists of three currents: potassium, sodium, and leak. Because the ratio of interior to exterior concentration for potassium and sodium is near unity, we can use equation 4.6 to represent their contribution to the change in membrane potential.

The leak current represents passive channels and therefore has no gating variable. Furthermore, it is not treated as ion-specific and tends to restore the membrane to the resting potential. The complete model is given by the equations

$$C_m \frac{dV}{dt} = g_{Na} m^3 h (V - E_{Na}) + g_K n^4 (V - E_K) + g_L (V - E_L), \tag{4.18}$$

where the gating variables have dynamics of the form

$$\frac{dx}{dt} = \frac{x_\infty(V) - x}{\tau_x} \tag{4.19}$$

Using the language developed thus far, we can see that the leak current is a passive or linear current. The potassium current is nonlinear and depends on the simultaneous meeting of four molecules of n. Likewise, the sodium current depends on the meeting of three molecules of m with one of h. The model used by Hodgkin and Huxley will stand in as our go-to neuron model in many cases.

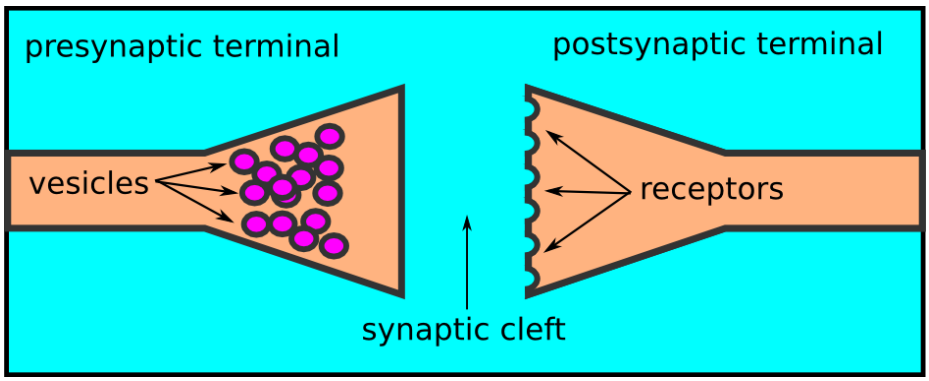

Figure 4.2: Illustration of the parts of a synapse including the presynaptic terminal, synaptic cleft, and postsynaptic terminal

4.4 Synaptic Transmission

Synapses are responsible for the transmission of signals between neurons. Here, we discuss the biochemical processes underlying the transmission of an action potential between synaptically connected neurons.

In the absence of a synapse, an electrical signal traveling between neurons would attenuate by around four orders of magnitude (Johnston et al. [27]). Therefore, to relay electrical signals efficiently between neurons a direct connection, or synapse, is required. Broadly speaking, there are two types of synapses: gap junctions and chemical synapses.

Gap junctions are fixed conductance connections which directly transmit an electrical signal. Gap junctions typically serve specialized purposes (e.g. high speed processing, glial cells, synchronization, retina). For our purposes they are a mathematical convenience and are not intended to reflect the underlying anatomy. Chemical synapses are variable conductance, or 'plastic' connections which transmits action potentials indirectly through chemical relays. A change in the potential of the synaptically neuron releases compounds known as neurotransmitters which induce a change

in the postsynaptic potential. For the discussion below, the word synapse refers to a chemical synapse.

Anatomically and functionally, a synapse can be thought of as divided into 3 parts as shown in Figure 4.2. The synapse consists of a presynaptic terminal, a postsynaptic terminal, and the synaptic cleft separating the the two. The presynaptic terminal contains a reservoir of vesicles can release neurotransmitter for which the postsynaptic neuron has receptors. The incoming electrical signal is translated into a chemical response and back very quickly. In many neurons the transmission process takes $\sim 1\ ms$ but can be as low as 100 μs in insects. The process consists of seven parts:

1. An Na^+ or K^+ generated action potential enters the presynaptic terminal.

2. Depolarization activates voltage gated Ca^{2+} channels through largely unknown processes. Because the extracellular concentration of calcium is much higher, when the channels open calcium ions flood in due to the steep concentration gradient. This process occurs in around 800 μs after the arrival of the action potential.

3. Unknown processes causes neurotransmitter vesicles to fuse to the cell membrane in a process called exocytosis. The fusion of vesicles induces the release of neurotransmitter into the synaptic cleft. It takes no more than 200 μs from the introduction of increased levels of calcium to trigger this release. The amount of neurotransmitter released by the binding of a single vesicle is called a quanta and is the basis for the so called quantum hypothesis.

4. Neurotransmitters rapidly diffuse across the synaptic cleft in $1 - 2\mu s$. Since this

portion of the transmission process accounts for only around .1% of the total time for synaptic transmission, it is usually treated as instantaneous in models.

5. The neurotransmitters bind to specific receptors in postsynaptic membrane. How this process occurs depends on the details of the receptors and will be discussed in greater detail in the neurotransmitters section.

6. Ligand gated ion channels in the postsynaptic neuron rapidly open leading to a change in the potential of the postsynaptic neuron. For some neurons this happens by way of a more complicated mechanism in which neurotransmitter binding to postsynaptic receptors releases second-messenger molecules. The second-messengers then modulate the ion channels. As one might expect, the second-messenger pathway is typically a slower transmission mechanism.

7. Vesicles pinch off from the membrane to be recycled in a process called endocytosis.

Before going into detail about the processes underlying synaptic transmission, it is worth pausing to consider the experimental basis for the theory of synapses. Experimentally, much of what is known comes from the neuromuscular junction and squid giant synapse. Therefore it is likely that the precise details will be different for synapses in the avian brain. Below we provide a short summary of important experimental results used in forming a theory for synapses.

The presence of the action potential in the presynaptic terminal is critical for synaptic transmission, though whether it is due to Na^+ and K^+ or direct stimulation is unimportant. Many experimental results underlie the importance of calcium in synaptic processes. It has been shown that the release rate of neurotransmitter can be

adjusted experimentally through baths of Ca^{2+} and Mg^{+}. The details of how calcium causes release are not fully known, but the general role of calcium is very important for synaptic transmission and plasticity.

After calcium interacts with vesicles, neurotransmitter particles are released into the synaptic cleft. Most fast excitatory synapses use glutamate as transmitter, which then binds to N-methyl-D-aspartate (NMDA) receptors in the post-synaptic cell. For our purposes, we will consider excitatory synapses associated with the transmission of α-amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid (AMPA), which mimics the effects of glutamate. Additionally, we will look at the inhibitory neurotransmitter γ-aminobutyric acid (GABA) fall under this category.

4.4.1 Synaptic Models

The quantitative model used to describe the synaptic current looks much like the models used for other ion channels. In particular, the synapse is treated a being gated by a single-particle barrier. While the actual synaptic current is determined by the outward flow and reuptake of neurotranmitter, the phenomenological model shown below can reproduce the qualitative form of the current and provides a simpler model.

Specifically, we model the synaptic current according to

$$I_{syn}(t) = g_{syn} r(t)[V_{post} - E_{rev}], \tag{4.20}$$

where the variable $r(t)$ is given by

$$\frac{dr(t)}{dt} = \alpha T(V)[1 - r(t)] - \beta r(t), \tag{4.21}$$

and

$$T(V) = \frac{T_{max}}{1 + exp(-[V_{pre}(t) - \theta_p]/\sigma_p)}. \tag{4.22}$$

In this model, the presynaptic voltage determines the probability that the gate is open through the function $T(V)$. With the gate open, the flow of synaptic current depends on the difference between the reversal potential of the neurotransmitter and the membrane potential of the post-synaptic cell. Excitatory neurotranmitters are defined as having a reversal potential above the spiking threshold (and thus tend to induce spiking in the postsynaptic cell). Conversely, inhibitory neurotransmitters have a reversal potential below the spiking threshold and thus inhibit postsynaptic spikes.

4.5 Experiments

4.5.1 Electrophysiology

When examining neurons, one typically measures the electrical signals produced by the neurons through electrophysiological experiments. In general, one can record directly from within a single cell (intracellular measurements) or in the space between cells (extracellular measurements). Here, we will need only consider intracellular recordings.

The potential of an individual neuron is recorded using a device called a patch clamp. In its most basic form, the patch clamp consists of a device such as a glass pipette which can be used to physically attach to a neuron, within which are contained a voltmeter, a variable current source, and an amplifier. The clamp can be operated in one of two modes. In current-clamp mode, the current supplied by the clamp is varied according to some experimental protocol and the voltage is recorded. Conversely, in

voltage-clamp mode the voltage is fixed and any voltage difference between the clamp and the cell is passed to the amplifier to produce a current which maintains a constant voltage. The current produced by the clamp is then equal and opposite to the current produced by the cell, allowing for recording the net current passing in our out of the neuron.

4.5.2 Synaptic Measurements

It is substantially more difficult to make electrophysiological recordings from synapses than from individual cells. Recording from a synapse essentially involves the techniques described above applied simultaneously to two synaptically connected neurons. In most experiments, the post-synaptic neuron is voltage clamped at some voltage V_{cl}, so that V_{post} becomes fixed. Voltage clamping the post-synaptic cell accomplishes two things. First, because the voltage of the post-synaptic cell is no longer changing in time, the various ion currents in the post-synaptic reach some equilibrium, and the dynamics of the post-synaptic neuron cease to impact any measurements. Second, the voltage clamp can measure the steady state ion current. In order to maintain a fixed voltage, the clamp must supply a current that exactly counterbalances the net ion current in or out of the cell. The current required to maintain this constant voltage is called the holding current I_H and is measured by the clamp. For example, when the post-synaptic cell is at its resting potential, by definition it requires no external current to maintain this potential and so $I_H = 0$. If the clamp is set to any other voltage value, the holding current is non-zero but does not vary in time.

In the next step of the experiment, with the post-synaptic cell voltage-clamped,

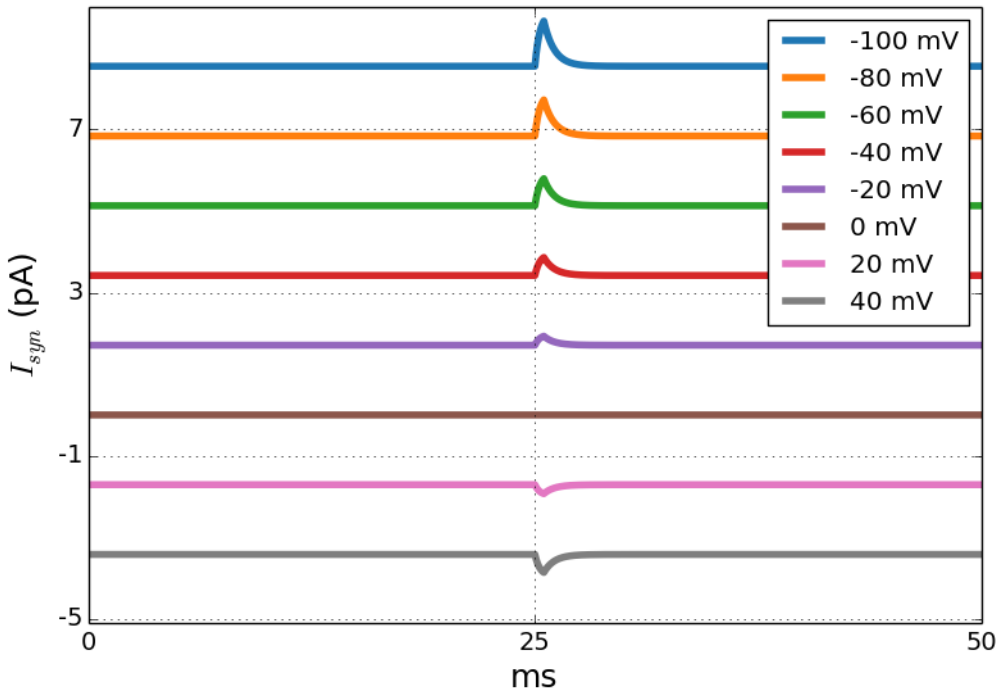

Figure 4.3: The synaptic current given by equation 4.20 for various voltage clamp values. Note that the synaptic current vanishes when the post-synaptic neuron is clamped at $E_{rev} = 0$.

the pre-synaptic cell is stimulated so that a synaptic signal is triggered. When the synapse transmits a signal, an additional current I_{syn} arrives at the voltage-clamped post-synaptic cell. In order to maintain a fixed voltage, the voltage clamp must now supply a total current $I_{cl} = I_H + I_{syn}$. Since the voltage of the post-synaptic cell is fixed at the clamp's value, the magnitude of the holding current will be the same as before the synaptic signal arrived. Thus, after measuring the total clamp current, the holding current subtracted off in order to yield a measurement of the synaptic current alone.

The reversal potential can be determined simply repeating this procedure and stepping through various values of voltage for the post-synaptic cell. When the

post-synaptic cell is voltage clamped at a value equal to the reversal potential, then according to 4.20 the synaptic current should vanish. Thus, by identifying the value at which the synaptic current vanishes, the synaptic reversal potential can be determined. This procedure was carried out numerically using the synaptic model and the results are shown in Figure 4.3.

In order to determine maximal synaptic conductance, g_{syn}, the presynaptic neuron is depolarized with a pulse long and strong enough that the gates will be fully open, so that the factor $r(t)$ in 4.20 goes to one. Now, the maximum value of the synaptic current in time is recorded. Dividing the maximum of the synaptic current by the difference $[V_{post} - E_{rev}]$ then gives the maximal conductance. By repeating this procedure for various values of V_{post}, an I-V plot can be generated and a line fitted through the data to determine the best value for g_{syn}. This portion of the experiment was also replicated numerically and is shown in Figure 4.3.

Chapter 5

HVC Single Neuron Models

5.1 Introduction

This chapter discusses results of using the data assimilation framework presented in earlier chapters to estimate the state and parameters for a dynamic single neuron model. The data used are from voltage traces recorded during electrophysiological experiments with zebra finch HVC neurons. First, we present an overview of the relevant biological and anatomical background regarding birdsong production and the zebra finch. This section is intended to provide motivation and justification for the neuronal models appearing later on. Next, we introduce the models used, taking care to justify the currents which appear. Finally, we consider the results of assimilation on two different models and examine the conclusions that can be drawn from each.

5.2 HVC Background

The songbird is an attractive candidate for the study of learning and memory due to numerous parallels between human and oscine behavior. In particular, the ability of a bird to sing requires learning from other birds and is often seen as a model for human language acquisition. The effects of many stimuli on the learning and production of birdsong have been well studied through experiments using techniques such as playing auditory feedback, lesion studies, and modifying the temperature as reviewed in Knudsen and Gentner [31]. However, while many aspects of the production of birdsong are beginning to be understood, the mechanism by which song information is encoded at the cellular level remains mysterious.

While many results apply to songbirds in general, our focus will be on models of the zebra finch (*Taeniopygia guttata*). The zebra finch is studied owing to the fact that it learns but a single song consisting of highly stereotyped vocal gestures during a distinct learning period. Young birds learn specific songs by listening to adult tutors and forming memories which lead to song development (Margoliash [43]). The song itself consists of a $0.5 - 1.0\,s$ motif repeated a number of times. Each motif in turn consists of syllables (individual bursts of sound) approximately $100\,ms$ in length and occurring in a precise order (Long and Fee [38]).

Production, perception, and learning of birdsong occur through a set of forebrain nuclei, called the *song system*. The song system is a collection of bilaterally coordinated brain structures, organized into both feedforward and feedback pathways. Birdsong has remarkably precise and hierarchically organized structure mediated by a number of distinct, well-studied motor nuclei. Within the song system, the telencephalic

nuclei HVC [1] play the role of coding for syllable order and controlling the temporal structure of the song (Hahnloser et al. [22]). HVC is located within the avian analog of the mammalian premotor cortex, producing instructions through the motor pathway leading to highly stereotyped, precise, learned vocalization. It has been shown that there are three distinct neuronal populations with HVC, which can be categorized morphologically, electrophysiologically, or functionally. The first type, HVC_{RA} neurons project into an area known as the robust nucleus of arcopallium. The second type, X projecting neurons, project from HVC to area X in the avian basal ganglia. Lastly, cells called interneurons connect neurons within HVC to other HVC neurons and provide a nearly constant background of inhibitory stimuli. Only data from neurons projecting to area X (HVC_X) will be considered in detail here.

In addition to providing a trove of single-cell voltage data, HVC provides an excellent theatre in which to investigate synaptic changes associated with learning and memory. It is known that songbirds learn to sing via audition-dependent vocal plasticity (Mooney and Prather [45]). HVC has projection neurons that give rise to pathways specialized for song patterning, or necessary for audition-dependent vocal plasticity. This underlines the importance of HVC in singing and song learning and makes it a promising place to probe for synaptic interactions.

The details of how HVC neurons encode song information are poorly understood, owing in part to a lack of understanding of the properties of individual neurons. HVC neurons can be investigated through electrophysiological experiments, of which we will focus on *in vitro* intracellular recordings. Due to this lack of understanding, HVC presents an opportunity to see if data assimilation can assist neurobiology in

[1] once High Vocal Cortex, but now a proper name

characterizing neurons.

5.3 Assimilation Procedure

Here we present a summary of results collected from HVC_X data that has been available and analyzed using variational optimization on patch clamp data. The available data consisted of intracellular patch clamp recordings generated from three X-projecting neurons extracted from HVC in the zebra finch. In each case, the data was sampled at 50 kHz and various currents were injected according to established current injection protocols designed to explore the full dynamic range of the model (Abarbanel et al. [4]; Kostuk [32]). The injected currents typically consist of a combination of step currents and currents based on chaotic waveforms. The three neurons will be referenced hereafter as N1, N2, and N3 and there is no particular significance to the ordering of these names. For each neuron the data were divided into epochs consisting of several milliseconds of recorded data each. For N1 a total of 213 epochs were available, for N2 174, and for N3 209.

The analysis detailed below was performed by running IPOPT on 30,000 data points from each epoch in the collections for each neuron. Because the beginning of each epoch consisted of a period during which no current was injected, the first 10,000 data points (200 ms) of each epoch were discarded before the optimization procedure was performed.

After the optimization was completed, the model was integrated forward for an additional 50,000 time points in order to check the validity of the estimates. As many apparently good estimations resulted in poor predictions, the prediction process provides an important way to test the validity of the state and parameter estimations

that result from the optimization procedure.

5.4 A4 Model

The A4 model is a single compartment model and consists of 10 different ion currents and incorporates calcium dynamics and is based on the model presented in Daou et al. [11] and draws from several others (Destexhe et al. [12]; Dunmyre et al. [13]; Hodgkin and Huxley [26]; Terman et al. [62]; Wang et al. [66]). The state of the model is given by the differential equation,

$$C_m \frac{dV}{dt} = I_{NaT} + I_{NaP} + I_K + I_{CaL} + I_{CaT} + I_{SK} + I_h + I_A + I_L + \frac{I_{inj}}{I_{SA}}. \tag{5.1}$$

Additionally, because a current modulated by calcium (I_{SK}) is included in the model, it is necessary to explicitly model the calcium dynamics. The form used here is a very rough approximation to the complicated processes underlying the change in internal calcium concentration. The calcium dynamics are given by the equation

$$\frac{d[Ca^{2+}]}{dt} = -f\left[\epsilon\left(I_{CaL} + I_{CaT}\right) + k_{Ca}\left([Ca^{2+}]_i - b_{Ca}\right)\right], \tag{5.2}$$

where f is the fraction of free-to-total cytosolic $[Ca^{2+}]$ (unitless), ϵ combines effects of buffers,cell volume, and molar charge (units of $mM/pA \cdot ms$), k_{Ca} is the pump rate (units of ms^{-1}) and b_{Ca} is the basal level of$[Ca^{2+}]$(units of mM). Conceptually, the first term in parentheses in equation (5.2) tells us that decreases in the calcium concentration can be attributed to the two calcium currents. The second term represents an attempt by the calcium pumping mechanism to return the concentration

to the basal level.

The dynamics of each of the ion currents are specified in the table below. Note that sodium dynamics given in Daou et al. [11] are not included, nor is a sodium dependent potassium current. The sodium dynamics and related potassium current were found to have little impact on the model from previous studies and were therefore excluded from this model.

I_{NaT}

There are two functionally distinct sodium currents, differing in their inactivation properties. The first, I_{NaT} is essentially the sodium current from the Hodgkin-Huxley model. Looking at the gating variables for I_{NaT}, we can see that the overlap of the m_∞ and h_∞ curves signifies voltage values for which the sodium current responds strongly. For this reason, this form of the current is called a *window current*, due to the response in the window of overlap. Since the reversal potential for sodium is $\approx 50\,mV$, the sodium currents are depolarizing.

I_{NaP}

I_{NaP} is a persitent, or non-inactivating, sodium current that plays a role in repetitive firing and subthreshold behavior. Biophysically, I_{NaP} may represent a distinct ion channel or a different mode of the same I_{NaT} channel. The functional form of the current comes from Dunmyre et al. [13]. The non-inactivating behavior is included in the dynamical equation for I_{NaP} through an instantaneous activation variable and a slow inactivation variable. The depolarization block and plateau potential that some HVC neurons exhibit in response to positive current pulses

(Kubota and Saito [33]) is understood to indicate some long-lasting inward current and so I_{NaP} is included here.

I_K

The typical I_K, such as appears in the Hodgkin-Huxley model is called the delayed-rectifier current and has relatively slow activation and may show voltage-dependent inactivation. Because the reversal potential ($\approx 90\,mV$) for potassium is well below the threshold voltage, this current acts to reset the potential after a spike.

I_A

I_A is the transient potassium current. Like the I_K current, I_A participates in spike repolarization and contributes to resting potential. Because it activates rapidly and then inactivates, it prevents neurons from responding to fast depolarizations, and contributes to a delay in spiking response. As there is some experimental evidence of delay (Kubota and Taniguchi [34]; Mooney and Prather [45]), this current is included to account for it.

I_h

I_h is the hyperpolarization current, with the important properties that it is activated by hyperpolarization and undergoes no inactivation. The hyperpolarization current accounts for the phenomenon of sag as found in Kubota and Taniguchi [34], Kubota and Saito [33], and Dutar et al. [14]. While the effect of the hyperpolarization current is well understood, the purpose of this current is largely mysterious. I_h has unusual kinetics and so the model used for this current is based on the dynamic

description provided in Destexhe et al. [12].

I_{SK}

I_{SK} is a calcium gated potassium current found in Kubota and Saito [33]. It functions in much the same way as a typical ohmic current, but with the calcium concentration playing the role typically reserved for voltage in activating the channel. Unlike the other ion currents, the gating variable for the I_{SK} current is characterized by insensitivity to voltage. Activation of I_{SK} limits the firing frequency of action potentials and can be used to produce spike frequency adaptation (Daou et al. [11]). Additionally, SK channels are thought to be involved in synaptic plasticity and therefore play important roles in learning and memory (Stackman et al. [59]).

I_{CaL}

Due to the drastic difference between intracellular and extracellular calcium concentrations, calcium currents cannot be well represented by the usual Nernst equation, and require the full GHK form. There are 4 pharmacologically distinct $[Ca^{2+}]$ currents L, T, N, and P of which L and T are of interest here.

I_{CaL} is a high-threshold long-lasting (hence L) and was historically the main $[Ca^{2+}]$ current measured. As the name implies, this current is activated at high potentials and deactivation depends on $[Ca]_{in}$ but not on voltage. L-type calcium currents have been shown to exist in HVC neurons (Kubota and Saito [33]; Long et al. [39]).

I_{CaT}

I_{CaT} is a low-threshold transient current, suggested by Llinas and Yarom [37] and Llinas and Yarom [36]. It exhibits strong voltage-dependence at depolarized potentials and contributes to spontaneous burst firing and subthreshold activity (Terman et al. [62]). In HVC neurons, I_{CaT} is thought to be responsible for rebound firing after a period of inhibition (Daou et al. [11]).

I_L

The leak current used here is essentially the same as what appears in the Hodgkin-Huxley model. The channel for the leak current is passive, and hence the equation for the leak current has no gating variables, and is intended as a kind of catch-all for ions that leak out of the cell. Because of its low reversal potential ($E_L \approx -70\,mV$), I_L is a hyperpolarizing current important for spike repolarization.

5.4.1 Units

Currents are measured in picoAmps and voltages in millivolts. The kinetics for the gating variables in the A4 model taken on several different possible forms and are given in the table below. The parameter θ_x is the half-activation voltage for the gating variable x and σ_x determines the slope of the gating variable x.

5.4.2 A4 Model Optimization Estimating Conductances

The table below summarizes the results of the attempted optimization procedure for the A4 model in which only the state variables and values of the maximal

Table 5.1: A4 model ion currents including activating and inactivating gating variables associated with each

Current	Dynamics	Activating	Inactivating
I_{NaT}	$g_{NaT} m_\infty^3 h (E_{Na} - V)$	m	h
I_{NaP}	$g_{NaP}(mp_\infty)(hp)(E_{Na} - V)$	mp	hp
I_{CaL}	$g_{CaL} V s_\infty^2 \left(\frac{Ca_{ex}}{1 - e^{\frac{2FV}{RT}}}\right)$	s_∞	–
I_{CaT}	$g_{CaT} V [a_{T_\infty}]^3 [b_{T_\infty}]^3 r_T \left(\frac{Ca_{ex}}{1 - e^{\frac{2FV}{RT}}}\right)$	$a_{T\infty}$	$b_{T\infty}$
I_{SK}	$g_{SK} \left(\frac{[Ca^{2+}]^2}{[Ca^{2+}]^2 + k_s^2}\right) (E_K - V)$	–	–
I_L	$g_L (E_L - V)$	–	–
I_H	$g_H \left(k_r r_f + (1.0 - k_r) r_s\right) (E_H - V)$	–	–
I_A	$g_A a_\infty e (V - V_K)$	a_∞	e

Table 5.2: Dynamical equations as functions of voltage for A4 model gating variables

Gating Variable	Kinetics
$m_\infty, mp_\infty, a_{T_\infty}, s_\infty, n, e, hp, r_T$	$x_\infty(V) = \frac{1}{1 + e^{\frac{V - \theta_x}{\sigma_x}}}$
n, h, e, hp	$\frac{dx}{dt} = \frac{x_\infty(V) - x}{\tau_x}$
h	$h(V) = \frac{0.128 e^{\left(\frac{V+15}{-18}\right)}}{0.128 e^{\left(\frac{V+15}{-18}\right)} + \frac{4}{1 + e^{\left(\frac{V+27}{-5}\right)}}}$
n, hp	$\tau_x(V) = \frac{\tau_X}{\cosh \frac{V - \theta_x}{2\sigma_x}}$
$a_{T_\infty}, r_{T_\infty}$	$x_{T_\infty} = \frac{1}{1 + e^{\frac{V - \theta_{x_T}}{\sigma a_T}}}$
b_{T_∞}	$b_{T_\infty} = \frac{1}{1 + e^{\frac{r_T - \theta_b}{\sigma_n}}} - \frac{1}{1 + e^{\frac{-\theta_b}{\sigma_n}}}$

Table 5.3: A4 model optimization results including total size of available data set and how many runs resulted in good results

Folder	Recordings	Completed	Good Pred	Points Est	Points Pred
N1	213	177	57	30000	50000
N2	174	172	48	30000	50000
N3	209	151	67	30000	50000

conductances are estimated. The table follows the same format as used to present the results of the B4 model.

Compared to the B4 conductance only estimates, the A4 model can produce remarkably good predictions from estimating conductances alone. To produce the predictions seen below, 30000 data points were used in the assimilation step. As before, because the voltage trace recordings typically begin with a period during which no current is injected, the first 10,000 data points from each recording were discarded and were not used during the optimization procedure. All kinetic parameters were fixed at the values given in the MATLAB file specifying the A4 model.

Unsurprisingly, because the conductance only estimation process is a linear optimization problem, and therefore substantially more straightforward, many more of the IPOPT runs completed for the the conductance only A4 model. A total of 500 runs completed, compared to 288 for the B4 model in which all parameters were estimated.

Furthermore, compared to the case of estimating all kinetic parameters in the B4 model in which a total of 125 runs produced good predictions, for the A4 conductance only estimation 172 runs produced good predictions. This suggests that when good fixed values for kinetic parameters are chosen, varying maximal conductances alone is sufficient to find good estimation and prediction. Four examples of good predictions are shown in Figure 5.1, and compare well with the predictions

generated from estimating all parameters in the B4 model.

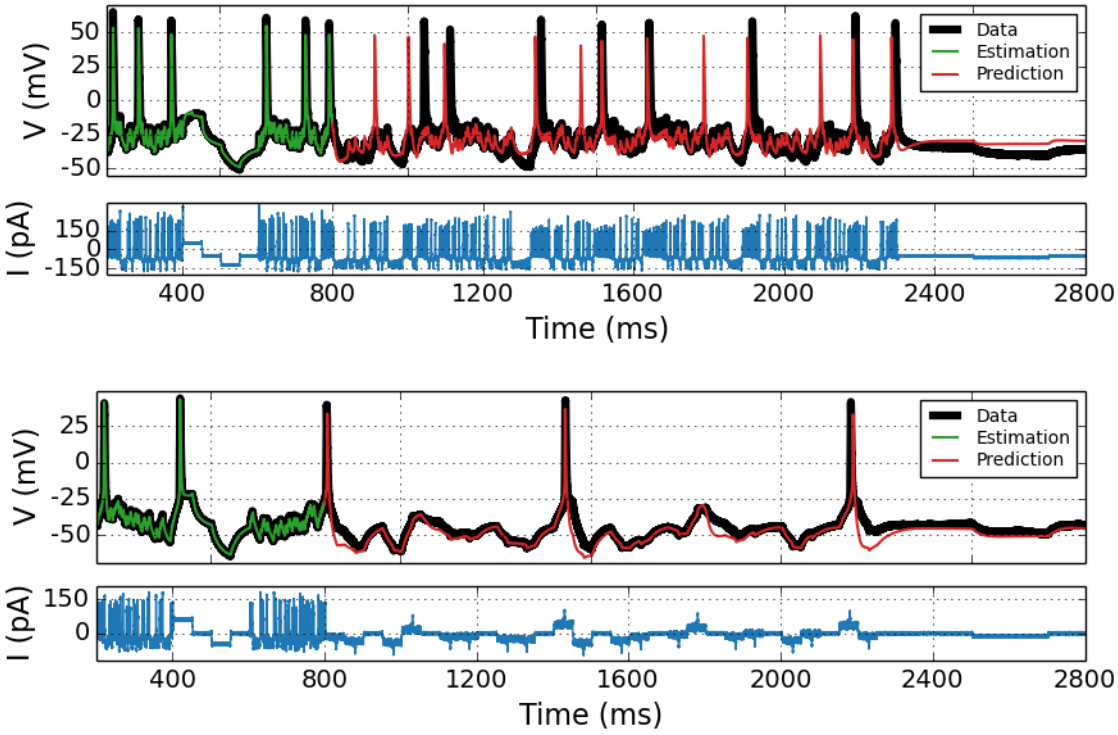

Figure 5.1: Clockwise from top left: N1aj1_23 and N2aj1_29. Prediction of the A4 model after using 30000 data points to estimate all of the parameters and state variables of the model.

As in the case of the B4 model in which all states and parameters were estimated, an analysis of the maximal conductance distribution for the A4 model was performed. Normalized histograms giving the maximal conductances for values estimated for each epoch are shown below. The histograms were created by binning the data from the 172 runs which produced good predictions into 15 bins representing equally spaced regions between the upper and lower bounds of each parameter.

The distribution of the maximal conductances shows several promising trends. The first important feature to note is that virtually none of the estimations hit the lower bound of the estimation range. This trend is further emphasized when looking at

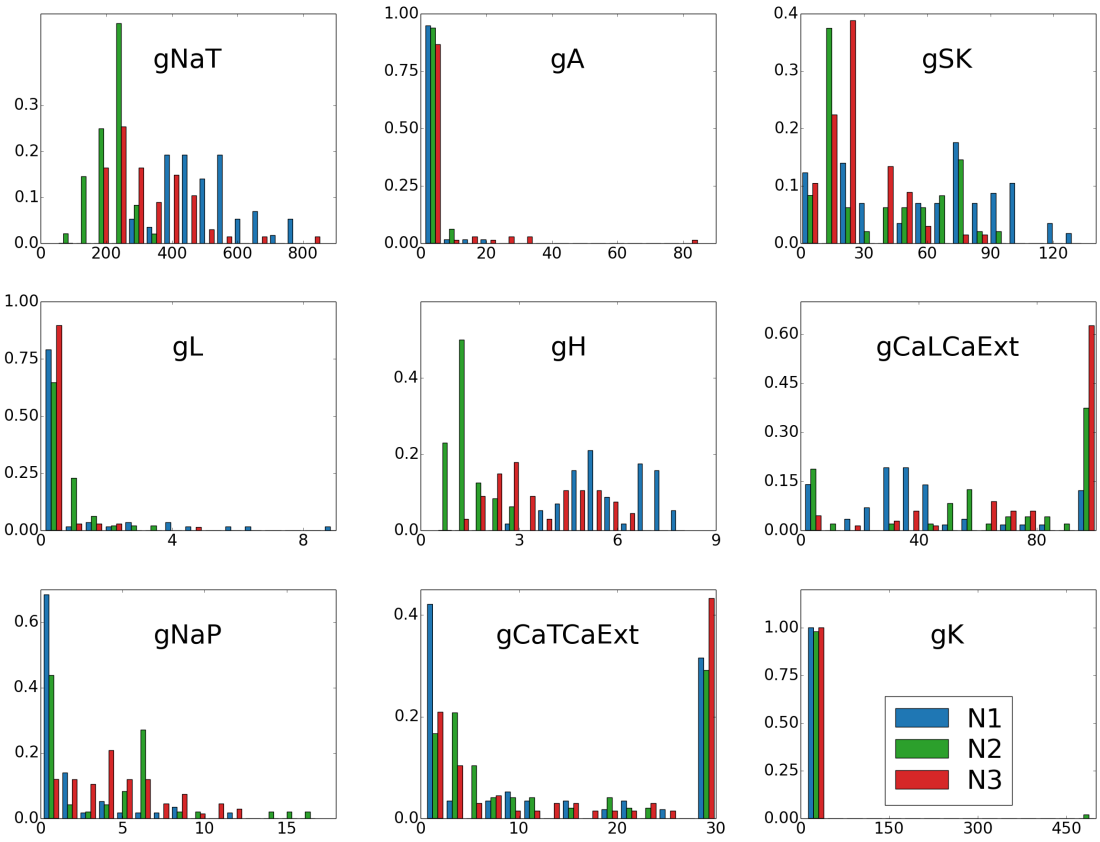

Figure 5.2: Histograms of maximal conductance estimates for data sets from three different neurons

histograms that have been created with a smaller bin size. For example, the estimates for the maximal conductance of the potassium channel g_K all cluster within the $0 - 100$ range, but none of the predictions deemed to be good estimate the maximal conductance to be exactly zero. Instead, all the estimates for g_K cluster in the $20 - 40$ range in good agreement with biophysically plausible values. The non-zero values for all of the maximal conductances imply that all the ion currents included in the model are contributing in some way to the model.

The second important trend to notice within the normalized maximal conductance histograms is that the results seem to cluster into one of two distributions. While it is admittedly risky to draw conclusions from a sample size of only 3 neurons,

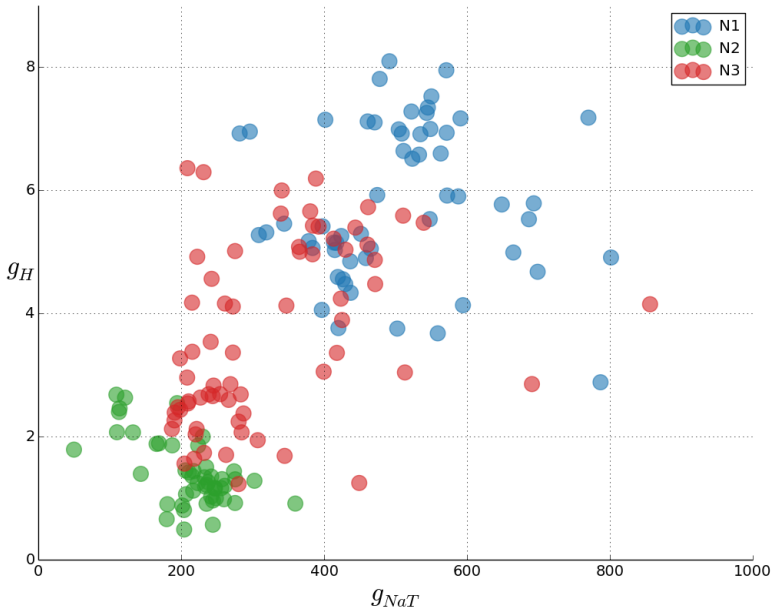

Figure 5.3: Maximal conductance estimates from optimizing the A4 model on N1, N2, and N3, projected into a 2 dimensional conductance space.

it appears from the histograms that the distributions resulting from the estimates of N2 and N3 generally agree with one another, while the estimates from N1 form a second distribution. Again, judging from the histograms, this trend was observed most clearly for $gNaT$ and gH, which motivated the creation of projections of the estimation results into a 2-dimensional maximal conductance space along $gNaT$ and gH axes.

The results from looking at the 2D maximal conductance projections are shown in Figure 5.3. The data show that the A4 model consistently puts maximal conductance estimates for a given neuron within a particular region of conductance space, with the values generated from N2 and N3 clustering in the lower left of the $gNaT$ and gH axes, while the estimates from N1 cluster more broadly in the upper right corner.

This diagram contains two important clues that the model appears to be working as intended. First, the results from different epochs generated from the same neuron are self consistent. This can be seen most clearly in the tight grouping generated from N2, but is also apparent in the clustering of estimates generated from N1 and N3. Secondly, the A4 model appears to be capable of differentiating between different neurons. While it is difficult to draw definitive conclusions from only 3 neurons, it appears that both N2 and N3 cluster in the same general region of conductance space, while N1 occupies a second region. If this result were to be replicated using a larger number of neurons, it would suggest that there exist at least 2 subclasses of cell within HVC X-projecting neurons.

5.5 B4 Model

The B4 model is a single-compartment conductance based model drawn from Meliza et al. [44] and consists of 9 different ion currents in addition to an injected current. The state of the model is given by the differential equation

$$C_m \frac{dV}{dt} = I_{NaT} + I_{NaP} + I_{K1} + I_{K2} + I_{K3} + I_h + I_L + I_{CaL} + I_{CaT} + \frac{I_{inj}}{I_{SA}}., \quad (5.3)$$

where the parameter I_{SA} is used to set the scale of the injected current. The dynamics for each of the ion currents are specified in the table below. It should be noted that the activation and inactivation variables are different for each ion current (i.e. the m appearing in the equation for I_{NaT} is not the same as the m appearing in the equation for I_{NaP}), but are called m and h respectively in order to easily compare the behavior of different ion currents.

Table 5.4: B4 model ion currents including activating and inactivating gating variables associated with each

Current	Dynamics	Activating	Inactivating
I_{NaT}	$g_{NaT}m^3h(E_{Na}-V)$	m	h
I_{NaP}	$g_{NaP}m(E_{Na}-V)$	m	–
I_{K1}	$g_{K1}m^4(E_K-V)$	m	–
I_{K2}	$g_{K2}m^4h(E_K-V)$	m	h
I_{K3}	$g_{K3}m(E_K-V)$	m	–
I_h	$g_h h(E_h-V)$	–	h
I_L	$g_L(E_L-V)$	–	–
I_{CaL}	$p_{rlt}m^2\left(\frac{p_{gou}-0.001e^{\frac{V}{13.0}}}{e^{\frac{V}{13.0}}-1.0}\right)V$	m	–
I_{CaT}	$p_{rht}m^2h\left(\frac{p_{gou}-0.001e^{\frac{V}{13.0}}}{e^{\frac{V}{13.0}}-1.0}\right)V$	m	h

each of the gating variables (both activating and inactivating) $x = m, h$ in the conductance-based model described above are given by the differential equation

$$\frac{dx}{dt} = \frac{0.5\left(1+\tanh\left(\frac{V_x-V_{1/2}}{\kappa}\right)-2x\right)}{\tau_o+\tau_{max}\left(1-\tanh^2\left(\frac{V_x-V_{1/2}}{\sigma}\right)\right)}, \tag{5.4}$$

where the hyberbolic tangent form has been used to improve numerical stability. In this equation $V_{1/2}$ is the half-activation voltage, κ is the slope of the activation function between the closed and open state, τ_o is the minimum relaxation time, $\tau_{max}+\tau_o$ is the peak relaxation time, and σ is the width of the relaxation time function.

5.5.1 B4 Model Optimization Results Estimating All Parameters

The table below summarizes the results of the optimization procedure for the B4 model in which all parameters and the state of all variables are estimated. The number of epochs is the number of available data sets, and agrees with the count given

Table 5.5: B4 model optimization results including total size of available data set and how many runs resulted in good results

Folder	Recordings	Completed	Good Pred	Points Est	Points Pred
N1	213	63	22	30000	50000
N2	174	97	44	30000	50000
N3	209	128	59	30000	50000

in the introduction. IPOPT was run for the B4 model on all 596 epochs available. After all epochs were run, a two part process was used to separate good predictions from failures.

When run on a given data set, IPOPT can succeed or fail to complete the optimization. The number in the *Completed* column is the number of epochs for which IPOPT completed the optimization procedure. A completed run does not guarantee that IPOPT produced a good or even reasonable answer, merely that the optimization completed and returned a set of parameters for that data set. There was no single obvious feature of the data which could explain why IPOPT succeeded for some epochs but not for others. The size of the data set used to do the assimilation certainly impacts the number of runs which succeed or fail (with the general trend that when more data are used, fewer runs completed), but there is nothing that distinguished successes and failures for a fixed amount of data.

Not all of IPOPTs completions produce good predictions. For this reason the number of IPOPT results which produce a reasonable prediction is recorded in the *GoodPred* column. The judgment of what constitutes a reasonable prediction is qualitative, but the the results thus far have typically been extremely good or extremely bad, making such judgment easier.

For each neuron in the data set, the B4 model was able to very accurately

predict spiking behavior at least some of the time. Nearly all of the *estimates* look very good, illustrating the importance of prediction in discriminating between results. An excellent looking estimate often gives rise to a poor prediction, but the converse almost never happens. In order to avoid presenting potentially misleading figures, all plots included here include both estimation and prediction windows alone.

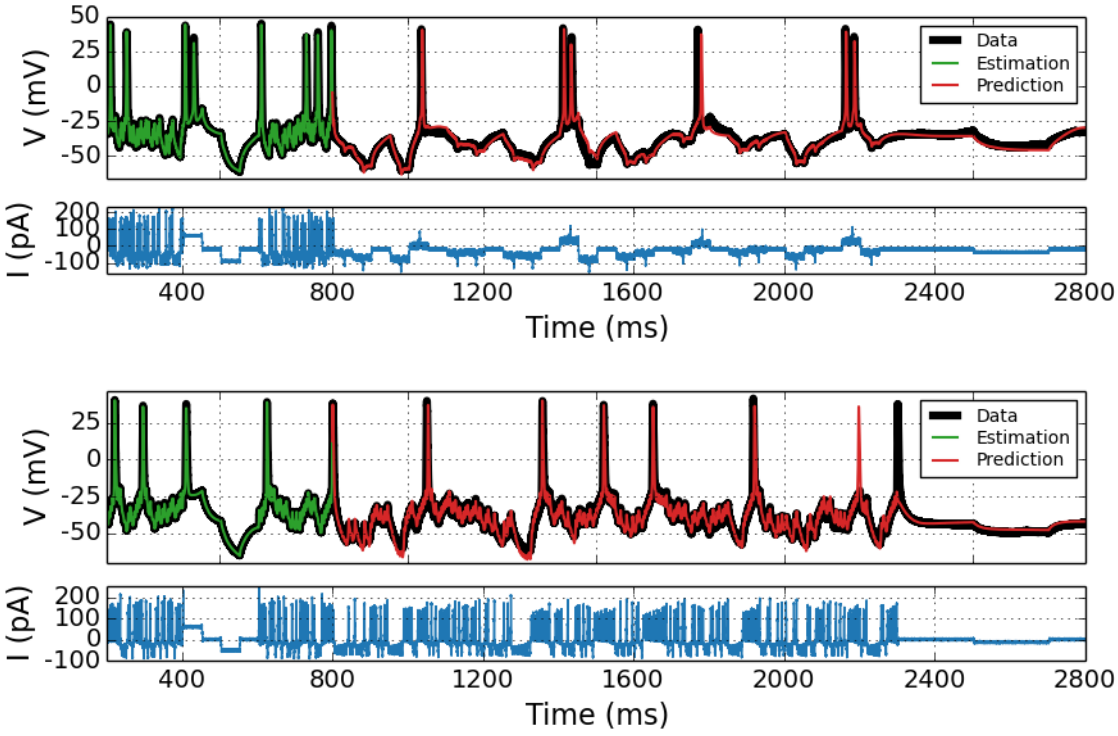

Figure 5.4: N3aj3_40 and N2aj1_29. Prediction of the B4 model after using 30000 data points to estimate all of the parameters and state variables of the model.

Using the parameter output from the 115 epochs which resulted in good predictions, it is possible to look at the statistical properties of successful assimilation runs. In both the B4 and A4 models used, we expect that the model is not as sensitive to the kinetic parameters as to the values of the maximal conductances. In light of this hypothesis, it makes sense to investigate the distribution of estimated maximal

conductance values to check both for self consistency within a given data set, as well as differentiation between different data sets.

In order to produce the maximal conductance histograms seen below, for each of the 115 epochs which produced good prediction the parameter values for maximal conductance were binned by neuron. The bounds constraining the possible values for each parameter are determined in the *specs.txt* file and define the upper and lower allowed values for each of the parameters. The interval between upper and lower limits was divided into 50 equally sized bins, which were then used to create the histograms seen below. The results do not appear to show clear differentiation between neurons, though appear to be reasonably self consistent. The histograms for $gA2$, *gou*, and *gin* show that many of the runs result in parameter estimations at the upper bound of the search range, which suggests that the upper bound for these conductance be increased in any future runs.

The conductances as estimated in this way do not appear to show any clear grouping or differentiation between neurons. This result is consistent with the results of the IPOPT assimilation procedure used in BioHH1. Several explanations of this result are possible. Since all 3 neurons used for assimilation were X-projecting, it is plausible that all 3 posses the same same general values of maximal conductance and therefore cannot be distinguished in this way.

A second possibility is that the model cannot adequately distinguish between traces from different neurons. Evidence for this is provided by the A4 conductance only estimation results shown below (see Figure 5.3), which illustrates that for at least some models conductance values can be used to distinguish between the 3 neurons in this data set.

From estimating the conductances alone it does not appear that the B4 model is able to correctly predict spiking behavior in the cases considered. As spike timing constitutes a crucial measure of a model's ability to predict accurately, this implies that a variable-conductance, but fixed-kinetics, version of the B4 model will not be able to reproduce experimentally observed voltage traces to the desired precision.

5.5.2 B4 Model Optimization Estimating Conductances

Attempting to fix the kinetic parameters and estimate only the maximal conductances in the B4 model did not result in good predictions for any of the cases considered. In the first attempt, the kinetic parameters were fixed at their initial guesses as specified in the *specs.txt* file. Most of the predictions agreed with the data for a very brief 20 ms at the beginning of the prediction window, before quickly diverging from the data trajectory. In many cases the prediction simply flat-lined from this point onward. Since we know that accurate predictions are possible for *some* set of kinetic parameters, it is likely that a better choice of kinetic parameters at which to fix the kinetic parameters would result in a model which could provide better predictions.

Another attempt was made at estimating maximal conductances using a better choice for the fixed values of the kinetic parameters. Kinetic parameter values were fixed based on values determined from a successful run of IPOPT in which *all* parameters were estimated. For instance, if run 14 is identified as the successful run, then all kinetic parameters are fixed at the values of the estimation procedure in run 14. After the kinetic parameters were fixed, IPOPT was asked to vary maximal conductances alone to find those values which produced the best estimation using the

values of kinetic parameters as determined by run 14. This second approach proved more successful than the first, though the predictions were not nearly as good as in the case for which all parameters were estimated.

Chapter 6

Use of Assimilation on Synaptic Plasticity Models

Synaptic plasticity is thought to provide the underlying biological mechanism for memory and learning. However, because of the complicated chemical pathways contributing to plasticity and the experimental challenge associated with isolating and recording data from a synapse, much about plasticity is still unknown. Plasticity experiments produce limited data, but plasticity models require values for states not observed. By choosing a suitable dynamic model for synaptic plasticity it is conceivable that data assimilation techniques can be used to reveal underlying features of plasticity which are not directly observable using laboratory techniques.

This chapter consists of two sections. The first section explains the scope of the models presented and provides background on developing models for plasticity. It then reviews important experimental results and findings that any putative plasticity model should replicate. Additionally, this section provides a coarse description of the experimental procedures used to obtain these results, which allows for realistic

considerations of what measurements are available for solving the data assimilation problem. The second section concerns the results of applying data assimilation techniques on a synaptic plasticity twin experiment. The model and procedures used are explained and results are presented.

6.1 Overview of Plasticity Phenomena

Synaptic plasticity refers to changes in the strength of a synaptic connection between neurons which modulate the synaptic signal. For modeling purposes, we include plasticity effects by introducing a weighting variable, whose dynamics determine the time course of changes in the maximal synaptic conductance. Synaptic plasticity can be divided into short term plasticity and long-term plasticity based on the duration of the effect. Typical short-term plasticity effects persist on the order of milliseconds to minutes, while long-term plasticity effects last an hour or more. Long-term plasticity is thought to provide a mechanism for memory and learning and is the form of plasticity that will be discussed here.

Biochemical and physiological changes may both contribute to long-term plasticity, though the chemical changes likely provide the dominant force. From the physiological standpoint, changes in the shape of the dendritic spine may contribute to long-term plasticity, by changing the electrical properties of the dendrites by altering their physical shape. However, from estimation of the size of the change in the maximal conductance it can be seen that the impact on the maximal conductance is small. Additionally, the impact of physiological changes in the dendrites would depend on the relative magnitude of the resistance of the dendrites compared to the soma and would vary across different neurons. For these reasons, the effect of physical size will

be neglected in the subsequent discussion and plasticity will be treated as a chemical phenomenon.

Long-term plasticity, can be subdivided into the categories of long-term potentiation (LTP) and long-term depression (LTD). LTP is defined as a long-lasting increase in the amplitude of a synaptic response following brief, high-frequency activity of a synapse and was described by Landfield and Deadwyler [35]. Here, high-frequency refers to a range of 50 Hz to 200 Hz for induction protocols lasting from 0.5 to 20 seconds. LTP can be separated into an induction phase during which the change in synaptic strength occurs and a maintenance phase during which the change is expressed. The onset kinetics of the induction phase are not reliably known in part because of the difficulty of separating the onset of long-term plasticity from the effects of STP. The expression of LTP begins within minutes of induction, and the effects can last over a time scale from days to weeks. Many of the studies of LTP come from the study of mammalian hippocampal neurons, so the results may not generalize to the avian case. Experimentally, the effects of LTP can be measured by recording the amplitude of the synaptic response before and after the stimulation protocol using field potential recordings.

The biochemical processes underlying LTP are complex. LTP can be modulated by a number of neurotransmitter systems and appears to be different at different synapses. Further complicating matters, it is likely that plasticity depends on second-messenger systems.

Moving deeper down the taxonomic tree, LTP can in turn be divided into NMDA-dependent and NMDA-independent forms. The distinction between these forms of plasticity comes down to the way in which Ca^{2+} enters into the postsynaptic neurons.

Broadly speaking, the Ca^{2+} concentration in the postsynaptic neuron through three route: voltage gated channels, Ca^{2+}-permeable NMDA receptors, and intracellular calcium stores. It is known that induction of LTP depends on the postsynaptic Ca^{2+} concentration. Experiments in which postsynaptic Ca^{2+} was blocked using Ca^{2+} chelators (chemicals which preferentially bond to Ca^{2+}, thereby preventing it from engaging in other reactions), showed that LTP was not induced in this case. Notably, if the Ca^{2+} was chelated after the induction phase, there was no observed effect on expression. Perhaps unsurprisingly, NMDA-dependent plasticity is thought to be caused by an influx of Ca^{2+} via NMDA receptors, when the depolarization of the postsynaptic neuron is sufficient to relieve the Mg^{2+} block that typically prevents an influx of Ca^{2+} through NMDA receptors. NMDA-independent plasticity is thought to occur via voltage gated channels, though this form has been studied less extensively.

From the perspective of neurotransmitters, it has been observed that depolarization of the presynaptic neuron causes glutamate release. The most prominent mechanism is persistent activation of one or more protein kinases associated with expression. Most experimental evidence favors increase in transmitter release accounting for at least part of increase in synaptic efficacy. From this perspective, the conductance of a synapse can be thought of as a sort of time average of the neurotransmitter release rate. If Ca^{2+} influx in the postsynaptic neuron triggers release of transmitter, then logically this process requires some sort of retrograde messenger to let the presynaptic neuron know about the Ca^{2+} influx in the postsynaptic neuron. While evidence suggests that the rise in the concentration of Ca^{2+} in the postsynaptic neuron is the initial trigger for LTD, it possible that small increase in Ca^{2+} concentration leads to LTD while larger increases lead to LTP.

LTD is also thought to be required for learning, because otherwise all synapses would saturate and more info can be stored in a system which can increase and decrease strength. It has also been observed that LTD sometimes occurs under non-Hebbian conditions.

6.1.1 Discussion of Model Features

From the considerable experimental evidence available, we must choose which features to attempt to replicate in our model. In order to do this, it is useful to first review relevant experimental evidence, before specifying a particular model and discussing which features are include and left out.

Evidence for LTP was first described in rabbit hippocampal neurons (Bliss and Lømo [8]). Typical plasticity experiments use extracellular field recordings to measure the amplitude of an excitatory postsynaptic potential (EPSP). The time course of the recorded extracellular field potential is then taken to be approximately equal to the transmembrane current. Using this technique, it is then possible to measure the amplitude of EPSPs both before and after an induction paradigm designed to invoke plastic changes. By comparing the maximal conductances measured before and after the induction protocols, one can quantify any changes observed in plasticity.

Usually a stimulus intensity threshold exists, below which LTP will not be observed. As one might imagine, LTP also exhibits saturation, meaning that synaptic strengths cannot be increased indefinitely. Many forms of long-term plasticity appear to require activity in both the presynaptic and postsynaptic neuron, a hypothesis which is referred to as Hebb's rule (Hebb [23]). There is a fair amount of evidence to support this hypothesis (Brown et al. [9]; Bi and Poo [6]),though there also exists

some evidence for plasticity which does not obey Hebb's rule.

Many important characteristics of synaptic plasticity are grouped together under the heading of spike timing dependent plasticity. STDP refers to dependence on the precise timing of presynaptic and postsynaptic activity. In typical STDP experiments, both pre- and postsynaptic neurons are stimulated to produce pairs of spikes separated by some time τ. The temporal separation between the spikes is varied and the effect on the response recorded (Bi and Poo [5]). This procedure produces a curve representing the change in maximal conductance as a function of the temporal separation between pre- and postsynaptic spikes. However, experiments performed using more than just 2 spikes (*triplet* and *quadruplet* induction paradigms) have suggested that there may be more complicated processes at work that can not be explained purely by a pairing model (Froemke and Dan [18]).

More recently, experiments have demonstrated the importance of precise timing of calcium in determining plasticity phenomena (Zucker [71]). A brief but relatively large increase in Ca can trigger LTP, while a more moderate prolonged rise induces LTD (Yang et al. [67]). The importance of calcium in plasticity has been established for some time, but the details of the model have yet to be worked out. Models for the effect of Ca levels on plasticity often involve complicated chemical pathways with many parameters that must be hand-tuned to achieve the expected results (Rubin et al. [54]). The role of calcium is an important consideration that is a worthy area of future investigation.

6.1.2 Phenomenological Models

Because of the complexity of and uncertainty surrounding the biological processes underlying plasticity phenomena, it is useful to consider phenomenological models for synaptic plasticity (Morrison et al. [46]). For our purposes, a model is deemed to be phenomenological if the states of the model are not presumed to have a direct correspondence to any underlying chemical signals. Instead, the model is intended to replicate general observed results, without concern for the precise description of the underlying mechanism. Attempts, such as the BCM rule (Bienenstock et al. [7]), have been made to unite experimental results under a common theory, but no single unifying rule has yet been found.

From the standpoint of data assimilation in dynamical systems, one issue with many of the models proposed is that they are discontinuous and thus many of the techniques discussed herein cannot be directly applied. It is therefore desirable to consider a dynamic model of plasticity.

6.2 Plasticity Model

The model we will use will consist broadly of two smaller models: a model for synaptic transmission and a model for spike-timing dependent plasticity. One issue inherent in such a model is that there are two widely separated timescales associated with the two parts described. The synapse itself transmits a signal on the order of 1 ms and is the fast part of the model. In contrast, plasticity is usually invoked by following a procedure that takes place on the order of minutes and thus has a much longer time scale. The separation between these two timescales poses both experimental and

numerical hurdles. Each part of the combined model is explored separately below and finally a combined result is obtained.

The basic plasticity model to start with is given in Abarbanel et al. [2]. This model has several desirable features, though it also has multiple limitations which are in disagreement with experimental observations.The model used is specified by three coupled differential equations given below.

$$\frac{dP}{dt} = f(V_{pre}) - \beta_P P \tag{6.1}$$

$$\frac{dD}{dt} = g(V_{post}) - \beta_D D \tag{6.2}$$

$$\frac{d\Delta g}{dt} = \gamma(PD^{\eta} - DP^{\eta}). \tag{6.3}$$

The functions f, g are here chosen to be

$$f(V_{pre}) = \frac{1}{2}\left(1 + \tanh(\frac{V_{pre} - \theta_P}{\sigma_P})\right) \tag{6.4}$$

$$g(V_{post}) = \frac{1}{2}\left(1 + \tanh(\frac{V_{post} - \theta_D}{\sigma_D})\right). \tag{6.5}$$

Before delving in to the numerical experiments, it is worth examining promintent features of this model as relating to the plasticity phenomena described above. The model given in equation 6.3 is Hebbian in the sense that both presynaptic activity (associated with the state variable P) and postsynaptic activity (associated with the state variable D) are required in order to produce changes in the maximal conductivity. If either P or D is zero, no change results. The state variables P and D decay to zero with a time scale set by the the value of $\beta_{P,D}$, and are only pulled toward non-zero value during times of electrical activity. When both P and D have non-zero values, a

nonlinear competition between the two variables occurs.

From the standpoint of shortcomings, the model does not include the feature of saturation, which refers to the finite strengthening of any given synapse. Similarly, the model can cause Δg to decrease to a negative value which is no longer meaningful as a conductance. Clearly, no explicit information about calcium is included here, though an extension of the model which included a calcium time course is presented in Abarbanel et al. [3].

The model given in equation (6.3) has several advantages. First, it is dynamical meaning that it can be tested against dynamical data assimilation techniques. Second, the model is simple, meaning that the nonlinear optimization problem is easier and the techniques applied are more likely to work. Also due to simpliicity it is easier to interpret the meaning of some of the parameters. For example, the parameter γ sets the time scale on which plasticity occurs. Finally, unlike many phenomenological models the model given in equation (6.3) does not explicitly depend on any particular matching scheme between pre and postsynaptic spikes. In principle, it can thus be used equally well on pair, triplet, or any other arbitrary plasticity induction protocols.

6.2.1 Verification of Model

Table 6.1: Numerical values for parameters used in simulations to verify the model given in 6.6.

Param.	Value
α_P	33.5
α_D	33.5
β_P	0.098
β_D	0.035
η_P	4.0
γ	1e-6

In order to verify the implementation of the model worked as expected. Results were compared with the paper. As many of the results cited in the paper are based on the limiting case of a very narrow spike, the equations for P and D were modified to read,

$$\begin{aligned} \frac{dP}{dt} &= \alpha_P \delta(t - t_{pre}) - \beta_P P \\ \frac{dD}{dt} &= \alpha_D \delta(t - t_{post}) - \beta_D D, \end{aligned} \tag{6.6}$$

and the model was integrated forward from the initial condition $[P, D, \Delta g] = [0, 0, 0]$. The parameters used are given in Table 6.1.

Note that since the functions f and g are delta function, the shape of the incoming spike doesn't matter in this model and it is sufficient to specify spike times alone. The results of using this model are in good qualitative agreement with those given in the paper as well as those cited in Bi and Poo [6] and are shown below.

6.2.2 Plasticity Twin Experiments

The simplest twin experiment we can perform on this model is one in which we specify the form of the driving voltage for P and D and ask if we can recover the parameters from measurements of Δg alone. Note that this procedure is not experimentally plausible, because techniques to inject two voltages and measure a conductance simultaneously do not exist. Nevertheless, this twin experiment is a direct test of whether or not the model given in 6.3 can be successfully subjected to data assimilation. In other words, if this first test does not work, then there is no point in attempting to use assimilation when including this plasticity rule in a larger

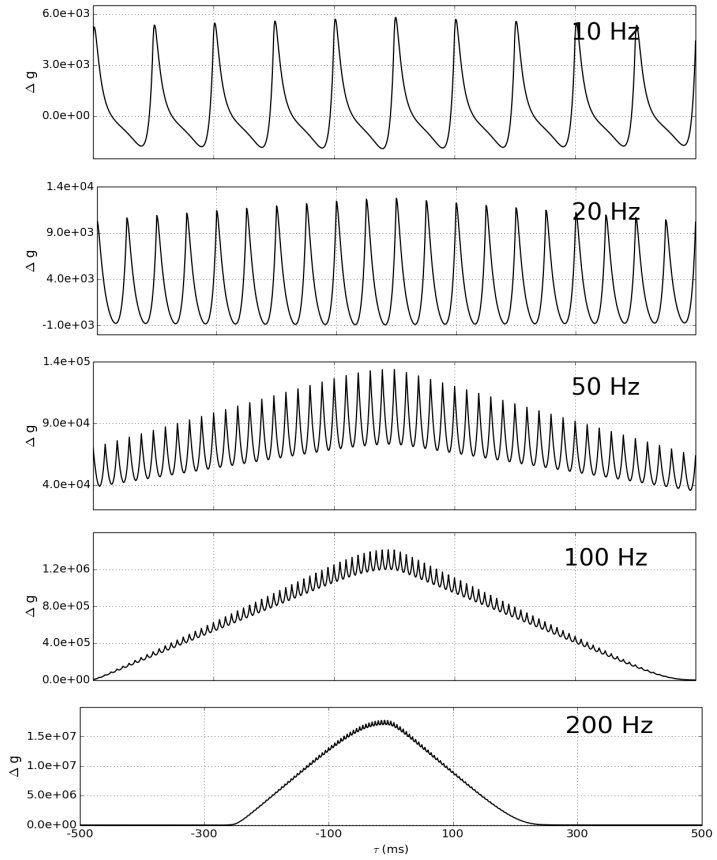

Figure 6.1: Implementation of the plasticity rule using eqn. 6.6 and the parameters specified in Table 6.1. The horizontal axis specifies the temporal offset between pre- and postsynaptic spikes, $\tau = t_{post} - t_{pre}$. Only the frequency of the input voltage is changed between plots.

network.

In performing twin experiments, voltage data were first created using a simple Hodgkin-Huxley neuron. Three different current injection protocols were used in order to test whether some injected current waveforms work better than others.

In each case, the basic procedure is the same. Two identical Hodgkin-Huxley model neurons were injected with the currents specified in the protocol in order to produce two voltage traces. One of these voltage traces was then used in the equation for P, while the other was used in the equation for D. Using the voltage traces

explained above, the full model was integrated forward from the initial condition $[P, D, \Delta g] = [0, 0, 0]$, noise was added to the data for Δg and the state of the model was written to a data file at an interval of $dt = 0.02\,ms$. Using 10,000 points from this data trace, data assimilation was performed using the annealing method and the results are shown in the tables below.

Comparison of Current Waveforms

In the first protocol, a constant injected current of $1.0\,nA$ was used to induce spiking at a regular interval and the simulated membrane potential was used as the voltage in the equation for P. A second voltage trace was then created using the same procedure, but shifted by a time of $\tau = 2\,ms$ and used as the voltage appearing in the differential equation for D. This procedure ensures that presynaptic spikes will always arrive a time τ before postsynaptic spikes. After the equations were integrated forward, Gaussian noise of amplitude A = 0.01 was added to the data for Δg. The estimated parameters for each of the three injection protocols are shown in Table 6.2 below.

In the second protocol, square impulses of $0.3\,nA$ were spaced so to induce spikes at different temporal separation. The spacing of the current pulses varied from $4\,ms$ down to $0\,ms$ in increments of $0.5\,ms$. After a pause of $20\,ms$, the procedure was repeated a second time with the order of the current impulses reversed. This protocol was designed to cause first a strengthening of synaptic strength and then a weakening. After the equations were integrated forward, Gaussian noise of amplitude A = 0.01 was added to the data for Δg.

In the third protocol, the Lorenz 63 model was used to generate two different

chaotic waveforms for current varying between a maximum value of $1.0\,nA$ and a minimum value of $-0.2\,nA$. After the equations were integrated forward, Gaussian noise of amplitude A = 0.01 was added to the data for Δg. The results of using this protocol on the dynamic synapse model are shown in Figure 6.2.

The results from the parameters indicate reasonably good agreement with the true values of the parameters. The discrepancy between the two is likely due at least in part to the close fit between the data and model. It is likely that using either a longer data trace, or a more complicated voltage waveform would result in better estimations.

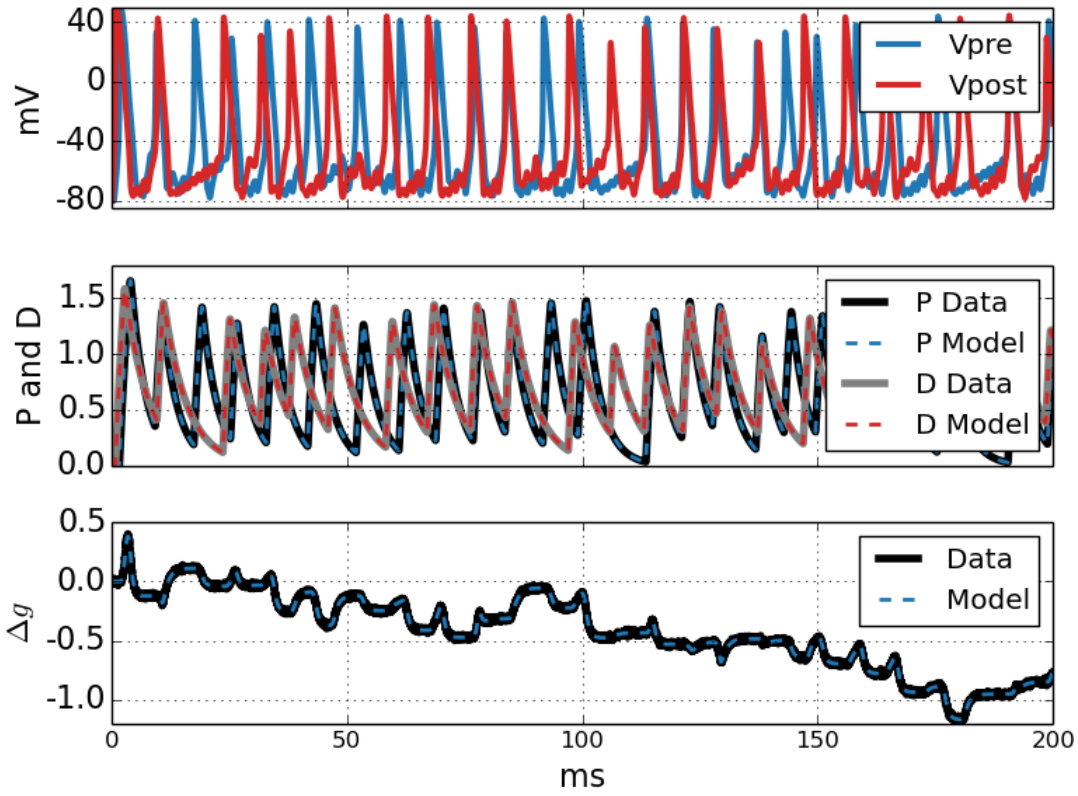

Figure 6.2: Simulation of the plasticity rule given by eqn. 6.3 and current injection protocol 3.

Table 6.2: Parameter estimates from twin experiment using the three current injection protocols described above.

Param.	Correct	Estimate 1	Estimate 2	Estimate 3
θ_P	-20.0	-19.6	-20.0	-20.0
$1/\sigma_P$	0.25	0.137	0.159	0.270
β_P	0.3	0.296	0.299	0.299
θ_D	-5.0	-5.29	-5.13	-5.18
$1/\sigma_D$	0.2000	0.182	0.131	0.163
β_D	0.2	0.200	0.200	0.200
γ	0.1	0.099	0.098	0.099
η	4.0	4.000	4.051	4.019

6.3 Unified Model

6.3.1 Synaptic Model

The synaptic model used here is taken from Gibb et al. [19] and given by the equation,

$$I_{syn}(t) = g_{syn} r(t)[V_{post} - E_{rev}], \tag{6.7}$$

where

$$\frac{dr(t)}{dt} = \alpha T(t)[1 - r(t)] - \beta r(t), \tag{6.8}$$

and

$$T(t) = \frac{T_{max}}{1 + exp(-[V_{pre}(t) - \theta_p]/\sigma_p)}. \tag{6.9}$$

In order to have a model of an actual synapse that exhibits plasticity, it is necessary to combine the synapse model given in 6.7 with the model for plasticity given in 6.3. What we would like is an equation of the form,

$$I_{syn}(t) = g_{syn} w(t) r(t)[V_{post} - E_{rev}], \tag{6.10}$$

where $r(t)$ specifies the gating behavior of the synapse (a short time scale), and $w(t)$ is a weighting variable that describes any changes due to long term potentiation or depression. Saturation specifies that w(t) should be bounded below by zero and above by some maximum strength.

In our model for plasticity, Δg represented the percentage change in synaptic strength and is bounded (in principle) below by -100 and above by whatever the maximum strength is. $\Delta g = 0$ represents no change in synaptic strength and corresponds to $w = 1$. Thus, we can relate w and g linearly through the equation w = (g/100 + 1). A full model for a plastic synapse can then be described by the dynamical equations

$$\frac{dr(t)}{dt} = T(t)[1 - r(t)] - \beta r(t) \tag{6.11}$$

$$\frac{dP}{dt} = \frac{1}{2}\left(1 + \tanh(\frac{V_{pre} - \theta_P}{\sigma_P})\right) - \beta_P P \tag{6.12}$$

$$\frac{dD}{dt} = \frac{1}{2}\left(1 + \tanh(\frac{V_{post} - \theta_D}{\sigma_D})\right) - \beta_D D \tag{6.13}$$

$$\frac{dw}{dt} = \gamma_w(PD^\eta - DP^\eta). \tag{6.14}$$

At this stage the model can begin to represent plausible experiments. In a typical plasticity experiment, the postsynaptic neuron is voltage clamped and the presynaptic neuron is stimulated in order to produce a synaptic current. The experiment then proceeds in two alternating phases. In a measurement phase, low frequency pulses are used to determine the maximal conductance as described above. A low frequency is selected so that plasticity mechanisms will not be activated during the measurement phase. After the conductance has been measured, a series of higher frequency pulses are used in order to elicit plasticity phenomena. After the high

frequency pulses, the measurement phase is repeated in order to ascertain any change in the conductance.

This procedure has several drawbacks where numerical methods could offer improvements. First, much of the information contained in the waveform of the synaptic current is wasted. In order to determine the maximal conductance, only the maximum value of the current is used. Information about the shape of the synaptic current pulse and how it changes as a function of time is neglected in the experimental procedure, but can be used to determine synaptic parameters numerically. Second, the experimental procedure is more limited in resolution. Long series of pulses (on the order of minutes) are needed in order to see a detectable change in the synaptic current. Data assimilation techniques could potentially bypass this limitation by picking out smaller changes than can be easily seen by eye. A third and related difficulty is that the experimental procedure is time consuming. If data assimilation techniques can be used to determine relatively smaller changes in synaptic strength, the length of the experimental protocols could be correspondingly reduced.

From looking at the equations given in 6.14, we can see that the time scale of the plasticity variable is set by γ_w and is typically much longer than any of the other relevant time scales in the problem. This issue is manifested numerically and experimentally in the long times needed to perform plasticity experiments. In numerical experiments, we can get around this by using an artificially higher value for γ_w.

6.3.2 Unified Model Twin Experiments

As with the synaptic and plasticity components of the model, twin experiments were performed on the unified model. The first series of twin experiments is not intended to reflect current experimental capabilities, but still illustrates how data assimilation using the synaptic current can determine the parameters of the underlying system. The second series of twin experiments is intended to be experimentally plausible and uses a voltage-clamped postsynaptic neuron. While voltage-clamping the postsynaptic neuron makes the dynamics less interesting, this twin experiment represents a viable experiment.

Non-voltage clamped case

Varying the voltage of both the pre- and postsynaptic neurons allows us to explore a greater dynamical range for the model and likely makes estimating all state and parameters easier. An example of a twin experiment performed using these criteria is shown in Figure 6.3. In this experiment, only the synaptic current is measured and all unmeasured states and parameter are estimated using only this measurement. The maximal conductance and reversal potential are fixed at their correct values, because these parameters can be accurately measured using the experimental techniques describe above. Since the values of these parameters are taken to be fixed, they are not included in the results appearing below.

The twin experiment was performed using the following procedure. The Hodgkin-Huxley equations were integrated forward using an injected current modeled on the Lorenz 63 equations. The simulated voltages produced using this procedure were then used as V_{pre} and V_{post} appearing in equation 6.14 and the differential equations

were integrated forward in time. Plugging the results of this integration into equation 6.10 , a simulated synaptic current was produced. From the simulated synaptic current, 20,000 data points of spacing $dt = 0.02\,ms$ were then used to assimilate the model using the annealing procedure.

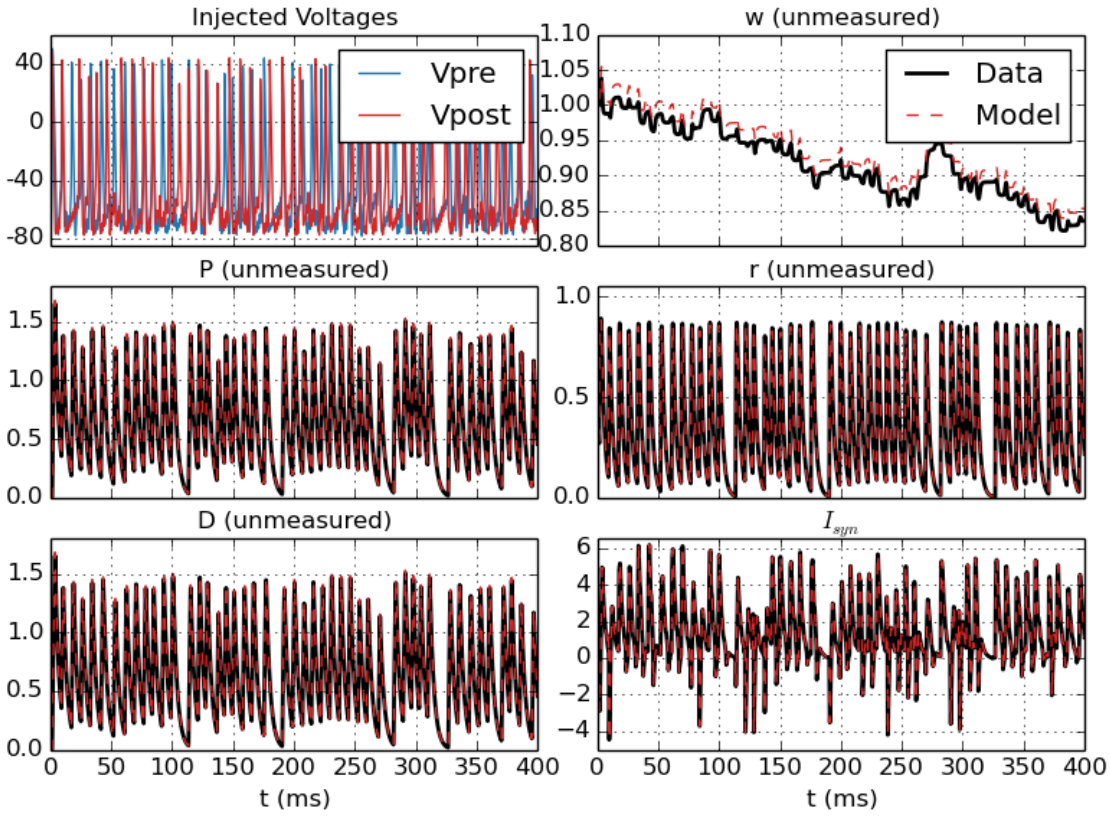

Figure 6.3: Implementation of the combined synapse model using equation 6.14 and current injection protocol 3.

Using this procedure, parameters in the model can be accurately estimated and an example of these estimates is shown in Table 6.3. While some of the parameters in the model gave relatively poor estimates (the parameters $\theta_{P,D}$ and $\sigma_{P,D}$ appearing in the hyperbolic tangent function proved particularly difficult to estimate), there are two mitigating factor which make this inaccuracy less problematic. First, it can be observed that varying the parameters in the hyperbolic tangent does not much

change the dynamics of the P and D state variables, and therefore has even less impact on the behavior of the model synaptic current. Second, although the model is phenomenological and none of the parameters directly correspond to any chemical, many of the parameters which are well estimated can be more easily interpreted. For example, the value of η can be related through the law of mass action to the number of molecules of some substances P and D that ultimately contribute to plasticity. Similarly, the value of γ_w sets the time scale on which plasticity occurs and is therefore also of interest.

Table 6.3: Parameter estimates from twin experiment using the three current injection protocols described above.

Param.	Correct	Estimate
θ_P	-20.0	-20.0
$1/\sigma_P$	0.250	1.000
β_P	0.300	0.299
θ_D	-5.00	-5.490
$1/\sigma_D$	0.200	0.198
β_D	0.200	0.202
γ_w	0.110	0.112
η	4.000	4.163
$a * T_{max}$	3.300	3.089
θ_w	2.000	2.538
$1/\sigma_w$	0.100	0.094
β_r	0.380	0.379

Voltage clamped case

In a viable experiment, the postsynaptic neuron is voltage clamped in order to allow for measurement of the synaptic current. However, because the voltage is now fixed the state associated with the postsynaptic neuron, D, now goes to a fixed point. This makes it impossible to estimate any of the parameters appearing the dynamical equations for this state. On the other hand, the dynamical equations

become simplified. First, we can solve for the fixed point of the dynamic equation for D by setting $\frac{dD}{dt} = 0$. This gives,

$$D_o = \frac{1}{2\beta_D}\left(1 + \tanh\left(\frac{V_{post} - \theta_D}{\sigma_D}\right)\right), \tag{6.15}$$

a constant value. If we imagine clamping the postsynaptic neuron and consider only the steady-state behavior, we can thus effectively treat D as a parameter rather than a state variable. If we now estimate only the fixed value of D, we reduce the dimension of the the system of equations given in 6.14 by one and eliminate having to estimate 3 parameters. Because this version of the equations requires only measuring the synaptic current in the case of a voltage-clamped postsynaptic neuron, the biological analog to this numerical experiment should be entirely feasible. The limitation to this procedure is that because the postsynaptic voltage is fixed, the spike-timing dependent effects cannot be investigated directly. The more limited dynamical range explored by the model results in greater difficulty estimating the parameters.

Using a fixed voltage value for the postsynaptic neuron, a twin experiment was performed using 20,000 data points with a separation of $dt = 0.02ms$. The parameter estimates for this procedure are shown in Table 6.4 and are generally less accurate than the case in which the postsynaptic voltage is also allowed to vary. From looking a the estimates given in Figure 6.4, this is likely because the measured synaptic current can be well fit with the parameters estimated.

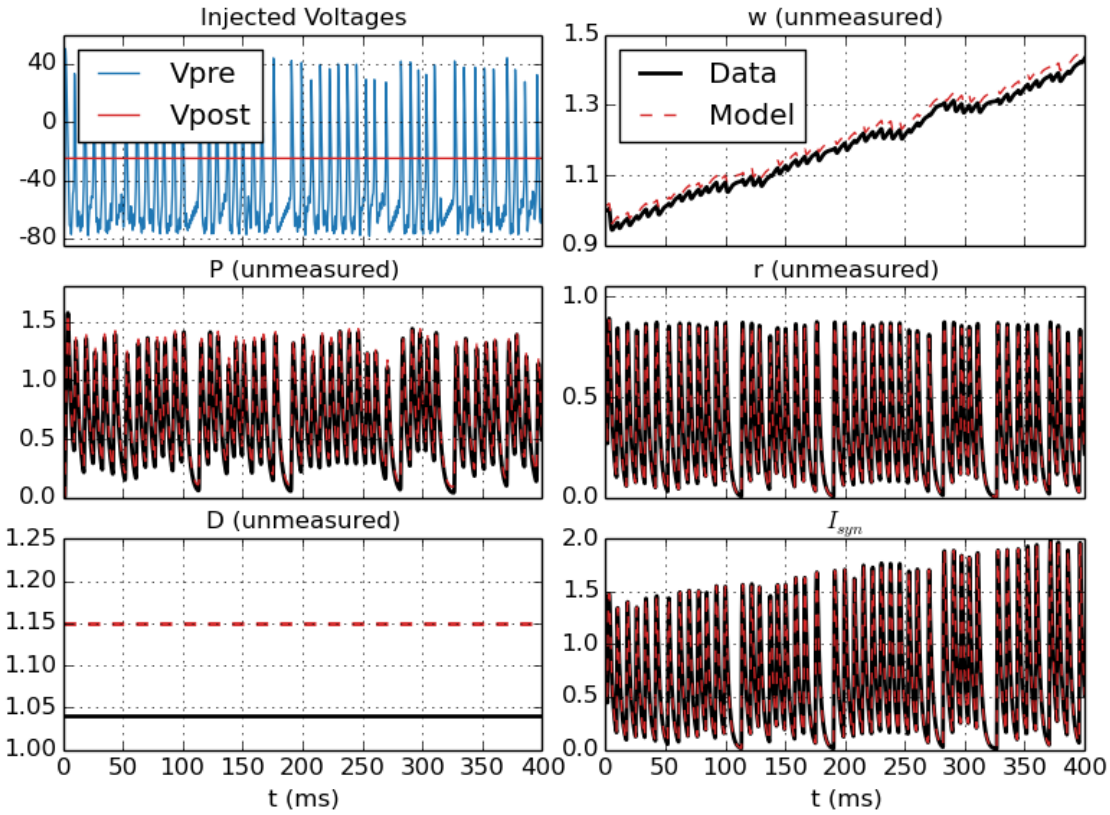

Figure 6.4: Implementation of the combined synapse model using equation 6.14 and a fixed postsynaptic voltage value of $V_{post} = -25mV$.

Table 6.4: Parameter estimates from twin experiment with fixed postsynaptic voltage.

Param.	Correct	Estimate
θ_P	-20.0	-21.3
$1/\sigma_P$	0.066	0.051
β_P	0.300	0.295
D_o	1.040	1.149
γ_w	0.010	0.003
η	4.000	5.933
$a * T_{max}$	3.300	3.096
θ_w	2.000	2.593
$1/\sigma_w$	0.200	0.097
β_r	0.380	0.379

Bibliography

[1] Henry Abarbanel. *Analysis of observed chaotic data*. Springer Science & Business Media, 1997.

[2] Henry DI Abarbanel, R Huerta, and MI Rabinovich. Dynamical model of long-term synaptic plasticity. *Proceedings of the National Academy of Sciences*, 99 (15):10132–10137, 2002.

[3] Henry DI Abarbanel, Leif Gibb, Ramón Huerta, and Mikhail I Rabinovich. Biophysical model of synaptic plasticity dynamics. *Biological cybernetics*, 89(3): 214–226, 2003.

[4] Henry DI Abarbanel, P Bryant, Philip E Gill, Mark Kostuk, Justin Rofeh, Zakary Singer, Bryan Toth, and Elizabeth Wong. Dynamical parameter and state estimation in neuron models. *The dynamic brain: an exploration of neuronal variability and its functional significance*, 2011.

[5] Guo-qiang Bi and Mu-ming Poo. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *The Journal of neuroscience*, 18(24):10464–10472, 1998.

[6] Guo-qiang Bi and Mu-ming Poo. Synaptic modification by correlated activity: Hebb's postulate revisited. *Annual review of neuroscience*, 24(1):139–166, 2001.

[7] Elie L Bienenstock, Leon N Cooper, and Paul W Munro. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *The Journal of Neuroscience*, 2(1):32–48, 1982.

[8] Tim VP Bliss and Terje Lømo. Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *The Journal of physiology*, 232(2):331–356, 1973.

[9] Thomas H Brown, Edward W Kairiss, and Claude L Keenan. Hebbian synapses: biophysical mechanisms and algorithms. *Annual review of neuroscience*, 13(1): 475–511, 1990.

[10] J Cessna, C Colburn, and TR Bewley. Multiscale retrograde estimation and forecasting of chaotic nonlinear systems. In *Decision and Control, 2007 46th IEEE Conference on*, pages 2205–2210. IEEE, 2007.

[11] Arij Daou, Matthew T Ross, Frank Johnson, Richard L Hyson, and Richard Bertram. Electrophysiological characterization and computational models of hvc neurons in the zebra finch. *Journal of neurophysiology*, 110(5):1227–1245, 2013.

[12] Alain Destexhe, Agnessa Babloyantz, and Terrence J Sejnowski. Ionic mechanisms for intrinsic slow oscillations in thalamic relay neurons. *Biophysical Journal*, 65 (4):1538, 1993.

[13] Justin R Dunmyre, Christopher A Del Negro, and Jonathan E Rubin. Interactions of persistent sodium and calcium-activated nonspecific cationic currents yield dynamically distinct bursting regimes in a model of respiratory neurons. *Journal of computational neuroscience*, 31(2):305–328, 2011.

[14] Patrick Dutar, Huan M Vu, and David J Perkel. Multiple cell types distinguished by physiological, pharmacological, and anatomic properties in nucleus hvc of the adult zebra finch. *Journal of Neurophysiology*, 80(4):1828–1838, 1998.

[15] Robert M Fano and David Hawkins. Transmission of information: A statistical theory of communications. *American Journal of Physics*, 29(11):793–794, 1961.

[16] Richard Phillips Feynman and Albert R Hibbs. *Quantum mechanics and path integrals*, volume 2. McGraw-Hill New York, 1965.

[17] Andrew M. Fraser and Harry L. Swinney. Independent coordinates for strange attractors from mutual information. *Phys. Rev. A*, 33:1134–1140, Feb 1986. doi: 10.1103/PhysRevA.33.1134. URL http://link.aps.org/doi/10.1103/PhysRevA.33.1134.

[18] Robert C Froemke and Yang Dan. Spike-timing-dependent synaptic modification induced by natural spike trains. *Nature*, 416(6879):433–438, 2002.

[19] Leif Gibb, Timothy Q Gentner, and Henry DI Abarbanel. Inhibition and recurrent excitation in a computational model of sparse bursting in song nucleus hvc. *Journal of neurophysiology*, 102(3):1748–1762, 2009.

[20] Philip E Gill, Walter Murray, and Margaret H Wright. *Practical optimization*. Academic press, 1981.

[21] David E. Goldman. Potential, impedance, and rectification in membranes. *The Journal of General Physiology*, 27(1):37–60, 1943. doi: 10.1085/jgp.27.1.37. URL http://jgp.rupress.org/content/27/1/37.abstract.

[22] Richard HR Hahnloser, Alexay A Kozhevnikov, and Michale S Fee. An ultra-sparse code underliesthe generation of neural sequences in a songbird. *Nature*, 419(6902):65–70, 2002.

[23] Donald O Hebb. The organization of behavior, 1949.

[24] Bertil Hille et al. *Ion channels of excitable membranes*, volume 507. Sinauer Sunderland, MA, 2001.

[25] AL Hodgkin and AF Huxley. Propagation of electrical signals along giant nerve fibres. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, pages 177–183, 1952.

[26] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544, 1952.

[27] Daniel Johnston, Samuel Miao-Sin Wu, and Richard Gray. *Foundations of cellular neurophysiology*. MIT press Cambridge, MA, 1995.

[28] Kevin Judd and Thomas Stemler. Failures of sequential bayesian filters and the successes of shadowing filters in tracking of nonlinear deterministic and stochastic systems. *Physical Review E*, 79(6):066206, 2009.

[29] Matthew B Kennel, Reggie Brown, and Henry DI Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical review A*, 45(6):3403, 1992.

[30] Donald E Kirk. *Optimal Control: An Introduction*. Prentice-Hall, 1970.

[31] Daniel P Knudsen and Timothy Q Gentner. Mechanisms of song perception in oscine birds. *Brain and language*, 115(1):59–68, 2010.

[32] Mark Kostuk. *Synchronization and Statistical Methods for the Data Assimilation of HVc Neuron Models*. PhD thesis, University of California San Diego, 2012.

[33] M Kubota and N Saito. Sodium-and calcium-dependent conductances of neurones in the zebra finch hyperstriatum ventrale pars caudale in vitro. *The Journal of physiology*, 440(1):131–142, 1991.

[34] Michinori Kubota and Ikuo Taniguchi. Electrophysiological characteristics of classes of neuron in the hvc of the zebra finch. *Journal of Neurophysiology*, 80(2): 914–923, 1998.

[35] Philip W Landfield and Sam A Deadwyler. *Long-term potentiation from biophysics to behavior*, volume 35. Liss, 1988.

[36] R Llinas and Y Yarom. Properties and distribution of ionic conductances generating electroresponsiveness of mammalian inferior olivary neurones in vitro. *The Journal of Physiology*, 315(1):569–584, 1981.

[37] Rodolfo Llinas and Y Yarom. Electrophysiology of mammalian inferior olivary neurones in vitro. different types of voltage-dependent ionic conductances. *The Journal of physiology*, 315(1):549–567, 1981.

[38] Michael A Long and Michale S Fee. Using temperature to analyse temporal dynamics in the songbird motor pathway. *Nature*, 456(7219):189–194, 2008.

[39] Michael A Long, Dezhe Z Jin, and Michale S Fee. Support for a synaptic chain model of neuronal sequence generation. *Nature*, 468(7322):394–399, 2010.

[40] Andrew C Lorenc and Tim Payne. 4d-var and the butterfly effect: Statistical four-dimensional data assimilation for a wide range of scales. *Quarterly Journal of the Royal Meteorological Society*, 133(624):607–614, 2007.

[41] Edward N Lorenz. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2):130–141, 1963.

[42] Edward N Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1, 1996.

[43] Daniel Margoliash. Behavioural biology: Name that tune. *Nature*, 432(7018): 682–683, 2004.

[44] Daniel Meliza, Mark Kostuk, et al. Dynamical state and parameter estimation validated by prediction of experimental membrane voltages for conductance-based models of individual neurons. Unpublished, 2012.

[45] Richard Mooney and Jonathan F Prather. The hvc microcircuit: the synaptic basis for interactions between song motor and vocal plasticity pathways. *The Journal of neuroscience*, 25(8):1952–1964, 2005.

[46] Abigail Morrison, Markus Diesmann, and Wulfram Gerstner. Phenomenological models of synaptic plasticity based on spike timing. *Biological cybernetics*, 98(6): 459–478, 2008.

[47] James C Nearing et al. *Mathematical Tools for Physics*. Dover Publications, 2003.

[48] Steven A Orszag and CM Bender. Advanced mathematical methods for scientists and engineers. *Mac Graw Hill*, 1978.

[49] Valery Iustinovich Oseledec. A multiplicative ergodic theorem. lyapunov characteristic numbers for dynamical systems. *Trans. Moscow Math. Soc*, 19(2):197–231, 1968.

[50] Louis M Pecora and Thomas L Carroll. Synchronization in chaotic systems. *Physical review letters*, 64(8):821, 1990.

[51] Louis M Pecora, Linda Moniz, Jonathan Nichols, and Thomas L Carroll. A unified approach to attractor reconstruction. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 17(1):013110, 2007.

[52] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007. ISBN 0521880688, 9780521880688.

[53] OE Rossler. An equation for hyperchaos. *Physics Letters A*, 71(2):155–157, 1979.

[54] Jonathan E Rubin, Richard C Gerkin, Guo-Qiang Bi, and Carson C Chow. Calcium time course as a signal for spike-timing–dependent plasticity. *Journal of neurophysiology*, 93(5):2600–2613, 2005.

[55] Marco Sandri. Numerical calculation of lyapunov exponents. *Mathematica Journal*, 6(3):78–84, 1996.

[56] Simo Särkkä. *Bayesian filtering and smoothing*. Cambridge University Press, 2013.

[57] Paul So, Edward Ott, and WP Dayawansa. Observing chaos: Deducing and tracking the state of a chaotic system from limited observation. *Physical Review E*, 49(4):2650, 1994.

[58] Francesco Sorrentino and Edward Ott. Using synchronization of chaos to identify the dynamics of unknown systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 19(3):033108, 2009.

[59] Robert W Stackman, Rebecca S Hammond, Eftihia Linardatos, Aaron Gerlach, James Maylie, John P Adelman, and Thanos Tzounopoulos. Small conductance ca2+-activated k+ channels modulate synaptic plasticity and memory encoding. *The Journal of neuroscience*, 22(23):10163–10171, 2002.

[60] Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. Westview press, 2014.

[61] Floris Takens. *Detecting strange attractors in turbulence*. Springer, 1981.

[62] D Terman, JE Rubin, AC Yew, and CJ Wilson. Activity patterns in a model for the subthalamopallidal network of the basal ganglia. *The Journal of neuroscience*, 22(7):2963–2976, 2002.

[63] Andre Nikolaevich Tikhonov, AV Goncharsky, VV Stepanov, and Anatoly G Yagola. *Numerical methods for the solution of ill-posed problems*, volume 328. Springer Science & Business Media, 2013.

[64] Nicolaas Godfried Van Kampen. *Stochastic processes in physics and chemistry*, volume 1. Elsevier, 1992.

[65] Andreas Wächter. *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*. PhD thesis, Carnegie Mellon University, 2002.

[66] Xiao-Jing Wang, Yinghui Liu, Maria V Sanchez-Vives, and David A McCormick. Adaptation and temporal decorrelation by single neurons in the primary visual cortex. *Journal of neurophysiology*, 89(6):3279–3293, 2003.

[67] Shao-Nian Yang, Yun-Gui Tang, and Robert S Zucker. Selective induction of ltp and ltd by postsynaptic [ca2+] i elevation. *Journal of neurophysiology*, 81(2): 781–787, 1999.

[68] Jingxin Ye, Paul J Rozdeba, Uriel I Morone, Arij Daou, and Henry DI Abarbanel. Estimating the biophysical properties of neurons with intracellular calcium dynamics. *Physical Review E*, 89(6):062714, 2014.

[69] Jingxin Ye, Daniel Rey, Nirag Kadakia, Michael Eldridge, Uriel I Morone, Paul Rozdeba, Henry DI Abarbanel, and John C Quinn. Systematic variational method for statistical nonlinear state and parameter estimation. *Physical Review E*, 92 (5):052901, 2015.

[70] J. Zinn-Justin. Quantum Field Theory and Critical Phenomena, 2002. URL https://hal.archives-ouvertes.fr/hal-00120423. International Series of Monographs on Physics 113, 1054 pp. (2002), Fourth Edition.

[71] Robert S Zucker. Calcium-and activity-dependent synaptic plasticity. *Current opinion in neurobiology*, 9(3):305–313, 1999.