UNIVERSITY OF CALIFORNIA SAN DIEGO

Nonlinear Inference in Partially Observed Physical Systems and Deep Neural Networks

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

in

Physics

by

Paul J. Rozdeba

Committee in charge:

Professor Henry D. I. Abarbanel, Chair
Professor Jeffrey L. Elman
Professor Michael J. Holst
Professor John A. McGreevy
Professor Clifford M. Surko

2018

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

Chair

University of California, San Diego

2018

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to thank my advisor Professor Henry D. I. Abarbanel for his support and guidance throughout my years as a graduate student. Henry's unending and ever-increasing energy and enthusiasm for science has been a great source of inspiration to me. I am grateful for the freedom he offered me to explore problems that I found interesting and, ultimately, the trust that he placed in me as a student and a scientist.

I would also like to thank Professor Alexandre Chorin for being my mentor while at Lawrence Berkeley National Laboratory, encouraging me to explore data assimilation from a new perspective, laying the foundation for a new research direction presented in Chapter 4. He and his postdoc at the time, now Professor Fei Lu, provided me with invaluable guidance and mathematical know-how (thanks Fei!).

I thank the members of my dissertation committee, Professors Jeffrey L. Elman, Michael J. Holst, John A. McGreevy, and Clifford M. Surko for dedicating their valuable time and patience.

Thanks to my colleagues in our research group, especially to Nirag for all the enlightening conversations and feedback, to Dan Rey for showing me what it is to be a great, dedicated scientist, to Mike and Uri for riding it all out with me, and to Sasha and Zheng for all their input to the machine learning project.

Finally, thanks to my parents for all their love and support through the years, teaching me the values of passion and dedication to whatever I do, to my sisters who are true friends and were always there to support me, to my friends in San Diego and elsewhere, and my friends back home who always made me feel like I'd never left.

Chapter 5, in part, has been adapted from the material submitted for publication as it may appear in H. D. I. Abarbanel, P. J. Rozdeba, and S. Shirman, *Machine Learning; Deepest Learning as Statistical Data Assimilation Problems*, Neural

Computation. I was a co-author for this publication.

VITA

2010	Bachelor of Arts, New York University
2010-2011	Junior Research Scientist, New York University CCPP
2011-2017	Teaching Assistant, University of California, San Diego
2013-2017	Graduate Student Researcher, University of California, San Diego
2015	U.S. DOE Office of Science Graduate Student Researcher (SCGSR), Lawrence Berkeley National Laboratory
2018	Doctor of Philosophy, University of California, San Diego

PUBLICATIONS

H. D. I. Abarbanel, P. J. Rozdeba, and S. Shirman. *Machine Learning, Deepest Learning: Statistical Data Assimilation Problems.* Publication under review, preprint posted on arXiv:1707.01415v1 [cs.AI] (2017).

J. Ye, N. Kadakia, P. J. Rozdeba, H. D. I. Abarbanel, and J. C. Quinn. *Improved variational methods in statistical data assimilation.* Nonlin. Processes Geophys. **22**, 205-213 (2015).

J. Ye, D. Rey, N. Kadakia, M. Eldridge, U. I. Morone, P. J. Rozdeba, H. D. I. Abarbanel, and J. C. Quinn. *Systematic variational method for statistical nonlinear state and parameter estimation.* Phys. Rev. E **92**, 052901 (2015).

J. Ye, P. J. Rozdeba, and H. D. I. Abarbanel. *Estimating the biophysical properties of neurons with intracellular calcium dynamics.* Phys. Rev. E **89**, 062714 (2014).

ABSTRACT OF THE DISSERTATION

Nonlinear Inference in Partially Observed Physical Systems and Deep Neural Networks

by

Paul J. Rozdeba

Doctor of Philosophy in Physics

University of California, San Diego, 2018

Professor Henry D. I. Abarbanel, Chair

The problem of model state and parameter estimation is a significant challenge in nonlinear systems. Due to practical considerations of experimental design, it is often the case that physical systems are partially observed, meaning that data is only available for a subset of the degrees of freedom required to fully model the observed systems behaviors and, ultimately, predict future observations. Estimation in this context is highly complicated by the presence of chaos, stochasticity, and measurement noise in dynamical systems.

One of the aims of this dissertation is to simultaneously analyze state and parameter estimation in as a regularized inverse problem, where the introduction of a model makes it possible to reverse the forward problem of partial, noisy observation; and as a statistical inference problem using data assimilation to transfer information from measurements to the model states and parameters. Ultimately these two formulations achieve the same goal. Similar aspects that appear in both are highlighted as a means for better understanding the structure of the nonlinear inference problem.

An alternative approach to data assimilation that uses model reduction is then examined as a way to eliminate unresolved nonlinear gating variables from neuron models. In this formulation, only measured variables enter into the model, and the resulting errors are themselves modeled by nonlinear stochastic processes with memory.

Finally, variational annealing, a data assimilation method previously applied to dynamical systems, is introduced as a potentially useful tool for understanding deep neural network training in machine learning by exploiting similarities between the two problems.

Chapter 1

Introduction

1.1 The Inference Problem

As computational capabilities and the availability of large data sets has grown over the past several decades, the problem of state and parameter inference for constructing predictive models of time-evolving systems using data has become ubiquitous in many fields of science. Many of these methods, falling under the umbrella of data assimilation [2, 1, 43], have their roots in numerical weather prediction [22, 71]; other examples include applications in fields such as geophysics and climate modeling [10, 44, 20], biology [81, 48, 59, 36, 54] and astrophysics [4].

In the context of dynamical systems under which this dissertation falls, the backdrop to the inference problem is a time-evolving system which is under observation, and for which data about these observations may be collected. A model of the system is chosen to reflect whatever knowledge of intuition the experimenter has about the system's properties. For example, in neurobiology it is common to conduct experiments on single neurons where they are stimulated by a known input electrical current, and the response of the cell is observed by measuring the electric potential across its membrane over time. In this case, there is a well-established point neuron model called Hodgkin-Huxley [40] which models a neuron's membrane voltage as a set of ordinary differential equations (ODEs), that includes variables representing the opening and closing action of "gating variables" that allow for the flow of charged ions into and out of the cell. This model contains four variables evolving in time, called states, and 18 static, adjustable values called parameters. The goal is to use inference methods to use recordings of the cell's membrane potential to estimate parameter values, as well as the state of *all* of the variables in the model, to then predict the cell voltage further in time.

The Hodgkin-Huxley model is an example of a nonlinear system. These nonlinearities complicate the inference task significantly [2, 1], and various methods have been developed specifically to deal with these problems [2, 41, 15]. The problem is essentially that there may be many sets of parameter values and model states which look nearly equivalent or valid, given the observed data and the model, but many of which may actually perform quite poorly when the model is eventually used to make predictions. The situation is even more complex in chaotic systems,, which are nonlinear systems that display extreme sensitivity to errors in initial conditions, so that small errors in the estimated states and parameters might lead to drastically worsened predictions. In fact, it will be shown in Chapter 3 that even the parameterization of the inference method itself must be carefully chosen to maximize the model's predictive capabilities.

It is possible, in fact very likely, that the model chosen for the system will be wrong, in the sense that it does not capture every single one of its possible properties or behaviors. Dealing with model error is a significant challenge in data assimilation. In climate systems, for example, errors arise due to unresolved dynamics resulting from finite temporal and spatial resolution of observation and modeling, often due to practical limitations of measurement capabilities. One approach to solving the model error problem is presented in Chapter 4, in which the model error *itself* is given a model, with the goal of recovering the full range of behaviors observed in the true system, but not in the model.

Another broad class of inference problems fall under machine learning [31, 53, 67], many of which have gained significant traction in recent years, especially the field of "deep learning" which has, as the reader may know, become extremely popular in industrial applications. While this may seem as a bit of a non-sequitur because we have been discussing inference for dynamical systems, the final chapter presents recent work which shows that the two problems share an equivalence. This equivalence is exploited to show that a method of variational data assimilation may prove useful to the neural network training problem.

This chapter will establish some of the basic concepts present in every data assimilation problem, and introduce inference both as an inverse problem as well as a statistical problem, although these two approaches are related. The variational method for data assimilation used widely through the dissertation, which is a path integral-based approach, is then defined at the end of the chapter.

1.1.1 States, Measurements, and Models

There are several important pieces to the problem of building and using predictive models for physical systems which form the central theme of this dissertation. First, all physical systems have a *state*, which for our purposes will usually be described by a scalar quantity x, or a vector $\boldsymbol{x} \in \mathbb{R}^D$, continuously parametrized by another quantitiy s or set of quantities $s_1, s_2, \ldots$ (henceforth generically referred to as s for a single parameter *or* set of parameters). For example,

$$\boldsymbol{x}(t) = (q_1(t), q_2(t), q_3(t), p_1(t), p_2(t), p_3(t))$$

where $q_i(t)$ is a position in the ith spatial coordinate, $p_i(t)$ is the corresponding momentum, and the parameter t is time, could be used to describe the motion of a ballistic object like a rocket being tracked as it flies through the high atmosphere, or in low orbit. This kind of state is of particular interest to the rest of this discussion as we will primarily be examining the problems of measurement and inference in physical dynamical systems (where all that needs to be said, for now, is that a dynamical system is a physical system which evolves in time). Eventually, we will see that this sort of construction is also incredibly powerful in describing classification and prediction machines in the context of machine learning, drawing a correspondence between time and layers in neural network models.

Another kind of state, which has multiple parameters s and is called a *field*, is useful for describing the configuration of extended physical objects which are also changing in time:

$$\boldsymbol{\varphi}\left(t, \boldsymbol{x}\right) = \left(\varphi_1(t, \boldsymbol{x}), \varphi_2(t, \boldsymbol{x}), \ldots\right).$$

The individual field components might represent other properties of the system besides position or momentum, and in fact position itself, $\boldsymbol{x} = (x_1, x_2, x_3)$, is a parameter for the field in addition to time. In a geophysical system these field components could be fluid velocity, temperature, pressure, or any other number of important or relevant physical quantities. In an example presented later in Chapter 2, a scalar field variable $\sigma(\boldsymbol{x})$ will be used to represent the distribution of electric charge on a three-dimensional surface.

Another important concept is that information about a physical state is gathered from *measurements* through the act of observation. Measurements contain information about a system that are “agnostic” on some level: the measuring device interacts with the measured system some way, ultimately resulting in a number displayed on the device’s readout. This viewpoint is not particularly useful for making predictions if the measurements do not reflect the entire state of the system, especially if the system is governed by nonlinear dynamics.

The third ingredient to the inference problem is thus a *model*, which is required in order to interpret measurements as a dynamical variable which interacts with other quantities in the system. Data assimilation extracts the information from measurements and translates them into state and parameter values so that, eventually, one can predict new observations of the system using the model.

1.2 Statistical Data Assimilation

Data assimilation is the process by which data, consisting of a time series of observations of a physical system, is presented to a model in order to infer the true values of the system’s state and parameters. Generally speaking, the goal is to use these estimates to make predictions about the system in question. Data assimilation is often necessary for many reasons, including:

- The observations are noisy, so the current state of the system is uncertain,
- There are fewer measured variables than model variables,
- The model parameter values are uncertain,

among others.

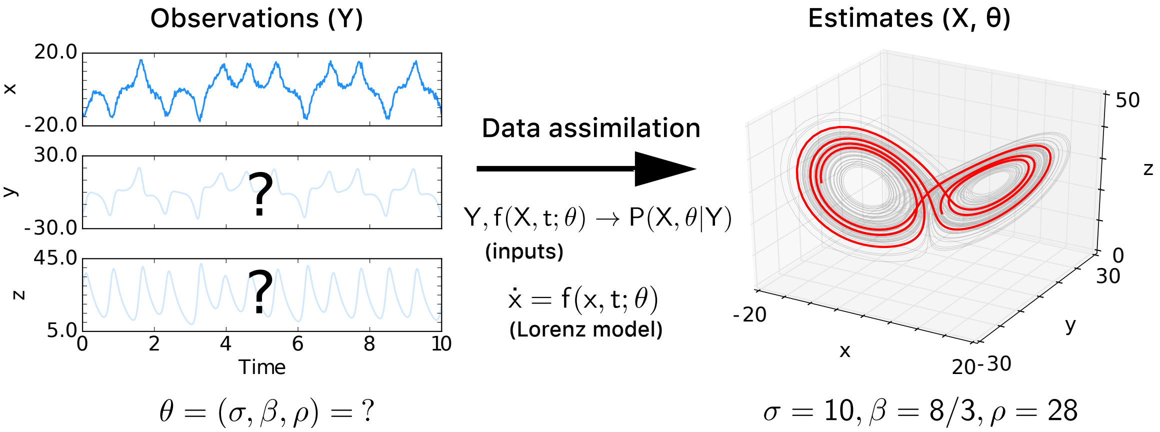


Figure 1.1: Sketch of a data assimilation problem. Left: The observations (shown in blue) comprise a 1-dimensional noisy time series $Y = \{\boldsymbol{y}^1, \ldots, \boldsymbol{y}^N\}$. These are assumed to be the x variable of the Lorenz system (so $\boldsymbol{y}^n = x(t^n)$), which is a 3-dimensional ODE with state variables x, y, and z, and 3 parameters σ, β, and ρ. The parameter values are unknown. Center: Data assimilation feeds the observations into the model, generating a probability distribution $P(X, \theta|Y)$, conditioned on the observations, for model states and parameters. Right: The estimates include a time series $X = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N\}$ (shown in red), and parameters $\boldsymbol{\theta}$, where $\boldsymbol{x}^n = (x(t^n), y(t^n), z(t^n))$ and $\boldsymbol{\theta} = (\sigma, \beta, \rho)$.

A sketch of a data assimilation problem is shown in fig. 1.1. In this example, incoming data is a stream of noisy, 1-dimensional observations which are discrete in time. The goal is to make accurate predictions of future observations of this same quantity. The observed system is modeled using a *dynamical system*, which are ODE models that describe the time evolution of a point in phase space - the vector space in which the state of the observed system is defined. The state of the system is described by the D-dimensional vector $\boldsymbol{x}(t)$, which evolves in time according to the model of the system:

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}, t; \boldsymbol{\theta}), \quad \boldsymbol{x} \in \mathbb{R}^D \quad \text{(dynamical system)}. \tag{1.1}$$

In this equation, $\boldsymbol{\theta}$ is a collection of static parameters for the model. These parameters control the types of behavior exhibited by the model. In nonlinear systems, these behaviors may change rapidly as $\boldsymbol{\theta}$ is altered and the system undergoes a bifurcation. In Chapter 5, it is shown that neural networks in machine learning share a similar enough structure to dynamical systems that the data assimilation procedure presented here carries over with only slight modifications.

Note that this is a vector equation for every state x_i in the system, so each variable has its own dynamical model f_i These are functions of the entire state vector $\boldsymbol{x}$, however, so that in general the variables are coupled and they interact with each other. This is how information is transferred from measured to unmeasured states in data assimilation, although there are limitations to this information transfer when the number of observed variables shrinks.

In order for data assimilation to work, it is assumed that the data Y is actually a time series of observations of trajectories described by the model. The sketch in figure 1.1 is a specific example of a case where the data consists of measurements of the x variable in the Lorenz model [56]:

$$\frac{dx}{dt} = \sigma\left(y - x\right), \quad \frac{dy}{dt} = x\left(\rho - z\right) - y, \quad \frac{dz}{dt} = xy - \beta z. \tag{1.2}$$

The model is an *assumption* in the sense that we don't actually know, a priori, what the other variables in the physical system are, because they are not observed.[1] Thus, the model must be chosen based on some knowledge or intuition about the system in question. For example, if one sits and observes wave heights near a coastline for some time, then choosing a geophysical fluid model to infer the state of the ocean at some distance from the coastline is a reasonable assumption to make. On the contrary, without a model, there is almost no reason to suspect that the other two states y

[1] In this case, the data was actually generated by sampling a solution to the Lorenz equations with a known initial state and parameter values. This setup is what will later be defined as a *twin experiment*, which is a powerful tool for testing data assimilation procedures.

and z even exist! Without the constraint imposed by choosing a model that links x, y, and z together, it is equally reasonable to believe almost any trajectory that one could imagine for the unobserved states. One could do equally poorly by choosing a *bad* model for the system, however, where enforcing it too strongly might actually be detrimental to inferring the true states of even the observed variables. From this perspective, data assimilation may be alternatively thought of as an inverse process to the act of measurement, where the model is introduced to regularize the inversion. These competing interests are at the heart of data assimilation; dealing with them in a quantitative way is part of the scope of this dissertation.

Ultimately, the purpose of selecting a model in the first place is to learn enough about the observed system to make predictions about its future. In the Lorenz example, one must know the state of *all three* variables x, y, and z, as well as all of the parameter values σ, β, and ρ, in order to provide an initial condition for forward integration of the system. Guessing any of these values is problematic because the Lorenz model is chaotic, so that any errors in these guesses will grow rapidly in time.

This is where data assimilation comes into play: the goal is to transfer the information available in the x observations (the data) into the model to *infer* the values of y, z, σ, β, and ρ. Using the data assimilation approach that will be presented later, the data shown in fig. 1.1 is sufficient to make this inference; the end result is the 3-dimensional trajectory shown in red, in addition to the inferred parameter values. These two go hand-in-hand, however. With a poor guess for the parameters, the true trajectory may not make any sense, and *vice versa*. Ultimately, the measure of success in the real world is the accuracy of *predictions* made using these estimates, because the true state and parameter values are never directly accessible for comparison with the inferred estimates.

Ultimately data assimilation is a statistical procedure. The measurements contain noise, which blur the information about the true states of measured variables; the number of measurements may be sufficiently low that many estimates may appear equally valid as model solutions or parameter values; and the model itself may be uncertain because of noise or, worse, missing pieces of the ODE model or unmodeled variables.

In more specific terms, data comes in the form of a time series Y, which is a "path vector" consisting of N L-dimensional discrete observations in time:

$$Y = \left\{ \boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^N \right\}, \quad \boldsymbol{y}^n \in \mathbb{R}^L \tag{1.3}$$

where $\boldsymbol{y}^n$ is shorthand for the vector $(y_1(t_n), y_2(t_n), \ldots, y_L(t_n))$. The observations are separated by a constant interval Δt, so $t_n = t_1 + (n-1)\Delta t$. The time series which we assume to be indirectly observing, called X, is also a path vector but composed of higher-dimensional vectors at each time if the observations are assumed to be partial:

$$X = \left\{ \boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N \right\}, \quad \boldsymbol{x}^n \in \mathbb{R}^D \quad (D \geq L). \tag{1.4}$$

Data assimilation is still useful if $D = L$: the true state of the system is not exactly known because observation noise still induces uncertainty in the state values. Note that this definition of the path variable X will usually be meant to include the model parameters:

$$X = \left\{ \boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N, \boldsymbol{\theta} \right\}, \quad \boldsymbol{x}^n \in \mathbb{R}^D \quad (D \geq L), \quad \boldsymbol{\theta} \in \mathbb{R}^{D_\theta} \tag{1.5}$$

where D_θ is the number of parameters of the model. Beyond considerations of notational convenience, inclusion of the model parameters into the path variable is justified by the viewpoint that static parameters are also variables of the system, but which obey the dynamical equations:

$$\frac{d\boldsymbol{\theta}}{dt} = 0. \tag{1.6}$$

It should be noted that the more general case for data assimilation is that one is observing a field ϕ that is a function of space and time governed by the partial differential equation (PDE)

$$\frac{\partial \phi_\mu(t, \boldsymbol{r})}{\partial t} = \mathcal{F}_\mu(\phi, t, \boldsymbol{r}) \tag{1.7}$$

where μ is an index for the field components. $\mathcal{F}$ contains functions and spatial derivatives of ϕ, as well as time t and spatial position $\boldsymbol{r}$ if the equations are inhomogeneous. From a practical perspective, data assimilation is usually carried out numerically on a computer rather than analytically. In this case it is natural to discretize the field equation, for example using a finite difference method:

$$\phi_\mu(t, \boldsymbol{r}) \quad \rightarrow \quad \phi_\mu(t, i\Delta r_1, j\Delta r_2, k\Delta r_3) \tag{1.8}$$

where i, j, and k are indices that label nodes at discrete spatial locations separated by Δr_1, Δr_2, and Δr_3. The dynamics are now defined for a set of discrete variables $\boldsymbol{x}$, governed by a set of ordinary differential equations (ODEs) in time:

$$\frac{dx_i}{dt} = f_i(\boldsymbol{x}, t; \boldsymbol{\theta}), \quad (i = 1, \ldots, D). \tag{1.9}$$

$\boldsymbol{x}$ is the vector of D model states which evolve in time, and $\boldsymbol{\theta}$ are the D_θ static parameters which should, in principle, be derived from the form of $\mathcal{F}$. These equations also inherit the explicit time-dependence of $\mathcal{F}$ as nonautonomous terms, which physically might be due to an external forcing of the system.

Further discretizing the equations in time, one is left with a set of difference equations $\boldsymbol{g}$ of the form

$$\begin{aligned}\boldsymbol{g}(\boldsymbol{x}^n, \boldsymbol{x}^{n+1}, t^n, t^{n+1}; \boldsymbol{\theta}) &= \boldsymbol{x}^{n+1} - \boldsymbol{x}^n - \int\limits_{t^n}^{t^{n+1}} dt\, \boldsymbol{f}(\boldsymbol{x}, t; \boldsymbol{\theta}) \\ &\equiv \boldsymbol{x}^{n+1} - \boldsymbol{F}(\boldsymbol{x}^n, \boldsymbol{x}^{n+1}, t_n, t_{n+1}, \boldsymbol{\theta}). \end{aligned} \qquad (1.10)$$

In practice, $\boldsymbol{F}$ is an approximation to the time integral using some numerical scheme, for example with a Runge-Kutta method [68, 77, 51]. This is generally a necessity because there may not be a closed form for $\boldsymbol{F}$ if $\boldsymbol{f}$ is nonlinear. Additionally, if the scheme is explicit then $\boldsymbol{F}$ is only a function of quantities at time t_n; later it will be shown that the structure of the variational data assimilation method we use here can be exploited to use an implicit discretization scheme in a very natural way.

It was already established that a model for the time evolution of the system is required for inference, but there is another model implicitly present, which is the measurement function $\boldsymbol{h}$ that connects the observations to the states. In general, this function is nonlinear and stochastic:

$$\boldsymbol{y}(t_n) = \boldsymbol{h}[\boldsymbol{x}(t_n), t_n] + \boldsymbol{g}[\boldsymbol{x}(t_n), t_n] \cdot \boldsymbol{\xi}^n, \quad \boldsymbol{h} : \mathbb{R}^D \to \mathbb{R}^L \qquad (1.11)$$

Here, the measurement noise is represented by $\boldsymbol{\xi}^n$ is a set of L random variables at every observation time t_n. The form for the measurement function given in eq. 1.11 is quite general; for the duration of this dissertation, however, it will be assumed that $\boldsymbol{h}$ is linear and independent of time, and that the measurement errors are additive, so that

$$\boldsymbol{y}(t_n) = \mathbf{H}\boldsymbol{x}(t_n) + \boldsymbol{\xi}^n \tag{1.12}$$

where $\xi_i^n \sim N(0, \sigma_i^2)$ is a Gaussian measurement noise term, just one possible example of noise. In this formula, $\mathbf{H}$ is just an $L \times D$ matrix, which is defined for the duration of this dissertation to be a *direct observation*.[2] A direct observation is one for which which $x_\ell^n = y_\ell^n$; for example, in a 3-dimensional system in which the variables x_1 and x_3 are observed:

$$\boldsymbol{y}^n = \boldsymbol{h}(\boldsymbol{x}^n) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1^n \\ x_2^n \\ x_3^n \end{pmatrix} + \begin{pmatrix} \xi_1^n \\ \xi_3^n \end{pmatrix} \equiv \mathbf{H}\boldsymbol{x}^n + \boldsymbol{\xi}^n \tag{1.13}$$

Estimating the full state of the system and its parameter values, X, is in this sense an inverse problem of finding the inverse to the measurement function $\boldsymbol{h}$. However, inverting $\boldsymbol{h}$ is in general an ill-posed problem; for a direct observation it essentially amounts to inverting a projection matrix. This is equivalent to the earlier idea that without a model, anything is believable if it isn't measured.

This inverse problem, however, can be "regularized" with the introduction of a model for the system; this idea is further explored in Chapter 2 in the context of linear systems. Choosing a model provides the extra information needed to carry out this inversion, where this information comes in the form of coupled dynamics between all of the desired state variables. Chapter 2 introduces some ideas about how this inversion is affected by introducing the model. The next section will show how data is transferred to the model in a stochastic process, leading to a variational form of data

[2] Including amplitudes $A_\ell \neq 1$ is more general, but these amplitudes can be absorbed into the dynamical equations. Thus, they are assumed to be equal to one throughout.

assimilation that will mostly be used in Chapter 3 and beyond.

1.2.1 Data Assimilation as a Stochastic Process

The approach to data assimilation presented here is founded on the notion that it is an inherently statistical problem: the values of the model states $\boldsymbol{x}^n$ are unknown, or known only up to a probability distribution, and the same may be said for the model parameter values, $\boldsymbol{\theta}$. Noise in the observations induces additional uncertainty in these estimates. Given a data stream Y, it is thus most appropriate to track the time evolution of *distributions* of states rather than any individual trajectory. If the parameters are unknown, they should also be estimated up to a probability distribution.

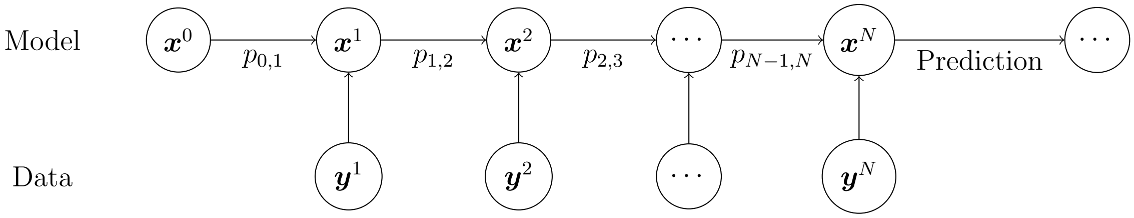


Figure 1.2: A sketch of the statistical data assimilation process. The transition probabilities from time t^n to t^{n+1}, $p_{n,n+1}$, functionally contain the dynamical equations $\boldsymbol{f}$. At each time the state $\boldsymbol{x}^n$ is informed by the observation $\boldsymbol{y}^n$. At the end of the estimation window, a state is drawn from the distribution $P(\boldsymbol{x}^N|\boldsymbol{x}^{N-1},\ldots,\boldsymbol{x}^0,\boldsymbol{y}^N,\ldots,\boldsymbol{y}^1)$ and used for forward prediction.

The form of this distribution is the result of modeling data assimilation as a statistical process like the one shown in fig. 1.2. In this model, one considers the process of time evolution of the model state $\boldsymbol{x}$ as a stochastic process with transition probabilities $p_{n,n+1}$, which is the probability of state $\boldsymbol{x}^{n+1}$ occurring given past values of $\boldsymbol{x}$. The probability of state $\boldsymbol{x}^{n+1}$ is additionally influenced by an observation at that time, $\boldsymbol{y}^{n+1}$. Thus, it is appropriate to formulate data assimilation in terms of a conditional probability distribution:

$$P(\boldsymbol{x}^n|\boldsymbol{x}^{n-1},\ldots,\boldsymbol{x}^1,\boldsymbol{y}^n,\ldots,\boldsymbol{y}^1)\equiv P(\boldsymbol{x}^n|X^{n-1},Y^n) \tag{1.14}$$

which is the probability distribution for the value of the current state, $\boldsymbol{x}^n$, conditioned upon the states at all previous times, and all of the past measurements up to the current time.

Alternatively, one might work with the *joint* distribution for the estimate of the states and parameters over the entire trajectory, conditioned on an entire trajectory of data; this is of the form

$$P(X,\boldsymbol{\theta}|Y)=\boldsymbol{\mathcal{L}}(Y|X)\Pi(X,\boldsymbol{\theta}) \tag{1.15}$$

where the likelihood, $\boldsymbol{\mathcal{L}}$, is where information from the data enters into P; and the prior, Π, contains the dynamical equations and is where the probability of the *full* model state and parameters comes into play. Whereas filtering methods tend to work sequentially in time with the conditional probability (1.14), variational methods consider the joint probability (1.15). In nonlinear and chaotic systems, filtering methods [22, 43, 41] tend to suffer from instabilities that accumulate as the sequential data assimilation proceeds in time, whereas variational methods are more robust in the face of these instabilities. The trade-off is that sequential methods are much cheaper from a computational cost perspective. This section will focus on deriving the joint distribution from the sequential one, because variational methods will be the tool of choice for most of the numerical experiments considered in this dissertation.

It is perhaps simpler to start with the distribution for $\boldsymbol{x}^n$ conditioned only on previous measurements, Y^n (the path of measurements up to time n):

$$\begin{aligned}
P(\boldsymbol{x}^n|Y^n) &= \frac{P(\boldsymbol{x}^n, Y^n)}{P(Y^n)} \\
&= \frac{P(\boldsymbol{x}^n, \boldsymbol{y}^n, Y^{n-1})}{P(\boldsymbol{y}^n, Y^{n-1})} \\
&= \frac{P(\boldsymbol{x}^n, \boldsymbol{y}^n|Y^{n-1})P(Y^{n-1})}{P(\boldsymbol{y}^n|Y^{n-1})P(Y^{n-1})} \\
&= \frac{P(\boldsymbol{x}^n, \boldsymbol{y}^n|Y^{n-1})}{P(\boldsymbol{y}^n|Y^{n-1})} \frac{P(\boldsymbol{x}^n|Y^{n-1})}{P(\boldsymbol{x}^n|Y^{n-1})} \\
&\equiv \exp\left[\mathrm{CMI}(\boldsymbol{x}^n, \boldsymbol{y}^n|Y^{n-1})\right] P(\boldsymbol{x}^n|Y^{n-1}). \qquad (1.16)
\end{aligned}$$

It is useful to stop at this point to examine the distribution in terms of the CMI, or conditional mutual information, between $\boldsymbol{x}^n$ and $\boldsymbol{y}^n$ conditioned on all the previous measurements. This is a measure of how independent $\boldsymbol{x}^n$ and $\boldsymbol{y}^n$ are in the conditional sense. If they are in fact statistically independent, then the CMI goes to zero and $P(\boldsymbol{x}^n|Y^n) = P(\boldsymbol{x}^n|Y^{n-1})$ trivially.

Continuing with the derivation, the second term in the last line of (1.16) is rewritten as a marginalized integral over the state at time $\boldsymbol{x}^{n-1}$:

$$\begin{aligned}
P(\boldsymbol{x}^n|Y^{n-1}) &= \int d\boldsymbol{x}^{n-1} P(\boldsymbol{x}^n|\boldsymbol{x}^{n-1}, Y^{n-1}) P(\boldsymbol{x}^{n-1}|Y^{n-1}) \\
&= \int d\boldsymbol{x}^{n-1} P(\boldsymbol{x}^n|\boldsymbol{x}^{n-1}) P(\boldsymbol{x}^{n-1}|Y^{n-1}) \qquad (1.17)
\end{aligned}$$

where the first line was simplified by assuming the Markov property for the model system. This is the correct assumption to make, because in most of the examples considered here, the model of the system is an ODE which is local in time. In Chapter 5, a multi-layer neural network model for machine learning is introduced where the layers play an equivalent role to time, and layers are similarly connected sequentially so that this Markov property holds there, too.

Substituting this identity into the second term in the last line of (1.16) yields

$$P(\boldsymbol{x}^n|Y^n) = e^{\mathrm{CMI}(\boldsymbol{x}^n,\boldsymbol{y}^n|Y^{n-1})} \int d\boldsymbol{x}^{n-1} P(\boldsymbol{x}^n|\boldsymbol{x}^{n-1}) P(\boldsymbol{x}^{n-1}|Y^{n-1}) \tag{1.18}$$

The last term in this integral, $P(\boldsymbol{x}^{n-1}|\boldsymbol{y}^{n-1})$, can be rewritten in terms of this identity in a recursive process until all time points in the observation window have been exhausted. This yields the expression

$$\begin{aligned} P(\boldsymbol{x}^n|Y^n) &= e^{\mathrm{CMI}(\boldsymbol{x}^n,\boldsymbol{y}^n|Y^{n-1})} \int \prod_{m=0}^{n-1} d\boldsymbol{x}^m e^{\mathrm{CMI}(\boldsymbol{x}^m,\boldsymbol{y}^m|Y^{m-1})} P(\boldsymbol{x}^{m+1}|\boldsymbol{x}^m) P(\boldsymbol{x}^0|\boldsymbol{y}^0) \\ &\equiv \int \prod_{m=0}^{n-1} d\boldsymbol{x}^m e^{-A(X^n,Y^n)} \end{aligned} \tag{1.19}$$

where A is the desired action of variational data assimilation. Note that $P(\boldsymbol{x}^0|\boldsymbol{y}^0) = P(\boldsymbol{x}^0)$ if no measurement is made at $t = t_0$, which is just a prior distribution on the initial state of the system.

The data assimilation action is thus defined as follows:

$$A(X^n, Y^n) = -\sum_{m=0}^{n} \mathrm{CMI}(\boldsymbol{x}^m, \boldsymbol{y}^m|Y^{m-1}) - \sum_{m=0}^{n-1} \log P(\boldsymbol{x}^{n+1}|\boldsymbol{x}^n) - \log P(\boldsymbol{x}^0). \tag{1.20}$$

This expression is somewhat simplified if the measurements are made independent of each other in time, in which case $\mathrm{CMI}(\boldsymbol{x}^m, \boldsymbol{y}^m|Y^{m-1}) = -\log P(\boldsymbol{y}^n|\boldsymbol{x}^n)$. If the measurements are modeled by a measurement function with additive noise,

$$\boldsymbol{y}^n = \boldsymbol{h}(\boldsymbol{x}^n) + \boldsymbol{\xi}^n \Rightarrow P(\boldsymbol{y}^n|\boldsymbol{x}^n) = P_\xi(\boldsymbol{y}^n - \boldsymbol{x}^n) \tag{1.21}$$

which the distribution of the measurement noise. If this noise is Gaussian, then the final form for the CMI term is

$$\sum_{m=0}^{n}\sum_{l,k=1}^{L}\frac{(R_m^n)_{lk}}{2}\left(h_l(\boldsymbol{x}^n)-y_l^n\right)\left(h_k(\boldsymbol{x}^n)-y_l^n\right). \tag{1.22}$$

A Gaussian approximation is also made to simplify the model error. This is equivalent to assuming additive noise in the dynamics, in which case:

$$\boldsymbol{x}^n = \boldsymbol{F}(\boldsymbol{x}^{n-1}) + \boldsymbol{\eta}^n \tag{1.23}$$

which similarly allows the transition probability terms to be rewritten as

$$-\log P(\boldsymbol{x}^n|\boldsymbol{x}^{n-1}) = \sum_{m=0}^{n}\sum_{i,j=1}^{D}\frac{(R_f^n)_{ij}}{2}\left(x_i^{n+1}-F_i(\boldsymbol{x}^n)\right)\left(x_j^{n+1}-F_j(\boldsymbol{x}^n)\right). \tag{1.24}$$

This Gaussian approximation is also used throughout the dissertation as an approximation to the model error. In the limit that the observed system is deterministic, there is assumed to be no model error, which is recovered in the Gaussian approximation in the limit that $R_f \to \infty$. If there is model error, for example because the system contains noise and the dynamics are described by a Langevin equation [91, 92], or if the model is incomplete so that $\boldsymbol{f}$ does not fully capture the dynamics of the observed system, then the deterministic limit is not appropriate. However, in the path integral formulation presented in the next section, in which data assimilation is cast in terms of expectation values of functions $\Phi(X)$ acting on paths of model trajectories and parameters, it is sufficient that the ratio $R_f/R_m \gg 1$ to carry on with state and parameter estimation. Additionally, it is shown in Chapter 3 that even in the case where the model error is not Gaussian, it is possible to recover somewhat by carrying out estimation with carefully chosen values of the coefficient R_f.

1.2.2 Path Integral Formulation of DA

The action defined in the previous section allows one to compute expectation values of quantities in the path space using a path integral formulation, based on considerations in [27, 90]:

$$\mathrm{E}\left[\Phi(X)|Y\right] = \frac{\int DX \; \Phi(X) e^{-A(X,Y)}}{\int DX \; e^{-A(X,Y)}}. \tag{1.25}$$

These integrals represent the statistics of state and parameter estimates, which is essentially the goal of data assimilation from a statistical perspective. Note that this is not a "true" path integral, usually, because the states and measurements are defined at discrete times. When X an Y are defined on a discrete grid of N time points:

$$\int DX = \int \prod_{n=1}^{N} \prod_{i=1}^{D} dx_i^n \prod_{j=1}^{D_\theta} d\theta_j \tag{1.26}$$

so that this integral is actually taken in the $D \times N + D_\theta$ dimensional space of state trajectories and parameters. However, it is still potentially very difficult to evaluate. It is not particularly difficult to evaluate if the dynamical model, $\boldsymbol{f}$, is linear; essentially these integrals can just be done by hand, because the action is quadratic and thus the integral is simply a Gaussian integral. When $\boldsymbol{f}$ is nonlinear, on the other hand, A contains arbitrary non-quadratic terms and the integral is no longer Gaussian. In this case, the expectation values must be estimated somehow.

Several methods exist for doing this approximation, including Monte Carlo sampling [60, 39, 74]; here we focus on using the Laplace approximation [52, 5], which turns the approximation into an optimization problem. Under the Laplace approximation, integrals of a form such as (1.25) are dominated by the integral within small regions near minima of A.

To see why this is true, first consider the integral in the denominator of (1.25) for the Gaussian action, in which the measurement and model errors are assumed to be Gaussian and independent of each other in time:

$$\begin{aligned}&\int DX\, e^{-A(X,Y)}\\&=\int DX\ \exp\left[-\frac{R_m}{2}\sum_{n=1}^{N}\sum_{\ell=1}^{L}(x_\ell^n-y_\ell^n)^2-\frac{R_f}{2}\sum_{n=1}^{N-1}\sum_{i=1}^{D}\left(x_i^{n+1}-F_i(\boldsymbol{x}^n,\boldsymbol{\theta})\right)^2\right].\end{aligned} \tag{1.27}$$

In the limit that $R_f/R_m \to \infty$ (the limit of deterministic dynamics), this integral is dominated by contributions in small regions near minima of A, which are the peaks of $e^{-A(X,Y)}$. In Chapters 2 and 3, it is shown that this limit is detrimental to estimation even when the model of the observed system is known exactly. However, the values of R_f/R_m which are found to be optimal for minimizing path estimation error, as well as prediction error, are of the order of 100 or 1000. The approximation still holds validity in this regime, because contributions from the measurement error to the above integral are exponentially suppressed compared to contributions from the model error.

If A contains just a single minimum located at $X = X_0$, then

$$\begin{aligned}\int DX\, e^{-A(X,Y)} &\simeq \int DX\, e^{-A(X_0,Y)-\frac{1}{2}(X-X_0)^{\mathrm{T}}\mathbf{A}''(X_0,Y)(X-X_0)+\mathcal{O}(X^3)}\\&\simeq e^{-A(X_0,Y)}\int DX\, e^{-\frac{1}{2}(X-X_0)^{\mathrm{T}}\mathbf{A}''(X_0,Y)(X-X_0)}\end{aligned} \tag{1.28}$$

where $\mathbf{A}''(X_0,Y)$ is the Hessian of A evaluated at X_0.[3] This is now in the form of a Gaussian integral, which may be evaluated explicitly as:

$$\int DX\, e^{-A(X,Y)} \simeq e^{-A(X_0,Y)}\sqrt{\frac{(2\pi)^{ND+D_\theta}}{|\mathbf{A}''(X_0,Y)|}}. \tag{1.29}$$

[3] Note that because this expansion is evaluated near a minimum of A, $\boldsymbol{A}' = 0$ and thus the leading-order term is quadratic in $(X - X_0)$.

The numerator of (1.25) is similarly approximated; again, if $A(X,Y)$ contains just a single minimum in X, then

$$\begin{aligned}\int DX\;\Phi(X)\,e^{-A(X,Y)} &\simeq \int DX\;\Phi(X_0)\,e^{-A(X_0,Y)-\frac{1}{2}(X-X_0)^{\mathrm{T}}\mathbf{A}''(X_0,Y)(X-X_0)} \\ &= \Phi(X_0)\,e^{-A(X_0,Y)}\sqrt{\frac{(2\pi)^{ND+D_\theta}}{|\mathbf{A}''(X_0,Y)|}}. \qquad (1.30)\end{aligned}$$

In the case of nonlinear systems, however, it is likely that A has many local minima $\{X_0, X_1, \ldots, X_M\}$. The complex local minimum structure of A in chaotic systems is explored in detail in [42, 47, 2]. If these minima are well-separated in the path space, then the approximation that the integral is dominated by the peaks of $e^{-A(X,Y)}$ holds, and the above integral becomes

$$\int DX\;\Phi(X)\,e^{-A(X,Y)} \simeq \sum_{m=1}^{M}\Phi(X_m)\,e^{-A(X_m,Y)}\sqrt{\frac{(2\pi)^{ND+D_\theta}}{|\mathbf{A}''(X_m,Y)|}}. \qquad (1.31)$$

Combining the integrals of the numerator and denominator yields:

$$\mathrm{E}\left[\Phi(X)|Y\right] \simeq \frac{\sum_{m=1}^{M}\Phi(X_m)\,e^{-A(X_m,Y)}\sqrt{\frac{(2\pi)^{ND+D_\theta}}{|\mathbf{A}''(X_m,Y)|}}}{\sum_{m=1}^{M}\,e^{-A(X_m,Y)}\sqrt{\frac{(2\pi)^{ND+D_\theta}}{|\mathbf{A}''(X_m,Y)|}}}. \qquad (1.32)$$

When A has a single dominant minimum X_0, then the higher-action minima may be neglected in this approximation. Comparing the action for X_0 and the next-highest minimum X_1, the ratio $e^{-A(X_1,Y)}/e^{-A(X_0,Y)} = e^{-[A(X_1,Y)-A(X_0,Y)]} \ll 1$ when $A(X_1,Y) \gg A(X_0,Y)$. The approximation in (1.32) thus simplifies to

$$\mathrm{E}\left[\Phi(X)|Y\right] \simeq \frac{\Phi(X_0)\,e^{-A(X_0,Y)}\sqrt{\frac{(2\pi)^{ND+D_\theta}}{|\mathbf{A}''(X_0,Y)|}}}{e^{-A(X_0,Y)}\sqrt{\frac{(2\pi)^{ND+D_\theta}}{|\mathbf{A}''(X_0,Y)|}}} = \Phi(X_0). \qquad (1.33)$$

Data assimilation in the path integral formulation is thus equivalent to the problem of maximum *a posteriori* estimation, where conditional expectation values $\mathrm{E}\left[\Phi(X)|Y\right]$ are computed by finding the mode of A, X_0, and evaluating $\Phi(X_0)$. This is seen to be true in an example chaotic system in Chapter 3 when a sufficient number of model variables are observed.

On the contrary, when the number of variables is insufficient to produce a well-separated minimum, expectation values should be estimated using the full form of the approximation in (1.32). The evaluation of (1.32) is not difficult from a theoretical perspective, but may be computationally expensive. The time complexity of computing the determinant of an $n \times n$ matrix, using Gaussian elimination or the LU decomposition, is $\mathcal{O}(n^3)$ where $n = ND + D_\theta$ [68]. More efficient algorithms exist for determinant calculation with sparse matrices [18, 17]. The Hessian $\mathbf{A}''$ is, in fact, sparse because the off-diagonal derivatives $\partial^2 A/\partial x_i^n \partial x_j^m$ are only nonzero when $n = m$, $m - 1$, or $m + 1$; the remaining derivatives involving parameters, such as $\partial^2 A/\partial x_i^n \partial \theta_j$, are few in number in comparison, leaving most of the Hessian elements equal to zero.

In this dissertation, however, these expansions are not calculated explicitly in underobserved systems, for example in Chapter 3. Rather, the quality of estimates which are found to minimize A by searching the action surface, starting from a large number of initializations of the minimization procedure, are selected from a distribution of estimates found at various R_f/R_m values by assessing their ability to make accurate predictions. It is found that a careful choice of R_f/R_m can produce estimates which reduce the prediction error by an order of magnitude or more, albeit for relatively short times compared to the case where L is sufficiently large in comparison to D. As a future extension to this project, a more detailed analysis where the full Laplace approximation is used is warranted to augment these estimates.

1.3 Twin Experiments

Before moving on to the next chapter, it is worth describing the setting in which many of the numerical experiments in this dissertation are performed. A twin experiment is defined as a situation in which state and parameter estimation is done with synthetic data generated by a *known* model. In the case of a system modeled by ODEs, this amounts to integrating the (known) equations forward in time, and simulating measurements by sampling the solution at discrete time intervals, adding noise to each sample to simulate measurement error. This is, of course, not a particularly realistic situation, but it is frequently used to validate the inference method itself. This is because, if the result of inference is poor estimation or prediction, with a twin experiment the source of the data is at least exactly known. This allows one to study how the model $\boldsymbol{f}$, the particular data trajectory Y, and the way in which the estimation itself is carried out in a controlled fashion. The effects of changing $\boldsymbol{f}$, Y, or aspects of the method are all explored in one way or another in this dissertation.

Chapter 2

Inverse Problems and Dynamical Regularization

In Chapter 1, a statistical view of data assimilation was used to derive a discrete-time action A, serving as a cost function for state and parameter estimates given a time series of data and a model for the observed system. A path integral formulation was described in which the statistics of path estimates are computed using integrals of path-space functions over the conditional probability distribution $P(X|Y) \sim e^{-A(X,Y)}$ of state and parameter sets given a trajectory of observed data. The Laplace approximation is employed to compute these integrals, requiring the identification of minima of A; these are peaks of the conditional distribution $P(X|Y)$, making the Laplace approximation equivalent to maximum *a posteriori* estimation methods from a Bayesian perspective.

At a fundamental level, however, data assimilation amounts to the inversion of the measurement function $\boldsymbol{h}$: the system's state is to be computed from measurements. In partially observed systems, this inversion is an ill-posed problem because the system is underdetermined, requiring the construction of pseudoinverses to $\boldsymbol{h}$ that make the problem fully determined through the introduction of a model, which includes all of the variables and parameters whose trajectories or static values the experimenter wishes to infer. This chapter investigates the nature of this pseudoinverse construction through the lens of regularization of linear inverse problems.

First, some theory is developed on how singular value decompositions (SVDs) can be used to analyze regularization as a filtering problem. This will be followed by an example in which noisy voltage measurements are used to estimate a static charge distribution, establishing some general properties of regularization and choosing a model, as well as a methodology for evaluating the quality of an estimate that will become useful in Chapter 3. Finally, the concept of dynamical regularization is introduced in the context of state estimation with noisy observations of a linear oscillator. A spectral analysis of the pseudoinverse will provide insight into how the introduction of a model affects state estimation. These results are summarized, ready to be finally linked to the nonlinear problem examined in Chapter 3.

2.1 Measurement and Inference: Inverse Problems

The acts of observing a system, and inferring the state of the system from observations, are problems which are inverse to one another. Mathematically speaking, going back and forth between observations and the true state amounts to evaluating a measurement function and its inverse. The act of observation is commonly referred to as a *forward problem*, whereas the *inverse problem* is estimating the true state from observations. In a sense the forward problem is easy: mathematically it corresponds to applying a measurement function to the state of the system, which is just "plug-

and-chug" once $\boldsymbol{x}$ is known:

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}). \tag{2.1}$$

$\boldsymbol{y}$ is the data collected from measurements of the system in question, whereas $\boldsymbol{x}$ is the state of a *model* of the system. The function $\boldsymbol{h}$ is also a model, which models the process of measuring $\boldsymbol{x}$. It is usually referred to as the *measurement function* for the system. The ultimate goal is to invert $\boldsymbol{h}$ to *infer* $\boldsymbol{x}$ from $\boldsymbol{y}$; this is the inverse problem.

$\boldsymbol{h}$ is a function, or functional, of $\boldsymbol{x}$ which may be nonlinear and/or stochastic, introducing some difficulties into the problem of inferring the full state of partially observed systems that are elaborated upon throughout the dissertation. The true state of the system is represented by the variable $\boldsymbol{x}$, while $\boldsymbol{y}$ represents an observation of $\boldsymbol{x}$. These quantities are in bold to denote that they are in general vector quantities or functions, where $\boldsymbol{x} \in \mathbb{R}^D$, $\boldsymbol{y} \in \mathbb{R}^L$, and $\boldsymbol{h} : \mathbb{R}^D \to \mathbb{R}^L$. Note that L is not necessarily equal to D, corresponding to a *partial observation* of the system. The primary focus of this dissertation will be on problems in which $L < D$.

The essential difficulty of the inverse problem is that the only accessible information about the true state is that which is passed through $\boldsymbol{h}$. The problem may be underdetermined if $\boldsymbol{h}$ is rank-deficient, so that there is a large degree of degeneracy in the choice of the unobserved states; or it may be overdetermined in which case there is no solution to the inverse problem. Additionally, in the class of problems considered here where $\boldsymbol{h}$ acts to smooth features of $\boldsymbol{x}$, inverting $\boldsymbol{h}$ amplifies noise in the measurements or errors in the measurement model, so that estimates of $\boldsymbol{x}$ will contain artificial high-frequency components not present in the true system. Our task is thus to modify the inverse problem by *regularizing* it, introducing a model to provide the additional information from the data which is not immediately apparent from the observations themselves. The resulting estimates of $\boldsymbol{x}$ should be accurate and stable to perturbations in $\boldsymbol{y}$, and ultimately be useful for making high-quality predictions of *new* observations. In the case of dynamical systems, this goal is cast as finding estimates of the model state and parameters which predict accurately for as far in the future, beyond the end of the estimation window, as possible.

2.2 Regularization of Inverse Problems

Linear inverse problems [38] are those in which the corresponding forward problem, in this case the act of observation, is represented by a linear function(al) $\boldsymbol{h}(\boldsymbol{x})$. When $\boldsymbol{x}$ and $\boldsymbol{y}$ are functions of continuous variables, the linear forward problem is represented by an integral of the form:

$$\boldsymbol{y}(s) = \int ds'\, \mathbf{K}(s, s')\, \boldsymbol{x}(s') + \text{noise} = \boldsymbol{h}(\boldsymbol{x}) + \text{noise} \tag{2.2}$$

where the integration kernel $\mathbf{K}$ defines the measurement model. s and s' could themselves be vectors, for example if they represent spatial position. One example of a linear forward problem is the measurement of electric potential V at a point $\boldsymbol{r}$ in the region surrounding an electric charge distribution ρ:

$$V(\boldsymbol{r}) = \frac{1}{4\pi} \int\limits_{\text{source}} d^3\boldsymbol{r}'\, \frac{\rho(\boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|}. \tag{2.3}$$

This model is valid when the distribution is sufficiently isolated from other charges, as well as any conductors or polarizable materials. The inverse problem is to estimate ρ from observations of V. This example will be explored in detail in a subsequent section.

While the problem of designing and implementing a measurement device, as well as actually making accurate and precise observations, may be a significant experimental challenge, there is no theoretical complication in the sense that there is no ambiguity once the measurement function is chosen and implemented. The theoretical challenges arise in real system where either the observations have errors, so $g = h[f] + \eta$ where η is a discrepancy defined on the same range as g; or if K is rank-deficient, in which case there is not enough information present *a priori* to have a unique solution to the inverse problem.

It is often natural to discretize forward and inverse problems, so that rather than being continuous functions of the parameter(s) s, $\boldsymbol{x}$ and $\boldsymbol{y}$ exist on a grid and the integral in (2.2) turns into a matrix acting on X; here, the "path vector" notation established in Chapter 1 is used to denote the discretization of the vector $\boldsymbol{x}$ over the coordinate grid. This is common practice, because continuous problems must be discretized in some way to be solved on a computer. For physical or engineering problems, this discretization is commonly done using finite differencing or a finite element decomposition [68, 55].

In path space, the measurement function becomes a matrix acting on the true state of the system:

$$Y = HX + \xi \tag{2.4}$$

where ξ is the measurement noise for each element of $Y \in \mathbb{R}^M$ in path space, and the measurement function $H \in \mathbb{R}^M \times \mathbb{R}^N$ is, again, a matrix in path space. Here, M is the number of measurements and N is the number of true states of the system.

The inference problem is to find a solution to $HX = Y$, but when $M < N$ the problem is underdetermined and there is not necessarily a unique solution. The inverse problem is thus ill-posed for a partially observed system; the next step is to introduce a model to cure this ill-posedness, described in the next section as a *regularization* of the inversion.

2.2.1 Regularization as a Smoother

Consider the linear least-squares problem:

$$\boldsymbol{x}^{(\mathrm{est})} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \left(\mathbf{A}\boldsymbol{x} - \boldsymbol{y}\right)^2$$

$$\left(\boldsymbol{x} \in \mathbb{R}^N,\ \boldsymbol{y} \in \mathbb{R}^M,\ \mathbf{A} \in \mathbb{R}^M \times \mathbb{R}^N\right) \tag{2.5}$$

Here, $\boldsymbol{y}$ are the observed data points, and $\mathbf{A}\boldsymbol{x} = \boldsymbol{y}$ is a model of the observed system where $\boldsymbol{x}^{(\mathrm{est})}$ is a noise-free estimate of the "true" signal. This is the essence of many problems in science and engineering, for example signal reconstruction or image processing from noisy data [70, 30]. Similar challenges arise here [38] to the formulation of the inverse problem in terms of a matrix inversion when the minimization problem is ill-posed because $\mathbf{A}$ is rank-deficient or ill-conditioned, or there is noise in the data. This is not surprising, because these two formulations are actually equivalent in that the solutions are the same.

A commonly used method for dealing with this ill-posedness is Tikhonov regularization [79, 38], in which an extra term is introduced into the problem statement in eq. (2.5):

$$\boldsymbol{x}^{[\gamma]} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \left\{\left(\mathbf{A}\boldsymbol{x} - \boldsymbol{y}\right)^2 + \gamma\boldsymbol{x}^2\right\}. \tag{2.6}$$

This is usually referred to as the regularized linear least-squares problem, where $\boldsymbol{x}^{[\gamma]}$ is an estimate of the true signal that is now dependent on the *hyperparameter* γ. Introducing this regulating term has several effects: first, it acts as a low-pass filter by damping high-frequency components in the estimated signal, thus acting as a smoother on the estimated solutions. Second, it tempers the ill-conditioning of $\mathbf{A}$ by balancing its eigenvalue spectrum. Finally, regularization makes the inversion of $\mathbf{A}$ well-defined if it is rank-deficient, which means there is a unique solution for $\boldsymbol{x}^{[\gamma]}$. A more general form of Tikhonov regularization replaces $\gamma\boldsymbol{x}^2$ with $\mathbf{L}(\boldsymbol{x}-\boldsymbol{x}_0)$, where $\mathbf{L}$ is a linear operator (in the discretized case, it is a matrix). This more general form of regularization will be used later in analyzing a linear dynamical system, but for the moment the simpler form in (2.6) is kept to examine the filtering properties of the regularization term.

It is instructive [38] to write solutions to the regularized problem in terms of the SVD of $\mathbf{A}$, which leads to an expression for solutions $\boldsymbol{x}^{[\gamma]}$ that highlight the smoothing properties of regularization. To begin, the least-squares problem is recast as a matrix problem. Rewriting (2.6) as a matrix equation,

$$\boldsymbol{x}^{[\gamma]} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \left\| \begin{pmatrix} \mathbf{A} \\ \sqrt{\gamma} \end{pmatrix} \boldsymbol{x} - \begin{pmatrix} \boldsymbol{y} \\ 0 \end{pmatrix} \right\|_2 \tag{2.7}$$

The equivalent linear algebra problem is

$$\begin{pmatrix} \mathbf{A} \\ \sqrt{\gamma} \end{pmatrix} \boldsymbol{x} = \begin{pmatrix} \boldsymbol{y} \\ 0 \end{pmatrix} \quad \Rightarrow \quad \boldsymbol{x}^{[\gamma]} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A} + \gamma\mathbb{I}\right)^{-1} \mathbf{A}^{\mathrm{T}}\boldsymbol{y}. \tag{2.8}$$

What remains is to choose a value for γ. If it is too small, the estimate trusts the data too much and will contain too much error due to noise. If it is too large, then the solution is oversmoothed and important features in the data may have been eliminated. Rewriting $\mathbf{A}$ in terms of its SVD will highlight how this occurs:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}} = \sum_{i=1}^{M} u_i \sigma_i v_i^T \tag{2.9}$$

where the columns of $\mathbf{U} \in \mathbb{R}^M \times \mathbb{R}^M$ are the *left-singular vectors* of $\mathbf{A}$; the columns of $\mathbf{V} \in \mathbb{R}^N \times \mathbb{R}^N$ are the *right-singular vectors* of $\mathbf{A}$; and $\mathbf{\Sigma} \in \mathbb{R}^M \times \mathbb{R}^N$ is a matrix whose diagonal entries, σ_i, are the square roots of the non-zero eigenvalues of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ (and, consequently, $\mathbf{A}\mathbf{A}^{\mathrm{T}}$). If the singular values are ordered by value, and labeled as $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_M$, then the singular vectors associated with the largest singular values are associated with low-frequency components of the decomposition. Conversely, the small-σ singular vectors are associated with high-frequency components of the decomposition [38].

Substituting the SVD of $\boldsymbol{A}$ into the least-squares solution (2.8):

$$\begin{aligned}
\boldsymbol{x}^{[\gamma]} &= \left(\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^{\mathrm{T}} + \gamma\mathbf{V}\mathbf{V}^{\mathrm{T}}\right)^{-1}\mathbf{V}\mathbf{\Sigma}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}\boldsymbol{y} \\
&= \mathbf{V}\left(\mathbf{\Sigma}^2 + \gamma\mathbb{I}\right)^{-1}\mathbf{\Sigma}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}\boldsymbol{y} \\
&= \sum_{i=1}^{M} \frac{v_i \varphi_i^{[\gamma]} u_i^T}{\sigma_i}\,\boldsymbol{y}, \quad \varphi_i^{[\gamma]} \equiv \frac{1}{1+\gamma/\sigma_i^2}
\end{aligned} \tag{2.10}$$

where the $\varphi_i^{[\gamma]}$ are referred to as Tikhonov filtering factors. Thus, singular vectors of the SVD for which $\gamma \gg \sigma$ become small with regularization, so that the series is effectively truncated at whichever i this threshold is crossed. Keeping γ small (i.e. $\gamma \ll \sigma_1$) means that the estimate includes contributions from all the high-frequency components of the SVD, while increasing γ filters these high-frequency contributions first and tends to smooth the estimates.

This shows that Tikhonov regularization acts like a smoother on estimated solutions to the linear least-squares inverse problem. It is important to note that the “model” in this case is $\boldsymbol{x} = 0$, which is apparent from taking the limit $\gamma \to \infty$ so that the term $\gamma \boldsymbol{x}^2$ dominates over the original measurement model, $\mathbf{A}\boldsymbol{x} = \boldsymbol{y}$. In the discussions that follow, models which are more motivated by the properties of the observed system are introduced as regularization terms, while maintaining the linearity of the regularization so that a spectral analysis of the pseudoinverse to the measurement function may be performed.

2.2.2 Linear Inference in a Static System: Electric Charge Distributions and Voltage Measurements

In this section, an example in which measurements of the electric potential around an object with known size and geometry are used to estimate the distribution of electric charge on its surface to illustrate some of the features of regularization in a linear inverse problem. It is assumed that these distributions are static in time, which avoids the additional complexities associated with modeling and estimating a dynamical system, but maintains most of the essential features of the regularization problem. It will be shown that incorporating some very simple, physically-reasonable assumptions about the system into a regularization term makes this estimation feasible, improving the quality of charge distribution estimates and, by extension, predictions of *new* voltage measurements. Importantly, it is also shown that there is an optimal regularization strength for maximizing estimation and prediction accuracy.

This problem is examined when the charge distribution is “glued” to two different surfaces: on a rectangular plane embedded in three dimensions, and on the surface of a sphere. Additionally, two kinds of regularization are considered, stemming from two different models of the distribution: one in which the distribution is assumed to not vary much from the average density, in which case them model is $\sigma = \bar{\sigma}$; and the other is that the distribution is smooth, enforced by $\boldsymbol{\nabla}^2\sigma = 0$. In each case a collection of M noisy measurements of the electric potential (or voltage) V constitute the observations from which σ is to be estimated. While the charged objects under observation are "real" physical objects and thus have charge distributions on their surfaces modeled as continuous functions of the spatial coordinates, the estimation problem is to be carried out on a computer and thus these distributions are naturally discretized in space. This means that σ is estimated in *patches* on the surface, returning a discrete sampling of σ rather than a continuous function as the result.

Without loss of generality, the plane is taken to line in the $z = 0$ plane with side lengths of $L_x = 1$ and $L_y = 1$, oriented to be centered at $x = 0$, $y = 0$ and with its edges aligned with the x and y axes. The spatial discretization of σ is in a rectangular grid oriented in the x and y directions, with grid spacings δx and δy. As a matter of terminology, the grid resolution in each direction is defined as $L_x/\delta x$ and $L_y/\delta y$, meaning that small grid spacings correspond to high resolutions and vice versa. On the sphere, which sits centered at the origin and has a radius $R = 1$, the grid is also "rectangular" but in the θ (polar) and ϕ (azimuthal) angular coordinates. This means that the grid spacings, $\delta\theta$ and $\delta\phi$, are defined on the angular coordinates and are constant functions of θ and ϕ. The effect of this is that the grid is more concentrated near the poles of the sphere than near the equator.

The actual charge densities themselves that will be "measured" in this example are displayed in Figure 2.1; while in reality these distributions would be created by some natural physical process, for the purposes of these numerical experiments they are defined by the following functions:

$$\begin{aligned} \textbf{Plane: } & \sigma(x,y) = \sigma_0 \cos\left[(0.83)2\pi x + \frac{\pi}{0.3}\right] \sin\left[(0.7)2\pi y - \frac{\pi}{0.15}\right], \\ \textbf{Sphere: } & \sigma(\theta,\phi) = \sigma_0\, Y_{21}(\theta,\phi). \end{aligned} \tag{2.11}$$

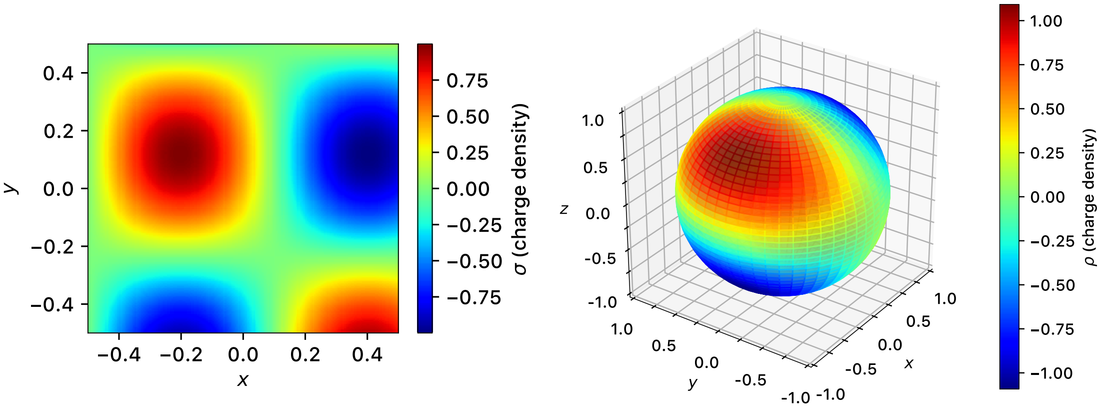


Figure 2.1: The true charge densities for the plane (left panel) and sphere (right panel) in the inverse problem example of inverting voltage measurements for charge density, $V \to \sigma$. These surfaces are discretized with square grids in x, y and θ, ϕ, respectively.

$Y_{21} = \frac{1}{\sqrt{2}}\left(Y_2^{-1} - Y_2^1\right)$ is a real-valued linear combination of spherical harmonics with $l = 2$. For the sake of simplicity, σ_0 is set to 1 (with units of charge/unit area).

These charge distributions naturally have an electric potential associated with the electric field surrounding them. Measurements of the electric potential V are modeled by Couloumb's law, which says that given a static, confined charge distribution ρ, the electric potential in the region surrounding the distribution is given by

$$V(\boldsymbol{r}) = \frac{1}{4\pi} \int\limits_{\text{source}} d^3x' \, \frac{\rho(\boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|}. \tag{2.12}$$

In the case of a charge distribution defined on a *surface* in three-dimensional space, the integral in (2.12) simply changes into

$$V(\boldsymbol{r}) = \frac{1}{4\pi} \int\limits_{\text{surface}} d^2x' \frac{\sigma(\boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|} \tag{2.13}$$

where σ has dimensions of charge/unit area. Here, $\boldsymbol{r}$ is the location of the probe where V is measured (which corresponds to s in (2.2)), and $\boldsymbol{r}'$ is the location of the source charge density $\sigma(\boldsymbol{r}')$. This is in the exact form of the generic linear measurement modeled by eq. (2.2), with a source σ taking the place of f (the true state of the system); the observation V in place of g; and the Couloumb kernel $1/4\pi|\boldsymbol{r}-\boldsymbol{r}'|$ acting as the measurement kernel K. This gives rise to the following translation table to relate to the more general quantities defined previously:

$$f(s') \to \sigma(\boldsymbol{r}'), \quad g(s) \to V(\boldsymbol{r}), \quad K(s,s') \to \frac{1}{4\pi|\boldsymbol{r}-\boldsymbol{r}'|}. \tag{2.14}$$

While in this example, the true state of the system (σ) is actually known exactly, in reality one must consider the situation where only noisy measurements of V are accessible in an experiment. Despite the fact that one has access to this normally “hidden” information in a twin experiment, ultimately one should base their judgment on the quality of a state estimate by the *predictions* they produce. In this case, the predictions are values of V at previously unmeasured locations $\boldsymbol{r}$. This sets up the twin experiment paradigm for validation of this method: the data are generated using a known functional of σ, and the quality of estimates is judged by the errors of estimation of σ, and the prediction error of novel V values.

As was stated previously, the estimation problem is to be carried out numerically and thus should be discretized in space. We have already discussed the strategy of discretizing σ on a grid covering either surface, but the M voltage measurements are made at a collection of discrete points in space and thus the measurement function requires no further discretization. This translates into the integral over the source coordinates $\boldsymbol{r}'$ being transformed into a sum over the discrete spatial grid, whereas the voltage is evaluated at arbitrary points in space off of these grids, $\boldsymbol{r}$. Furthermore, the measurement function, represented by the Coulomb integral, is linear and thus represented by a matrix operator in discrete coordinates. If we consider $\sigma_j = \sigma(\boldsymbol{r}'_j)$

to be the charge density at point $\boldsymbol{r}'_j$, and $V_i = V(\boldsymbol{r}_i)$ the measured voltage at point $\boldsymbol{r}_i$, then the original integral representation of the forward problem (measurement) transforms into:

$$V(\boldsymbol{r}_i) = \frac{1}{4\pi} \int\limits_{\text{source}} d^2x' \, \frac{\sigma(\boldsymbol{r}')}{|\boldsymbol{r}_i - \boldsymbol{r}'|} \to V_i = \sum_j K_{ij}\sigma_j. \tag{2.15}$$

In matrix notation, the above equation will be written as $\boldsymbol{V} = \mathbf{K}\boldsymbol{\sigma}$.

This linear operator $\mathbf{K}$ is a matrix equal to the discretization of the Coulomb integral operator. The choice of discretization for $\mathbf{K}$ is not unique; for example, if the spatial discretization is to occur by quadrature with a grid defined on the spatial coordinates of the source distribution, then the choice of this grid is by no means trivial. In this example we simply choose to use rectangular grids, which the benefit of not obscuring other important details of the inverse problem behind the complexity of the grid construction.

$$\begin{aligned}
\textbf{Plane: } V(\boldsymbol{r}_i) &= \frac{\delta x\, \delta y}{16\pi} \sum_j \left[\frac{\sigma(x'_j, y'_j)}{D(\boldsymbol{r}_i, x'_j, y'_j)} + \frac{\sigma(x'_j, y'_j + \delta y)}{D(\boldsymbol{r}_i, x'_j, y'_j + \delta y)} \right. \\
&+ \left. \frac{\sigma(x'_j + \delta x, y'_j)}{D(\boldsymbol{r}_i, x'_j + \delta x, y'_j)} + \frac{\sigma(x'_j + \delta x, y'_j + \delta y)}{D(\boldsymbol{r}_i, x'_j + \delta x, y'_j + \delta y)} \right], \\
&\equiv \sum_j K_{ij}\sigma_j \\
D(\boldsymbol{r}_i, x'_j, y'_J) &\equiv \sqrt{(x_i - x'_j)^2 + (y_i - y'_j)^2 + z_i^2}.
\end{aligned} \tag{2.16}$$

For the sphere:

$$
\begin{aligned}
\textbf{Sphere: } V(\boldsymbol{r}_i) &= R^2\,\frac{\delta\theta\,\delta\phi}{16\pi}\sum_j\Bigg[\frac{\sin(\theta'_j)\,\sigma(\theta'_j,\phi'_j)}{D(\boldsymbol{r}_i,\theta'_j,\phi'_j)}+\frac{\sin(\theta'_j)\,\sigma(\theta'_j,\phi'_j+\delta\phi)}{D(\boldsymbol{r}_i,\theta'_j,\phi'_j+\delta\phi)}\\
&\quad+\frac{\sin(\theta'_j+\delta\theta)\,\sigma(\theta'_j+\delta\theta,\phi'_j)}{D(\boldsymbol{r}_i,\theta'_j+\delta\theta,\phi'_j)}+\frac{\sin(\theta'_j+\delta\theta)\,\sigma(\theta'_j+\delta\theta,\phi'_j+\delta\phi)}{D(\boldsymbol{r}_i,\theta'_j+\delta\theta,\phi'_j+\delta\phi)}\Bigg],\\
&\equiv\sum_j K_{ij}\sigma_j\\
D(\boldsymbol{r}_i,\theta'_j,\phi'_j) &\equiv\sqrt{(x_i-R\sin\theta'_j\cos\phi'_j)^2+(y_i-R\sin\theta'_j\sin\phi'_j)^2+(z_i-R\cos\phi'_j)^2}.
\end{aligned}
\tag{2.17}
$$

The sum is performed over all of the patches on the plane with corners at (x'_j, y'_j); or the patches on the sphere cornered at (θ'_j, ϕ'_j). Now that the forward problem is finally in the form $V_i = \sum_j K_{ij}\sigma_j$, which is linear and discretized, the goal is to somehow compute the solution to the *inverse* problem $\sigma_i = \sum_j K^{-1}_{ij} V_j$. As discussed previously, there are potentially severe problems with this inversion that make the calculation untenable or just impossible in the first place.

Regularizing the Inverse Problem $\boldsymbol{V} \to \boldsymbol{\sigma}$

The problem of estimating the charge distribution $\boldsymbol{\sigma}$ from $\boldsymbol{V}$ is ill-posed. One way to show this is by brute force (see top row of fig. 2.3), attempting the inversion through linear least-squares without any regularization. In this context linear least-squares is:

$$
\boldsymbol{\sigma}_{\text{est}} = \underset{\boldsymbol{\sigma}}{\operatorname{argmin}}\,(\mathbf{K}\boldsymbol{\sigma}-\boldsymbol{V})^2 \iff \boldsymbol{\sigma}_{\text{est}} = \left(\mathbf{K}^\top\mathbf{K}\right)^{-1}\mathbf{K}^\top\boldsymbol{V}. \tag{2.18}
$$

The discretized integration operator $\mathbf{K}$ does not, in general, have a well-defined or unique inverse. If the problem is over- or under-determined, then $\mathbf{K}$ is not an invertible matrix. Because $\mathbf{K}^{\top}\mathbf{K}$ has the same rank as $\mathbf{K}$, $\left(\mathbf{K}^{\top}\mathbf{K}\right)^{-1}$ is also not well-defined. In addition to this, $\mathbf{K}$ approximates an integral and thus tends to smooth the features of $\boldsymbol{\sigma}$, so inversion of $\mathbf{K}$ tends to amplify high-frequency components of the estimated solution and thus may be extremely sensitive to perturbations in the data $\boldsymbol{V}$. To attempt to solve these problems, inversion of $\mathbf{K}$ is to be made well-defined through regularization. This is done by adding additional terms to the cost function (2.18), but the question remains of what appropriate regularization terms are.

Previously, in §2.2, it was shown that introducing Tikhonov [38] regularization into an ill-posed inverse problem had the effect of filtering spurious high-frequency components in the estimated solution that were the result of a rank-deficient measurement function. This was done by analytically inserting the Tikhonov regularization term, $\gamma \boldsymbol{x}^2$, into the least squares cost function, and then rewriting the solution in terms of the singular value decomposition (SVD) of the measurement function. Here we examine the use Tikhonov regularization as a model of a nearly-uniform distribution; additionally, a regularization term involving the Laplacian of the charge distribution is used as a model of smoothness. It will be shown that both of these types of regularization lead to fairly accurate estimates of σ with the right choice of regularization strength γ.

When it is assumed that the distribution does not vary much from its average value $\bar{\boldsymbol{\sigma}}$:

$$\boldsymbol{\sigma}^{[\gamma]} = \underset{\boldsymbol{\sigma}}{\operatorname{argmin}} \left[\left(\mathbf{K}\boldsymbol{\sigma} - \boldsymbol{V}\right)^2 + \gamma \left(\boldsymbol{\sigma} - \bar{\boldsymbol{\sigma}}\right)^2 \right] \tag{2.19}$$

where $\bar{\boldsymbol{\sigma}}$ is the average charge density on the surface. The average charge density is actually measurable in an experiment from the monopole moment of V, which is known from measuring V at a sufficiently far distance from the distribution, thus further justifying the use of such a model. This is just Tikhonov regularization in its more general form, $\mathbf{L}(\boldsymbol{x}-\boldsymbol{x}_0)$ with $\mathbf{L}(\boldsymbol{x}-\boldsymbol{x}_0)=\gamma\,(\boldsymbol{x}-\boldsymbol{x}_0)^{\mathrm{T}}\,\mathbb{I}\,(\boldsymbol{x}-\boldsymbol{x}_0)$, and $\boldsymbol{x}_0=\bar{\boldsymbol{\sigma}}$.

The second method, in which the Laplacian of σ, $\nabla^2\sigma$, is included to enforce local smoothness of the distribution, will be referred to as Laplacian regularization. The regularized cost function thus takes the form:

$$\boldsymbol{\sigma}^{[\gamma]}=\underset{\boldsymbol{\sigma}}{\operatorname{argmin}}\left[(\mathbf{K}\boldsymbol{\sigma}-\boldsymbol{V})^2+\gamma\left(\boldsymbol{\nabla}^2\boldsymbol{\sigma}\right)^2\right] \tag{2.20}$$

where the Laplacian operator,

$$\boldsymbol{\nabla}^2=\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}, \tag{2.21}$$

is understood to be discretized in space. Here we choose to use a second-order finite difference discretization scheme on the same grid over which σ is estimated:

$$\begin{aligned}\boldsymbol{\nabla}^2\sigma(x,y)\simeq\ &\frac{\sigma(x+\delta x,y)+\sigma(x-\delta x,y)-2\sigma(x,y)}{(\delta x)^2}\\&+\frac{\sigma(x,y+\delta y)+\sigma(x,y-\delta y)-2\sigma(x,y)}{(\delta y)^2}.\end{aligned} \tag{2.22}$$

Rather than discretizing the Laplacian over the surface of the sphere, only Tikhonov regularization will be employed in that case. This will highlight some qualitative features of the regularized inversion process, whereas a more quantitative analysis involving estimation and prediction error will be done using the plane charge distribution.

Figure 2.2 shows the evolution of charge distribution estimates as the Tikhonov regularization strength increases when the number of voltage measurements $M=100$ for the spherical distribution. These measurements are randomly distributed around the sphere, within an outer shell ranging from $r=1.2$ to $r=2$. When γ is small

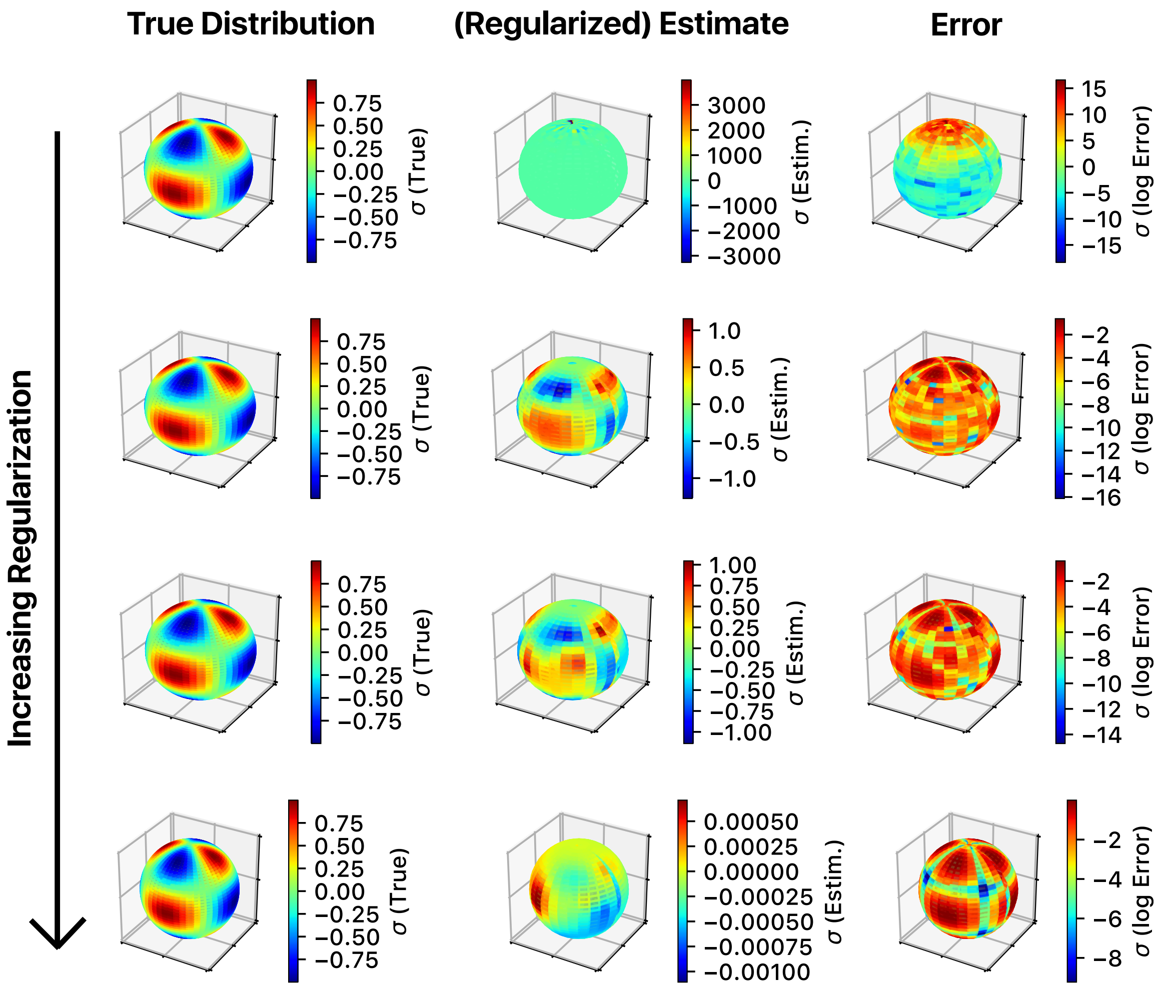


Figure 2.2: Measurements of the electric potential in the region surrounding an electrically charged sphere are used to reconstruct the true distribution of charge density. Left column: the true charge distribution. Center column: the reconstructed distribution, estimated with a regularized inverse of the Coulomb integral. Right column: squared estimation error (see text for definition). Going from top to bottom, the coefficient on the regularization term increases, showing $\gamma = 5 \times 10^{-4}$, 10^{-1}, 1.5, and 8×10^{2}. Introducing an "optimal" amount of regularization leads to an estimate with locally minimal and somewhat evenly-distributed error.

the estimation errors are quite large, especially at the poles where the variations in σ are the most rapid (apparently the increased resolution at the poles was not enough to compensate for this). Conversely, when γ is large the regularization dominates and, quite expectedly, the estimate of σ is fairly uniform and close to the average density of 0. It is in the intermediate regime that the original distribution is maximally recovered.

For this model the choice of some intermediate γ value was logical. With no model the problem is already known to be ill-posed. When γ is large, the distribution estimates tend towards $\boldsymbol{\sigma} = 0$, which is not a particularly useful result if one at all believes that there is a non-uniform distribution on the surface. Essentially, the model here is "wrong", but if one is careful then the model can be enforced on the charge estimates with just the right strength between undersmoothing at oversmoothing. Later in the dissertation, this is shown to be an important consideration to take into effect when the model is a chaotic dynamical system.

Estimation and Prediction for the Planar Distribution

A similar calculation was carried out to estimate the true planar distribution from voltage measurements, but this time with Tikhonov as well as Laplacian regularization. The estimation error is calculated as an average over the surface of the squared discrepancies $(\boldsymbol{\sigma}_{est} - \boldsymbol{\sigma}_{true})$. The prediction error is similarly computed by comparing predictions of electric voltage against a validation data set V_{val}, similarly generated to the twin experiment data but which was not used for estimating the distribution.

Fig. 2.3 shows the $\boldsymbol{\sigma}$ estimation and $\boldsymbol{V}_{\text{val}}$ prediction errors when using Tikhonov and Laplacian regularization. The plane is divided into a $N_x \times N_y = 32 \times 32$ evenly spaced grid so that $\boldsymbol{\sigma}$ is effectively a 1024-dimensional vector. To generate data for estimating the charge distribution, V was randomly sampled at M points in the region surrounding the plane, and integrating Couloumb's law over a fine grid in comparison to the estimation grid. A small amount of noise was additionally added to each measurement to simulate measurement error in a real measurement. Estimates were

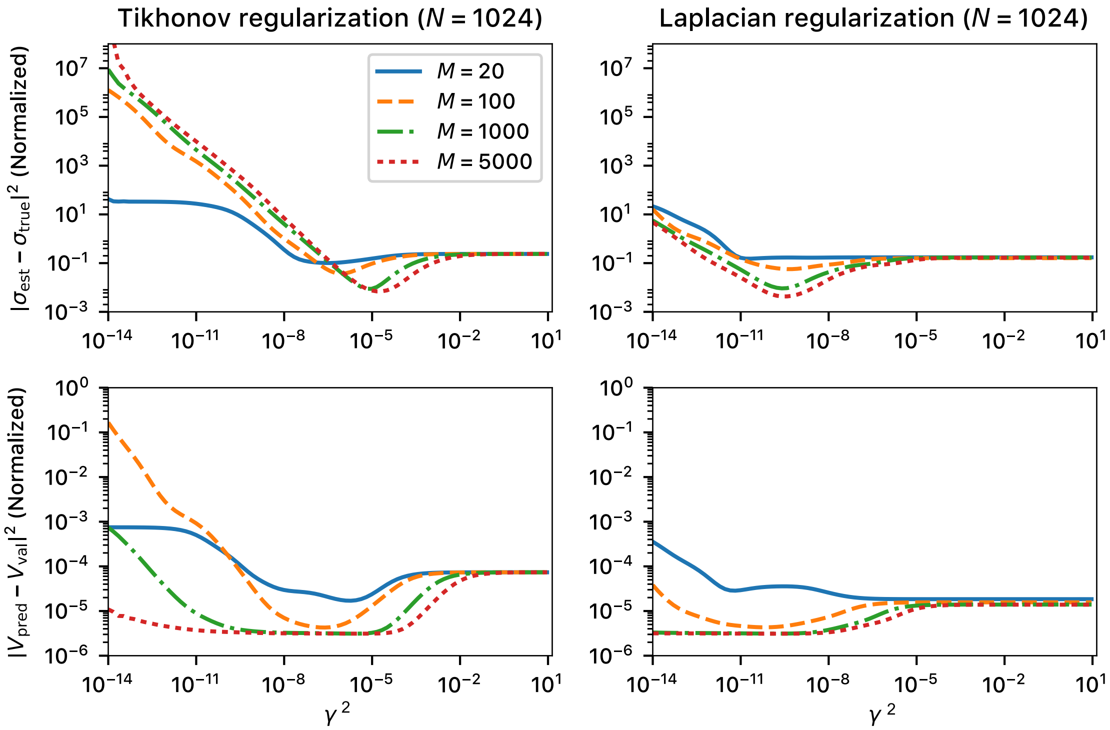


Figure 2.3: Estimation (top row) and prediction (bottom row) errors for the planar charge distribution, $\boldsymbol{\sigma}$, and validation set voltages, $\boldsymbol{V}_{\text{val}}$, respectively. The left column shows errors when using Tikhonov regularization, where $\boldsymbol{g}(\boldsymbol{\sigma}) = \boldsymbol{\sigma}$, and the right column when using Laplacian regularization, with $\boldsymbol{g}(\boldsymbol{\sigma}) = \boldsymbol{\nabla}^2\boldsymbol{\sigma}$ (the Laplacian is discretized on the xy grid using finite differences). Note that both methods achieve similar minimum levels of error. However, Laplacian regularization is somewhat less sensitive to its value when considering prediction error.

compared to the true distribution σ at each grid point, and the squared error was averaged over the plane. Predictions are done on a validation set of voltage values "measured" at 1000 new locations (not part of the set from which σ was estimated).

It is apparent, yet again, from the estimation and prediction errors that one must carefully choose the regularization strength to avoid problems with under- or oversmoothing the solution. In this case, the voltage prediction errors were found to be at least an order of magnitude smaller for intermediate γ values compared to the regime of strong regularization. There did not seem to be any advantage to choosing Laplacian regularization, however.

It is not necessary to go into much more detail this. What is important is that regularization has been established as a smoother through numerical experiments, and has a significant effect on the quality of estimation and prediction. Furthermore, it is seen that carefully selecting the regularization strength is necessary to produce an optimal prediction. Here it is basically known that the model is wrong; in real settings, however, it is not necessarily known how wrong the model is, or if it is wrong at all. In a chaotic system in Chapter 3, this will be shown to warrant a careful numerical analysis of the properties of the estimation and prediction problem, and that some very general features of this problem carry over to the case where the regularization term is a nonlinear function.

2.3 Data Assimilation in a Linear Dynamical System as a Regularized Inverse Problem

It will now be shown that the methods of regularization introduced in the previous example are equivalent to state estimation in a linear dynamical system using variational data assimilation. In the path integral formulation of variational data assimilation, statistics of state and parameter estimates are computed using high-dimensional integrals of the conditional distribution $P(X|Y) \propto e^{-A(X,Y)}$, which is a function of the action defined in Chapter 1. Under the Laplace approximation, these integrals are computed using the peaks of this distribution, which correspond to minima of the action. Model state and parameter estimation thus takes the form of a regularized linear inversion like the one presented above, with the difference being that spatial smoothness and uniformity were (perhaps reasonably) assumed as properties of the true distribution under observation, leading to regularization terms which minimized the L_2-norm of $\boldsymbol{\sigma}$ and its Laplacian, $\boldsymbol{\nabla}^2\boldsymbol{\sigma}$.

We now consider problems in which the corresponding assumption is that the underlying process generating the observed time-series data is a dynamical system. The corresponding term replacing $\boldsymbol{\nabla}^2\boldsymbol{\sigma}$, for example, will be one which minimizes the error in the forward mapping $\boldsymbol{F}(\boldsymbol{x}^n, \boldsymbol{x}^{n+1}) = \boldsymbol{x}^{n+1}$, which regularizes the inverse problem by introducing information into the system in the form of constraints in space *and* time. This form of regularization is henceforth referred to as *dynamical regularization.*

In addition, the dynamical model is chosen to be the *right* model for the system, in that it is the same model that was used to generate the data. Despite this fact, one must also carefully choose γ so as to not oversmooth the estimates, leading to worsened predictions. This carries over to the analysis of a nonlinear system in Chapter 3.

2.3.1 State Estimation in Linear Dynamical Systems

The observed system in this example is a linear oscillator with mass m and oscillation frequency ω. The data consists of noisy observations of the oscillator's position over time. The system is modeled as a simple harmonic oscillator (SHO), which is governed by the second-order ODE:

$$m\frac{d^2q}{dt^2} = -m\omega^2 q \equiv -kq. \tag{2.23}$$

In this equation, k is the familiar spring constant if the oscillator is actually an object with mass m connected to a totally stationary object by a spring. This system can easily be rewritten as two first-order ODEs by defining a new variable $v = \dot{q}$ (which is just the velocity of the oscillator), in which case the dynamical equations become

$$\begin{aligned} \frac{dq}{dt} &= v, \quad \frac{dv}{dt} = -\omega^2 q \\ \text{or} \quad \frac{d\boldsymbol{x}}{dt} &= \frac{d}{dt}\begin{pmatrix} q \\ v \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\omega^2 & 0 \end{pmatrix}\begin{pmatrix} q \\ v \end{pmatrix} \equiv \mathbf{K}\boldsymbol{x}. \end{aligned} \tag{2.24}$$

When $k > 0$, $\mathbf{K}$ has imaginary eigenvalues and the system is thus a linear oscillator with constant energy $E = \frac{1}{2}mv^2 + \frac{1}{2}m\omega^2 q^2$.

If the position is observed at regular intervals Δt, with measurement noise $\xi^n \sim N(0, \sigma^2)$ in each observation, then Y is a path vector given by:

$$Y = \left\{ q^1 + \xi^1, q^2 + \xi^2, \ldots, q^N + \xi^N \right\}. \tag{2.25}$$

The measurement function, H, is a matrix which projects $\boldsymbol{x}$ to the position coordinate at each observation time:

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 1 & \cdots \\ & & \vdots & & & \ddots \end{pmatrix} \quad \Rightarrow \quad Y = HX + \Xi. \tag{2.26}$$

where Ξ is the path variable for the measurement noise. By inspection it is clear that H is highly singular: every other column is filled with zeros, meaning that the rank of H is equal to just half the dimension of the space of the full trajectory, which is $2N$. The inverse problem $X = H^{-1}Y$ thus has no unique solution, and any pseudoinverse for H which we construct will almost certainly be highly sensitive to perturbations in the data. This last fact is problematic for the realistic situation in which the data is noisy.

The ill-posedness of this inverse problem will be solved through regularization, similarly to the example presented above. The effect of regularization will be examined analytically in terms of the eigenvalue decomposition of the regularized inverse measurement operator, as well as the projections of the regularized estimates of X and the data Y onto its eigenvectors. These results will be compared with a numerical analysis which uses a prediction-based metric for assessing the optimality of regularization that is analogous, again, to what was used previously in the static charge problem.

The Regularized Problem

Similarly to the problem of static charge estimation from voltage measurements, physical knowledge or intuition about the system under observation will be used to construct regularization terms for the cost function, now equal to the discrete-time action, A, defined in Chapter 1. The difference between this example and the previous static charge example is that a dynamical model for the system based on the forces acting on the oscillator (or, equivalently, the potential function for the system) is already being considered as the model of the system, meaning that it is unnecessary to make assumptions about the structure or composition of the system beyond the dynamical model, given in (2.24).

Yet again, however, one needs to consider the fact that the inverse problem is to be done numerically on a computer. The ODE model for the oscillator is already "discretized" in space, but the data is sampled in discrete time, leading to a discretization of model trajectories in time. Normally this requires approximating the forward integral with some numerically-computed forward mapping $\boldsymbol{F}$:

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \int\limits_{t_n}^{t_{n+1}} ds\ \boldsymbol{f}(\boldsymbol{x}(s);\boldsymbol{\theta}) \simeq \boldsymbol{F}(\boldsymbol{x}^n, \boldsymbol{x}^{n+1}; \boldsymbol{\theta}). \tag{2.27}$$

When considering a *linear* system $\boldsymbol{f}$, as is the case here, no approximation is necessary because the above integral can be done analytically. For the sake of comparison to later problems, however, which are *nonlinear* and thus $\boldsymbol{F}$ has to be estimated, we maintain the position that numerical integration of $\boldsymbol{f}$ is necessary. For the purposes of this example the trapezoid rule [68] is used for the discretization, giving:

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \frac{\Delta t}{2}\,\mathbf{K}\left(\boldsymbol{x}^n + \boldsymbol{x}^{n+1}\right). \tag{2.28}$$

This method is implicit because $\boldsymbol{x}^{n+1}$ is a function of $\boldsymbol{x}^{n+1}$, which means that in order to get $\boldsymbol{x}^{n+1}$ an algebraic equation must be solved, rather than simply plugging in the values of the current state as is the case with explicit integration schemes. In this case the equation happens to be extremely easy to solve:

$$\boldsymbol{x}^{n+1} = \left(\mathbb{I} - \frac{\Delta t}{2}\mathbf{K}\right)^{-1}\left(\mathbb{I} + \frac{\Delta t}{2}\mathbf{K}\right)\boldsymbol{x}^n. \tag{2.29}$$

In general the inversion of eq. (2.28) to produce an analytic expression like eq. (2.29) is not necessarily possible, in particular when the ODE system $\boldsymbol{f}(\boldsymbol{x})$ is not linear and thus cannot be written in the form $\boldsymbol{f}(\boldsymbol{x}) = \mathbf{A}\boldsymbol{x}$. Implicit methods must be solved numerically for $\boldsymbol{x}^{n+1}$, using Newton's method [68, 19] for example.

There is no need, however, to explicitly solve for $\boldsymbol{x}^{n+1}$ because the action treats all $\boldsymbol{x}^n$ in the model error sum equally as independent variables. This means that it is only necessary to substitute the (implicit) expression for $\boldsymbol{x}^{n+1} - \boldsymbol{x}^n$ into the model error:

$$\boldsymbol{x}^{n+1} - \boldsymbol{x}^{n} = \frac{\Delta t}{2}\,\mathbf{K}\left(\boldsymbol{x}^{n} + \boldsymbol{x}^{n+1}\right) \equiv \boldsymbol{g}\left(\boldsymbol{x}^{n}, \boldsymbol{x}^{n+1}; \boldsymbol{\theta}\right). \tag{2.30}$$

This is also more in line with the analysis of nonlinear problems. An implicit method for the action discretization is purposefully chosen because it does *not* need to be solved explicitly, and implicit methods have the additional benefits of increase solution stability compared to explicit methods of the same order.

In the action, the model error is a sum taken over time of the discrepancy $\boldsymbol{g}$; in this example the discrepancy at each time is given by (2.30), and the model error is

$$\frac{R_f}{2}\sum_{n=1}^{N-1}\left[\boldsymbol{g}\left(\boldsymbol{x}^{n}, \boldsymbol{x}^{n+1}; \boldsymbol{\theta}\right)\right]^2 \tag{2.31}$$

where $\boldsymbol{g}^2 = \sum_i g_i g_i$ is a scalar. In this problem, $\boldsymbol{g}$ may be written as a matrix, and by extension the path-space version, G, can be written as a $(N-1)D \times (N-1)D$ dimensional matrix. One simply has to ensure that applying G to the path vector X produces the right result, which means

$$G = \frac{\Delta t}{2}\left(\begin{array}{ccccccccc} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & \\ -\omega^2 & 0 & -\omega^2 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & \\ 0 & 0 & -\omega^2 & 0 & -\omega^2 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & \\ 0 & 0 & 0 & 0 & -\omega^2 & 0 & -\omega^2 & 0 & \\ & & & \vdots & & & & & \ddots \end{array}\right). \tag{2.32}$$

Applying G to a path space state vector gives:

$$GX = \frac{\Delta t}{2}\left\{v^1 + v^2,\ -\omega^2\left(q^1 + q^2\right),\ v^2 + v^3,\ -\omega^2\left(q^2 + q^3\right),\ \ldots\right\}. \tag{2.33}$$

Now, each term in the action is written in path space variables; and because the measurement function and model for the observed system are both linear, all terms are of the form $V^2 = \sum_\alpha V_\alpha V_\alpha$ where V is some arbitrary path space vector[1]:

$$A = \frac{R_m}{2}\left(HX - Y\right)^2 + \frac{R_f}{2}\left(GX\right)^2 \equiv \frac{R_m}{2}\sum_\alpha \left[\left(HX\right)_\alpha - Y_\alpha\right]^2 + \frac{R_f}{2}\sum_\alpha \left(GX\right)_\alpha^2. \tag{2.34}$$

Finally, because the location of the minima of A, $X_{\text{min}} = \underset{X}{\operatorname{argmin}}\, A$, are unchanged by multiplication of a constant factor, the action is rescaled as

$$\begin{aligned} A' &= \frac{2}{R_m}A = \left(HX - Y\right)^2 + \frac{R_f}{R_m}\left(GX\right)^2 \\ &\equiv \left(HX - Y\right)^2 + \gamma\left(GX\right)^2 \\ \Rightarrow \quad X^{[\gamma]} &= \underset{X}{\operatorname{argmin}}\, A'. \end{aligned} \tag{2.35}$$

Comparing this to the least-squares cost function in the static charge example, $\gamma = R_f/R_m$ again takes on the role of the regularization strength. Like that example, the regularization couples system variables together and acts to remove the degeneracy of state estimates which leads to non-uniqueness of solutions, but now it simultaneously connects system variables in time.

Putting the estimation problem $\underset{X}{\operatorname{argmin}}\, A'$ in the form of a matrix problem allows it to be studied using several distinct approaches. First, in terms of how the

[1] The sum is written with indices for the sake of clarity, but V^2 should be understood to be $\sum_\alpha V_\alpha V_\alpha$ from now on

regularization affects the construction of a pseudoinverse for H from the perspective of its singular values and the resulting spectral decomposition of estimates $X^{[\gamma]}$ and the data Y. These results can thus be analyzed in terms of the regularization acting as a smoother on solutions. Second, estimation-error and prediction-error based metrics for optimality of regularization strength are introduced to analyze the accuracy of estimated solutions. This is treated purely as a numerical problem so that it is readily extensible to problems with nonlinear models $\boldsymbol{f}$, where H^{-1} cannot be written analytically in terms of H and $\boldsymbol{f}$.

Spectral Analysis of Dynamical Regularization

Starting from eqs. (2.35), in which state estimation is posed as a regularized least-squares problem in the path-space state variable X, one may follow the procedure in [38] , also presented in the introduction to this chapter, to write the solution in terms of an analytical expression for the pseudoinverse of H:

$$X^{[\gamma]} = \left(H^{\mathrm{T}}H + \gamma G^{\mathrm{T}}G\right)^{-1} H^{\mathrm{T}}Y \equiv \rho(\gamma)\, Y. \tag{2.36}$$

In the above expression, $\rho(\gamma) = \left(H^{\mathrm{T}}H + \gamma G^{\mathrm{T}}G\right)^{-1} H^{\mathrm{T}}$ is a pseudoinverse of H formed by introducing the regularization term $\gamma\,(GX)^2$ into the least-squares cost function, and thus is a function of the regularization strength γ. ρ is just a matrix in the discrete path space, making it possible to examine the effects of regularization in terms of its singular values and vectors, as well as the spectral content of state estimates $X^{[\gamma]}$, as a function of γ. The advantage to starting with this kind of analysis for a linear system is that the properties of the pseudoinverse of H can be studied *independently* of a particular data set or state estimate. This luxury is not afforded when dealing with nonlinear systems, where a construction of the pseudoinverse such as the one shown in (2.36). However, some ideas for how this kind of analysis could be possible in a hybrid approach, combined with numerical optimization of A, are presented at the end of this chapter.

The oscillation frequency ω in the model is first set equal to one, which is generally possible for a simple harmonic oscillator under rescaling of time. This is done to eliminate the need for choosing ω based on observation data, allowing for an analysis that generalizes to oscillators with arbitrary ω. It should be noted, however, that it is assumed observations are made frequently enough to satisfy the Nyquist criterion [78, 66], beyond which state estimation would almost certainly be impossible.

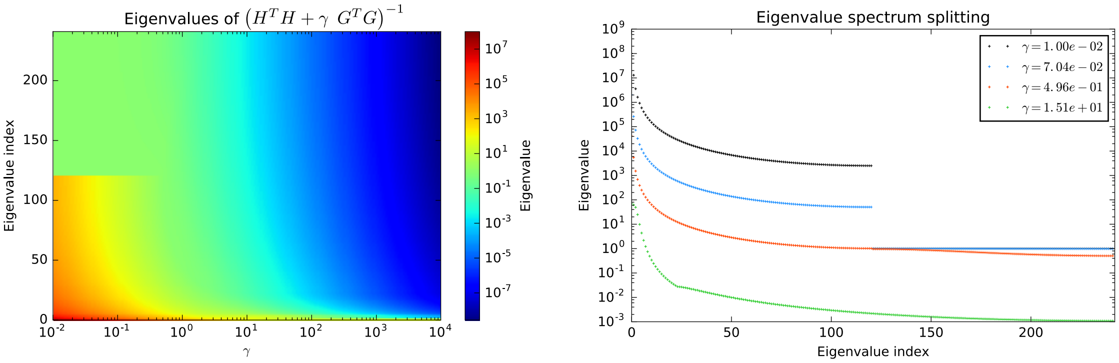


Figure 2.4: Eigenvalue spectrum for the regularized inverse of the measurement operator in the simple harmonic oscillator problem, in which only the position of the oscillator is measured.

First, the eigenvalue spectrum of the regularized inverse measurement operator is examined as a function of the dynamical regularization strength, γ. Examining the surface presented in figure 2.4, the eigenvalue spectrum of the regularized inverse is split between very large values near 10^6 or 10^7 when γ is very small, and the other half of the spectrum which is near 1. The regularized state estimates $X^{[\gamma]}$ are linear combinations of projections of the data path vector Y (actually, $H^{\mathrm{T}}Y$ which simply acts to double the length of Y without filling in the extra entries) onto the eigenvectors of the inverse. Clearly, there is information which is destroyed by taking the measurement: in this case the small eigenvalues correspond to portions of the measurement function which project the velocity to some negligibly small value. This splitting is even more apparent in the right panel of figure 2.4. Note that there is an intermediate value for γ near $1/2$ where the two halves of the spectrum join, and beyond which the larger eigenvectors begin to dominate again in comparison.

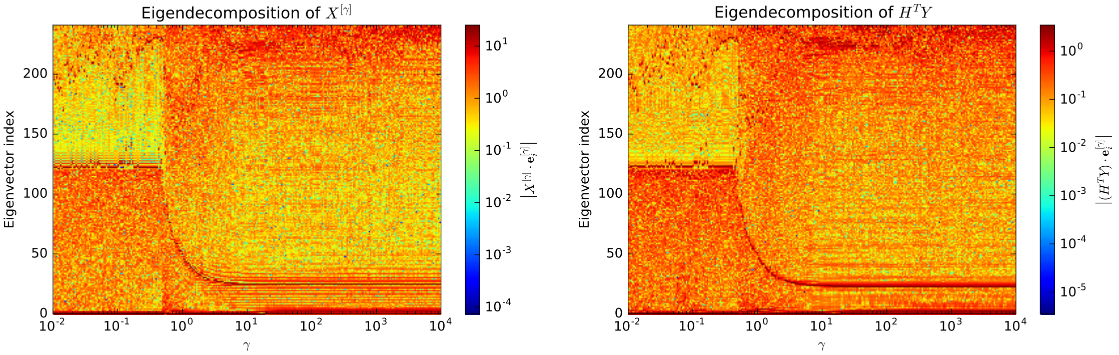


Figure 2.5: Coefficients of the decomposition of $X^{[\gamma]}$ and $H^{\mathrm{T}}Y$ onto the eigenvectors of the regularized inverse measurement operator, $\left(H^{\mathrm{T}}H + \gamma G^{\mathrm{T}}G\right)^{-1}$. As γ is increased, there is a dominant contribution that appears from a low-frequency eigenvector corresponding to smoothing of the estimate.

Figure 2.5 shows the coefficients of the projections of H^TY and $X^{[\gamma]}$ onto the eigenvectors of the regularized inverse measurement function. The most notable feature of this plot is that there is a strong peak in these coefficients that closely follows a line in this plane. At low γ, this line is horizontal and located near the highest-index eigenvector whose eigenvalue is not zero (more precisely, negligibly small in comparison to the lower-index eigenvalues). As soon as the two halves of the spectrum join, this peak begins to track a line that moves towards lower-index eigenvectors until again becoming horizontal.

Clearly, in the intermediate regime the solution to the inverse problem is changing significantly. Using the knowledge that the high-γ solutions are smoothed by regularization, the horizontal line at high γ corresponds to a smoothed solution dominating the estimate. It is suspected that the solutions in the intermediate regime

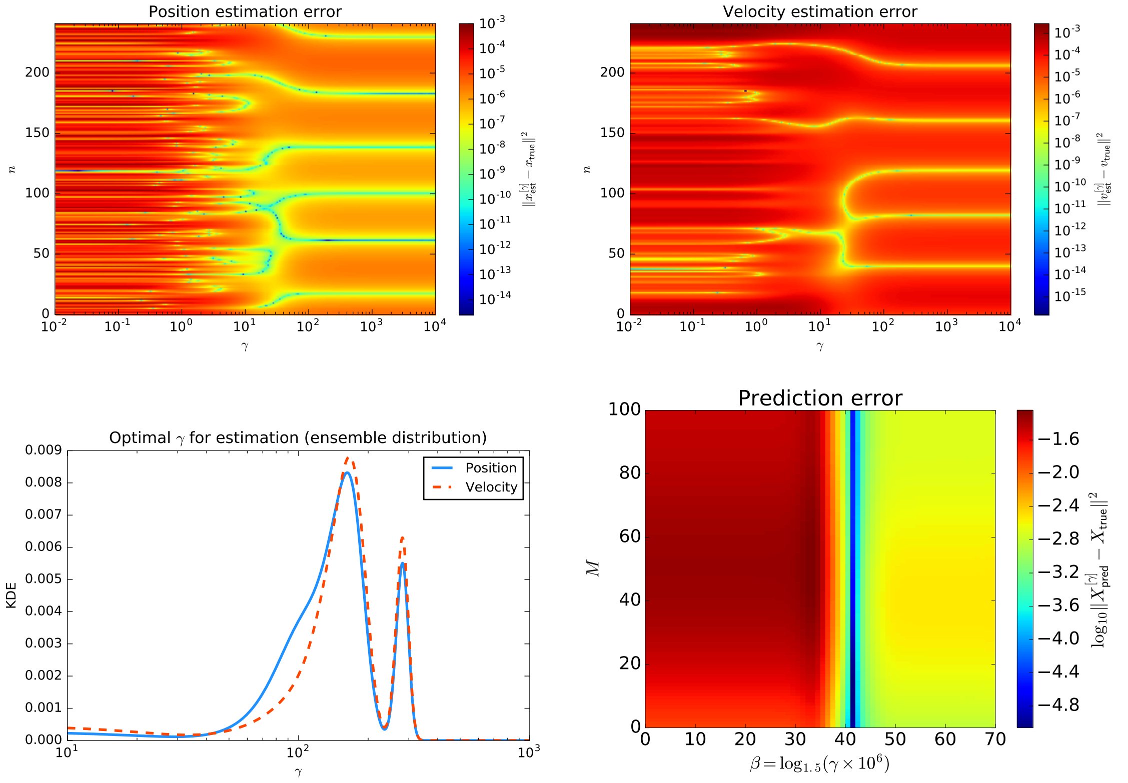


Figure 2.6: Top row: position and estimation errors for a single instantiation of the measurement noise. The measurement noise was drawn from the large-noise ($\sigma = 0.1$ m) ensemble. Each point in the graphs represents the squared error in the estimated path $X^{[\gamma]}$ at time t_n, compared to the true solution. Bottom row: KDE of γ values found to produce optimal estimates, and the position and velocity prediction error (averaged). Note the second peak at high γ: occasionally there was no locally optimal intermediate γ for estimation.

may prove better for estimation and prediction. While this was clearly the case for the σ-estimation problem where the model was wrong, this is shown to still be true with the "right" model in the next part of the discussion.

Numerical Analysis: Effects of Regularization on Prediction

The effect on state estimation and prediction error of adding in the dynamics as a regulization term is shown in fig. 2.6. The estimation and prediction errors are computed as time series over the estimation and prediction windows, at each value of the regularization strength γ:

$$\begin{aligned} \epsilon^n &= \left(q_{\text{est}}^{[\gamma],n} - q_{\text{true}}^n\right)^2 \quad \text{(estimation error).} \\ \Delta^n &= \left(q_{\text{pred}}^{[\gamma],n} - q_{\text{true}}^n\right)^2 \quad \text{(prediction error).} \end{aligned} \tag{2.37}$$

with similar definitions for v. Note that when γ is small, the estimates closely approximate the data, which is to say they inherit the noise in the data. This is apparent from examination of ε^n, which fluctuates rapidly over the estimation window (its average, though not shown, is approximately σ^2 which is the RMS value of the measurement noise in the data). As γ increases, these fluctuations disappear at a critical value where smoothing takes over. Note that at the top of each estimation error plot in fig. 2.6, which corresponds to the end of the estimation window, there is a sharp drop in the estimation error for a narrow range of γ values. The location of this local minimum is found to be dependent on the data, however, warranting a statistical analysis of these local minima locations.

If such an experiment is repeated many times, with different random instantiations of the measurement error, then this drop occurs frequently at an intermediate value of γ just below $\gamma = 200$. This is evident from examining the bottom panel in fig. 2.6, which shows the empirical distribution of γ values which minimize the estimation error at the end of the estimation window, ε^N, over the ensemble of measurement noise. However, this peak has a non-negligible width and a second peak is present at slightly higher values of γ.

The prediction error is similarly impacted by the choice of γ. A local minimum which is located near the peak of the distribution of optimal γ for estimation is observed to produce a prediction which is orders of magnitude more accurate than at neighboring γ. However, recall that the system in question is the simple harmonic oscillator. The motion of the system is periodic with no dissipation or instabilities, so while errors in the initial condition estimate for prediction do not grow exponentially in time (the case for chaotic systems), a slight error in the initialization translates to an oscillator with a slightly different amplitude and phase. However, this difference in amplitude or phase is propagated forward in time forever, so the prediction error never recovers if the initialization is wrong. Similarly, choosing the optimal γ value gives a prediction which never gets worse.

It should be noted that this value of γ does not reflect the size of the discretization error of the scheme used for time disctreization, shown in (2.28). This error, equal to $|\boldsymbol{g}|^2$, is several orders of magnitude smaller than the optimal γ^{-1}. Thus, one must consider a careful selection of γ for estimation and prediction beyond considerations of the numerical discretization scheme.

2.3.2 Dynamical regularization for nonlinear systems

The above analysis was conducted for a linear dynamical system, but many of the features of varying the regularization strength carry over to the nonlinear case. This is shown in numerical experiments in Chapter 3. One pathway to studying a nonlinear system in a similar fashion involves linearizing the model error to make it approximately in the form of a Tikhonov regularizing term. Starting from the nonlinear model error, where $\boldsymbol{g}^n$ is the model discrepancy at time t_n (for the Gaussian action defined in (1.24), $\boldsymbol{g}^n = \boldsymbol{x}^{n+1} - \boldsymbol{F}(\boldsymbol{x}^n)$), one may linearize the model error about a minimum of A, $X = X_0$:

$$\begin{aligned}\frac{R_f}{2}\sum_{n=1}^{N-1}\sum_{i=1}^{D}(g_i^n(\boldsymbol{x}^{n+1},\boldsymbol{x}^n))^2 &\to \frac{R_f}{2}\left[G(X)\right]^2 \\ &\simeq \frac{R_f}{2}\left[G(X_0) + G'(X_0)\left(X - X_0\right)\right]^2 \\ &= \frac{R_f}{2}\left[G(X_0) - G'(X_0) + G'(X_0)X\right]^2 \\ &\equiv \frac{R_f}{2}\left[G'(X_0)X + B(X_0)\right]^2 . \end{aligned} \tag{2.38}$$

Under the change of coordinates $Q \equiv G'(X_0)X + B$, the action in path variables may be written as

$$A = \frac{R_m}{2}\left[H(G'(X_0))^{-1}(Q - B) - Y\right]^2 + \frac{R_f}{2}Q^2 \tag{2.39}$$

and the (linearized) nonlinear problem is now in the form of Tikhonov regularization, but for the estimation of Q rather than X. The main hurdle in such a linearized analysis is that the regularized inverse to H is now dependent on X_0 itself, so that studying its properties requires minimization of A to find minima in the first place. Additionally, it may be required to expand the regularization term beyond linear order to properly capture the effects of nonlinearities in the model.

This provides the basis for the connection between dynamical regularization with linear and nonlinear systems. In the next chapter, it is seen that there are similar effects to varying the regularization strength R_f on estimation and prediction in a chaotic system. In fact, the method is even more sensitive to the choice of $\gamma \equiv R_f/R_m$ as the number of measured model variables decreases, and one finds a significant benefit to choosing the optimal value.

The analysis in the next chapter is performed entirely in terms of numerical experiments, but as a future research direction it may be fruitful to examine the problem using the construction presented above. This is because it is unknown exactly *why* introducing a particular number of measurements regularizes the system sufficiently to reduce the number of local minima of A and makes the global minimum so easy to discover through minimization. Additionally, it is not well understood why estimates and, thus, predictions are so sensitive to γ (especially for low L), where taking the deterministic limit $\gamma \to \infty$ is often highly detrimental. In the linear case, introducing the model was seen to have the clear effect on the regularized inverse of the measurement function of a recovery of the lost information associated with an unresolved dynamical variable. It is speculated that this kind of analysis of the nature of the measurement function itself in the nonlinear case would provide similarly valuable insight to answering these questions.

Chapter 3

Data Assimilation in Nonlinear Systems

In the previous chapter, data assimilation in a partially observed linear dynamical system was explored in the context of regularized inverse problems. A model of the observed system was used to cure the ill-posedness of inverting a rank-deficient measurement operator, acting as a projection of the full state of a linear oscillator onto its position variable. Even in the case where the *correct*, known model of the system was used, it was seen that the strength of the "dynamical regularization" term introduced to regularize the inversion, γ, had to be carefully chosen to optimize for prediction quality, rather than resorting to the deterministic limit $\gamma \to \infty$.

Using the variational data assimilation formulation discussed in Chapter 1, the model is introduced into the action function, A, in an equivalent manner to the more general procedure of Tikhonov regularization used to solve underdetermined linear inverse problems. This allowed for a detailed study of state estimation formulated in terms of a linear operator acting on measurements of the dynamical system, giving additional insight into the effect of the model on state estimates beyond examining the solutions in terms of estimation or prediction errors.

This chapter explores variational data assimilation for chaotic dynamical systems, in which case there is no equivalently simple reduction of the state and parameter estimation problem. We resort to numerical methods for finding estimates which minimize A, which will rely primarily on metrics that directly compare solutions to observed data. It will be shown, however, that these methods are still powerful tools for data assimilation, and similar considerations need to be taken into account to properly regularize the problem for optimal prediction quality.

First, some of the difficulties inherent in data assimilation in nonlinear systems will be discussed as background, motivating the use of an algorithm for minimizing A that reliably produces high-accuracy state and parameter estimates from data. This algorithm, referred to as "variational annealing" (VA) was introduced by Jack Quinn in his Ph.D. dissertation [69] and improved upon in subsequent work by Ye, et al [87, 88]. This discussion is warranted as VA is used heavily in the remainder of this dissertation. Next, numerical experiments will be used as examples to establish the efficacy of VA, as well as properties of the action, and state and parameter estimates themselves, as the inverse problem is increasingly regularized. Finally, a prediction-based metric will be introduced to improve the accuracy of VA solutions, especially when the number of observed variables is thought to be insufficient for reliable estimation, and to some degree when the model contains a large number of unresolved dynamical quantities present in the observed system.

3.1 Variational Annealing

In Chapter 1, the complex structure of the action function A for variational data assimilation was discussed as a consequence of introducing a chaotic model. The surface of A becomes more complex as R_f is increased [42, 47, 2] (thus introducing the model as a stronger constraint on solutions to $\underset{X}{\operatorname{argmin}} A$), posing a significant challenge to using variational methods due to the difficulties associated with numerical optimization of functions with many local minima in high-dimensional spaces [2]. Additionally, in chaotic systems the global minimum of A, if it exists, becomes increasingly isolated as the well structure around the minimum narrows and deepens. This is a consequence of the fact that chaotic systems are extremely sensitive to perturbations in their trajectory, as well as changes in model parameter values. Identifying the global minimum is in general an NP-complete problem [63].

Variational annealing (VA) is an algorithm designed to alleviate some of these difficulties by introducing the model error into A in a slow, controlled fashion. This algorithm is described extensively in [88, 87], in which it is shown to be an effective tool for characterizing the minimum structure of A, and for reliably finding low-action minima or, in the case of a twin experiment, the global minimum as long as enough variables are observed in the system. The algorithm is explained in this section, as well as the particular implementation details used throughout much of this dissertation. The example in the next section, in which VA is used to perform data assimilation in a Lorenz 96 system, will illustrate how the algorithm works in practice.

Recall the definition of the discrete-time action for data assimilation presented in Chapter 1:

$$A(X, Y) = \frac{R_m}{2} \sum_{n=1}^{N} \sum_{\ell=1}^{L} \left[h_\ell \left(\boldsymbol{x}^n\right) - y_\ell^n\right]^2 + \frac{R_f}{2} \sum_{n} \sum_{i=1}^{D} g_i(X)^2 \tag{3.1}$$

where, as a reminder, X is the path vector of model states at all times $t_n, n = 1, \ldots, N$ in the observation window, as well as model parameters $\boldsymbol{\theta}$; and Y is similarly the path vector of measurements in the observation window. In the model error term, the sum over n is intentionally left ambiguous, along with the full path vector X rather than $\boldsymbol{x}^n$ at any particular time(s) t_n, because $\boldsymbol{g}$ potentially acts on states at multiple times simultaneously depending on the choice of numerical discretization of the action integral. One potential issue for the numerics of minimizing A as it stands is that both terms grow with N, and L (measurement error) or D (model error), meaning that A could potentially grow very large as the system size increases. For the model error, this is assuming that R_m is chosen to be equal to the variance in the measurement error at a single time t_n. It will be shown shortly that in VA, R_f is essentially a free parameter, meaning that it can always be chosen to include N and D as normalization factors to accommodate for different system sizes. However, to maintain the connection with the constituent terms in the model error as a representation of the level of model error at particular times t_n, R_f is left independent of the system size, despite the equivalence of the two pictures in VA.

The terms in A are thus slightly modified to include extra normalization factors for N, L, and D:

$$A(X,Y) = \frac{R_m}{NL}\sum_{n=1}^{N}\sum_{\ell=1}^{L}\left[h_\ell\left(\boldsymbol{x}^n\right) - y_\ell^n\right]^2 + \frac{R_f}{ND}\sum_{n}\sum_{i=1}^{D} g_i(X)^2 \tag{3.2}$$

Now, when the "right answer" is plugged into A, the measurement error goes to 1 at the global minimum assuming that the errors in the measurement device are Gaussian, and that they were properly characterized by the experimenter. This is due to the fact that $\langle(\boldsymbol{x}_\ell^n - \boldsymbol{y}_\ell^n)^2\rangle = \sigma_{meas}^2$, so as long as one chooses R_m equal to the observed inverse covariance of the noise then the expected value of the measurement error is 1. This means that the measurement error is now independent of system size, which will be seen shortly to make the choice of "good" R_f values for VA easier for different system sizes.

Finally, because the location of the minima is unaffected by multiplying the action by a constant, it is instructive to examine solutions and action levels as a function of $\gamma \equiv R_f/R_m$, which indicates the relative size of the coefficients on the model and measurement error terms in (3.2). The action levels, as well as estimation and prediction errors, are generally plotted in terms of this ratio.

Now consider A when $\gamma = 0$, in which case there is effectively no model for the system. Minimizing A thus becomes a routine linear least-squares minimization, but *only for the observed variables.* The cost of setting $R_f = 0$ is that the minimization problem is now completely degenerate in the unmeasured variables and parameters, because there is no cost to altering their values. The solution is thus unique for the measured variables, which simply become equal to the measurements (A is positive-definite, and when $x_\ell^n = y_\ell^n$ at all times t_n, $A = 0$ uniquely), but the unmeasured variables and parameters may take on any desired value. Thus, while A is extremely easy to minimize when $R_f = 0$, the solutions are essentially worthless because they tell nothing more about the observed system than what is already known, namely the time series of observations Y.

One of the keys to the effectiveness of VA, which is a type of numerical continuation algorithm [3], is the observation that when R_f is non-zero but small, or more precisely $R_f/R_m \ll 1$, minimization of A is essentially linear least-squares regression with a slight correction coming from the model error term, thus estimates are not completely degenerate in the unmeasured variables and parameters. As R_f/R_m is increased, these minima which were computed using the "easy", nearly-least-squares action, are tracked as the position of the minima move in path space.

3.1.1 Variational Annealing Algorithm

Variational annealing is initialized by setting $R_f/R_m \ll 1$; practically speaking, it is generally found to be sufficient to set R_f/R_m to a value like 10^{-6} as long as the modified normalization of A shown in (3.2) is used. R_f is treated as a free parameter, and over the course of VA it is increased to a large value such that $R_f/R_m \gg 1$ (on the order of 10^8, practically speaking). The schedule under which R_f is increased is chosen by defining R_f as a function of an algorithmic hyperparameter β (essentially the algorithmic time for VA), specifically that $R_f = R_f(\beta) = R_f(0)\alpha^\beta$. $R_f(0)$ is the initial value for R_f chosen so that R_f/R_m is small at the beginning of the procedure. $\alpha > 1$ so that R_f increases as β is increased, and for many problems $1.1 < \alpha < 1.5$ has been found to be adequate. Choosing α to be too small or too large can be significantly detrimental; this will be elaborated upon following the definition of the algorithm presented below.

An initialization for the path variable of model states and parameters must also be chosen before starting VA. How to select a good initial path estimate X_{init} is described in more detail below, in §3.2.1. For the moment, it suffices to say that the measured states should be set equal to the values of the measurements in Y at all observation times, the unobserved states are drawn from random distributions selected to reflect the size of the model attractor, and initial parameter guesses from distributions that numerically encompass ranges of values thought to reflect the system's observed behavior.

Variational annealing proceeds by minimizing A with $R_f = R_f(0)$, and $X = X_{init}$. The choice of numerical optimization routine is up to the user.[1] The result

[1] Common choices may include a gradient-based method like nonlinear conjugate gradient; L-BFGS which is a quasi-Newton method; or a more sophisticated method such as IPOPT, an interior-point method for optimization with nonlinear constraint functions. §3.1.2 elaborates on the implementation details used for the numerical experiments in this dissertation.

of the optimization is the path estimate $X^{[0]}$, which is used to seed the next step of VA where optimization is performed by increasing R_f to the value $R_f(1) = \alpha R_f(0)$. Again, the result of this optimization seeds the next step, and VA proceeds until the final value $R_f(\beta_{max})$ is reached. β_{max} should be chosen according to the criterion described above, which is that $R_f(\beta_{max})/R_m >> 1$.

To summarize:

1. Set $R_f = R_f(0)$ and $X = X_{init}$.

2. Minimize A. Store the result; at step number β of the algorithm, this is the path estimate $X^{[\gamma(\beta)]}$.

3. Multiply R_f by α, and minimize A starting from $X^{[\gamma(\beta)]}$.

4. Store the result, and return to step 3. Terminate this loop when β reaches β_{max}.

It has been shown previously [88, 87] that following this procedure makes VA very effective at identifying global minima of A, and subsequently tracking them through large values of R_f/R_m. This is in contrast to optimization performed with hard constraints, exactly enforcing the dynamical model on estimated solutions, which creates an extremely complex cost surface over $\boldsymbol{x}^1$ (the initial state in the trajectory) and $\boldsymbol{\theta}$ arising from instabilities associated with the chaotic dynamics in the estimation window. Even if the minimization is weakly constrained, but R_f/R_m is not chosen to be small enough initially, the minimization cannot identify the global minimum with comparable frequency to using VA with a low initial value $R_f(0)/R_m$.

3.1.2 Implementation Details

For the numerical experiments described in this dissertation, the Hermite-Simpson (HS) collocation method [80] is used to discretize the model error integral.

HS is a high-order implicit discretization scheme which uses a midpoint approximation at each triplet of times (t_n, t_{n+1}, t_{n+2}). This gives rise to a specific definition for the model error term in the discrete-time action:

$$\frac{1}{ND}\sum_{n \text{ odd}}^{N-2}\sum_{i=1}^{D} R_{f,i}\left\{x_i^{n+2} - x_i^n - \frac{\delta t}{6}\left[f_i(t_n, x^n) + 4f_i(t_{n+1}, x^{n+1}) + f_i(t_{n+2}, x^{n+2})\right]\right\}^2 \\ + \left\{x_i^{n+1} - \frac{1}{2}\left(x_i^n + x_i^{n+2}\right) - \frac{\delta t}{8}\left[f_i(t_n, x^n) - f_i(t_{n+2}, x^{n+2})\right]\right\}^2. \tag{3.3}$$

Note that in this construction, the midpoint times t_{n+1} are actually observation times themselves. This simplifies matters somewhat for nonautonomous systems with external stimulus functions, because it is only required to insert values for measurements $\boldsymbol{y}^n$ and stimuli $\boldsymbol{s}^n$, corresponding to nonautonomous terms in the ODEs, into the action at the available observation times.

The numerical experiments themselves were carried out using VarAnneal [76], a publicly-available Python package. While VarAnneal was written by the author of this dissertation, it is primarily a front-end for computing derivatives of A using ADOL-C, a widely-used implementation of automatic differentiation written in C++, developed by Andreas Griewank and Andrea Walther [33] and maintained by the COIN-OR project [32]. The Python front-end for ADOL-C used by VarAnneal is PYADOLC [83], which is also a publicly-available software package. The optimization itself is performed using L-BFGS-B as implemented in SciPy; L-BFGS-B is a bounded version of the popular L-BFGS-B (**L**imited-Memory **B**royden-**F**letcher-**G**oldfarb-**S**hanno) algorithm, which is a quasi-Newton method that approximates the inverse Hessian of A rather than using a dense $N \times N$ matrix representation (hence, **L**imited-Memory). The details of this method are elaborated in [8, 89, 62].

Automatic differentiation, or AD [84, 65], is a popular tool in many fields of numerical optimization, used for computing derivatives of functions defined by complex computer codes in optimization problems. AD has the distinct advantages that it does not require computing symbolic forms for the derivatives of cost functions, but also does not use numerical approximations, so that derivatives may be efficiently computed to any desired level of precision. This is a big advantage to using AD in VarAnneal, because the action may be re-defined at the whims of the user with little to no cost in additional coding, due to the definition of A and its derivatives not being “baked in” to the code. In principle, this allows one to explore the effects of introducing different kinds of errors into the action. This is outside the scope of this dissertation, however, as all calculations are carried out using the Gaussian action defined previously with HS discretization for the model error. Instructions for installing and using VarAnneal are available in the GitHub repository in which it is hosted. In particular, it is worth noting that VarAnneal includes two versions of the code: one, used in this chapter and the next, for data assimilation in dynamical systems modeled by ODEs; and the second version for training neural networks, which will be used exclusively for the numerical experiments in Chapter 5.

3.2 Data Assimilation in a Chaotic System

The Lorenz 96 system was developed by Edward Lorenz in 1996 as a simple model for numerical weather prediction [57]. The variables in this model are meant to represent spatially-coupled atmospheric quantities which vary in time, such as temperature or the vorticity of atmospheric fluid rolls. Essentially, it was developed by Lorenz to study questions of predictability in complex dynamical systems. Despite the simplicity of the model’s definition, it exhibits chaotic behavior under certain parameterizations, and is thus widely used as a toy model in numerical studies of chaos and data assimilation [86, 14, 45].

The Lorenz 96, or L96, system is a polynomial ODE model defined in D dimensions by the following equations:

$$\frac{dx_i}{dt} = x_{i-1}\left(x_{i+1} - x_{i-2}\right) - x_i + K, \quad i = 1, \ldots, D \tag{3.4}$$

where the state index i is cyclic, so that $i = D + 1 = 1$. The parameter K represents a constant external forcing parameter, which takes on the same value for each state variable. It should be noted that this is a simplified version of the model originally proposed by Lorenz [57], which contained an additional set of "fast" variables coupled to the variables x_i in the equation above. This simplified version, however, still displays a large degree of complexity such as extensive chaos [45].

In this section, L96 will be used as an example system to display some general features of data assimilation in chaotic systems, specifically with the use of an the unconstrained path-integral formulation of DA presented in Chapter 1. Some of the results presented here have been previously studied in [88, 87, 47], revealing fundamental bounds on how sparsely such a system may be observed before state and parameter estimation becomes intractable. Namely, when the number of observed variables decreases below a particular level, finding accurate estimates appears to become unreliable enough that data assimilation is impractical.

However, it will be shown that there are ways to counteract this intractability to some extent by carefully studying the variational problem in terms of a prediction-based metric for choosing estimates. The result is that one should counterintuitively use solutions which do not strictly enforce a model of the observed system, even in a twin experiment setting where the underlying model is known exactly. In the next chapter, this methodology is briefly extended to a system with complex model errors arising from missing dynamical components (in fact, the "full" L96 system including fast and slow variables).

3.2.1 Action Structure and the Effects on Estimation

This problem will first be studied as a twin experiment, where it is assumed that the underlying dynamical model generating observed trajectories is exactly known, namely Lorenz 96 as defined in (3.1) with $D = 20$ and K known to be equal to 8.17. With this choice of model dimension and parameterization, the Lorenz 96 model is chaotic, which is evident from the presence of positive Lyapunov exponents (the full Lyapunov exponent spectrum is shown in figure 3.1). This fact is important to this discussion, which aims to explore some difficulties of the state and parameter inference problem in chaotic systems arising from the complex structure of the action.

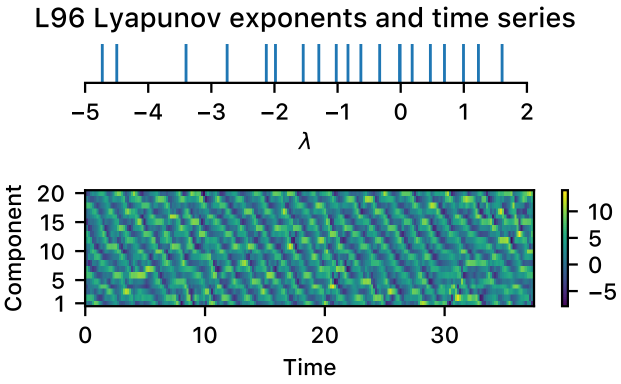


Figure 3.1: Lorenz 96 Lyapunov spectrum and phase space trajectories. Top panel: The presence of positive Lyapunov exponents indicates that the system is chaotic. Bottom panel: Chaos results in aperiodic behavior in the system.

A subset of $L < D$ variables are actually observed to generate the data. The observations are “direct” as defined in §1.13, meaning that the measurement function h is simply a linear projection operator $\mathbf{H} : \mathbb{R}^D \to \mathbb{R}^L$ with unit entries. It was previously shown in [47] that about 40% of the L96 variables must be observed in order to produce reliably accurate state and parameter estimates, meaning that one should expect some sort of threshold of observability near $L = 8$. This will be shown to have an effect on the minimum structure of the action, and thus the reliability with which accurate estimates can be found. The primary tool for exploring these effects is variational annealing, using a large number of state and parameter initializations for the algorithm.

Twin Data Generation

To produce twin data for this example, the L96 equations were integrated forward in time from a randomly chosen initial condition in the 20-dimensional state space, with D and K set equal to 20 and 8.17, respectively. As was discussed previously, this choice of D and K makes the system chaotic. The initial condition for twin data generation was chosen to lay within a box in the state space with side lengths of 20, and centered at 0. This corresponds to the normal bounds of the L96 attractor, thus in principle reducing the time required to integrate out transient behavior, as well as the risk of producing solutions outside of the attractor's basin of attraction. To approximate the forward integral, a widely-used fourth order explicit Runge-Kutta method [77, 51, 68], commonly called RK4, was used with a step size of $\Delta t = 0.001$. This integration step size is far lower than the time interval used to discretize the action integral for VA, which also uses the higher-order Hermite-Simpson (HS) rule. However, the discretization error in the HS method for the true solution, as well as the lowest-action estimates, is on average calculated to be approximately 10^{-7} which is much smaller than the typical γ^{-1} which optimized estimates or predictions.

After carrying out numerical integration of the model, the solution was saved every 25 time steps so that $\Delta t = 0.025$ for the data. This measurement frequency was chosen based on prior observations while carrying out numerical experiments for the work presented in [88, 87] that it is sufficiently small to find accurate state and

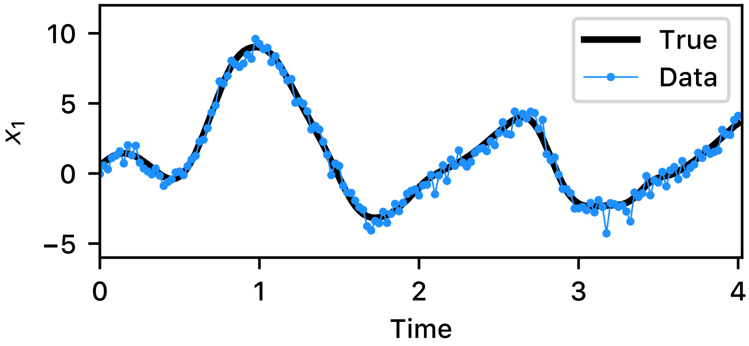


Figure 3.2: An example of time series data collected from a solution to the Lorenz 96 system for a twin experiment, representing data for just one out of L measured variables. The true solution is drawn in black under the measured data (in blue).

parameter estimates. Reducing the time step further was found to not appreciably improve estimation accuracy or the reliability of producing well-separated low action levels, and thus was chosen at this level as decreasing the time step needlessly increases the complexity of the optimization procedure by increasing the dimension of the path space. In order to simulate error in a measurement device, Gaussian noise was added to the states at each time point with a chosen variance of $\sigma^2 = 0.25$, where the noise was independently drawn for each variable at every time.[2] An example of such a time series of twin data for one observed Lorenz 96 variable is shown in figure 3.2.

The measurements in this example are direct observations of the system variables (defined in Chapter 1), giving rise to a measurement error term in A of the

[2] This i.i.d. property of the measurement noise avoids introducing additional complexities to the problem associated with correlated noise; an interesting question to address, but outside of the scope of this chapter, where the focus is on how L and the level of model error affect estimates and predictions.

form

$$A(X,Y) = \frac{R_m}{NL}\sum_{n=1}^{N}\sum_{\ell=1}^{L}\left[\sum_{i=1}^{D} H_{\ell i}x_i^n - y_\ell^n\right]^2 + \text{Model Error}$$
$$= \frac{R_m}{NL}\sum_{n=1}^{N}\sum_{\ell=1}^{L}[x_\ell^n - y_\ell^n]^2 + \text{Model Error} \tag{3.5}$$

and the model error uses the Hermite-Simpson discretization elaborated on in the previous section. For the sake of simplicity, the measurement function $\mathbf{H}$ will not appear explicitly in A in the case of direct measurements, and x_ℓ^n is used as shorthand for the measured variable corresponding to y_ℓ^n at time t_n instead.

Computing Action Levels with VA

Variational annealing was used to explore the cost structure of A for several values of L, namely $L = 5$, 7, 8, and 10. These values were chosen based on the observation for Lorenz 96 that approximately 40% of the system variables must be observed in order to produce reliably accurate estimates of all the states and the forcing parameter [47]. Thus, it is expected that a transition in the minimum structure of A should be observed as L is swept through these values, which will be seen shortly to be the case. The structure of the action for this problem was previously discussed in [88, 87], but the analysis is performed again here as it is important to the discussion of improving predictions later.

Each VA calculation was seeded by randomly choosing $N_{init} = 100$ different trajectories over the length of the observation window in the 20-dimensional phase space, as well as a value for the forcing parameter K. For the measured model components, the states were initialized to be equal to the data itself. The unmeasured components were randomly drawn from a uniform distribution defined over the closed interval $[-10, 10]$ in each variable direction; at each time point the unmeasured variables are independently drawn from these uniform distributions. The forcing parameter estimate was drawn from a uniform distribution on the closed interval $[6, 10]$.

These choices for the initializations are essentially motivated by a combination of some knowledge of the model dynamics and solution structure, as well as on empirical grounds. The state variable trajectories are observed to be bounded roughly within the range $[-10, 10]$; whereas the parameter range $[6, 10]$ encompasses a range of chaotic parameterizations for Lorenz 96, which is something that needs to be determined by actually studying the properties of solutions for varying K values. In a twin experiment, the true values of all these quantities are known, of course; but in the real world where data is collected by actually observing a system evolving in time, and the model is chosen on physical grounds, this sort of thinking must be employed in order to seed VA with reasonable estimates that are not hopelessly far from the truth.

The results of carrying out variational annealing N_{init} times are displayed in figures 3.3 and 3.4, which shows the value of A tracked over the course of the algorithm for all path vector initializations at the various chosen values of L. Also shown in some panels is the expected value of the action for the global minimum at large γ. Note that while minimization is carried out over discrete values of β, corresponding to action values at discrete γ values, the action levels are displayed as solid lines to track how different initializations jump between action values as γ increases.

The result is that a dominant action level only appears as L, the number of observed variables, approaches the threshold value $L = 8$. This separation is important if one wishes to approximate the discrete path space integrals $\mathrm{E}\left[G(X)|Y\right] = \int DX\, e^{-A(X,Y)} G(X) / \int DX\, e^{-A(X,Y)}$: as was discussed in Chapter 1, under the Laplace approximation, contributions to this integral are exponentially suppressed by the value

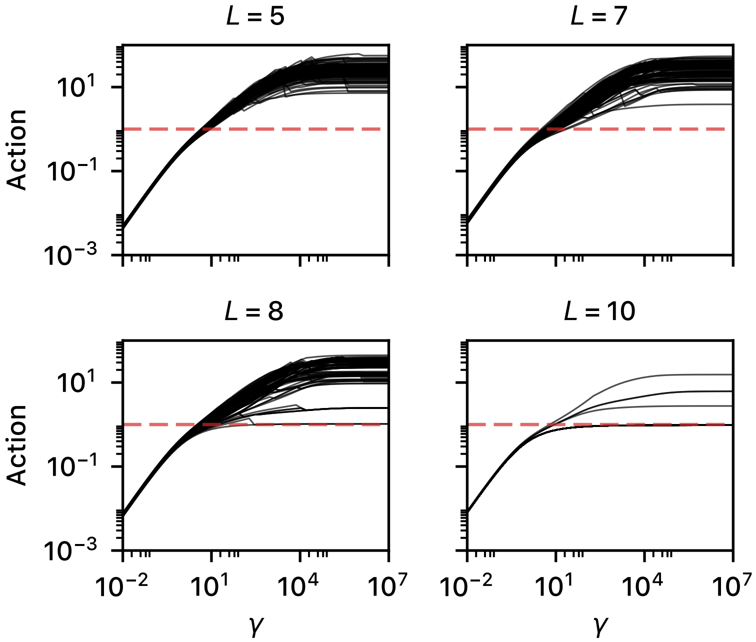


Figure 3.3: Action levels for the $D = 20$ Lorenz 96 system, with various L values corresponding to the number of measured state variables. Marked in red is the expected RMS value of the measurement noise for the true solution, which is reached by the global minimum estimate at high $\gamma = R_f/R_m$. Note the appearance of a distinct lowest action level when $L = 7$, just below the expected threshold of $L = 8$, at which point the lowest level splits off sufficiently from the higher levels to dominate the path integral for calculating estimated state and parameter statistics.

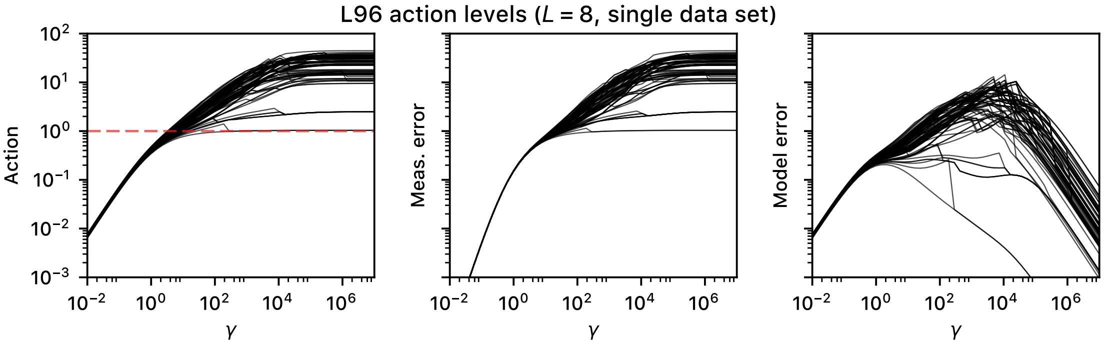


Figure 3.4: Detailed views of the action, as well as its constituent measurement and model errors, for the $L = 8$ case. At low $\gamma = R_f/R_m$, the action is dominated by model error, the result of estimated solutions closely tracking the data in the measured components, effectively oscillating with high frequency about the relatively smooth true solution. The measurement error dominates at high γ, which reaches the expected RMS value (in red) of the measurement error when the estimate "collapses" to the underlying smooth solution.

of A. This means that a well-separated, low-action estimate will dominate the approximation, and higher-action estimates may be neglected as their contribution becomes negligible. In the $L = 8$ case, this becomes an accurate approximation because the ratio e^{A_1}/e^{A_0} between the lowest level, with $A = A_0$, and the next-highest level with $A = A_1$, is slightly larger than 20 when γ is larger than 100 or so. Under the assumption that the observed system exactly obeys the $D = 20$ Lorenz 96 model, one should in principle use path estimates $X^{[\gamma]}$ for which $\gamma \gg 1$, in which case the estimate associated with the A_1 level is suppressed to contribute less than 5% to path integral estimates. Assuming that this level of accuracy is deemed to be sufficient, the estimates associated with the higher levels contribute ever more negligibly, allowing one to neglect them as well.

Note that the lack of separation of levels, which is observed in the $L = 5$ case, is not an entirely hopeless situation. In the next section, it will be shown that the situation can be improved significantly if one uses a procedure to select estimates at intermediate values of γ, rather than those corresponding to $\gamma(\beta_{max})$.

In order to glean some more information about the nature of the estimates as they are tracked over the course of the annealing process, the values of the measurement and model errors are displayed in figure 3.4 for the $L = 8$ case. Note that, at low values of γ, the action is dominated by the model error. In fact, this is expected to be the case: when $\gamma \ll 1$, $R_m \gg R_f$, corresponding to the measurements being enforced in the path estimates much more strongly than the dynamical model. The estimates of the measured variables closely track the data, so that at each measurement time, $(\boldsymbol{x}^n_{meas} - \boldsymbol{y}^n)^2 \simeq 0$.

The unmeasured variable estimates remain scattered about phase space, with estimated solutions exhibiting high-frequency oscillations over the estimation window. These high-frequency oscillations result from the measured variables tracking the noisy data, which appears to be a high-frequency signal oscillating about the true solution. This artificially introduces high-frequency components into the *entire* trajectory estimate, because the unmeasured variables are coupled to the measured ones through the dynamics. While this is logically consistent with expectations resulting from the structure of A, in the next section the estimates will be studied in more detail to provide a more convincing argument for the occurrence of this phenomenon.

Comparing State and Parameter Estimates in Different Levels

While examination of the action levels provide motivation for which action level estimates should be selected for using the Laplace approximation to compute statistics of path estimates, the estimates themselves are examined here in order to provide further justification for neglecting high-A estimates. Additionally, it will become apparent that if estimation errors for the states and parameters are tracked during annealing, new considerations appear regarding the value of γ at which estimates should be selected, even when the global minimum of A is found. This will motivate the topic of the next section, which establishes a criterion for selecting values for γ to produce optimal estimates and, ultimately, predictions.

These estimates are plotted in figures 3.5 through 3.9. In each case, the estimated trajectories for just one observed and unobserved variable (x_1 and x_2) are shown; the effects of varying L and γ on the rest of the states in this system are similar and thus omitted. In each figure, the true solution (from which the twin data was generated) is displayed, as well as the estimated trajectory corresponding to all of the action levels shown in figure 3.3. The lowest-action trajectory estimate is additionally highlighted for clarity in each case. There are several important qualitative trends for these estimates which shed additional light on why higher-action levels may be neglected, as well as why it is important to take care in selecting the value of γ from

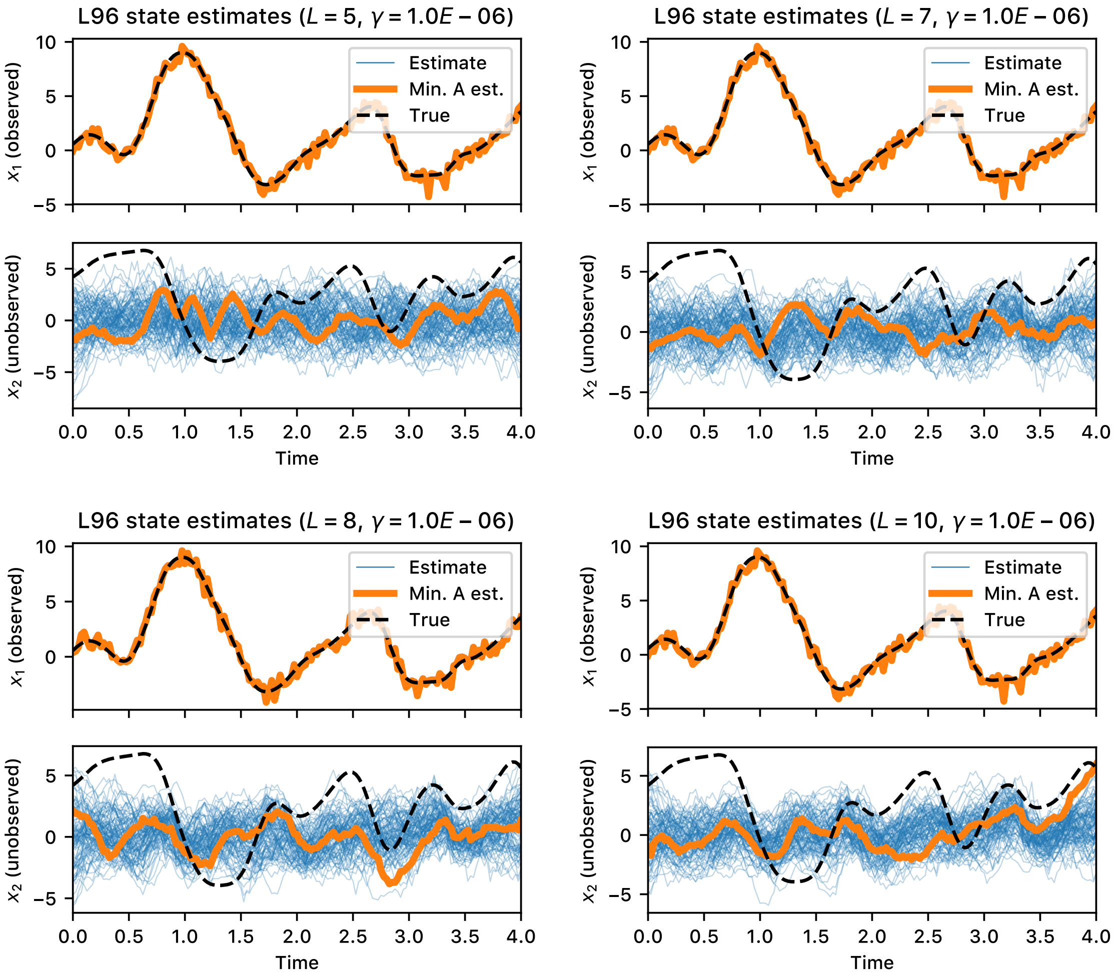


Figure 3.5: State estimates for partially observed Lorenz 96 with $\gamma \ll 1$, for various values of L (the number of observed model variables). The true solution is marked by dashed black lines, the minimum-action estimate in thick solid gold, and the higher-action estimates in thin blue. The model is weakly enforced in the action compared to the measurements, leading to the observed variable trajectories matching closely with data. The unobserved variable estimates have little to do with the true solution, and inherit some of the high-frequency oscillations associated with noise in the data. Additionally, there is little difference in the estimates for different L values, reflecting the lack of information transfer from data to the model regardless of the number of observations.

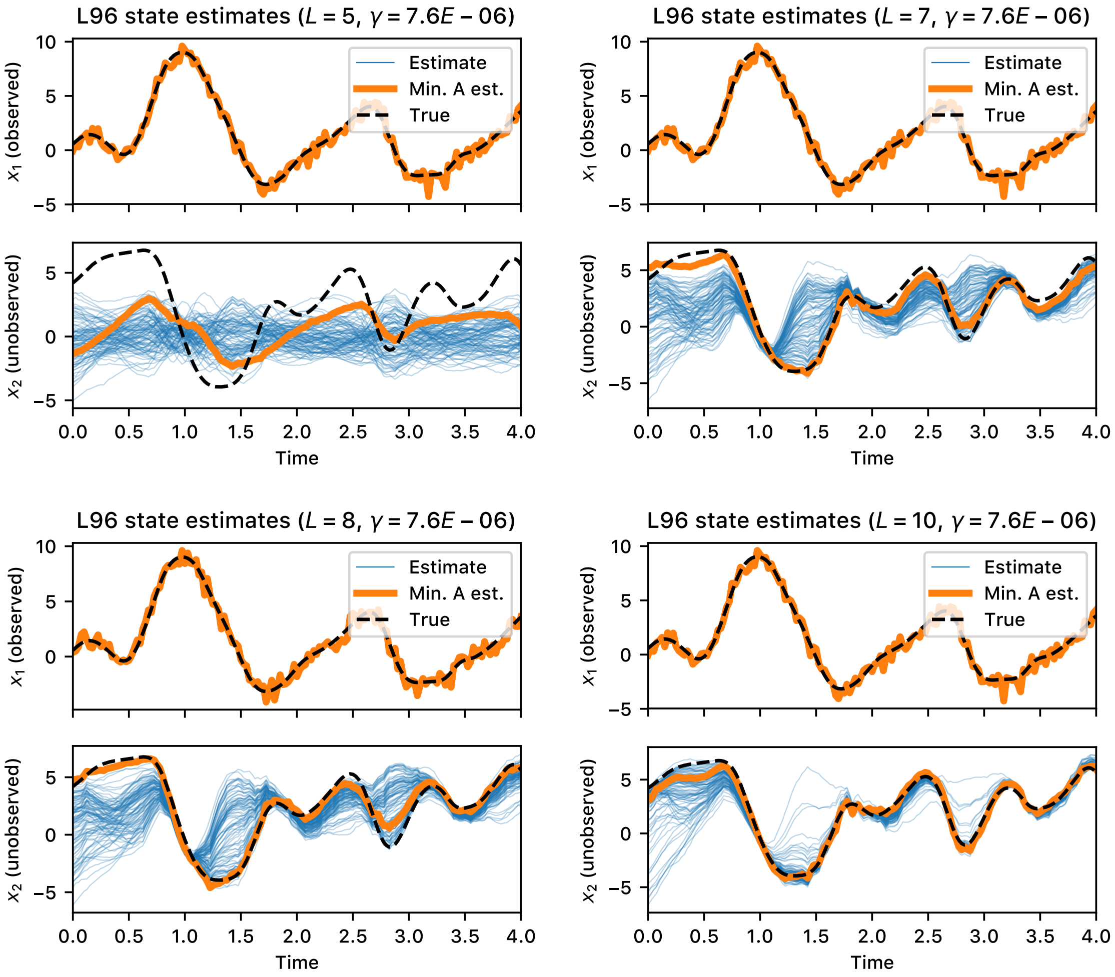


Figure 3.6: State estimates for partially observed Lorenz 96 for various values of L (the number of observed model variables), with γ increased slightly after a few steps of variational annealing. The true solution is marked by dashed black lines, the minimum-action estimate in thick solid gold, and the higher-action estimates in thin blue. The model error is slowly being introduced as a stronger and stronger penalty; estimates of the unobserved variables are beginning to more closely track the true solution, especially the lowest-action estimates. When $L = 5$, the estimation error is still significant, with the estimates at all action levels still having little to do with each other indicating a high degree of near-degeneracy of solutions. As L approaches 8, the minimum-action estimate begins to closely track the true solutions; a weaker trend appears of the higher-A estimates beginning to track the true state somewhat. When $L = 10$, even the higher action levels begin to track the true solution well near the end of the estimation window.

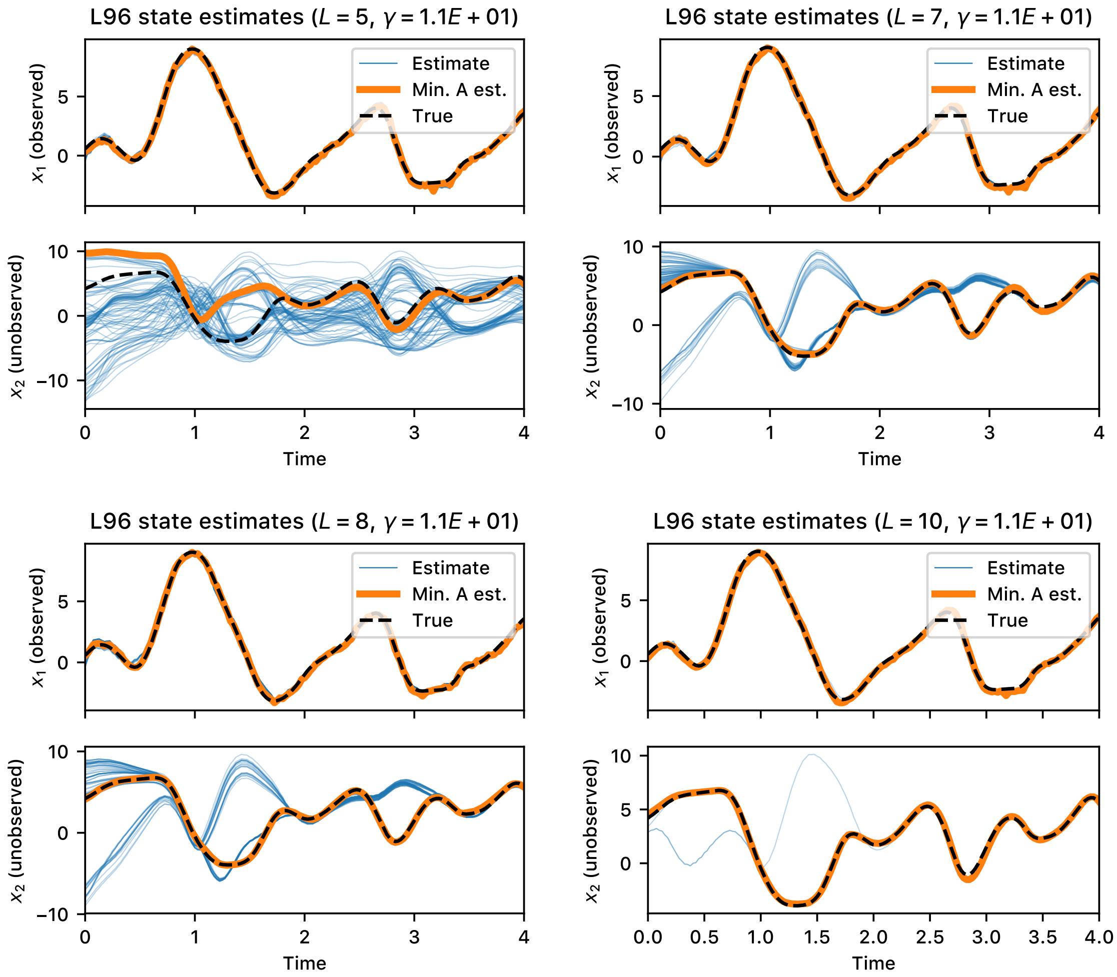


Figure 3.7: State estimates for partially observed Lorenz 96 for various values of L (the number of observed model variables), with γ approaching the value where the action begins to become independent of R_f/R_m. The true solution is marked by dashed black lines, the minimum-action estimate in thick solid gold, and the higher-action estimates in thin blue. When $L \geq 8$, the lowest-action estimate appears to be tracking the global minimum of the system, where the model error goes to zero and the measurement error is 1 (the RMS value of the measurement noise). Note that, when $L = 5$, the minimum-action estimate is actually tracking the true solution quite closely near the end of the observation window.

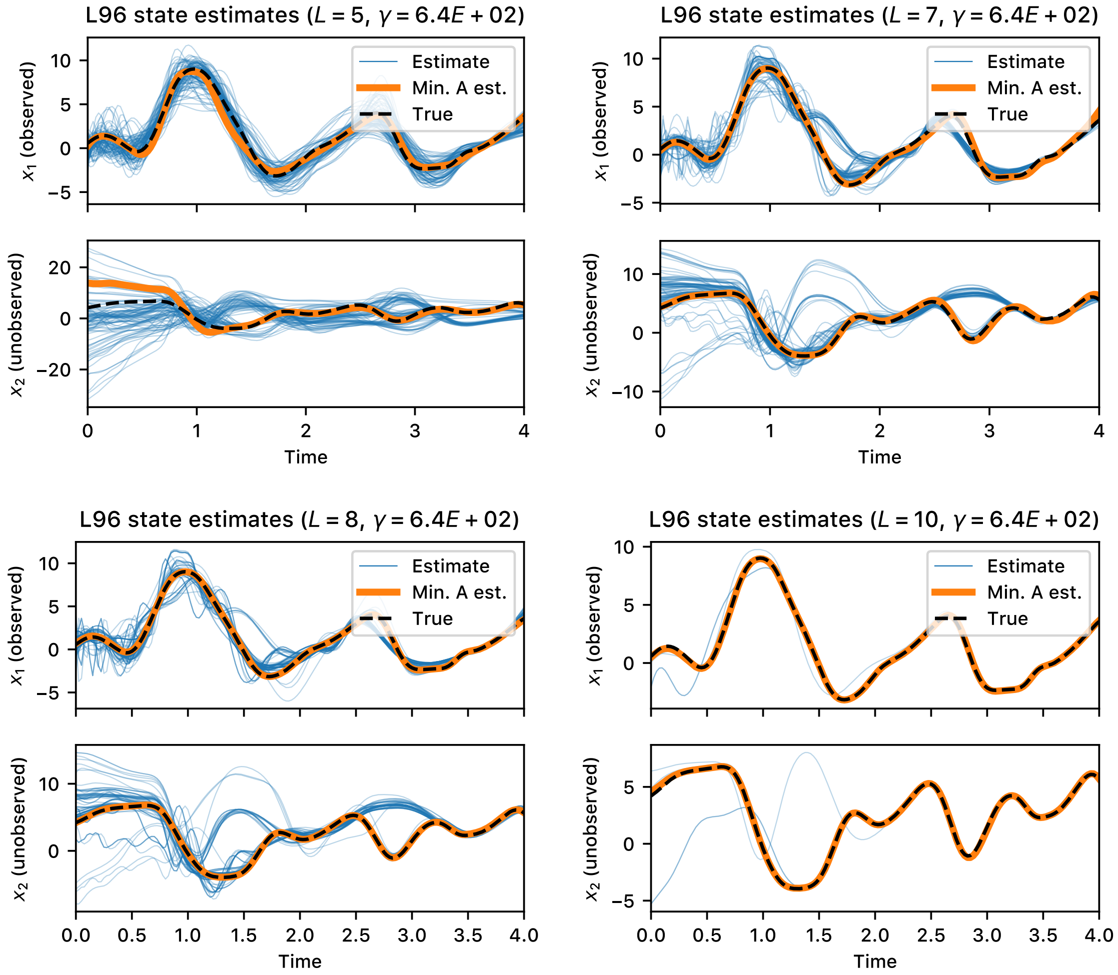


Figure 3.8: State estimates for partially observed Lorenz 96 for various values of L (the number of observed model variables), with γ approaching the value where state and parameter estimation error is often found to be at a local minimum, especially when $L = 5$.. The true solution is marked by dashed black lines, the minimum-action estimate in thick solid gold, and the higher-action estimates in thin blue.

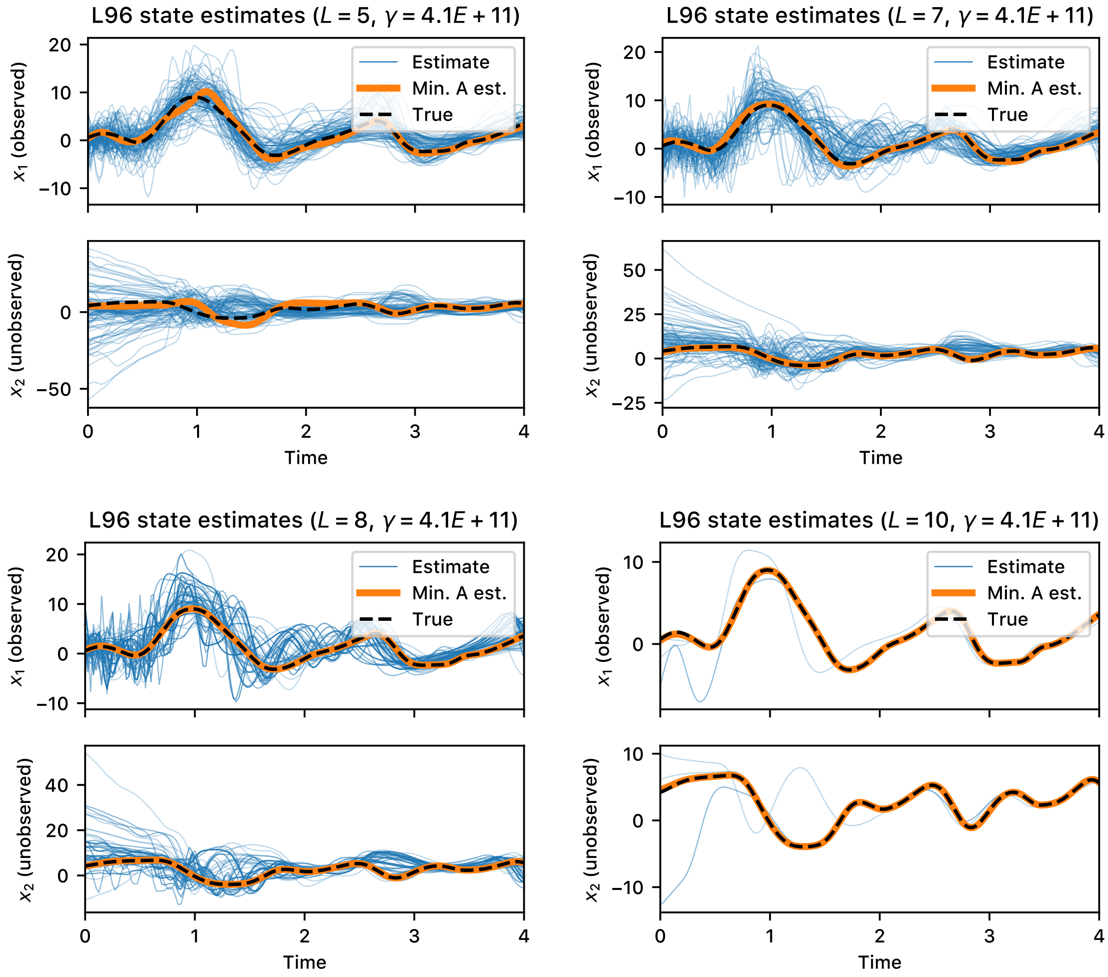


Figure 3.9: State estimates for partially observed Lorenz 96 for various values of L (the number of observed model variables), where γ has reached a high value and the model is strictly enforced through the model error. The true solution is marked by dashed black lines, the minimum-action estimate in thick solid gold, and the higher-action estimates in thin blue. This is the deterministic limit of the action, formally speaking. The minimum-action solutions continue to closely track the true solution; however, it will be shown that the estimates of K and $\boldsymbol{x}^N$ have been hurt in this limit compared to a more intermediate value of $\gamma \sim 10^3$.

which estimates should be drawn in order to maximize prediction quality.

First consider the effect of different values for L in each case. Focusing on estimates corresponding to a single value for γ, it is apparent that when L is low, there is significantly more degeneracy in the solutions than for high L values. This is indicated by the tangle of estimates corresponding to the higher action levels, which are nearly valid trajectories over short stretches of time, but often stray significantly from the true solution which is tracked much more closely by the lower-action estimates. This explains why the action is higher for these sometimes-correct estimates. The model error is accumulated over the course of the estimation window; sections of the trajectory which differ significantly from the true solution as they tend to violate the model, whereas the portions of time over which estimates are nearly correct contribute negligibly to the error.

While in many cases these higher-action trajectories approach the true solution near the end of the estimation window, and thus may appear to be usable for the purposes of prediction because they provide an accurate seed for forward integration, the *parameter* estimates have not been considered and, in fact, these higher-action solutions tend to produce poor predictions in comparison.

On the other hand, examining the estimates as a function of γ reveals the effects of introducing the model error term in A with increasing strength. When $\gamma = \gamma(0) = 10^{-6}$, the observed variable estimates closely track the data as expected. Note that the data is not explicitly shown in these figures, but this tracking effect is apparent from noting that the estimates themselves oscillate rapidly about the true solution. The unmeasured variable estimates are essentially nowhere near the true solution, and in some cases exhibit oscillatory behavior of a higher frequency than is characteristic of the Lorenz 96 solutions with this choice of parameterization ($K = 8.17$, as a reminder). Also note that increasing L does little to help state estimation quality; the model error is simply too weakly enforced for a significant amount of information to be transferred to the model.

As γ increases, as shown in figure 3.6, the effect of introducing the model error becomes apparent as the trajectories of the unmeasured state variables begin to more closely track the true solution. This is somewhat true even for the high-action levels when $L \geq 7$, and when $L = 10$ these estimates come reasonably close to the true solution even just a few steps into the annealing algorithm, when $R_f/R_m \sim 7.6 \times 10^{-6}$. Note that the measured states continue to closely track the data, because the measurement error penalty is still relatively large compared to that for the model error. Signs of this are exhibited in the unmeasured variable estimates, which still contain some high-frequency oscillatory components over portions of the estimation window.

When γ reaches a value of approximately 10, shown in figure 3.7, the measured variable estimates are seen to be nearly smoothed and close to the true solution, which is the effect of the model error becoming a significant penalty comparable to the measurement error. Also note that the near-degeneracy of estimates in the higher action levels has been significantly reduced, especially for $L \geq 7$ where it almost becomes possible to count the number of distinct estimates. By this point, when $L = 10$ there are essentially just two distinct solutions present in the x_2 trajectory (this may seem inconsistent with the action level plots shown previously, but remember that there are 18 more variables in the system which are not displayed in these figures, meaning that the other two solutions are still present but not visible upon inspection of just x_1 and x_2).

What is perhaps surprising is that this degeneracy actually appears to *increase* when γ reaches higher values, especially at the end of the annealing procedure when $\gamma \sim 10^{11}$. The lowest-action estimate, however, remains close to the true solution even as the rest of the estimates begin to degenerate. This is in fact one of the primary advantages to using variational annealing: as long as α is chosen to be small enough, VA is able to effectively track estimate as R_f is increased, in particular the global minimum estimate if L is large enough.

This analysis has established some of the features of state estimation as more measurements are introduced into the action, as well as some of the effects of varying R_f, the coefficient of the model error term. In particular, it is now more apparent that high-action estimates may be neglected for estimating the discrete path space integrals for computing estimate statistics when the threshold value of $L = 8$ measurements is reached. Additionally, despite the fact that when $L = 7$, the global minimum estimate corresponding to tracking of the true solution is not found by VA, there are signs that the lowest level estimates may still be usable for prediction purposes. Finally, increasing R_f is shown to first produce estimates which tend to collapse towards, and then closely track the true solution; however, there are signs that beyond a certain point the estimates are negatively impacted. This is readily apparent by visual inspection of the high-action estimates, but not immediately so for the lowest-action estimate.

A more careful analysis of the state and parameter estimates is required to determine if, in fact, the minimum-A estimate is significantly impacted by increasing R_f beyond a certain point. In the next section of this chapter, this is carried out by examining the estimation errors of the lowest action levels as a function of γ. In some cases the result will be surprising, leading to the final portion of this chapter in which a new criterion for estimate selection is introduced, using a statistical analysis of the estimates for many data samples collected from different trajectories on the Lorenz 96 attractor.

3.3 Statistical Approach to Optimizing Predictions with Sparse Observations

To begin this discussion, the model state and parameter estimation errors are tracked during the annealing experiments carried out for the Lorenz 96 system described in the previous section. In particular, these errors are always tracked for the estimates corresponding to the minimum-action levels. When $L = 8$ or greater, this is justified by the fact that these estimates, which in fact are the global minima of the action, clearly dominate the Laplace approximation to discrete path space integrals. For smaller L, this is also found to be a fruitful analysis: even though the lowest-action estimates are not found to be clearly separated from the high-A ones (or, in the case of $L = 7$, the lowest level does not correspond to the global minimum of A), it will be shown that there is in fact a significant advantage to choosing estimates with a particular R_f value that overwhelmingly tend to produce estimates that significantly improve prediction accuracy.

When variational annealing is used for data assimilation, this provides the distinct advantage that, even if other estimates are not considered in the Laplace approximation, one may improve predictions by choosing the "right" value for R_f. This is advantageous for practical reasons, because computing the series expansion in the Laplace approximation requires separately computing the inverse Hessian of A, as explained in Chapter 1. This computation is costly in terms of calculation time and memory requirements, especially as the system size grows large: inverting a dense Hessian matrix of size $M \times M$ is an operation with a time complexity of $\mathcal{O}(M^3)$. From a theoretical perspective, however, these other estimates cannot be ignored because they *do* contribute significantly to path space integrals. The suggestion of the author, then, is to further develop this as a hybrid method, where considerations of optimal

R_f values are combined with improving the accuracy of the Laplace approximation by including higher action level estimates.

3.3.1 Tracking Estimation and Prediction Errors in Variational Annealing

The state and parameter estimates corresponding to the lowest action level at every value of γ during annealing is henceforth referred to as $X^{[\gamma]}$. While there are, of course, N_{init} different estimates at every γ, it is to be assumed that the lowest action estimate is selected for the purposes of prediction; thus, $X^{[\gamma]}$ is to refer to the lowest-action estimate without ambiguity.

In addition, rather than separately considering the estimation errors for each L value separately, only the case when $L = 8$ is analyzed here. The lessons learned from the $L = 8$ results are similar enough in each case, and later in the discussion the *prediction* errors will be studied in greater detail to lead to the final result, in which the statistics of estimates and predictions for a large collection of data samples are used to motivate the R_f-selection criterion.

There are actually multiple estimation and prediction errors to consider. In a more realistic setting, where the only data available to an experimenter is the measurements of the state trajectories in the observation window, the only errors that can actually be calculated are the errors which compare predictions to the data (an error function comparing the measured states to data is not defined here; this is just the value of the measurement error, which is already tracked during annealing). However, when conducting such an analysis in a twin experiment setting, the true state of the system is known for all of the model variables, as well as the true parameter values. The estimation errors compared to the true state of the system are thus examined in order to provide greater insight into how γ affects these errors, as well as stronger validation for the method itself.

These errors are defined as follows:

$$\begin{aligned}
\text{State estimation error:} \quad \varepsilon_x^n(\gamma) &= \frac{1}{D}\sum_{i=1}^{D}\left[(x_{\text{est}}^{[\gamma]})_i^n - (x_{\text{true}})_i^n\right]^2, \\
\text{Parameter estimation error:} \quad \varepsilon_\theta(\gamma) &= \frac{1}{D_\theta}\sum_{i=1}^{D_\theta}\left[(\theta_{\text{est}}^{[\gamma]})_i - (\theta_{\text{true}})_i\right]^2, \\
\text{Data prediction error:} \quad \Delta^n(\gamma) &= \frac{1}{L}\sum_{m=1}^{M}\sum_{\ell=1}^{L}\left[(x_{\text{pred}}^{[\gamma]})_\ell^{n-m} - y_\ell^{n-m}\right]^2.
\end{aligned} \tag{3.6}$$

The state and parameter estimation errors, as their name implies, are equal to the squared discrepancies between the estimated states and parameters and their true values. The prediction error compares the value of a prediction made using the estimate of the full model state at the end of the estimation window, $(\boldsymbol{x}_{\text{est}}^{[\gamma]})^N$, as well as the estimated parameter values $\boldsymbol{\theta}_{\text{est}}^{[\gamma]}$, to the *data* observed in the prediction window, which is actually the only information available in a real experiment. Additionally, the state variable errors are computed as time series rather than averaging over time; in the prediction error, this allows one to readily see at which point the prediction quality begins to decay. They are all implicitly functions of γ, as well, because the values of the estimates depend on γ; the predictions thus are implicitly functions of γ, too, because predictions are calculated by integrating forward in time from the estimated state at the end of the observation window, $(\boldsymbol{x}_{\text{est}}^{[\gamma]})^N$, as well as the estimated parameter values $\boldsymbol{\theta}^{[\gamma]}$.

In these errors, the definitions of N, L, and D are already known to the reader, while D_θ is the number of model parameters. The sums over m, which are sums over time, are introduced to smooth the errors when comparing estimates or predictions to noisy data. It is instructive to introduce this smoothing so that the errors when comparing to data do not oscillate rapidly in time, making the comparison closer to that with the (noiseless) true states. Finally, N_{pred} is the number of time points beyond the estimation window at which predictions are compared to data or the true solution. Predictions always start at $n = N + 1$ (the first time point beyond the estimation window) and terminate at $n = N + 1 + N_{\text{pred}}$; prediction errors are also defined starting at $n = N + 1$ and up to $n = N + 1 + N_{\text{pred}}$.

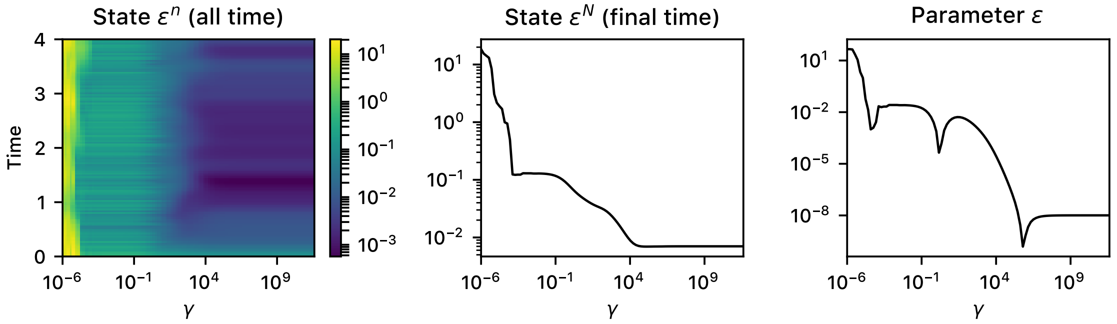


Figure 3.10: State and parameter estimation errors for the lowest-action path estimate in the $D = 20$ chaotic Lorenz 96 system, when $L = 8$. Left panel: time series of state estimation errors, as a function of $\gamma = R_f/R_m$. Center panel: state estimation error at the end of the observation window; there is a barely-perceptible local minimum near $\gamma = 10^4$; in other instances with a different data set Y, this minimum is more pronounced. Right panel: the parameter estimation error; the local minimum just above $\gamma = 10^4$ will often translate into greatly improved prediction accuracy.

Figure 3.10 shows the state and parameter estimation errors for the lowest-action estimate in the $L = 8$ case. These errors are computed using the same data set that was used for annealing in the previous section, where all of the computed action levels are shown in figure 3.4. The general features of the state estimation error surface are expected when the analysis performed in the previous section, in which state estimates were compared to data and the true solution, is taken into consideration. At small γ values, the estimation error is relatively large because the measured variable estimates match the data rather than the true solution, thus contributing an expected value of 1 (the RMS value of the measurement noise); but more importantly, the unmeasured variable estimates bear little resemblance to the true solution, which is why the error is closer to 10 near the smallest γ values. Note that this error is also fairly constant over time. This is also unsurprising because the dominant contributions to the error come from the unmeasured states, and with $\gamma \ll 1$ the model is weakly enforced, so one should not expect there to be any sort of structure in the time direction where the model forces estimates closer to the true solution at some times compared others.

As γ increases, the estimation error surface undergoes a transition as γ is increased past 1. The error becomes much smoother in time, and its value drops significantly (in some cases by several orders of magnitude). This is the result of the measured variable estimates collapsing towards the true solution, which contributes to the smoothing effect as well as the decrease in the value of the error; the unmeasured state estimates also tend to approach the true solution as γ approaches 1. The surface appears to be relatively constant as γ is increased further; however, upon closer inspection it will be seen that there is a local minimum in this error surface for an intermediate value of γ, indicating a state estimate that will perform better for the prediction stage.

The state estimation error is separately shown in figure 3.10 at the end of the estimation window. The general features described in the previous paragraph hold here as well, but the estimation at time N is “special” because it will be used to seed a prediction forward in time, thus warranting closer inspection.

The local minimum feature for the parameter estimation error appears to be much more drastic in comparison. The error drops by almost 10 orders of magnitude from the beginning of the annealing process, and contains fairly narrow local minima as a function of γ. As a reminder to the reader, however, this is a squared estimation error, so the absolute value of the difference between the estimated parameter value actually varies between approximately 3 (at the start of annealing) down to about

10^{-4} at later stages. Additionally, the error in the local minimum located near $\gamma = 10^5$ corresponds to a difference in the absolute value of the error which is approximately 10^{-4}.

These observed local minima in state and parameter estimation error may appear to be small enough to be ignored. However, one must remember that this is a *chaotic* system, which by definition is highly sensitive to perturbations in the state of the system as well as its parameter values.

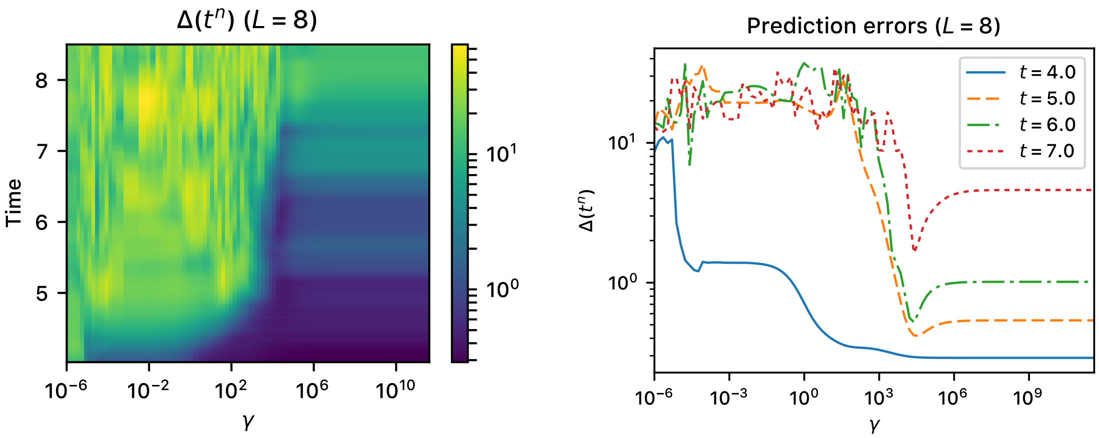


Figure 3.11: Prediction error surface for $L = 8$ observed variables in the $D = 20$ chaotic Lorenz 96 system. Left panel: prediction error surface as a function of γ and prediction time. Right panel: constant-time slices of the error. Note the presence of a local minimum just above $\gamma = 10^4$, where prediction error is reduced by a factor of 2 or 3 compared to those made with estimates at higher γ.

Examining the prediction errors as a function of γ in fact reveals that these local minima have a significant enough effect that one *must* pay careful attention to their existence. In figure 3.11, the prediction error surface compared to the data in the prediction window is shown, along with constant-time slices so that the values of the errors may be determined more precisely. Prediction quality is poor when seeded using small-γ estimates; this is expected because the estimation error was high in this regime for both the state variables and parameter value. When $\gamma \gg 1$, the prediction quality is significantly improved; for the first second or so, Δ is below 1, corresponding to a discrepancy of less than 5% compared to the true solution (in absolute value terms).

Perhaps the most interesting feature of this prediction error surface is the long, extended local minimum "valley" in γ near 10^4. Not coincidentally, this is the same value at which a local minimum in estimation error was found previously. What is perhaps surprising here is not that there is a local minimum in prediction error near this value, but rather the time duration over which this minimum is maintained. This is, however, consistent with the notion that trajectories in chaotic system are overly sensitive to errors in the initial state of the trajectory and the model parameter(s). These errors grow exponentially in time at a rate determined by the largest Lyapunov exponent of the system. Thus, if the estimation procedure is able to "nail it" and find just the right value for the state initialization and parameter value, then one should expect this prediction to match the observed trajectory well for a long time in comparison to even closely neighboring initializations.

Similarly, the prediction error surface for the other L values exhibit this local minimum feature to one extent or another. These are shown in figure 3.12. It remains fairly pronounced for the $L = 10$ case, but the error also remains low at larger γ values. This reflects the fact that introducing more measurements into the action transfers more information into the model through data assimilation, thus enhancing the estimate precision. However, the picture appears to be even more interesting when L is *smaller* than 8: the local minimum in Δ is much more localized when $L = 7$ compared to $L = 8$, so that predictions quickly lose their accuracy when γ is not chosen within this narrow well. When $L = 5$, predictions are mostly of very low accuracy across the surface of Δ. However, it should be noted that a local minimum still exists just below $\gamma = 10^3$, even though it exists for a relatively short period of prediction time. Despite its small size, this feature is still interesting if one considers

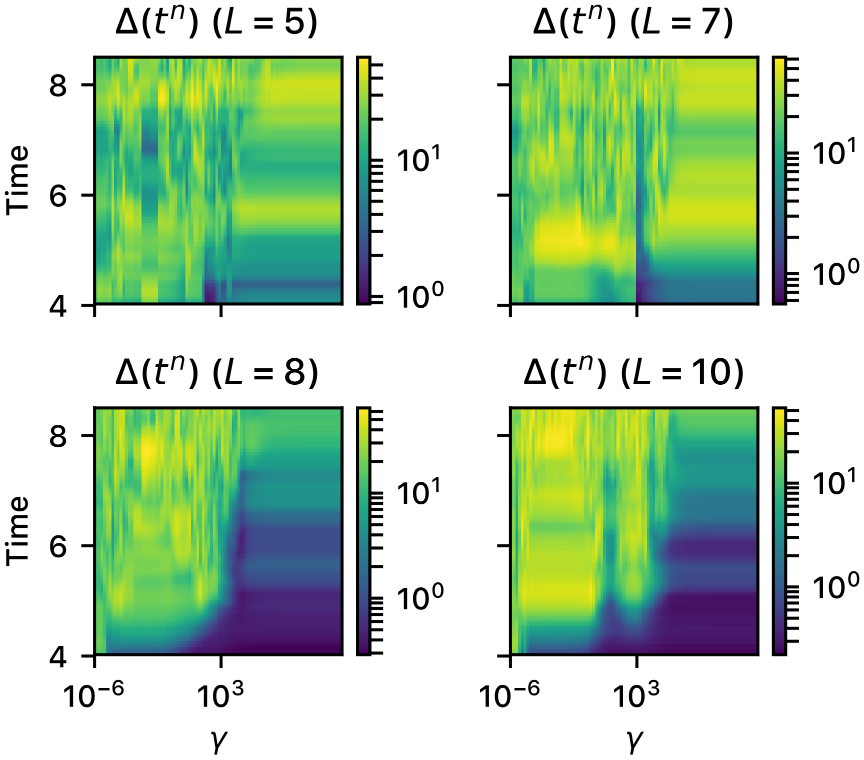


Figure 3.12: Prediction error surfaces, all L values. When $L < 8$, it is seen that there may be an even more significant advantage in choosing an estimate from the "right" choice of γ compared to higher γ values. When $L = 7$, the local minimum in prediction error stretches much further into the prediction window, and the disadvantage to choosing higher-γ estimates is more pronounced. When $L = 5$, predictions do not stretch very far in time with reasonable accuracy, but the local minimum is still present and, it will be shown, consistently located.

that this sort of analysis could be applied to an iterative data assimilation method, in which new measurements are incorporated to improve on previous state estimates as they become available. In other words, improving prediction accuracy for even a very short period of time is worthwhile, especially if one is stuck with the situation where only a small number of variables are practically observable.

3.3.2 Finding the Optimal Regularization Strength

In the previous section, it was shown that the errors in state and parameter estimates, as well as predictions, are sensitive to the choice of R_f in the action. In particular, these errors were shown to have a local minimum in R_f at a value which (numerically speaking) is far below the limit of $R_f/R_m = \gamma \to \infty$. The local minima in state estimation errors were not particularly deep, but importantly were highly pronounced in the parameter estimation error ε_θ. This led to local minima in the prediction error that were found to extend significantly further in prediction time than the errors in predictions computed using estimates from neighboring γ values. This minimum appeared to be especially pronounced and localized when $L = 7$, below the expected threshold of $L = 8$ for producing useful estimates to seed predictions.

However, these error surfaces, in particular the locations of local minima, are data-dependent. This means that if a different observed trajectory from the same system is used for state and parameter estimation, which in the twin experiment translates to using data generated by sampling a trajectory from the attractor at some earlier or later time, the value of γ at which these local minima are located will change. Thus, there is not just one “perfect” value of γ that can be used for prediction; rather, it is more useful to study the system to develop *probability distributions* for optimal γ values, and use them as a guide for selecting estimates at locally optimal γ given a new data set.

These distributions are empirically determined by repeating the above analysis many times for $S = 100$ different data sets Y, and in each case recording the optimal γ values for 1) minimizing state estimation error across the entire observation window, 2) minimizing parameter estimation errors, and 3) minimizing prediction error over the entire prediction window. The remaining task essentially amounts to binning these values in a histogram to generate an empirical probability distribution of optimal γ values for prediction. Actually, to be more precise, the empirical distributions are defined using a kernel density estimator, or KDE, with a Gaussian kernel function:

$$\text{Gaussian KDE:} \quad P(\gamma) = \frac{1}{S\sigma\sqrt{2\pi}} \sum_{s=1}^{S} e^{-(\gamma - (\gamma^{(opt)})_s^n)/2\sigma^2}. \tag{3.7}$$

where $(\gamma^{(opt)})_s^n$ is the optimal γ value found for data sample s (out of a total of S examples) at time n in the estimation or prediction window (the parameters are static in time, so the n index may be dropped in this definition). Choosing σ, or the "bandwidth" of the KDE, is essentially equivalent to choosing the bin width in a regular histogram. Various methods exist for choosing σ [72, 82, 50]. A KDE is used rather than a histogram because it is defined by a smooth, integrable function. Thus, the resulting distribution may be evaluated at any value of γ, and is also integrable against other smooth functions.

The end result of this is shown in figures 3.13 and 3.14, which display the KDEs of the empirically determined distributions of optimal γ values for state estimation and prediction. Examining these distributions, it is clear that the local minima in estimation and prediction errors found when examining a single data set was not a one-off phenomenon. In each L case, there is a clear peak in distributions for optimal state estimation and prediction at intermediate γ values; this holds true for the parameter estimation error as well. An additional peak in the state estimation

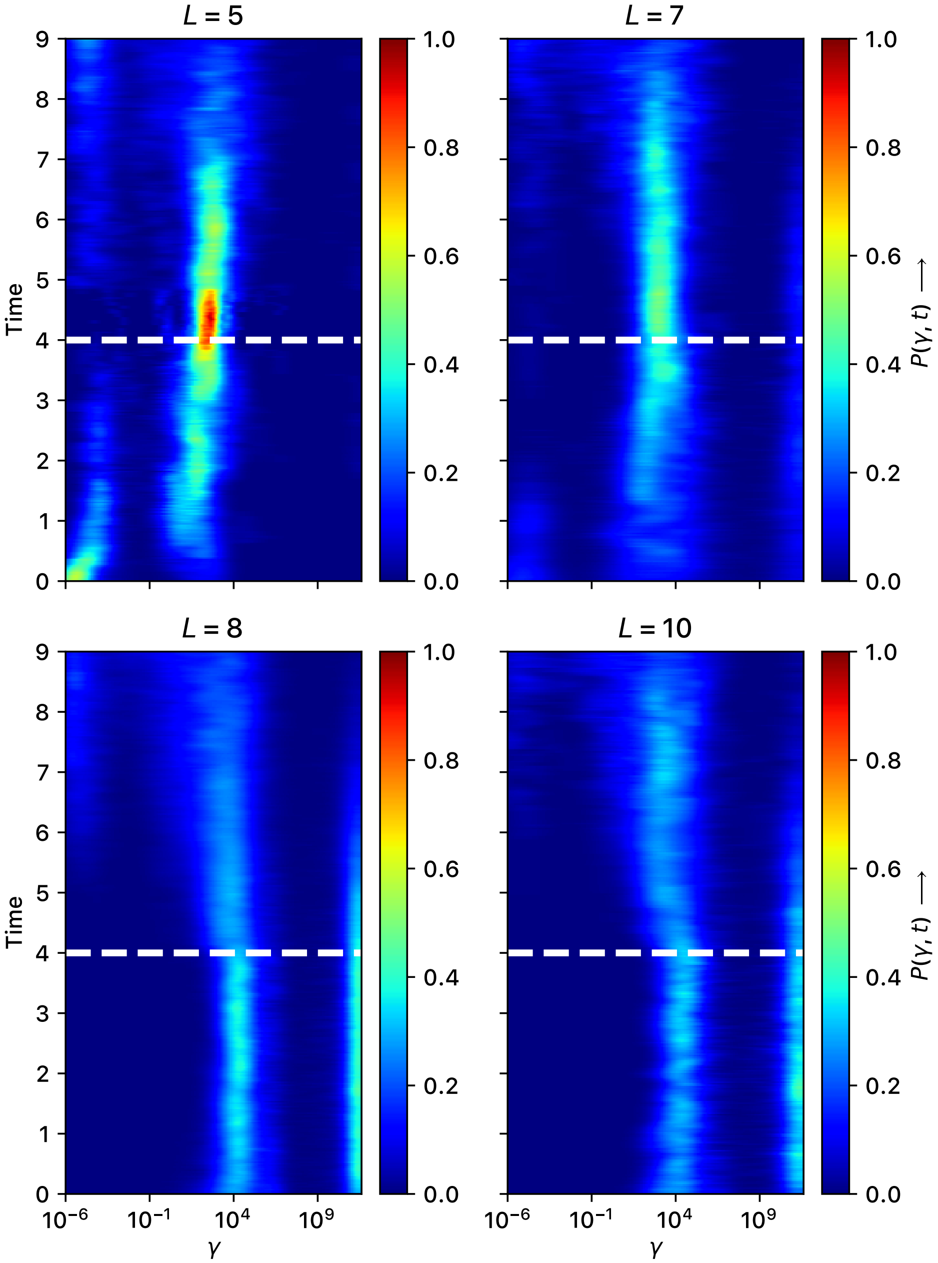


Figure 3.13: Empirical distributions of optimal γ values for model state estimation and prediction. Going clockwise from the top left panel: $L = 5$, 7, 10, and 8. At higher L values, where enough variables have been observed to consistently identify the global minimum of the system, it is about equally likely that the optimal γ value for estimation will be near 10^4, and near a much larger value ($\sim 10^{11}$). The peak near 10^4 becomes more dominant in the prediction window, however, As L decreases, this optimal value is more reliably located at intermediate γ values, with the high-γ peak essentially disappearing when $L = 5$.

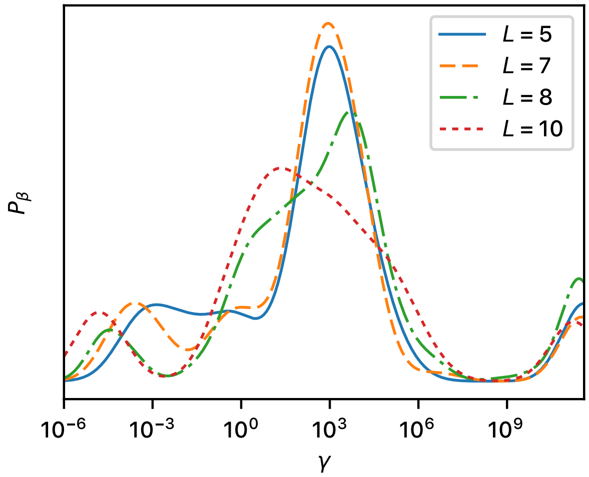


Figure 3.14: Empirical distributions of optimal γ values for parameter estimation. This is shown for reference to compare with the result in figure 3.13. As L decreases, the optimal γ value is more and more reliably located near 10^4, although the peak at high γ is similar at all L values. This explains the discrepancy between the high-γ peak in the estimation windows for $L = 8$ and 10, which disappears much more quickly than the intermediate-γ peak in the prediction window.

errors is observed at high γ values when $L = 8$ and 10, an indicator that a sufficient number of state variables have been observed to fully resolve the model, so that it is often also likely that the $\gamma \to \infty$ limit is appropriate. These peaks seem to disappear for lower L values, in which case the global minimum of the system was not identified. In fact, the intermediate-γ peak is strongly dominant in the $L = 5$ case, where the high γ peak seems to have all but disappeared.

The peaks in the parameter estimation distributions were similarly more well-localized for smaller L values. In addition, for all L cases the peak at high γ appeared to be much smaller compared to the intermediate γ peak. This is in contrast to the state estimation error distributions, which have high-γ peaks for $L = 8$ and 10 that are comparable to those at intermediate γ values. Note, however, that this high-γ peak tends to disappear more rapidly. While high γ values may often be optimal for state estimation, the parameter estimates are less likely to be accurate in this limit and, by extension, state predictions are less likely to be accurate. There is no such ambiguity for lower L, but one needs to remember that the prediction errors are generally higher in these cases despite the tight localization optimal-γ peaks.

These results are highly suggestive of a connection to the linear problem, in which a similar phenomenon was observed when the model was *right*. Understanding why this is so in this case demands a more direct analysis of the inverse problem in terms of the effects of the model on the inversion of $\mathbf{H}$; the framework for this analysis is presented at the end of Chapter 1.

3.4 Regularization with the Wrong Model

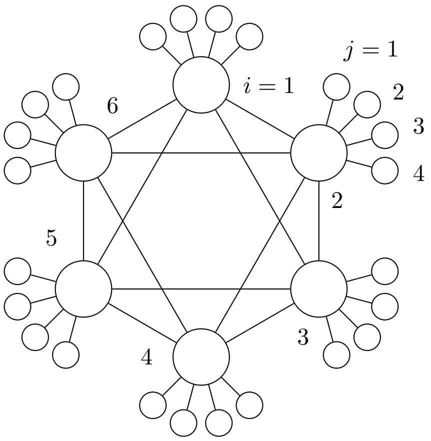


Figure 3.15: Sketch of the Lorenz 96 model with fast and slow variables $\boldsymbol{x}$ and $\boldsymbol{w}$, respectively. The large circles represent the individual $x_i \in \boldsymbol{x}$, whose dynamics obey the reduced L96 model studied in the previous chapter. The small circles are the fast variables $w_{j,i} \in \boldsymbol{w}$; with weak coupling h_x, they act like a background perturbation on the slow system.. They have similar dynamics to the slow variables but operate on a much faster time scale, have no explicit external forcing parameter (K in the slow system).

Consider an extension to the previous example of data assimilation for a Lorenz 96 model, in which a set of additional fast variables w, with similar dynamics to the "slow" model, but that oscillate with a much higher frequency, are coupled to the original system. Each of the D_s slow variables x_i, $i = 1, \ldots, D_s$, is coupled to its own set of J fast variables $w_{i,j}$, $j = 1, \ldots, J$

$$\begin{aligned} \frac{dx_i}{dt} &= x_{i-1}\left(x_{i+1} - x_{i-2}\right) - x_i + K + \frac{h_x}{J}\sum_{j=1}^{J} w_{j,i} \quad \text{(slow system)} \\ \frac{dw_{j,i}}{dt} &= \frac{1}{\varepsilon}\left[w_{j+1,i}\left(w_{j-1,i} - w_{j+2,i}\right) - w_{j,i} + h_y x_i\right] \quad \text{(fast system).} \end{aligned} \tag{3.8}$$

This is actually the model originally proposed by Lorenz [57]. A sketch of this system for $D_s = 6$ and $J = 4$ is shown in figure 3.15, where the nodes represent the state variables, and the vertices the couplings between state variables (couplings between the fast variables are omitted for clarity of the diagram; they should be inferred by the reader from the model equations in (3.8)).

In this example, the model used for estimate this system is purposefully chosen to be *wrong*. While data for the twin experiment is generated with the larger Lorenz 96 system with fast and slow variables, the estimated model contains only the slow variables. However, it will be shown that one may actually recover from this error somewhat by using a similar analysis to the previous example, where estimates and predictions are compared to observation as a function of the regularization strength R_f/R_m. This harkens back to the example in Chapter 2 where charge distributions were estimated from voltage data; there, the models introduced through regularization were also wrong, but it was possible to produce reasonably accurate predictions with the right regularization strength.

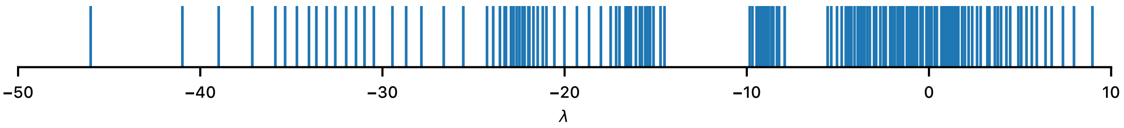


Figure 3.16: Lyapunov exponent spectrum for the Lorenz 96 system with fast and slow variables ($D_s = 20$, $J = 8$).

For the purposes of this analysis, the number of slow and fast variables (per slow variable) are set to $D_s = 20$ and $J = 8$, respectively. The parameter values are assigned at $K = 8.17$, $\varepsilon = 0.1$, $h_y = 1.0$, and $h_x = -0.1$. With these parameter choices, the fast variables operate on time scales which are approximately 10 times faster than the slow variables. Additionally, the model is observed to exhibit chaotic behavior, indicated by the presence of positive Lyapunov exponents, shown in figure 3.16.

This model is used to generate data for a state and parameter inference experiment, using a similar procedure to that presented in the previous section: the model system is integrated forward in time from a randomly selected initial condition in the $D = D_s(1 + J) = 180$ dimensional phase space, and after a period of integration time set to allow the system to settle to its attractor after a period of transient behavior, the trajectory is sampled every $\Delta t = 0.025$ time units. Many such data trajectories are collected for the purpose of carrying out a statistical analysis like the one presented in the previous chapter.

Predictions are made starting from $\boldsymbol{x}^N$ and using the estimated parameter value in the model for each lowest-action estimate at all values of γ. The details of this forward integration do not differ from the analysis of the previous section.

First, variational annealing was run with $N_{init} = 100$ initializations using a single data trajectory. The resulting action levels, as well as the measurement and model errors, are independently plotted in figure 3.17. Without the knowledge that the model is, in fact, wrong, it would appear that the estimation was highly successful. There is only one action level present, and it converges to the global minimum value of $\langle A \rangle = 1$ at high R_f.

Things start to look interesting, however, when the prediction errors are examined over the course of annealing. A much more pronounced minimum seems

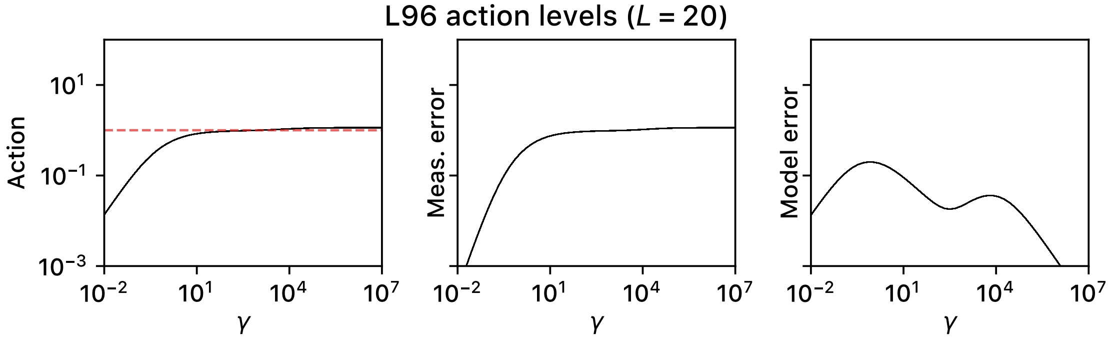


Figure 3.17: Action levels for estimation of the Lorenz 96 system with fast and slow variables, using the Lorenz 96 system containing only slow variables as the model. The slow system is fully observed, which leads to the identification of a single estimate that appears to be the global minimum. This is deceptive, and care must be taken to examine predictions as a function of γ to avoid the temptation of using estimates sampled from A as $\gamma \to \infty$.

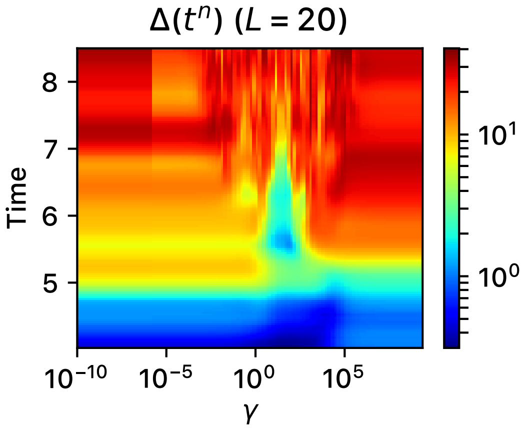


Figure 3.18: Prediction errors using the $D = 20$ Lorenz 96 system containing only slow variables, which is actually estimating the much larger system with 8 fast variables per slow variable. A local minimum in error consistently appears just below $\gamma = 10^4$.

to have appeared at an intermediate γ value than previously, and at high γ the predictions are distinctly worse, uniformly. When the model was correct, it was the case that predictions were of similar quality for about 1/4 of the prediction window at high γ compared to the local minimum. Repeated calculation of this prediction error surface, however, shows this to no longer be the case.

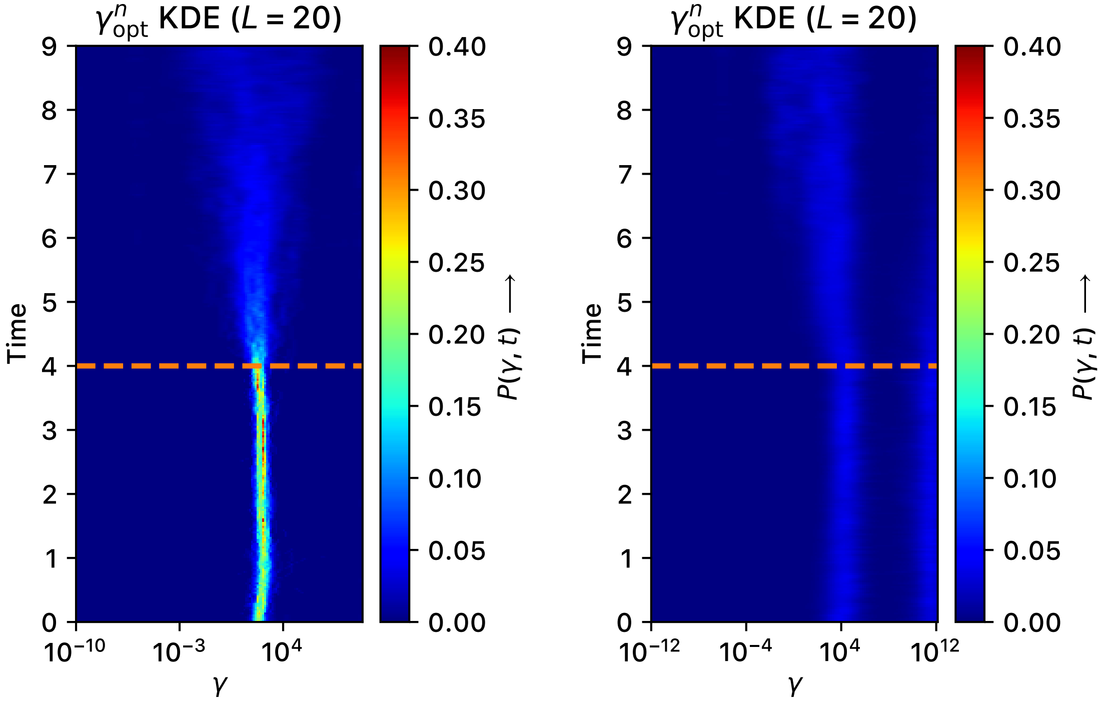


Figure 3.19: Distributions of optimal γ values for state estimation and prediction when the $D = 20$ Lorenz 96 system containing only slow variables is used to estimate the full fast/slow system (left panel), and the slow system (right panel). Left: a sharp peak appears just below $\gamma = 10^4$ in the estimation window, indicating that estimates are consistently optimal near this value and there is never an advantage to increasing γ to a very large value. Considering that the system appears to be fully observed with $L = 20$, this should be a clear sign that the model of the system is wrong in some way. Right: similarly to what was observed in the previous chapter, it is equally likely that estimates drawn from the intermediate-γ action will be optimal as from the high-γ action (depending on Y).

If this analysis is carried out many times, the optimal γ distribution looks *much* different than before. For comparison, the distribution for the $L = 20$ case in the slow system is shown alongside the KDE for the fast/slow version in figure 3.19. The peak in the optimal γ distribution is near the value which might be expected by looking at the prediction error in figure 3.18. While this local minimum would often move to different γ values in the fully-modeled system, depending on the data presented to the model, in this case the optimal value appears to be located within a *very* narrow range of values. Additionally, there is found to essentially never be any benefit to taking the $\gamma \to \infty$ limit.

The presence of unresolved system variables which are not contained in the model is an obvious limitation of this framework. However, this analysis shows that the situation is somewhat recoverable even with a partial model of the system if the model error is introduced with the proper regularization strength, similar to the voltage estimation problem when the model was also "wrong". Additionally, through a systematic characterization of the estimation problem in this system, it is found that the choice of regularization strength R_f/R_m for optimizing prediction accuracy is highly reliable, with the caveat that accurate predictions last for a short time in comparison to using a complete model for the system.

While the examples presented here and in Chapter 2 were ones in which the model was known to be wrong, and in fact one compare to the true model because it was also known, in reality it is often the case that the model is known to only be approximate but the unresolved dynamics are unknown. Thus, there is no way initially to model these errors, and one would have to approximate them or compensate for them in some way. This sort of characterization provides a method for doing this, where the model error is effectively modeled as an additive stochastic forcing to the dynamics (from the form of the Gaussian action), and predictions could be enhanced for at least some short time window into the future.

3.5 Conclusions

An extensive analysis of the estimation and prediction qualities associated with state and parameter estimates in the chaotic $D = 20$ Lorenz 96 system motivates a prediction-based criterion for selecting an optimal dynamical regularization, in the language of Chapter 2, in which a model system is characterized by computing state and parameter estimates, and their associated prediction errors, over an ensemble of observed trajectories. Using a distribution of γ values which were found to be optimal for comparing predictions with known future data, one is presented with a guide for choosing the regularization strength. In the Laplace approximation for evaluating the high dimensional integrals in the path space variational data assimilation formulation, one is to use well-separated, low-action estimates, ideally the global minimum of the action which is shown to exist for this problem when $L \geq 8$. Furthermore, the statistical interpretation of variational DA tells us that the limit $R_f/R_m = \gamma \to \infty$ in the action represents the limit of deterministic dynamical systems. It might be expected, therefore, that estimates which are found by numerical optimization of A should be selected from the limit $R_f \gg R_m$.

In Chapter 2, this was not found to be the case for a partially observed simple harmonic oscillator model by examining 1) the spectral structure of the regularized inverse measurement function, and 2) the state estimation and prediction error surfaces when data is introduced. The regularized H^{-1} was found to have a strongly-split eigenvalue spectrum at low γ values, indicative of a poorly regulated inversion. Examining the projections of the data onto the eigenvectors of the regularized inverse revealed well-defined track in the γ-(eigenvector) plane corresponding to the dominant contribution to the estimate. An analysis of estimation and prediction errors for varying regularization strengths was found to produce a well-defined distribution of optimal regularization strengths at an intermediate value near $\gamma = 100$. This optimal regularization strength was found to correspond to a point in this structure that indicated the transition between an undersmoothing and an oversmoothing regime had just completed.

Because Lorenz 96 is a nonlinear system, only the error analysis of the previous chapter could be mimicked. It was extended to empirically determine ensemble distributions of optimal values of γ for estimation and prediction over the entire estimation and prediction windows pictured in figures 3.13 and 3.14. This revealed a qualitatively similar result, which is that it is often optimal to select estimates $X^{[\gamma]}$ for which γ is within some intermediate range of values, rather than taking the "deterministic limit". In many cases, errors in the predictions were reduced by a factor of 10 compared to those produced by neighboring-γ estimates. This effect was especially pronounced for smaller values of L, in this case $L = 5$. Whereas the predictions did not last for nearly as long of a time with any kind of reasonable accuracy, compared to the higher-L cases, the location of the local minima in prediction error were much more reliably located around a particular value for different data inputs Y. Additionally, once the system has been characterized as in figures 3.13 and 3.14, it may be possible to use these distributions as a guide for optimal estimate selection. For larger L, the advantages of choosing a intermediate γ value seem to disappear: there are still apparently peaks near this intermediate value, but a strong peak also appears at high values of γ. This is not unreasonable from theoretical considerations, because it is assumed that as L increases, more information about the data is introduced to the model, so that taking the deterministic limit may be in fact be the more reasonable approach.

When the model of the system was wrong because it did not contain dynamical equations for a large number of the system variables, careful regularization of the problem provided a way to improve predictions for at least a short time. A similar effect was observed in Chapter 2 where accurate state estimates and, ultimately, predictions were recovered within small ranges centered about some finite value of γ.

This prediction-based criterion may prove to be useful as variational DA methods are applied to more fields, such as neurobiology and astrophysics, as well as in larger and more complex problems in more traditional fields for data assimilation, such as numerical weather prediction. When observing neurons in live networks, there is almost certainly a significant portion of the model which is missing if one uses a single-cell model for data assimilation as in [81, 48, 59]. Additionally, the method was found to be more reliable for the low-L system. Underobserved systems abound in the fields previously mentioned, making this potentially an even more fruitful application if the observability situation becomes worse in new problems.

A framework for understanding *why* certain choices of regularization from the viewpoint of inverting the measurement function was discussed in the conclusion to Chapter 2 for nonlinear systems. Combining this analysis with the prediction-based results could provide a much deeper level of insight into the nature of observability and stability for estimated chaotic systems. In particular, for understanding the value of measurements in a chaotic system; addressing the lower limits on how many measurements are required; or, given that the regularization is representative of a model error, why the optimal regularization chooses a particular value in terms of the structure and dynamics of the model.

Chapter 4

Reduced Models

The previous chapters presented a framework for model state and parameter estimation in partially-observed systems using a variational method of statistical data assimilation. In that context, a partial observation of $L < D$ of a model's D state variables are used to infer the model's full state and its parameter values for the purposes of forward prediction.

Here, an alternative formulation is presented in which the model itself contains *only* the L observed variables, and the remainder, which is the model error, is modeled through a discrete, stochastic model reduction approach presented in [14, 58]. These errors are treated as nonlinear stochastic processes, in general, with the hope of recovering the full range of dynamical behavior observed in the original system. This particular construction of the reduced model, based on the more general NARMAX model, sidesteps the difficulty associated with carrying out model reduction with the Mori-Zwanzig approach for continuous-time ODEs, but maintains some of the same general features. A significant challenge is in deciding how the error themselves should be modeled; this process is shown for a simple spiking neuron model, the Fitz-Hugh Nagumo system. Finally, a general framework for neuron model reduction with more complex stochastic models is presented. While the author is not prepared to present the results of preliminary work in this area, this introduction serves as the motivation for future research directions.

4.1 Stochastic Model Reduction: An Approach to Modeling Model Errors

In the approach to state and parameter inference that has been used up to this point of the dissertation, one always assumes an ODE model for the system in question,

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}, t; \boldsymbol{\theta}) \tag{4.1}$$

where $\boldsymbol{x} \in \mathbb{R}^D$ and $\boldsymbol{f} : \mathbb{R}^D \to \mathbb{R}^D$, i.e. the system is D-dimensional; one uses observations of $\boldsymbol{x}$ to estimate an initial condition for prediction, as well as values for the model parameters $\boldsymbol{\theta}$. (It is implied from now on that $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}, t; \boldsymbol{\theta})$, unless otherwise noted.) Only L of these variables are actually observed, and data assimilation method is used to estimate the trajectories of the remaining $D - L$ variables and model parameters $\boldsymbol{\theta}$. The metric used to assess the quality of the model is how well it can make predictions for the system, assuming one has reasonable estimates for the initial conditions.

To begin the discussion on how a reduced modeling approach is used, consider splitting up the ODE system into two pieces for the observed variables $\boldsymbol{x}$, and the unobserved variables $\boldsymbol{y}$:

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{z}(\boldsymbol{x}, \boldsymbol{y}), \quad \frac{d\boldsymbol{y}}{dt} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}) \tag{4.2}$$

In this construction, the “missing” dynamics are explicitly separated into an error term $\boldsymbol{z}$, without loss of generality. If one wishes to predict using such a model, it requires estimating the states of $\boldsymbol{x}$ and $\boldsymbol{y}$, and all of the parameters $\boldsymbol{\theta}_f$, $\boldsymbol{\theta}_g$, and $\boldsymbol{\theta}_z$, then predicting forward from the estimated initial condition $\boldsymbol{x}^N$ and $\boldsymbol{y}^N$ at the end of the observation window. In fact, this is just the variational data assimilation approach of the previous chapters.

In the *reduced modeling* approach that is presented here, one constructs a model which contains *only* observed quantities $\boldsymbol{x}$, while simultaneously acknowledging that a larger system is actually being observed, likely containint additional variables required to accurately describe the observed system. This, again, requires modifying $\boldsymbol{f}$ in some way to take into account dynamics of the *full* model containing the unmodeled variables, except that now there is no explicit dynamical equation for $\boldsymbol{y}$:

$$\frac{d\boldsymbol{x}}{dt} \simeq \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{z}(\boldsymbol{x}) \tag{4.3}$$

In other words, it is implicitly assuming that there are “hidden” variables present in the observed system, but that we wish to model their influence without explicitly including them in the model. The cost of such a construction is that $\boldsymbol{z}$ necessarily contains memory terms which take into account the past values of the hidden variables. This is because information takes time to propagate from one variable to another; for example, if two quantities x_1 and x_2 are coupled linearly in their dynamics:

$$\frac{dx_1}{dt} = \lambda_1 \left(x_2 - x_1\right), \quad \frac{dx_2}{dt} = \lambda_2 \left(x_1 - x_2\right) \tag{4.4}$$

then perturbations in x_2 take a time of order $1/\lambda_1$ to propagate to x_1, and vice versa.

The effect of memory can be explicitly shown in this linear system. Solving the x_1 equation:

$$\begin{aligned} &\frac{d}{dt}\left(x_1 e^{\lambda_1 t}\right) = \lambda_1 x_2 e^{\lambda_1 t} \\ &\Rightarrow \quad x_1(t) = x_1(t_0)\, e^{-\lambda_1(t-t_0)} + \lambda_1 \int\limits_{t_0}^{t} ds\, x_2(s)\, e^{-\lambda_1(t-s)}. \end{aligned} \tag{4.5}$$

If the system has evolved from the initial time t_0 to some short time t later, when $t - t_0 \ll 1/\lambda_1$,

$$x_1(t) \simeq x_1(t_0)\,[1 - \lambda_1(t - t_0)] + x_2(t_0)\,\lambda_1(t - t_0), \quad t - t_0 \ll 1/\lambda_1 \tag{4.6}$$

then $x_1(t)$ is, unsurprisingly, near its initial value, so the past behavior of x_2 hasn't had time to influence x_1 much. On the other hand, when $t \gg t_0$, $e^{-\lambda_1(t-t_0)}$ becomes small and

$$x_1(t) \simeq \lambda_1 \int\limits_{t_0}^{t} ds\, x_2(s)\, e^{-\lambda_1(t-s)} \tag{4.7}$$

so that x_1 is now *dominated* by the memory term. The memory kernel decays exponentially into the past, so that the system actually has a finite memory:

$$x_1(t) \simeq \lambda_1 \int\limits_{t_0}^{t} ds\, x_2(s)\, e^{-\lambda_1(t-s)} \simeq \lambda_1 \Delta t\, x_2(t-\tau)\,[1 - \lambda_1 \Delta t] \quad (\Delta t = t - \tau) \tag{4.8}$$

where τ reflects how far into the past the system "remembers" x_2 (the time beyond which contributions of order $(\lambda_1 \Delta t)^2$ can be ignored). Regardless, the memory term dominates x_1 when we are well beyond the initial time, so it must be taken into account for an accurate, reduced description.

It is actually possible to use this memory property to develop a *closed* system for x_1, which is a dynamical system for x_1 that contains *only* x_1 as a dynamical variable. Following the same logic as above, but for x_2:

$$x_2(t) \simeq x_2(t_0)\left[1 - \lambda_2(t - t_0)\right] + \lambda_2 x_1(t_0)(t - t_0), \quad t - t_0 \ll 1/\lambda_2. \tag{4.9}$$

Suppose that one wishes to simulate x_1 at regular times $t_0, t_1, \ldots$, where each time interval $t_{n+1} - t_n \equiv \Delta t$ is small. Then,

$$\begin{aligned} x_1(t_1) &= x_1(t_0)\left[1 - \lambda_1 \Delta t\right] + \lambda_1 x_2(t_0)\Delta t, \\ x_1(t_2) &= x_1(t_1)\left[1 - \lambda_1 \Delta t\right] + \lambda_1 x_2(t_1)\Delta t \\ &= x_1(t_1)\left[1 - \lambda_1 \Delta t\right] + \lambda_1 \left\{x_2(t_0)\left[1 - \lambda_2 \Delta t\right] + \lambda_2 x_1(t_0)\Delta t\right\}\Delta t \\ &\vdots \end{aligned} \tag{4.10}$$

This is a dynamical system (now in the form of a difference equation, rather than an ODE) for x_1 which is a function of x_1 alone, and is thus a closed system for x_1. The price we pay is that an initial condition is still required for x_2, which is reflective of the system's memory of x_2.

Discrete-Time Model Reduction

It is a far more complicated task to develop a reduced model for the more general case in which the full model system is nonlinear and/or stochastic. If the full system is split into observed and unobserved variables, $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively,

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}), \quad \dot{\boldsymbol{y}} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}) \tag{4.11}$$

then in the reduced system,

$$\dot{\boldsymbol{x}} \simeq \boldsymbol{f}_0(\boldsymbol{x}) + \boldsymbol{z}(\boldsymbol{x}) \tag{4.12}$$

the approximate dynamics $\boldsymbol{f}_0$ are obtained by a projection $\boldsymbol{f} \rightarrow \boldsymbol{f}_0$. This is almost certainly *not* one-to-one when $\boldsymbol{f}$ and/or $\boldsymbol{g}$ are nonlinear functions. In order to fully explore the solution space of the full model, then, $\boldsymbol{z}$ must be generalized to a nonlinear, stochastic function of $\boldsymbol{x}$ [14].

The Mori-Zwanzig (MZ) formalism [21, 12, 91, 92] is one approach to calculating expressions for the remainder term $\boldsymbol{f}_0$ and the error $\boldsymbol{z}$. MZ yields an exact expression for $\boldsymbol{z}$ containing a sum over noise and non-Markovian memory terms, which shows the necessity of including memory in a reduced modeling approach. Carrying out the calculations in the MZ formalism, however, is a difficult task, and there appears to be no more than one successful attempt in the literature [13] at using it for a nonlinear problem.

The formulation presented here follows a discrete-time model reduction approach developed by Chorin and Lu in [14, 58], where they showed that this approach is a powerful tool even for chaotic systems; in particular, the model is shown to be able to predict well in severely underobserved, chaotic systems. Here, one takes the viewpoint that most real data assimilation problems are naturally cast as discrete-time problems in the first place, especially since solutions are generally calculated numerically on a computer. It uses a NARMAX (Nonlinear AutoRegressive Moving Average with eXogenous inputs) model [6, 24], which has been widely used previously in nonlinear system identification [11, 37]. Starting with a discrete system sidesteps the difficulties associated with performing a reduction of the ODE system directly, à la the Mori-Zwanzig procedure.

First the dynamics are recast as a discrete forward mapping in time, which is in the form of a difference equation:

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \Delta t \left[\boldsymbol{R}_{\Delta t}(\boldsymbol{x}^n) + \boldsymbol{z}^{n+1}\right] \tag{4.13}$$

where $\boldsymbol{R}_{\Delta t}$ is the time integral of $\boldsymbol{f}_0$:

$$\boldsymbol{R}_{\Delta t}(\boldsymbol{x}^n) = \frac{1}{\Delta t} \int\limits_{t^n}^{t^{n+1}} dt \; \boldsymbol{f}_0(\boldsymbol{x}). \tag{4.14}$$

In practice this integral is performed by numerical approximation, so the equality above is not actually exact.

The remainder term, $\boldsymbol{z}^{n+1}$, absorbs the model error associated with the approximate projection mapping $\boldsymbol{f} \to \boldsymbol{f}_0$. By our previous logic, such a reduced model should have memory and contain stochastic elements. It should also be nonlinear if the model is to mimic the full dynamical range of the full system [14]. The NARMAX formulation has these properties, giving the remainder $\boldsymbol{z}$ the form[1]:

$$\begin{aligned} z_i^n &= \sum_{j=1}^{p} a_{i,j} z_i^{n-j} + \sum_{k=1}^{r} \sum_{j=1}^{s} b_{i,jk} Q_{i,j}(\boldsymbol{x}^{n-j}) + \sum_{j=1}^{q} c_{i,j} \xi_i^{n-j} + \xi_i^n \\ &\equiv \Phi_i^n + \xi_i^n. \end{aligned} \tag{4.15}$$

Note that the sums in this model are actually in time: z_i^{n-j}, for example, is the i-th component of $\boldsymbol{z}$ but j steps into the past. These sums are where the memory of the model explicitly appear. In addition to having a memory term for the error itself, which is the first term in (4.15), the model allow for nonlinearities through the functions $Q_{i,j}$; exactly how these functions are to be chosen will be shown in a

[1] This is the equation for just one component of $\boldsymbol{z}$; every component z_i has its own set of NARMAX parameters a_i, b_i, and c_i, as well as its own noise term $\xi_i \sim N(\mu_i, \sigma_i^2)$.

simple example later, but note that they should be motivated by knowledge of the physics of the *full* system. Finally, the stochastic part of the model is in the third and fourth terms, for which there is also explicitly a memory term. Note that the third term is not the only place where memory of the noise enters the model: memory of the noise enters *implicitly* through the first and second terms, which are otherwise deterministic.

This formulation introduces a new set of parameters into the model that must be estimated from data: the NARMAX coefficients a, b, and c, as well as the noise parameters μ and σ. In order to estimate these parameters, one introduces the notion of a probability distribution for the model parameters conditioned on a time series of data. The MLE minimizes the negative log-likelihood of this distribution, which is given by:

$$-\ln L(\boldsymbol{\theta}|\boldsymbol{x}^{1:N}) = \sum_{n=M}^{N} \frac{|\boldsymbol{z}^n - \Phi^n(\boldsymbol{\theta})|^2}{2\sigma^2} + \frac{N-M}{2} \ln \sigma^2 \tag{4.16}$$

where $\boldsymbol{\theta} = \{a, b, c, \mu, \sigma\}$ and $M = \max\{p+1, r, q\}$. This function can be minimized using any number of numerical optimization methods; we take advantage of the fact that the coefficients a, b, and c enter linearly into the model to precondition the estimate using the LSE (least-squares estimator).

Note that, despite the appearance of a "new" variable $\boldsymbol{z}$, (4.13) is still a closed system for $\boldsymbol{x}$. Simply use the definition of z component-wise in terms of x

$$z_i^n = \frac{x_i^n - x_i^{n-1}}{\Delta t} - R_{\Delta t, i}(\boldsymbol{x}^{n-1}) \tag{4.17}$$

on the right-hand side of (4.15), and substitute back into the original definition (4.13).

This yields:

$$x_i^n = x_i^{n-1} + \Delta t \left\{ R_{\Delta t,i}(\boldsymbol{x}^{n-1}) + \sum_{j=1}^{p} a_{i,j} \left[\frac{x_i^{n-j} - x_i^{n-j-1}}{\Delta t} - R_{\Delta t,i}(\boldsymbol{x}^{n-j-1}) \right] \right.$$
$$\left. + \sum_{k=1}^{r} \sum_{j=1}^{s} b_{i,jk} Q_{i,j}(\boldsymbol{x}^{n-j}) + \sum_{j=1}^{q} c_{i,j} \xi_i^{n-j} + \xi_i^n \right\}. \tag{4.18}$$

It may still be useful to keep track of $\boldsymbol{z}$ separately for the purposes of analyzing the error in the reduced model, i.e. where the reduced model is the most "wrong". This knowledge can aid in improving the model, whether through the introduction of additional Q terms, or adjusting the lengths of the memory in each term (p, q, and r).

Once the parameters of NARMAX have been selected by MLE on an observed data trajectory, it can ultimately be used for prediction, with a different approach for short-term vs long-term forecasts. For short-term predictions, the Gaussian noise terms in (4.15) are initialized such that $\xi^1 = \cdots = \xi^q = 0$; then, $\xi^{q+1}, \ldots, \xi^m$ are estimated using equation (4.15), where $m = \max\{p, r, q\} + 1$. This provides the necessary initialization for the first step in prediction, but beyond this step the prediction depends on the particular initializations of future values of ξ. Thus, an ensemble of predictions is made with different realizations for ξ, which may then be compared directly with each other, or analyzed statistically in the noise ensemble distribution.

Long-term forecasts do not require this careful initialization of ξ; as long as the system is ergodic, then the autocorrelation functions should become independent of the initialization after some sufficient time [14]. An ensemble of long term forecasts may appear to differ significantly from each other if one simply compares the discrepancy between them over time. However, it is often still of interest to study the statistics of a solution over long times rather than the exact trajectories. In the simple FHN neuron model example presented below, the distribution of voltage values as well as interspike intervals (the times between neighboring spikes) are examined and compared to the statistics of the full model.

4.2 Data Assimilation with a Reduced Model

The ultimate goal of constructing a reduced model like the one in eq. (4.13), with a model error $\boldsymbol{z}$ defined in (4.15), is to model and predict time series of a physical system with observed variables $\boldsymbol{x} \in \mathbb{R}^L$. This is a problem of data assimilation, where the parameters to be estimated from observed time series are the NARMAX coefficients a, b, and c. Additionally, we must estimate a good initial condition for the reduced model from observations, where "good" in this context means:

1. Producing accurate short-term predictions of new time series, and

2. Reproducing the *statistics* of the observed system faithfully.

The second consideration here is new, compared to the variational approach used in previous sections and chapters. The observed system is in a sense *inherently* stochastic now, under the assumption there are unmeasured variables in the system for which we have no model. Thus, the best one can *ever* hope to do in long-term predictions is to get the statistics right. Exactly which statistics are important varies from system to system; for example, one may simply wish to correctly reproduce the mean trajectories $\langle x_i \rangle$, or their variances $\langle x_i^2 \rangle - \langle x_i \rangle^2$. In the neuron example presented below, the statistics of the interspike interval are of particular interest in neurobiology and computational neuroscience [35, 46, 85, 9].

4.2.1 Stochastic FitzHugh-Nagumo: A Low-Dimensional Neuron Model

An example of a relatively simple, low-dimensional model for a spiking dynamical system, the FitzHugh-Nagumo (FHN) model [28, 64], is considered as a candidate for discrete stochastic model reduction. FHN is a simplified biological neuron model which exhibits some basic characteristics of more detailed models like the Hodgkin-Huxley (HH) model [40]. Important features of the HH model include, of course, spiking behavior, as well as a spike frequency which is dependent upon the magnitude of the injected current stimulating the cell. In the HH model, spiking behavior is explained by the presence of "gating variables", which represent the nonlinear bulk response of the voltage-activated protein gate structures present in the cell membrane. Extensions of the HH model abound in the literature, including the addition of calcium-dependent gating variables [23], synaptic structures [16], as well as nonadditive stochastic elements which describe the complex statistics of random opening and closing of individual ion gates in the membrane [29]. While many of these cases would surely provide solid motivation for a reduced model study, to the knowledge of the dissertation author there are no examples of the literature of this approach being applied to neuron models. Thus, this discussion starts small by considering the heavily simplified FHN model and establishing some results of the NARMAX model reduction method.

To wit, FHN is actually an approximation to HH. To understand how, consider the full Hodgkin-Huxley model system:

$$\begin{aligned}\dot{V} &= g_{\mathrm{Na}} m^3 h \left(E_{\mathrm{Na}} - V\right) + g_{\mathrm{K}} n^4 \left(E_{\mathrm{K}} - V\right) + g_{\mathrm{L}} \left(E_{\mathrm{L}} - V\right) + I(t) \\ \dot{x} &= \frac{x_\infty(V) - x}{\tau_x(V)}, \quad x \in \{m, h, n\} \end{aligned} \tag{4.19}$$

where V is the neuron's membrane potential; m and h are the activation and deactivation gating variables, respectively, for the sodium (Na^+) current; and n is the activation gating variable for the potassium (K^+) current. Rinzel made the observation in [73] that τ_m is small compared to τ_h and τ_n for all V; thus, $m \approx m_\infty(V)$ throughout the model's trajectory through phase space. This approximation is valid because τ_m corresponds to a very fast relaxation time for the m equation in (4.19), so m closely "tracks" m_∞ compared to h and n. Additionally, Krinsky and Kokoz in [49] found that $n \approx \alpha + \beta h$, where α and β are empirically determined constants which fit this approximation on a model trajectory. Rinzel's approximation effectively removes m as a dynamical variable from HH; Krinsky and Kokoz's approximation further reduces it by one variable.

This effectively leaves two variables behind in the model: V, the membrane voltage; and w, which is a linear combination of the old gating variables. The FHN model, which is a further reduction from the current state of the reduced HH model, is based on the simple observations that in the remaining (V, w) phase plane, the $\dot{V} = 0$ nullcline is approximately cubic, and the $\dot{w} = 0$ nullcline is approximately linear. Thus, we see now that the FHN model is a good approximation to HH when the m variable is fast compared to h and n, and when the model is operating near the nullclines in the phase space. When the model spikes, however, it is far from a nullcline; the consequence of this is that spike shape cannot be modified significantly by altering $I(t)$, whereas in HH it can.

What remains is the FHN model, a 2-dimensional ODE which contains a single "voltage" variable, u, and a "gating" variable w:

$$\dot{u} = u - \frac{u^3}{3} - w + I(t), \quad \dot{w} = \frac{u + a - bw}{\tau} \tag{4.20}$$

This model is an ideal candidate for the NARMAX model reduction scheme, as it is polynomial in u and w. The nonlinear Q functions in NARMAX can thus be reasonably restricted to contain only polynomial terms in u. This is shown to be true later in the analysis by more direct means.

It is of great interest in particular to study how this model behaves under stochastic forcing. Real neurons exhibit many effects of stochasticity in their behavior:

- Real neurons spike anomalously when the injected current is below the spiking threshold, and when other effects such as rebound spiking are taken into account,
- They have their spikes anomalously terminated above threshold,
- They exhibit phase drift, in which the interval between spikes fluctuates like a random variable, and
- They show effects reminiscent of stochastic resonance.

In this study, a white noise signal is injected into the FHN model. This is actually quite artificial from the perspective of sources of noise in nature. A real neuron may exhibit stochastic behavior due to noise in the gating variables, which is properly represented by multiplicative noise in a Langevin equation description [29]; or what appears to be stochastic behavior in a network when there are a large number of unknown inputs from other neurons. Regardless, this example is still illustrative in establishing model reduction as a technique for neuron systems, especially in the analysis of the complicated statistics of spike timings.

The purpose of analyzing a reduced NARMAX model for FHN is thus to gain a grasp on how one could reduce a more complex neuron model using the NARMAX construction. FHN is much simpler to analyze than Hodgkin-Huxley, as it does not require the inclusion of nonlinear Q terms in the NARMAX representation. However, it still shares important dynamical properties with HH like the dependence of spike frequency on the amplitude of $I(t)$. The gating variable, as we shall see, is simple to remove from FHN and replace with memory terms containing u alone. Eventually the gating variables must *all* be removed from HH if one is to construct a reduced model of an HH-like system, because the gating variables are not directly observable. The view taken here is thus that reducing HH is essentially a harder version of reducing FHN, and should be studied after FHN reduction is well-understood.

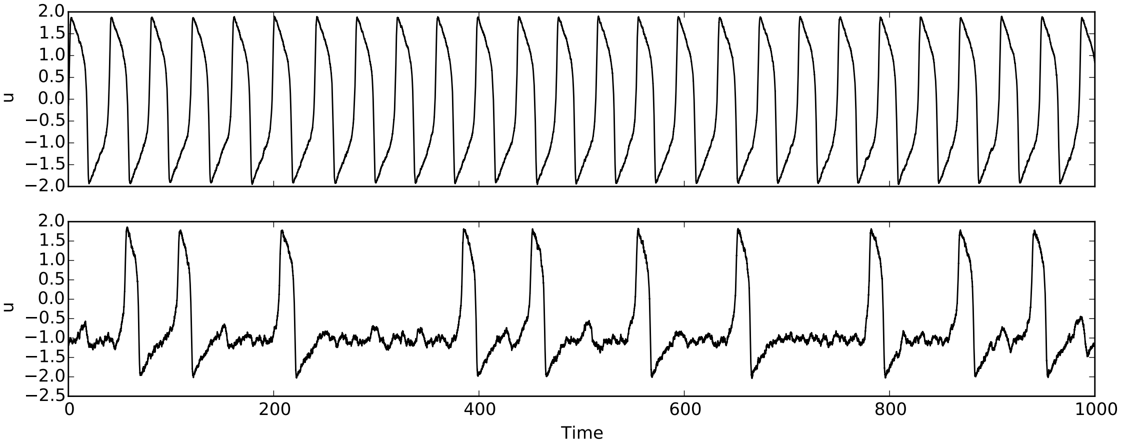


Figure 4.1: Examples of FHN model trajectories. In the top panel, the cell is held well above treshold with a DC injected current. A small amount of noise is added to this signal, causing the phenomenon known as "phase drift" in which the spiking frequency fluctuates over time. The bottom panel shows a very different behavior, induced by holding the cell just slightly below threshold with a DC current that has a much larger noise signal added to it. The result is a cell that appears to fire "anomalously" given that the injected current is nominally too small to produce spikes.

Before moving on, it is worth examining what solution to the FHN equations look like under a fairly simple form of stochastic forcing. In figure 4.1, the FHN system is subjected to a DC current with a small amount of noise added to it:

$$\begin{aligned} I(t) &= I_{\mathrm{DC}} + \eta(t), \\ \langle \eta(t) \rangle = 0, &\quad \langle \eta(t) \eta\left(t'\right) \rangle = \sigma^2 \delta(t - t'). \end{aligned} \tag{4.21}$$

The statistics of η means this is uncorrelated Gaussian (white) noise. In particular, in fig. 4.1, the values of I_{DC} and σ^2 are set in such a way as to "coax" two different behaviors from the FHN neuron. In one case, I_{DC} is set to be well above the threshold level for spiking, and σ is small so that large random fluctuations in the injected current are rare enough that the interruption of (nearly) monotonic spiking behavior is rarely seen. However, this small noise does have a measurable effect on the phase dynamics of the cell, where the time intervals between spikes are seen to oscillate about a mean value.

In the second case the model displays more complicated behavior. The constant term in the injected current is set to be just slightly below the threshold value so that the model would not normally spike on its own, but the noise is set to be large enough that random fluctuations in the current often temporarily push the neuron over the threshold for long enough that a spike occurs. While the white noise model used here is, again, almost certainly inadequate for describing the stochastic forcing of a realistic neuron, this case is studied because the model acts as a nonlinear filter for the input noise, thus in principle generating nontrivial statistics for the voltage or firing times. The goal is to test if the reduced model is able to reproduce these statistics.

Identifying Stochastic Memory Terms with NARMAX Model Reduction

In the FHN model, the gating variable w is considered to be unobservable on biological grounds, so its dynamical equation in (4.20) must thus be removed to form a reduced model which contains only the observable voltage variable, u. Consider solving the w equation by separation of variables:

$$\begin{aligned} \dot{w} &= \frac{u+a-bw}{\tau} \\ \Rightarrow \quad w(t) &= w(t_0)\, e^{-\frac{b}{\tau}(t-t_0)} + \frac{a}{b}\left(1 - e^{-\frac{b}{\tau}(t-t_0)/\tau}\right) + \frac{1}{\tau}\int_{t_0}^{t} ds\, e^{-\frac{b}{\tau}(t-s)}u(s). \end{aligned} \tag{4.22}$$

We see that $w(t)$ has an exponentially decaying dependence on its initial condition $w(t_0)$, tends to equilibrate towards a/b for $t \gg t_0$, and its dynamics have a memory of u which is of a time equal to approximately τ/b in the past. This solution is fairly simple because of the linear dependence of $\dot{w}$ on w, but this actually a general feature of gating variable dynamics in neuronal models. The difficulty arises in the nontrivial dependence of the gating variable dynamics on voltage; in this model $\dot{w}$ is *also* linear in u, so the solution to the $\dot{w}$ equation is in the form of a linear operator acting on u. This model will thus be much simpler to reduce than those containing more realistic gating variable dynamics, in which $\dot{w}$ would depend nonlinearly on u.

In the $\dot{u}$ equation of FHN,

$$\dot{u} = u - \frac{u^3}{3} - w + I(t),$$

there is a term linear in w which must be removed if we are to obtain a reduced model containing u alone. Consider a very basic approximate forward mapping for this equation:

$$\begin{aligned} u^{n+1} &= u^n + \int_{t_n}^{t_{n+1}} dt\, \dot{u} \\ &\simeq u^n + \Delta t \left[u^n - \frac{(u^n)^3}{3} - w^n + I^n \right]. \end{aligned} \tag{4.23}$$

This indicates that a discrete model, at the very least, requires a term which is linear in the immediate past state of w. Equation (4.22) says that this can be represented as a linear operator on past values of u. *Thus, it is reasonable to replace w in the discrete reduced model with terms which are linear in the past values of u.* This motivates the choice of a single $Q(\boldsymbol{x}^{n-j}) = u^{n-j}$ for the NARMAX representation. It remains to choose p, q, and r. These different choices should be analyzed by comparing estimates to data, as well as the trajectories of short-term predictions and statistics of long-term predictions. However, as stated previously, this analysis considers only the case where $p = q = r$. It was found that setting $q > 1$ led to instability in the solutions, which is almost certainly due to the effect of the memory term in amplifying the noise in the signal. p and r were also set to 1 because no significant advantage was found in increasing these values further.

To identify terms in the NARMAX representation that should be used for modeling the FHN system, once again a twin experiment is performed by generating synthetic data from the original FHN system. Again, this is done by integrating the FHN system forward in time. The injection of a noise term into the voltage equation requires that a stochastic integration method be used to generate these solutions. In both cases a Milstein [61] method of strong order 1.5 was used with an integration time step of $\Delta t = 0.001$. The data is saved with the coarser time step of $\Delta t = 0.1$ for $N = 10^6$ time steps. Two such trajectories are saved, where one is used for training the NARMAX parameters by the ML estimation approach described in the previous section, and the other as a validation set for testing the estimated NARMAX model through prediction. Only the first half of the training set is used for estimating the NARMAX parameters, while the second half is saved as a separate validation set for later. Samples of the trajectories for the two noise level cases are shown in figure 4.1.

One caveat in this case, however, is that the NARMAX method becomes very unstable when there is noise in the measurements, due to the presence of the memory terms which tend to amplify noise signals. Some preliminary work has been conducted by the author with his collaborators in this project, Alexandre Chorin and Fei Lu, but it remains an ongoing research problem. Thus, the twin experiment is carried out using noiseless synthetic data.

We thus examine the success of a reduced model in two regimes of FHN parameterization and forcing, which produce very different traces that will require reparameterizing the reduced model accordingly. In both cases, the injected current is constant in time, and a stochastic forcing is additionally introduced to model the effects of the neuron's environment as well as channel noise. A Gaussian noise term is added to a constant injection $I = I(t)$, where the statistics of the noise are given by

$$\langle \eta(t) \rangle_= 0, \quad \langle \eta(t)\eta(t') \rangle_= \sigma_c^2 \delta(t - t')$$

Altering the values of I and σ define the two cases under consideration:

1. $I > I_{\text{thr}}$, $\sigma_c = 0.03$: The model exhibits nearly monotonic spiking but with phase drift. The noise is small enough that anomalous spiking or spike termination is rare, but the interval between spikes is now a random variable.

2. $I \lesssim I_{\text{thr}}$, $\sigma_c = 0.07$: The injected current is just below threshold, so that with no additional forcing the voltage is constant, but a small positive perturbation will induce a spike. This is known as the *excitable* case, and when this level of stochastic forcing is introduced the spiking behavior is seemingly random.

4.2.2 Numerical Methods and Results

The NARMAX parameters were estimated using the synthetic data generated by the procedure described in the previous section. Minimizing the likelihood function

L was done by preconditioning the solutions with the iterative least squares algorithm, which was repeated until the least squares estimator was found to stabilize. This preconditioned estimate was used as the seed for numerical optimization of L. Derivatives of L were computed using PYADOLC [83], a Python wrapper around the ADOL-C [33, 32] library for automatic differentiation. The minimization was performed using L-BFGS [8, 89, 62] as implemented in ALGLIB [7].

The resulting parameter estimates for the two cases are shown in table 4.1. In both cases, a value of μ, the bias term, was estimated to be near 0. This is not terribly surprising, because the true value for the DC current I was known in the estimation step. A bit more surprising were the small sizes of b_1, which meant that the linear term for the immediate past value of u is neglected as a contribution to the model. The values of σ were close to the values used to generate the twin data, which is a good check on the validation of the procedure.

Finally, the results of short- and long-term predictions are compared to the training and validation data sets. In one case, the predictions is tested against the second half of the training set, while in the other it is compared to the validation set. The short term predictions were repeated over an ensemble of realizations for the measurement noise. The long term predictions were carried out by integrating the NARMAX model for a long time, and then directly computing statistics of the trajectory. The trajectories of the short-term predictions are shown in figure 4.4, while the long-term statistics are displayed in figures 4.2 and 4.3. These distributions were computed using a kernel density estimator (KDE), which was defined previously in

	a_1	b_1	c_1	μ	σ
Scenario 2	0.9992	-7.990×10^{-4}	-0.9991	-5.581×10^{-4}	0.2988
Scenario 3	0.9992	-7.982×10^{-4}	-0.9991	-5.583×10^{-4}	0.6990

Table 4.1: NARMAX parameter values in scenarios 1-3.

(3.7).

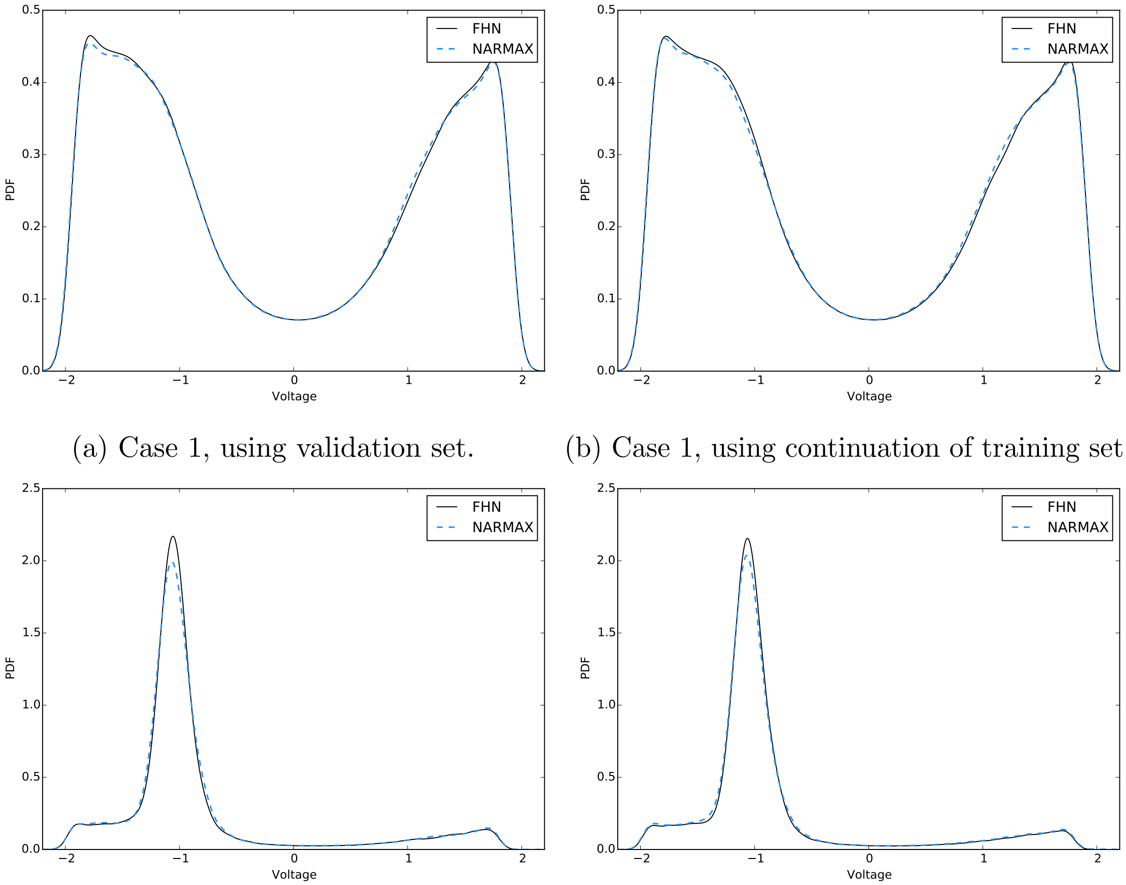
The interspike intervals were calculated by choosing a threshold value of 1, above which the model was considered to be in the onset of a spike. A criterion was set to ignore the trajectory as it decreased down back past 1 so that spikes would not be counted twice. This same method for counting spikes was applied to the predictions as well as the synthetic data used for training or validation.

4.2.3 Discussion

The reduced model for the FHN model with stochastic forcing was found to be a partial success by comparing the predictions made by the model with the time series data used to train the model, as well as a separate set of validation data which was generated by the same system, but with different initializations for the noise and the states u and w. Short term predictions were not successful: the model tended to spike almost immediately after initialization, even in the case where the trajectory was initially relatively flat (case 2).

However, some important general features of the trajectories were maintained from the original model in both cases. In case 1, the reduced model spiked nearly monotonically, but clearly exhibited phase drift based on the observed interspike interval distribution shown in figure 4.3, which was seen to have a finite width. The model for case 2 exhibited "anomalous" spiking just as the model used to generate the training data did. In fact, the empirical distribution of predicted interspike intervals matches that of the training and validation data with high accuracy. It was unclear why, in case 1, there is such a large mismatch in the two distributions when comparing to the continuation of the training set. This most likely warrants a simple re-checking of the calculation for errors.

The distributions of the predicted voltages themselves were found to be excellent



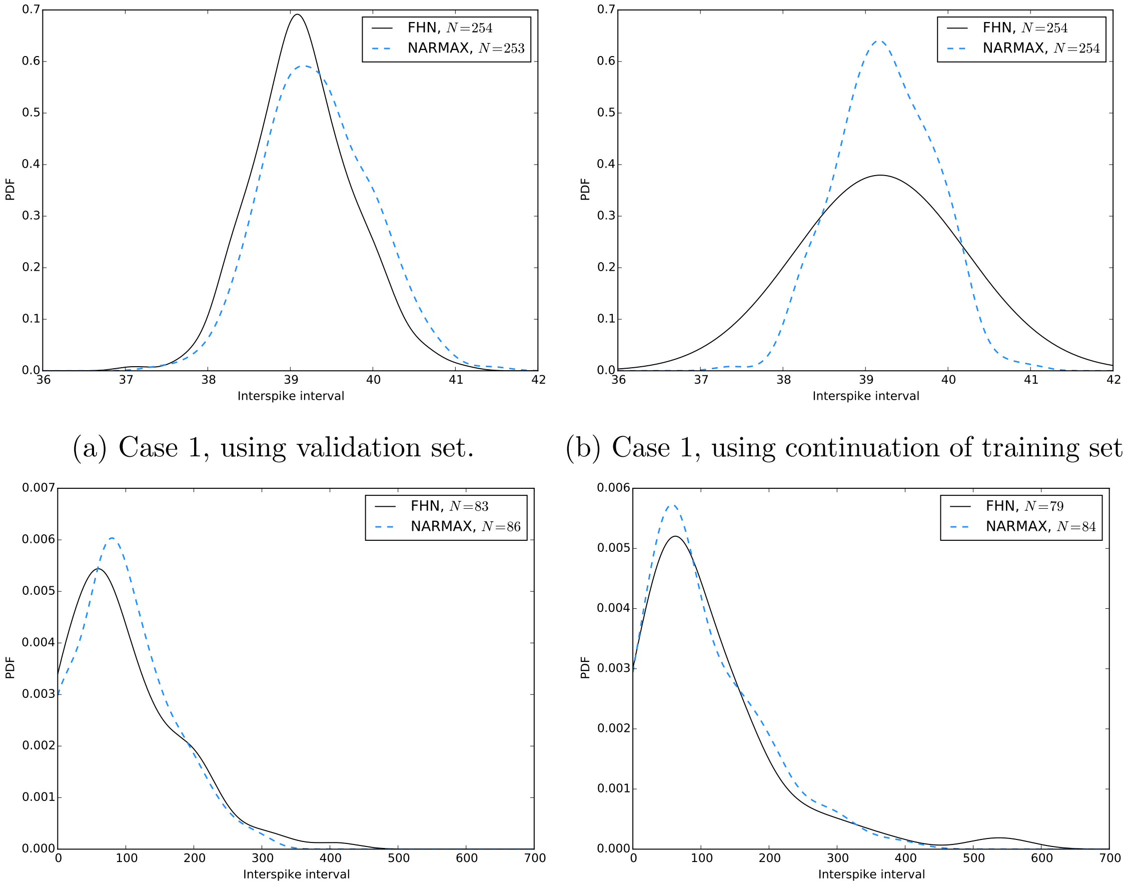
(a) Case 1, using validation set.

(b) Case 1, using continuation of training set.

(c) Case 2, using validation set.

(d) Case 2, using continuation of training set.

Figure 4.2: Observed distributions of the voltage variable u, calculated for long-term predictions using NARMAX (blue dashed line) and the original FHN system. In both cases (low- and high-noise systems), the reduced NARMAX model is able to recreate the original system's statistics well.



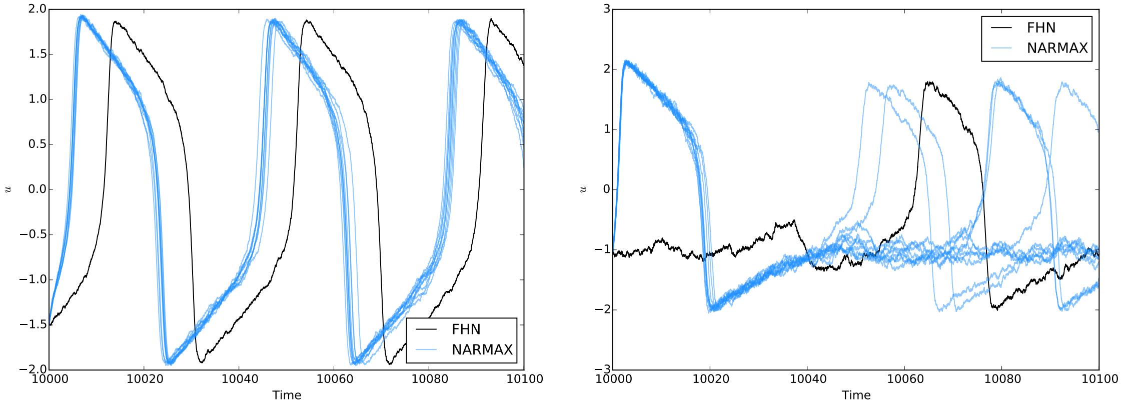
(a) Case 1, using validation set.

(b) Case 1, using continuation of training set.

(c) Case 2, using validation set.

(d) Case 2, using continuation of training set.

Figure 4.3: Interspike interval distribution comparison between full and reduced NARMAX models. Approximate distributions were computed from data using a Gaussian KDE; the legends in each figure indicate the number of spike pairs used to compute the KDE. In the cases shown in the top-left panel and the bottom panels, the distribution in the NARMAX model was a reasonably close match with the full model. In case 2 (high-noise), the NARMAX model did not appear to fully capture the tails of the distribution in the original system. The number of observed spike pairs is relatively low in each case, however; more experiments should be conducted to confirm if this is a failure of the NARMAX model, or a result of sampling error.



(a) Case 1

(b) Case 2

Figure 4.4: Ensemble predictions using the reduced model, vs a single time series from the full FHN model. Note that, in both cases above, the reduced model tends to spike immediately, but retains the qualitative features of the original model.

matches to the distributions for the training and validation data. This is less surprising than the result for the ISIs, but serves as a good validation check for the reduced models. The voltage in the full FHN model, in both cases of stochastic forcing, was found through preliminary calculations (not shown) to have a stationary distribution in the long-time limit. The larger discrepancies in the ISI distributions may be due to the fact that ISI does not have a stationary distribution over long times (this was not checked); but may also be due to sampling error since the number of spikes contained within the training and validations sets only numbered in the several hundreds.

The stochastic model reduction method presented in this chapter has thus been established for the reduction of neuron models. While the FHN model is not exceedingly complex, and does not contain strong nonlinearities or complex noise dynamics, it serves as a useful example for thinking about model reduction for more realistic neuron models like Hodgkin-Huxley, or even a slightly less complex model like Morris-Lecar but with the inclusion of channel noise. If these extensions to more complex models prove to be useful for prediction, this would already be a success because the nonlinear NARMAX formulation is generally well suited to reduced models of nonlinear systems.

Additionally, despite poor performance for short-term prediction, the reduced model was found to reproduce long-term statistics of the voltage and, more interestingly, the interspike intervals. These are important properties of a neurobiological system to preserve in a model, thus likely being of interest in biology and computational neuroscience.

A final comment is on two worthwhile extensions to this reduced modeling method: modifying it to be stable in the presence of measurement noise, so that it is more generally applicable in realistic experiments where measurement noise is almost always present; and also extending it so that the parameters of the dynamical model $\boldsymbol{f}_0$ which enters into the NARMAX definition can be estimated, beyond just estimating the coefficients of the NARMAX expansion.

4.3 Nonlinear Gating Variable Equations

Preliminary work has been performed on model reduction for neuron models with nonlinear gating variable equations. Successful reduction of such models is critical for modeling real neurons. FHN has some limitations in approximating the Hodgkin-Huxley model, discussed in the introduction to the previous section; beyond this the hope is to use this formalism to model more realistic sources of stochastic neuronal behavior, like gating variable noise.

FHN was still a useful starting point for this more general problem. The simplest models of neuron models with nonlinear gating variable dynamics are of the form

$$\frac{dV}{dt} = f(V, x_1, x_2, \ldots) + I(t), \quad \frac{dx_1}{dt} = g_1(V) - x_1 h_i(V), \quad \frac{dx_2}{dt} = g_2(V) - x_2 h_2(V), \ldots \tag{4.24}$$

where f is usually polynomial in the gating variables $x_1, \ldots$; g_i and h_i are nonlinear functions of V for gating variable x_i. Despite their nonlinear dependence on V, these equations are still in a form that can be solved using separation of variables:

$$x_i(t) = e^{-\int^t ds\, h_i(V(s))} \left[x_i(0) + \int^t ds_1\, g_i(V(s_1)) e^{\int^{s_1} ds_2\, h_i(V(s_2))} \right]. \tag{4.25}$$

In FHN the integral in brackets was greatly simplified and only depended linearly on past values of the voltage; no such simple statement can be made for arbitrary gating variable dynamics.

However, we consider the Morris-Lecar (ML) model as a starting point for constructing a reduced model that incorporates an approximation to this more complicated integral. This is a model with a single gating variable, and is defined as follows:

$$\begin{aligned} \frac{dV}{dt} &= g_{\mathrm{Ca}} m_\infty(V)\left(E_{\mathrm{Ca}} - V\right) + g_{\mathrm{K}} n \left(E_{\mathrm{K}} - V\right) + I(t) = f(V, n) + I(t) + z \\ \frac{dn}{dt} &= \frac{n_\infty(V) - n}{\tau_n(V)} \\ x_\infty(V) &= \frac{1}{2}\left[1 + \tanh\left(\frac{V - V_x^T}{V_x^S}\right)\right] \; (x \in \{m, n\})\,, \quad \tau_n(V) = \frac{\tau_n^0}{\cosh\left(\frac{V - V_n^T}{2V_n^S}\right)}. \end{aligned} \tag{4.26}$$

Using substitution of variables to solve for $n(t)$ yields

$$n(t) = e^{-\int^t ds\,[1/\tau_n(V(s))]} \left[n(0) + \int^t ds_1 \, \frac{n_\infty(V(s_1))}{\tau_n(V(s_1))} \, e^{\int^{s_1} ds_s [1/\tau(V(s_2))]} \right]. \tag{4.27}$$

Approximating the integrals in the above expression using the "right-side-rule" Riemann sum approximation yields

$$n(t_n) = e^{-\Delta t/\tau_n(V(t_n))} \left[n(t_{n-1}) + \frac{n_\infty(V(t_n))}{\tau_n(V(t_n))} \Delta t \right] \tag{4.28}$$

where $\Delta t = t_n - t_{n-1}$. Such an expression may be substituted into a discretized forward integration of the $\dot{V}$ equation, thus providing the form of the nonlinear term in the NARMAX construction.

What is surprising is that this simple construction appears to reproduce the spiking statistics of a ML model with channel noise [29]:

$$\begin{aligned}
\frac{dV}{dt} &= g_{\mathrm{Ca}} m_\infty(V)\,(E_{\mathrm{Ca}} - V) + g_{\mathrm{K}} n\,(E_{\mathrm{K}} - V) + I(t) = f(V, n) + I(t) + z \\
\frac{dn}{dt} &= \frac{n_\infty(V) - n}{\tau_n(V)} + \left[\frac{n_\infty(V) + (1 - 2n_\infty(V))\,n}{N_{\mathrm{K}} \tau_n(V)} \right]^{1/2} \frac{dW}{dt} \chi_n(n) \\
x_\infty(V) &= \frac{1}{2} \left[1 + \tanh\left(\frac{V - V_x^T}{V_x^S} \right) \right] \, (x \in \{m, n\}) \,, \quad \tau_n(V) = \frac{\tau_n^0}{\cosh\left(\frac{V - V_n^T}{2V_n^S} \right)}. \\
\chi_n(n) &= \begin{cases} A e^{-B/\left[1-(2n-1)^2\right]}, & 0 < n < 1 \\ 0, & \text{otherwise} \end{cases}
\end{aligned} \tag{4.29}$$

The noise in this model is multiplicative because it depends on n and V. Despite this fact, the reduced form of ML proposed for the deterministic model appears to recreate the spiking statistics of the multiplicative noise model, even though the functional form of the noise is not taken into account in the construction of the model. While the author is not prepared to report on these results yet, it opens an avenue of inquiry for the reduction of arbitrary neuron models, even if the observed system contains nontrivial multiplicative noise statistics. Additionally, ML just serves as a stepping stone for more complex neuron models containing additional gating variables, many of which follow the general form of (4.29).

Chapter 5

Deep Learning as Statistical Data Assimilation

5.1 Introduction

Thus far this dissertation has focused primarily on the application of data assimilation (DA) methods to solving nonlinear inference problems in physical dynamical systems. The use of variational and stochastic model reduction methods for state and parameter estimation in partially observed, chaotic dynamical systems, as well as in a simple stochastic spiking neuron model, was explored in some detail. In the case of variational DA, insights from linear inverse problems were used to develop a method for estimate selection to improve prediction quality of model estimates in nonlinear systems. These problems all had a similar overall structure, in which information is transferred from a time series of observations to a model of the observed system using to estimate the properties and dynamical state of a proposed physical model for the observed system. The models considered were in the form of (stochastic) ordinary differential equations (ODEs and SDEs), where physical properties manifested as model parameters, and the primary concern in estimating the state was to generate a set of initial conditions for prediction of the system's future state by way of forward integration in time.

In this final chapter, variational data assimilation is expanded to encompass a seemingly unrelated problem, which is the training and analysis of deep neural network (DNN) models for machine learning (ML). ML has rapidly grown as a field over the past several decades alongside advancements in computational capabilities and developments in deep learning [67, 31, 53]. The process of DNN training shares a surprising number of features with our previous task, which may equally well be re-framed as "training" a dynamical system to recognize the physical properties of an observed system from time series data, with the goal of choosing the optimal system based on its ability to predict forward in time from a large collection of partial, noisy initial conditions. This latter task is essentially the same as generalization in ML model training.

In this chapter it is shown that variational DA in dynamical systems and deep neural network training are essentially equivalent statistical physics problems. Numerical experiments with neural network training using variational assimilation are presented to argue the utility of this equivalence. Ultimately, it is shown that DA may be a powerful tool for gaining a deeper understanding of the training problem in ML, and by extension a potential supplement to existing training algorithms such as backpropagation.

5.2 Deep Neural Network Training

Deep neural networks (DNNs) [53, 31] are a class of neural network models which contain more than one hidden layer of neurons. One type of DNN is the multi-

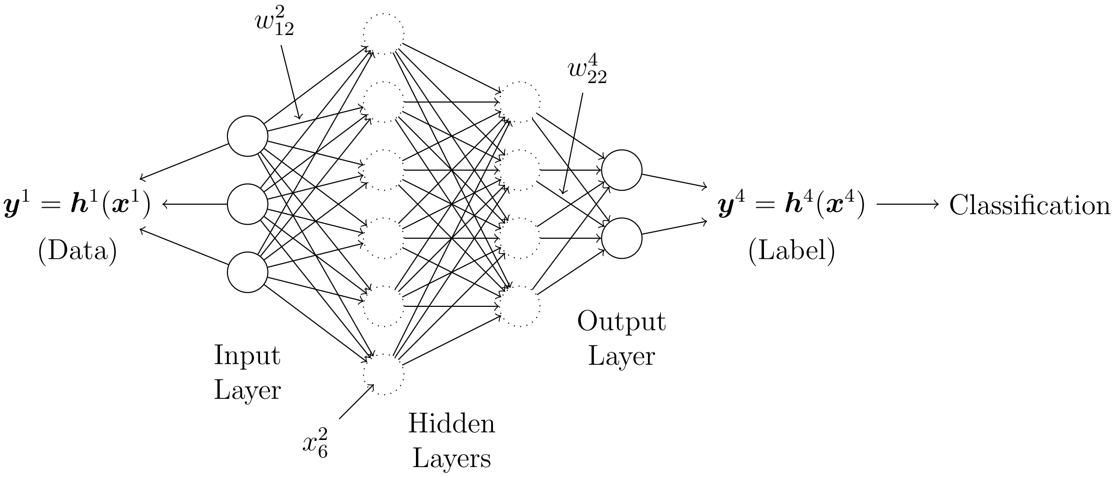

w^2_{12}
w^4_{22}
$\boldsymbol{y}^1 = \boldsymbol{h}^1(\boldsymbol{x}^1)$
(Data)
$\boldsymbol{y}^4 = \boldsymbol{h}^4(\boldsymbol{x}^4)$
Classification
(Label)
Output Layer
Input Layer
x^2_6
Hidden Layers

Figure 5.1: Diagrammatic sketch of the data classification process using a multi-layer perceptron, an example of a deep neural network. In this construction, the input data is assumed to be a measurement of the state of the input layer, $\boldsymbol{x}^1$; this propagates through "synapses" to the hidden layers, and is then projected into the 2-dimensional output layer which is measured as $\boldsymbol{y}^N$. The vertices represent synapses, defined by the activation function $\boldsymbol{a}$ and the weights $\mathbf{w}^n$. The input data is finally classified by choosing just one element of the label.

layer perceptron (MLP) [67, 31], itself a nonlinear extension of the older perceptron model first described by F. Rosenblatt in 1958 [75]. MLPs are neural networks that consist of layers of "neurons" connected by "synapses", in which neuron i in layer n is assigned a single real-valued activity variable x_i^n. The synapses are defined by activation functions which transfer these activities forward through the layers of the network. In the formulation presented here, these activation functions act on weighted sums of the activities in layer n to produce an activity in neuron i in layer $n+1$:

$$x_i^{n+1} = a\left(\sum_j w_{ij}^{n+1} x_j^n\right) \equiv a\left(z_i^n\right). \tag{5.1}$$

Several widely-used choices for these activation functions are described in [67, 31, 53]; a common choice is the sigmoid function:

$$x_i^{n+1} = \sigma(z_i^n) = \frac{1}{1 - e^{-z_i^n}}. \tag{5.2}$$

The parameters of this model are the weights at each layer, $\mathbf{w}^n$, which set the strengths of synaptic connections between one layer and the next. To preface the remainder of this discussion, training a neural network amounts to adjusting the network weights so that the model accurately predicts labels assigned to a collection of input data sets. For example, when neural networks are used in image recognition tasks, the input data is the pixel values that define the image, and the output is a label for the image. Examples of this image recognition task are described in [34, 26, 25].

A collection of K labeled data/label pairs are presented to the network at the input and output layers, respectively. These pairs are denoted by $\boldsymbol{y}_{(k)}^1$ (the kth example of input data) and $\boldsymbol{y}_{(k)}^N$ (the label for the kth example of input data). It is generally assumed that these data pairs are noisy; in connection with the theory behind data assimilation, $\boldsymbol{y}_{(k)}^1 = \boldsymbol{h}(\boldsymbol{x}_{(k)}^1) + \boldsymbol{\xi}$ is a measurement of the input layer corresponding to the kth data example, and $\boldsymbol{\xi}$ is the noise in the input data. Similarly, the data at the output layer $\boldsymbol{x}_{(k)}^N$, $\boldsymbol{y}_{(k)}^N$, is a measurement of the label for the data. Note that it was necessary to append example indices k to the activities in addition to the data pairs, because each input to the network generates a distinct set of activities throughout the layers. However, no such index is used for the network weights. This is because the goal of training a network is to identify a *single* collection of weights that describes the entire class of data represented by the K examples. This eventually allows for the prediction of labels for newly-presented input data which is unlabeled.

In a gradient based approach to training, these weights are “learned” by minimizing a cost function that compares the activity at layer n to the data at layer n. It is common to use a least-squares metric for this cost function:

$$C_K = \frac{1}{2KLN} \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{i=1}^{L} R_m^n \left(x_{(k),i}^n - y_{(k),i}^n \right)^2 . \tag{5.3}$$

In the machine learning context, data is typically only available for the input and output layers, so R_m^n is only nonzero when $n = 1$ or N. The optimization is carried out subject to the activation function defined in (5.1). The minima of C_K are network activities and synaptic weights which identify locally optimal machines for the labeling task.

In general, this cost function contains many local minima. The activation functions are nonlinear, and because the network model contains so many parameters and activation states, C_K is riddled with local minima corresponding to nearly-equivalent classification machines. In general there should also be a large number of local minima associated with suboptimal machines. Identifying the global minimum, if it exists, is an NP-complete problem [63].

To further complicate this task, the assumption that there is no error in the activation function model creates a strong constraint on the optimization. This led to difficulties with identifying low-action minima, or perhaps the global minimum, in the dynamical systems data assimilation problem; the similarities with the structure of this problem creates the expectation that there will be a similar difficulty encountered here. In the field of deep learning, strategies exist for dealing with these problems to some degree. In this chapter, it is shown that the neural network training problem is nearly equivalent to variational data assimilation with dynamical systems, warranting an extension of the DA method presented earlier in this dissertation to encompass NN training and sidestep these difficulties with an alternative approach.

5.3 Establishing the Equivalence with Data Assimilation

Consider the gradient-based approach to training when the strong model constraint on minimizing C_K is relaxed to allow for model errors at each layer in the network. Adding the model error into the cost function as a penalty term:

$$A_{ML}(X,Y) = C_K(X,Y) + \frac{R_f}{2NKD} \sum_{k=1}^{K} \sum_{n=1}^{N-1} \sum_{i=1}^{D} \left[x^{n+1}_{(k),i} - a(z^{n}_{(k),i}) \right]^2 \tag{5.4}$$

where $z^n_{(k),i}$ is the weighted sum over all the neurons in layer n defined in (5.1). The notation used in this definition of the extended cost function is highly suggestive: X and Y are defined as path vectors consisting of the ensemble of network states and weights, and the ensemble of labeled data pairs; and the new cost function A_{ML} is defined as the action of machine learning. Note that the original cost function is recovered when $R_f/R_m \to \infty$ (R_m is included in the definition of C_K), in correspondence with the same limit of the data assimilation action which enforces the dynamical model of the observed system as a hard constraint on path estimates.

This action for the path integral approach to variational data assimilation in dynamical systems, denoted separately here as A_{DS}, was already derived in Chapter 1. In that approach, the probability distribution $P(X|Y)$ for the model state and parameter estimates conditioned on the measurements $Y = \left\{ \boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^N \right\}$ made within the observation window $[t_1, t_N]$ at discrete times $t_1, t_2, \ldots, t_N$ is used to calculate statistics of state and parameter estimates. Under the Laplace approximation this becomes an optimization problem, in which estimates with low, well-separated action values dominate. Such estimates are chosen to define the estimated model, which is finally used to make predictions beyond the end of the estimation window. This procedure is essentially equivalent to validation by prediction in machine learning.

Finally, in the dynamical systems case, if the measurements are "direct" and independent from one another in time with Gaussian errors, with an error covariance equal to $R^n_{m,\ell}$ for the measurement of state x_ℓ at time t_n; and the model errors are taken to be Gaussian as well with inverse noise covariances $R^n_{f,i}$, then the action takes on the following functional form:

$$A_{DS}(X,Y) = \sum_{n=1}^{N}\sum_{\ell=1}^{L}\frac{R^n_{m,\ell}}{2}\left[x^n_\ell - y^n_\ell\right]^2 + \sum_{n=1}^{N-1}\sum_{i=1}^{D}\frac{R^n_{f,i}}{2}\left[x^{n+1}_i - f_i\left(\boldsymbol{x}^n;\boldsymbol{\theta}\right)\right]^2 \tag{5.5}$$

Comparing A_{ML} with A_{DS} establishes the near-equivalence of the data assimilation and network training problems under the substitution of time with layer number. The main difference between these two problems is that data is presented to a model as a time series in data assimilation, whereas the data is an ensemble of K pairs of input/output data in machine learning. In data assimilation, introducing a model error into A_{DS} thus acts as a smoother for state estimates; as more data is introduced to the system, more information about the solution becomes available and the estimates tend to "collapse" to trajectories about which the noisy data signal oscillates. It is unclear to the author whether or not R_f should act as a smoothing parameter for neural networks as well.

This equivalence motivates the use of variational annealing (VA), as it was presented in Chapter 3, for the neural network training task. The ultimate goal of training is to produce optimal neural network architectures for predicting labels of new data inputs, requiring accurate estimation of the conditional expectation values of functions $G(X)$ acting on the paths of activations and weights. Estimating these expectation values using the Laplace approximation presented in Chapter 1 requires finding minima of A_{ML}. While the machine learning action, A_{ML} is not a convex function in path space, it is sufficient to use minima which are well-separated in action from others with a much lower value of A_{ML}. These minimum action paths, X_0, dominate the approximation because the next-higher action path, X_1, contributes less by a factor of $\exp\left[A_{ML}(X_1) - A_{ML}(X_0)\right]$, and is thus exponentially suppressed.

Here it will be shown through numerical experiments using VA in the twin experiment setting that A_{ML} is not convex, but that a dominant minimum-action path may appear. This is expected to provide accurate approximations to the expectation values of path space functions, especially once sufficient information has been transferred to the model as data is presented to it. In the twin experiment, a collection of input/output pairs are generated by a neural network with a known architecture. The network that is trained by VA has a similar architecture, but contains a smaller number of hidden layers. The results of estimation and prediction using the estimated network are explored as the number of hidden layers, and thus the complexity of the network, as well as the number of examples presented to the networks are varied.

5.4 Training the Multi-Layer Perceptron

Here, a network with $N = 100$ layers and $D = 10$ neurons per layer, with sigmoid activation functions, is constructed to generate synthetic data. Another network, which is called the estimated network, is constructed with a similar architecture, except that $N < 100$ and is varied. In both cases the networks have one input and output layer, and $N - 2$ hidden layers. The data-generating network is assigned synaptic weights drawn from a uniform distribution on the interval $[-0.1, 0.1]$.

After assigning the weights to the data-generating network, a large collection of input/output pairs is created by generating inputs $\boldsymbol{x}^1_{(k)}$ which are fed to the input layer of the data-generating network. This generates a corresponding output $\boldsymbol{x}^N_{(k)}$.

Additionally, a small amount of Gaussian noise is added to each input and output, with mean 0 and variance 0.0025, which defines the data ensemble $\{\boldsymbol{y}^1_{(k)}, \boldsymbol{y}^N_{(k)}\}$. These pairs are saved for training and validating the estimated networks, where in each experiment K of these pairs are used for training, and a separate set of K_{val} are used to validate the network through prediction. In each case, all elements of $\boldsymbol{y}^1_{(k)}$ and $\boldsymbol{y}^N_{(k)}$ are presented to the estimated network during training, corresponding to $L = D = 10$ or full observation. The effects of reducing L to be less than D are not explored here, although this may lead to interesting consequences or applications. One idea, for example, is to use training to fill "gaps" in input data, such as in increasing the resolution of input images.

The model of the estimated network is purposefully chosen to have a smaller number of layers than the data-generating network to test the limitations of a "wrong" model. As N increases, thus increasing the complexity of the network model, it is expected that its ability to predict will improve as it approaches the size of the true network. Additionally, for a given estimated network size, the number of training pairs K is varied. The expectation is that introducing more data will provide more information about the true network to the estimated network and, thus, improve its ability to predict the outputs of new input data in the validation set.

With data presented to only the input and output layers, the machine learning action becomes

$$A_{ML}(X,Y) = \frac{1}{K}\sum_{k=1}^{K}\left\{\frac{R_m}{2L}\sum_{i=1}^{L}\left[\left(x^1_{(k),i} - y^1_{(k),i}\right)^2 + \left(x^N_{(k),i} - y^N_{(k),i}\right)^2\right]\right.$$
$$\left. + \frac{R_f}{ND}\sum_{n=1}^{N}\sum_{i=1}^{D}\left[x^{n+1}_{(k),i} - a_i\left(\sum_{j=1}^{D} w^n_{ij} x^n_{(k),j}\right)\right]^2\right\}. \qquad (5.6)$$

It should be noted again that there is only a *single* set of weights in this action for the K training data pairs, so that the end result is just one network architecture to use for prediction purposes. Using the same weights for each data pair k requires a distinct configuration of activations for each pair; this is reflected by indexing the activations $\boldsymbol{x}^n_{(k)}$ by k.

In variational annealing, $R_f/R_m = \gamma$ is initially chosen equal to 10^{-8}, and $R_f(\beta) = R_f\alpha^\beta$ with $\alpha = 1.1$. The implementation of VA is similar to that described in Chapter 3: `VarAnneal` [76] uses ADOL-C to evaluate derivatives of A_{ML}, and the optimization is carried out with the L-BFGS-B [8, 89, 62] algorithm. Note that, because the activation functions are already in the form of a forward mapping, the Hermite-Simpson approximation does not need to be employed in defining the model error. The action is simply of the form A_{ML} in 5.6.

Finally, the $K \times D \times N$ network activations and $D^2 \times (N-1)$ weights must be initialized before beginning the VA training procedure. The weights were randomly drawn from the uniform distribution on the interval $[-0.1, 0.1]$, while the activations were drawn from uniform distributions on the interval $[0.4, 0.6]$.

5.4.1 Validation by Prediction

The weights in the estimated network are set by training, after which it is validated in a prediction step, where a separate set of K_{val} data pairs not seen by the network during training are used. These K_{val} pairs are chosen from the larger data set generated earlier by the $N = 100$ network (thus, the estimated network is making predictions on new data inputs from the same system it was estimating during training). The validation set inputs are fed to the input layer of the estimated network, and the average prediction error is calculated as

$$E^2 = \frac{1}{LK_{val}} \sum_{k=1}^{K_{val}} \sum_{i=1}^{L} \left(x^N_{(k),j} - y^N_{(k),j} \right)^2 . \tag{5.7}$$

All $L = 10$ inputs are presented at the input layer.

5.4.2 Training and Validation Results

Training the estimated network was tested with $N = 10$, 20, and 50 layers, and each of these networks was presented with $K = 1$, 2, 5, and 10 examples of input/output pairs. VA was initialized with $N_{init} = 100$ paths in each case according to the choice of initialization values presented earlier.

The results of training with variational annealing are displayed in figures 5.2 through 5.4. Each initialization is tracked over the course of annealing, so that each plot contains as many as $N_{init} = 100$ distinct action levels. In all three cases, presenting just one example to the network is insufficient for identifying a distinct lowest action level; using the Laplace approximation to compute conditional expectation values of activations and weights most likely fails in this case. Additionally, the extremely small magnitude of the action for all of the levels indicates overfitting. This is true if, through training, the networks are arranged in weight configurations such that the data of the input and output pair are matched exactly by the estimated activations through the model.

This would explain why the measurement error term is so small, but for A_{ML} to be small the model error must also be small. It is suspected that this occurs because for an estimated network with N layers, there are $N \times D$ activation states and $(N-1) \times D^2$ weights which are all allowed to vary freely, meaning that the network is probably "flexible" enough to arrange itself into a valid configuration

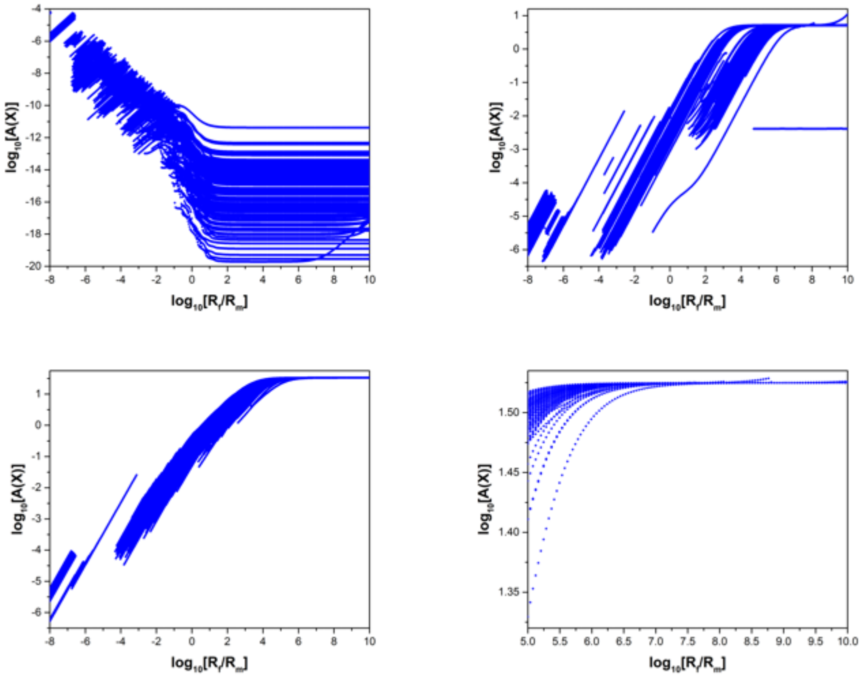


Figure 5.2: Action levels for training the activations and weights in the $N = 10$ estimated network with $D = 10$ neurons per layer, starting VA at $R_f/R_m = 10^{-8}$ with $N_{init} = 100$ randomly chosen initial network configurations, using synthetic data generated from an $N = 100$, $D = 10$ network. Top left: $K = 1$ examples are presented during training. There is no dominant, well-separated minimum, and the action is near zero for all estimates (a sign of overfitting to the single example). Top right: With $K = 2$ examples, different initializations to VA all eventually approach $A = 1$, becoming independent of R_f when $R_f/R_m \gg 1$. Bottom left: Only one action level remains for $K = 10$ examples when $R_f/R_m \gg 1$. Bottom right: Zoomed-in view of the $K = 10$ action levels for large R_f/R_m, highlighting a single remaining action level.

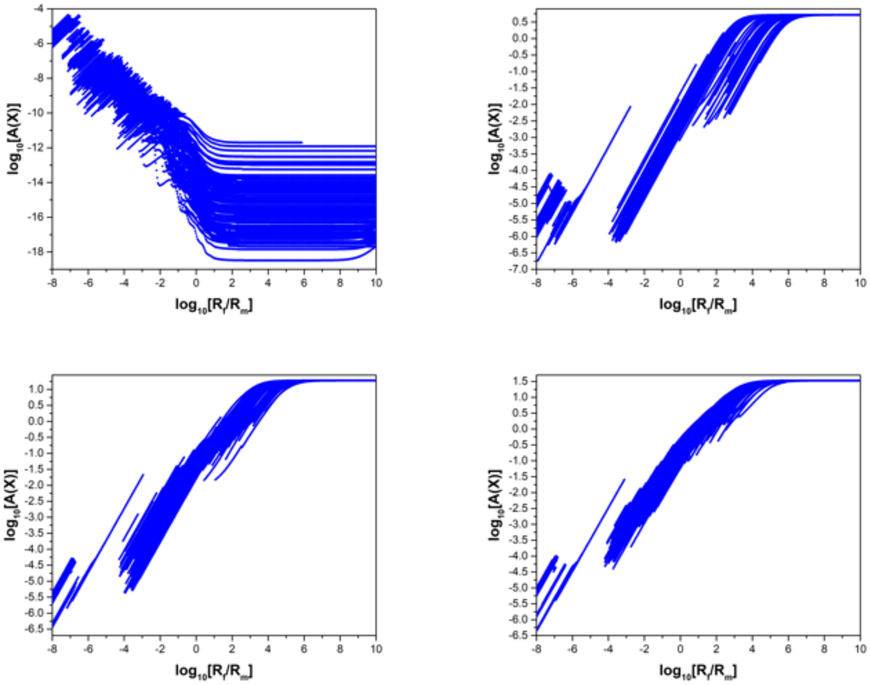


Figure 5.3: Action levels for training the activations and weights in the $N = 20$ estimated network with $D = 10$ neurons per layer, starting VA at $R_f/R_m = 10^{-8}$ with $N_{init} = 100$ randomly chosen initial network configurations, using synthetic data generated from an $N = 100$, $D = 10$ network. The result of training is qualitatively similar to the $N = 10$ case, with $K = 1$ (top left), $K = 2$ (top right), $K = 5$ (bottom left), and $K = 10$ (bottom right) input/output pairs presented during training.

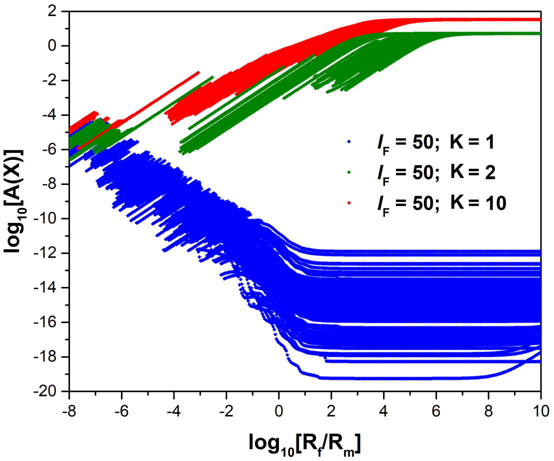


Figure 5.4: Action levels for training the activations and weights in the $N = 50$ estimated network with $D = 10$ neurons per layer, starting VA at $R_f/R_m = 10^{-8}$ with $N_{init} = 100$ randomly chosen initial network configurations, using synthetic data generated from an $N = 100$, $D = 10$ network. Here, the $K = 1$, 2, and 10 training pair cases are overlaid

(according to the model) to accommodate almost any single input/output pair during training. However, this configuration is specifically the one which was able to describe a particular input/output relationship, and there is no guarantee that such a network will be good at predicting the output of any other, new input. It is in fact shown below that the prediction error is about twice as large when the network is trained with just one example, compared to a larger number K.

When $K = 2$ and thus a second example is presented for training, there are still many distinct action levels for small R_f/R_m, but they mostly collapse to just one value by the end of the annealing. The value of the action also appears to become mostly independent of R_f/R_m when it is large. In the dynamical systems case, this was expected to happen for the global minimum, which tracks the true solution during annealing and, thus, the discrepancy function $\boldsymbol{g}$ of which the model error is composed goes to zero. The measurement error approaches an expected value equal to the RMS value of the measurement noise in this limit. This also appears to be the outcome in the

$K = 2$ case described here. The ML action is normalized such that the expected value of the measurement error is 1 when the model error goes to zero and is independent of R_f as it becomes large.

This indicates that the nature of the global minimum changes when more than one example of data is presented to the network. When $K = 1$, it appears that the network has enough flexibility to describe the input/output relationship between the *noisy* output and the *noisy* input, even when R_f/R_m becomes large and thus the model is strongly enforced on solutions. This is actually not dissimilar from what might happen in a dynamical system if, say, samples from just two neighboring time points are presented during estimation: if the model states and parameters are all allowed to vary, then it is likely a choice of parameterization exists which is able to exactly describe that single time step, even when the data is noisy. Finding the true, noiseless solution requires introducing a longer time series of data. With enough data the global minimum of the action is located on the true state of the system, at which point the solution is smoothed by the enforcement of a dynamical system.

Similarly, in the neural network case the global minimum appears to be the estimate which exactly describes the noisy input/ouput pairs when too few examples are presented, but the introduction of more data pushes this minimum into a different configuration which is suspected to eliminate the noise from the data. To show this, the estimates should be compared to the noiseless input/output pairs generated by the $N = 100$ network to see if the estimation error becomes small. This was not performed in this experiment; however, the theoretical argument remains compelling.

As more examples are introduced, the action similarly increases towards 1 and eventually becomes independent of R_f/R_m, but the action levels also start to cluster together more tightly. This indicates that introducing more examples reduces the number of local minima, so that there exist fewer valid network configurations which also match the input/output pairs well. This probably also means that the problem of over-fitting with $K = 1$ is reduced more and more as K increases.

Finally, the prediction capabilities of each estimated network were tested with new data pairs from the validation set. Again, this data was generated by the same $N = 100$ network as the training data. Each estimated network was presented with $K_{val} = 100$ data inputs, where the output was calculated by evaluating the transfer functions layer-by-layer. The resulting outputs were compared to the outputs in the validation data, using the error function defined in (5.2).

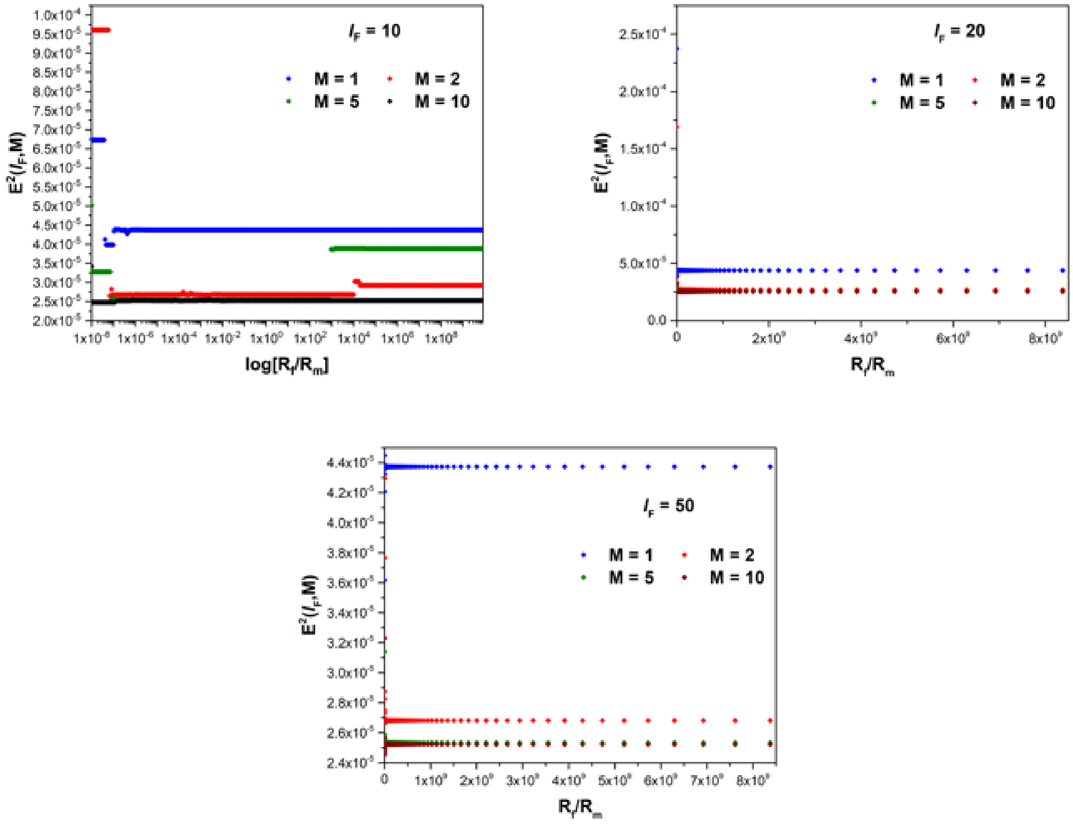


Figure 5.5: Prediction errors in the validation step for the $N = 10$, 20, and 50 estimated networks (in these plots, $l_f = N$ and $M = K$). In each case the lowest-action network configuration was used to predict with $K_{val} = 100$ input/output pairs, and the errors were averaged over K_{val} and the output activations. The prediction errors tend to decrease as more data is introduced to the estimated networks during training.

The results are displayed in figure 5.5. In each panel, the prediction error as a function of R_f/R_m is plotted for a given estimated network size N (in these plots, $N = l_f$ and $K = M$). It is seen that increasing the number of training pairs decreases the prediction error in the $N = 20$ and $N = 50$ networks, with a similar trend for the $N = 10$ network, which is another indicator that introducing more examples gives the estimated network more information about the true network.

5.5 Conclusions

In this chapter, an equivalence between variational data assimilation for dynamical systems and neural network training in machine learning was established. A machine learning action, A_{ML}, was derived as an extension of the typical L_2 cost function used in other training approaches such as backpropagation, where the strong constraint of exact enforcement of the activation functions in the neural network was relaxed and introduced as a penalty, as in the data assimilation action A_{DS}. Numerical experiments showed that variational annealing, an algorithm for computing minima of the action, identifies what is suspected to be the global minimum of the action.

The change in the structure of the action as more examples of data are introduced during training, in addition to changing the number of layers, was an indicator of the prediction capabilities of a given network, verified by prediction using a validation set of novel input/output pairs. A small number of layers and data examples was shown to be required to achieve small prediction errors for data generated by a much larger network. VA thus provides a potentially useful approach to network training which successfully estimates network with low prediction error, but is also potentially a tool for addressing questions about how much information or how complex of a network is required for this to be true.

This chapter was adapted from Henry D. I. Abarbanel, Paul J. Rozdeba, and Sasha Shirman, *Machine Learning; Deepest Learning as Statistical Physics Problems*, which is currently under review for publication, with the permission of the authors. The dissertation author was a co-author of this paper.

Bibliography

[1] H. D. I. Abarbanel. *Analysis of Observed Chaotic Data*. Springer, 2012.

[2] H. D. I. Abarbanel. *Predicting the Future: Completing Models of Observed Complex Systems*. Springer, 2013.

[3] E. L. Allgower and K. Georg. *Numerical continuation methods: an introduction*, volume 13. Springer Science & Business Media, 2012.

[4] E. Armstrong, A. V. Patwardhan, L. Johns, C. T. Kishimoto, H. D. I. Abarbanel, and G. M. Fuller. An optimization-based approach to calculating neutrino flavor evolution. *Physical Review D*, 96(8):083008, 2017.

[5] C. M. Bender and S. A. Orszag. *Advanced Mathematical Methods for Scientists and Engineers I: Asymptotic Methods and Perturbation Theory*. Springer, 2013.

[6] S. A. Billings. *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons, 2013.

[7] S. Bochkanov. ALGLIB. http://www.alglib.net.

[8] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

[9] W. H. Calvin and C. F. Stevens. Synaptic noise and other sources of randomness in motoneuron interspike intervals. *Journal of Neurophysiology*, 31(4):574–587, 1968.

[10] J. A. Carton and B. S. Giese. A reanalysis of ocean climate using simple ocean data assimilation (soda). *Monthly Weather Review*, 136(8):2999–3017, 2008.

[11] H. F. Chen. New approach to recursive identification for ARMAX systems. *IEEE Transactions on Automatic Control*, 55(4):868–879, 2010.

[12] A. J. Chorin and O. H. Hald. *Stochastic Tools in Mathematics and Science*, volume 3. Springer, 2009.

[13] A. J. Chorin, O. H. Hald, and R. Kupferman. Optimal prediction with memory. *Physica D: Nonlinear Phenomena*, 166(3):239–257, 2002.

[14] A. J. Chorin and F. Lu. Discrete approach to stochastic parametrization and dimension reduction in nonlinear dynamics. *Proceedings of the National Academy of Sciences*, 112(32):9804–9809, 2015.

[15] A. J. Chorin and X. Tu. Implicit sampling for particle filters. *Proceedings of the National Academy of Sciences*, 106(41):17249–17254, 2009.

[16] P. Dayan and L. F. Abbott. *Theoretical neuroscience*, volume 806. Cambridge, MA: MIT Press, 2001.

[17] J. Demmel. *Applied Numerical Linear Algebra.* Society for Industrial and Applied Mathematics, 1997.

[18] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. Liu. A supernodal approach to sparse partial pivoting. *SIAM Journal on Matrix Analysis and Applications*, 20(3):720–755, 1999.

[19] P. Deuflhard. *Newton methods for nonlinear problems: affine invariance and adaptive algorithms*, volume 35. Springer, 2011.

[20] H. Eibern and H. Schmidt. A four-dimensional variational chemistry data assimilation scheme for eulerian chemistry transport modeling. *Journal of Geophysical Research: Atmospheres*, 104(D15):18583–18598, 1999.

[21] D. J. Evans and G. P. Morriss. *Statistical Mechanics of Nonequilibrium Liquids (Academic, London, 1990).* ANU Press, 1995.

[22] G. Evensen. *Data Assimilation: The Ensemble Kalman Filter.* Springer, 2009.

[23] M. Falcke, R. Huerta, M. I. Rabinovich, H. D. Abarbanel, R. C. Elson, and A. I. Selverston. Modeling observed chaotic oscillations in bursting neurons: the role of calcium dynamics and ip 3. *Biological cybernetics*, 82(6):517–527, 2000.

[24] J. Fan and Q. Yao. *Nonlinear Time Series: Nonparametric and Parametric Methods.* Springer, 2003.

[25] H. Fang. Using deep learning to create professional-level photographs. Google Research Blog, https://research.googleblog.com/2017/07/using-deep-learning-to-create.html.

[26] H. Fang and M. Zhang. Creatism: A deep-learning photographer capable of creating professional work. arXiv:1707.03491 [cs.CV].

[27] R. P. Feynman, A. R. Hibbs, and D. F. Styer. *Quantum Mechanics and Path Integrals.* Courier Corporation, 2010.

[28] R. FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophys. J.*, 1(6):445–466, 1961.

[29] R. F. Fox and Y.-n. Lu. Emergent collective behavior in large numbers of globally coupled independently stochastic ion channels. *Physical Review E*, 49(4):3421, 1994.

[30] R. C. Gonzalez, R. E. Woods, et al. Digital image processing, 1992.

[31] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning.* MIT press, 2016.

[32] A. Griewank and A. Walther. ADOL-C. http://projects.coin-or.org/ADOL-C.

[33] A. Griewank and A. Walther. Getting started with ADOL-C. *Combinatorial Scientific Computing, CRC Press*, pages 181–202, 2012.

[34] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, 316(22):2402–2410, 2016.

[35] B. S. Gutkin and G. B. Ermentrout. Dynamics of membrane excitability determine interspike interval variability: a link between spike generation mechanisms and cortical spike train statistics. *Neural Computation*, 10(5):1047–1065, 1998.

[36] F. Hamilton, T. Berry, N. Peixoto, and T. Sauer. Real-time tracking of neuronal network structure using data assimilation. *Physical Review E*, 88(5):052715, 2013.

[37] E. Hannan. The identification and parameterization of armax and state space forms. *Econometrica: Journal of the Econometric Society*, pages 713–723, 1976.

[38] P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion.* SIAM, 1998.

[39] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[40] A. L. Hodgkin and A. F. Huxley. Currents carried by sodium and potassium ions through the membrane of the giant axon of loligo. *The Journal of Physiology*, 116(4):449–472, 1952.

[41] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[42] N. Kadakia. The dynamics of nonlinear inference. 2017.

[43] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[44] E. Kalnay, M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen, et al. The ncep/ncar 40-year reanalysis project. *Bulletin of the American meteorological Society*, 77(3):437–471, 1996.

[45] A. Karimi and M. R. Paul. Extensive chaos in the Lorenz-96 model. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(4):043105, 2010.

[46] J. Keating and W. Thach. Nonclock behavior of inferior olive neurons: interspike interval of purkinje cell complex spike discharge in the awake behaving monkey is random. *Journal of Neurophysiology*, 73(4):1329–1340, 1995.

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

[48] M. Kostuk, B. A. Toth, C. D. Meliza, D. Margoliash, and H. D. I. Abarbanel. Dynamical estimation of neuron and network properties II: path integral Monte Carlo methods. *Biological Cybernetics*, 106(3):155–167, 2012.

[49] V. Krinsky and Y. Kokoz. Analysis of the equations of excitable membranes. *Biofizika*, 18:506–511, 1973.

[50] J. M. Krisp, S. Peters, C. E. Murphy, and H. Fan. Visual bandwidth selection for kernel density maps. *Photogrammetrie-Fernerkundung-Geoinformation*, 2009(5):445–454, 2009.

[51] W. Kutta. Beitrag zur näherungweisen integration totaler differentialgleichungen. *Zeitschrift für Mathematik und Physik*, 46:435–453, 1901.

[52] P. S. Laplace. Memoir on the probability of the causes of events. *Statistical Science*, 1(3):364–378, 1986.

[53] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, May 2015.

[54] J. Liepe, P. Kirk, S. Filippi, T. Toni, C. P. Barnes, and M. P. Stumpf. A framework for parameter estimation and model selection from experimental data in systems biology using approximate bayesian computation. *Nature protocols*, 9(2):439–456, 2014.

[55] D. L. Logan. *A first course in the finite element method*. Cengage Learning, 2011.

[56] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, 1963.

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

[58] F. Lu, K. K. Lin, and A. J. Chorin. Data-based stochastic model reduction for the Kuramoto–Sivashinsky equation. *Physica D: Nonlinear Phenomena*, 340:46–57, 2017.

[59] C. D. Meliza, M. Kostuk, H. Huang, A. Nogaret, D. Margoliash, and H. D. I. Abarbanel. Estimating parameters and predicting membrane voltages with conductance-based neuron models. *Biological Cybernetics*, 108(4):495–516, 2014.

[60] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[61] G. Milshtejn. Approximate integration of stochastic differential equations. *Theory of Probability & Its Applications*, 19(3):557–562, 1975.

[62] J. L. Morales and J. Nocedal. Remark on algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization. *ACM Transactions on Mathematical Software*, 38(1):7, 2011.

[63] K. G. Murty and S. N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39(2):117–129, 1987.

[64] J. Nagumo, S. Arimoto, and S. Yoshizawa. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10):2061–2070, 1962.

[65] R. D. Neidinger. Introduction to automatic differentiation and MATLAB object-oriented programming. *SIAM Review*, 52(3):545–563, 2010.

[66] H. Nyquist. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644, 1928.

[67] R. Potember. "Perspectives on research in artificial intelligence and artificial general intelligence relevant to DoD". Technical Report JSR-16-Task-003, JASON, The MITRE Corporation, McLean, VA, USA, Jan. 2017.

[68] W. H. Press. *Numerical Recipes 3rd Edition: The Art of Scientific Computing.* Cambridge University Press, 2007.

[69] J. C. Quinn. *A Path Integral Approach to Data Assimilation in Stochastic Nonlinear Systems.* PhD thesis, University of California, San Diego, 2010.

[70] L. R. Rabiner and B. Gold. Theory and application of digital signal processing. *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975. 777 p.*, 1975.

[71] F. Rawlins, S. Ballard, K. Bovis, A. Clayton, D. Li, G. Inverarity, A. Lorenc, and T. Payne. The met office global four-dimensional variational data assimilation scheme. *Quarterly Journal of the Royal Meteorological Society*, 133(623):347–362, 2007.

[72] V. C. Raykar and R. Duraiswami. Fast optimal bandwidth selection for kernel density estimation. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pages 524–528. SIAM, 2006.

[73] J. Rinzel. On repeated activity in nerve. *Federation Proceedings*, 37(14):2793, 1978.

[74] C. P. Robert. *Monte Carlo Methods*. Wiley Online Library, 2004.

[75] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386, 1958.

[76] P. Rozdeba. VarAnneal. http://www.github.com/paulrozdeba/varanneal.

[77] C. Runge. Über die numerische auflösung von differentialgleichungen. *Mathematische Annalen*, 46(2):167–178, 1895.

[78] C. E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[79] A. N. Tikhonov. On the stability of inverse problems. In *Dokl. Akad. Nauk SSSR*, volume 39, pages 195–198, 1943.

[80] F. Topputo and C. Zhang. Survey of direct transcription for low-thrust space trajectory optimization with applications. In *Abstract and Applied Analysis*, volume 2014. Hindawi Publishing Corporation, 2014.

[81] B. A. Toth, M. Kostuk, C. D. Meliza, D. Margoliash, and H. D. I. Abarbanel. Dynamical estimation of neuron and network properties I: variational methods. *Biological Cybernetics*, 105(3-4):217–237, 2011.

[82] B. A. Turlach et al. *Bandwidth Selection in Kernel Density Estimation: A Review*. Université Catholique de Louvain Louvain-la-Neuve, 1993.

[83] S. F. Walter. PYADOLC. http://github.com/b45ch1/pyadolc.

[84] R. E. Wengert. A simple automatic derivative evaluation program. *Commun. ACM*, 7(8):463–464, Aug. 1964.

[85] W. J. Wilbur and J. Rinzel. A theoretical basis for large coefficient of variation and bimodality in neuronal interspike interval distributions. *Journal of Theoretical Biology*, 105(2):345–368, 1983.

[86] D. S. Wilks. Effects of stochastic parametrizations in the Lorenz '96 system. *Quarterly Journal of the Royal Meteorological Society*, 131(606):389–407, 2005.

[87] J. Ye, N. Kadakia, P. Rozdeba, H. Abarbanel, and J. Quinn. Improved variational methods in statistical data assimilation. *Nonlinear Processes in Geophysics*, 22(2):205–213, 2015.

[88] J. Ye, D. Rey, N. Kadakia, M. Eldridge, U. I. Morone, P. Rozdeba, H. D. Abarbanel, and J. C. Quinn. Systematic variational method for statistical nonlinear state and parameter estimation. *Physical Review E*, 92(5):052901, 2015.

[89] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560, 1997.

[90] J. Zinn-Justin. *Quantum field theory and critical phenomena*. Clarendon Press, 1996.

[91] R. Zwanzig. Nonlinear generalized langevin equations. *Journal of Statistical Physics*, 9(3):215–220, 1973.

[92] R. Zwanzig. *Nonequilibrium statistical mechanics*. Oxford University Press, 2001.