# Stochastic Modeling of Neuron Dynamics

A Major Qualifying Project Report

Submitted to The Faculty of

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science in Mathematical Sciences

by

Natalie Tierney

Project Advisor:

Prof. Andrea Arnold

April 2024

**Abstract**

Incorporating randomness is critical in mathematical modeling to best represent real-life phenomena, especially in the biological space. A common strategy to best capture these complex dynamics is to operate inside a stochastic framework. This project explores extensions of an ordinary differential equation (ODE) system to stochastic differential equations (SDEs), specifically to the Hodgkin-Huxley neuron model. Through numerical approaches such as the Euler-Maruyama (EM) method and the Ensemble Kalman Filter (EnKF), we aim to analyze how the neuron dynamics compare in an ODE vs. SDE structure. Data assimilation techniques are utilized motivated by the potential to apply EM and EnKF methods to inverse problems with real data.

# Executive Summary

Analyzing the spiking behavior of a neuron is a problem that has been investigated by researchers for decades. Starting with the work of Hodgkin and Huxley and 1952, modifications to dynamical neuron models have been explored in efforts to both mathematically optimize the model and improve the predictive power of estimating when a neuron generates an action potential. In particular, working inside a stochastic framework has gained popularity to accurately depict in-vivo neuron dynamics.

The goals of this project are to expand the Hodgkin-Huxley (H-H) neuron model to a stochastic differential equation framework to understand its dynamics and perform state estimation. We begin by comparing the different types of single biological neuron models and define the equations of the H-H neuron model. We provide mathematical background into deterministic and stochastic modeling, specifically on how we can formulate a system of differential equations and apply numerical methods to solve these coupled systems. Through a working example of a damped mass-spring oscillator, we demonstrate the effectiveness of these numerical methods and extend them to our H-H model through performing various simulations.

Motivated by a competition held in 2009 that challenged competitors to predict spiking behavior of a neuron given real data, we explore the capabilities of Kalman filtering to estimate our noisy model states. In particular, we focus on the ensemble Kalman filter (EnKF) as a data assimilation approach and detail its algorithm by example. We then compare the deterministic and stochastic H-H framework results of the EnKF via simulations that change how we create our prediction state ensemble and introduce noise.

Finally, we consider the situation where we predict state estimates with less data to increase the complexity of our inverse problem. From our original estimates, we consider observation time series that are 1/100 and 1/10 of the original predictions. The goal of this experiment is to see if we can still retain information of our unmeasured variables. We discuss the differences in our estimates and supply relevant explanations of our results.

# Acknowledgements

I would like give a sincere thank you to Dr. Andrea Arnold for her expertise and guidance throughout the duration of this project. Her encouragement made this project possible. Additionally, I would like to thank my peers, friends, and family for their unwavering support throughout this process.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

In mathematical modeling, incorporating randomness is critical to best represent real-life phenomena. One common approach to best capture these real-world dynamics is to work inside a stochastic framework. Stochasticity is one of the most important elements to understanding information processing in the cerebral cortex [41]. Specifically, it has been shown in the literature that incorporating stochastic elements in neuron models is crucial to accurately describing the overall dynamics of neuron spiking behavior [47]. Neuron models are complex, dynamical systems that measure the behavioral states of membrane potential [24]. There are several researched neuron models varying in their intricacies and simplifications [36, 25, 28, 27, 34, 22]. The three main classes of single neuron models in the biological space are: binary [36, 22], threshold [28, 54, 22], and dynamical models [34, 27, 25, 15, 22].

Binary single neuron models are models that simply depict whether a neuron is firing (Boolean value of 1) or not firing (Boolean value of 0, known as the quiescent state) [22]. The research of single binary neuron models originated in 1943 by Warren McCulloch and Walter Pitts with their efforts to drastically simplify the electric dynamics of a singular neuron [36]. The McCulloch-Pitts model takes a collection of inputs $x_i$ for $i = 1, ..., n$ and computes a weighted sum, using the weights on each input, denoted as $w_i$. Once the neuron reaches a certain defined threshold, the model outputs a 1 upon firing; 0 otherwise. A depiction of the McCulloch-Pitts model is shown in Figure 1.1. The binary neuron model is not commonly used by neuroscientists in practice today since it does not capture the complexities of single neuron dynamics [22]. Regardless, the work done by McCulloch and Pitts is the building block to more complex single neuron models and neural networks.

Threshold single neuron models depict the situation where voltage passes through the cell membrane as a function of time and crosses a certain threshold to generate an action potential [53]. The action potential is often referred to as the "firing" or "spiking" event of a neuron. The voltage is then reset to an initial value, and the process repeats. Some common types of threshold single neuron models are the leaky integrate-and-fire (LIF) model [22, 17] and the Izhikevich model [22, 28]. LIF models include a "leak" term denoting the proportion in which the voltage falls in relation to its initial value [22].

The last category of single neuron models are dynamical models. As its name suggests, these types of models aim to characterize the electrochemical dynamics of the flow of ions across the cell membrane. Dynamical models do not have a specified threshold and are more complex than threshold single neuron models [22]. Examples of dynamical models include the Morris-Lecar model (1981) [34] defined by the

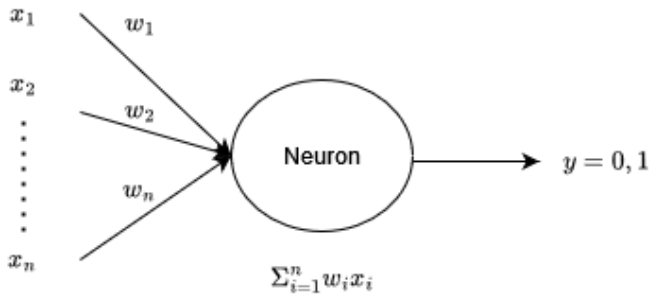


**Figure 1.1:** Single binary biological neuron model. Adapted from the work of McCulloch-Pitts [36].

following system of differential equations:

$$C\frac{dV}{dt} = I - \overline{g}_{Ca} m(v)(v - V_{Ca}) - \overline{g}_K w(v - V_K) - g_L(v - V_L) \tag{1.1}$$

$$\frac{dw}{dt} = \alpha(v)(1 - w) - \beta(v)(w) \tag{1.2}$$

Here, $V$ is the voltage or membrane potential, and $w$ is the recovery variable [22]. The model was derived from analyzing the voltage of a giant barnacle muscle fiber. Another dynamical single neuron model is the FitzHugh-Nagumo model (1961) [27], defined by the following equations:

$$\frac{dV}{dt} = \alpha(w + 3V - V^3 + I) \tag{1.3}$$

$$\frac{dw}{dt} = -(V - a + bw)/\alpha \tag{1.4}$$

where $V$ and $w$ hold the same meaning as they do in the Morris-Lecar model. The FitzHugh-Nagumo model is simpler to model and more mathematically tractable while still capturing important neuron behaviors [54]. Both the Morris-Lecar and FitzHugh-Nagumo models are simplifications of the Hodgkin-Huxley (H-H) neuron model. The H-H model, named after its creators Alan Hodgkin and Andrew Huxley, is a set of coupled differential equations describing the voltage and gating variable dynamics of a neuron. These equations were derived manually upon observations of a giant squid axon [25]. The model created by Hodgkin and Huxley in 1952 is still present in literature today to analyze neuron spiking behavior [33, 51, 7, 49, 41, 15]. For example, in the H-H model, ion channels are either in open or closed states. Fox and Lu reduce the dynamics of the ion channels and incorporate a noise term to their equations [15, 9]. Swietlicka et al. explore the potential of a multi-state representation of the potassium and sodium channels [49]. Parameter estimation [33, 51] and the incorporation of time-varying currents [7] further supplement to the understanding of neuron spiking behavior.

The first thought to introduce stochasticity into neuron models is to extend the model to a Markov process [22]. This includes modeling neural dynamics as probabilistic models of ion channel states [47, 49, 52], probabilistic spiking neural models [29], or through the addition of noise by stochastic resonance [35, 22]. Recently, more work of incorporating noise through extensions to stochastic differential equations, or SDEs, is appearing in the literature [15, 20, 9, 41]. This framework is more consistent to capturing the dynamics of our neuron model and less costly computationally compared to simply adding noise [9, 40, 15, 20].

An SDE is a differential equation that consists of one or more stochastic processes whose solution is also a stochastic process [24]. At the heart of SDEs is Brownian motion. A Brownian motion is a

stochastic process whose increments with respect to time are both independent and Gaussian. SDEs are used to model many practical applications stemming to areas of finance [32, 24, 43, 48], epidemiology [21, 24, 37], biology [48, 24, 7, 15, 20, 9, 41], and physics [24].

The solution to an SDE is derived by integration from the Fundamental Theorem of Calculus. However, due to Brownian motion not being differentiable, the integral has to be taken "with respect to Brownian motion" [24]. There are two forms of integration that we can apply to the stochastic framework: the Itô integral or Stratonovich integral. In this project, we focus on the Itô definition. With an SDE formulated from Itô calculus, we employ numerical methods to solve the system [24, 20, 26, 32, 6, 39, 23]. The most common numerical method for SDEs is an extension of Euler's method called the Euler-Maruyama (EM) method [24]. Other numerical methods used to find SDE solutions include Milstein's method, which uses a truncation of Taylor expansions [13, 6], second order Runge-Kutta methods [6, 39], and rooted trees [6].

With a model established, state estimation can be performed. State estimation of a physical system predicts the behavior of both measurable and unmeasurable states with both process noise and measurement noise. State estimation algorithms require a mathematical model and input data to provide a predicted estimate of the internal states. The most common algorithmic approach for state estimation in biological applications is Kalman filtering [37, 7, 30, 48]. Upon observing data, Kalman filters perform assimilation to update predictions. In the literature, the extended Kalman filter (EKF) [48, 33] and the ensemble Kalman filter (EnKF) [37, 7, 12, 51, 30] are validated as effective state estimation approaches for neuron models.

In this project, we have two main objectives:

- Through the incorporation of noise, can we fit a mathematical model that accurately describes a neuron's spiking behavior?
- Given a model and data, can we capture information about unmeasured variables?

To answer the first objective, we compare and contrast deterministic and stochastic models of the H-H system of equations. We define the H-H model and identify the SDEs of H-H using Itô calculus. Upon establishing the system, we compare Euler's method as the numerical solution to the H-H ordinary differential equations (ODEs) to that of the EM method as the numerical solution to the H-H SDEs. We detail these numerical methods and compare their approximations through simulations.

Next, after fitting an appropriate model, we perform state estimation of both the measured and unmeasured states of the H-H model. In particular, we explore the capabilities of the EnKF in the prediction of SDE H-H state estimates. To our knowledge, little research has been conducted comparing the results of EnKF between the H-H deterministic and SDE forms. We offer, through verification and validation of algorithmic constraints, a detailed review of the performance of the EnKF given noisy measurements inside of the H-H SDE framework. This project serves as a proof-of-concept for accurately capturing the action potential dynamics of a neuron under the H-H model with aspirations of performing state estimation with real-world data.

The paper is organized as follows. Chapter 2 reviews the mathematical background of deterministic and stochastic modeling. We introduce numerical methods following a working example of a damped mass-spring harmonic oscillator. Chapter 3 provides an overview of the H-H model, summarizing all relevant equations in both ordinary and SDE forms. We perform numerical experiments to our H-H systems as defined in Chapter 2. Chapter 4 presents a background on Kalman filtering and exercises described algorithms to our working example from Chapter 3. Chapter 5 applies the state estimation approaches described in Chapter 4 to the deterministic and SDE H-H equations through a collection of simulations. Chapter 6 offers a discussion of results and future work.

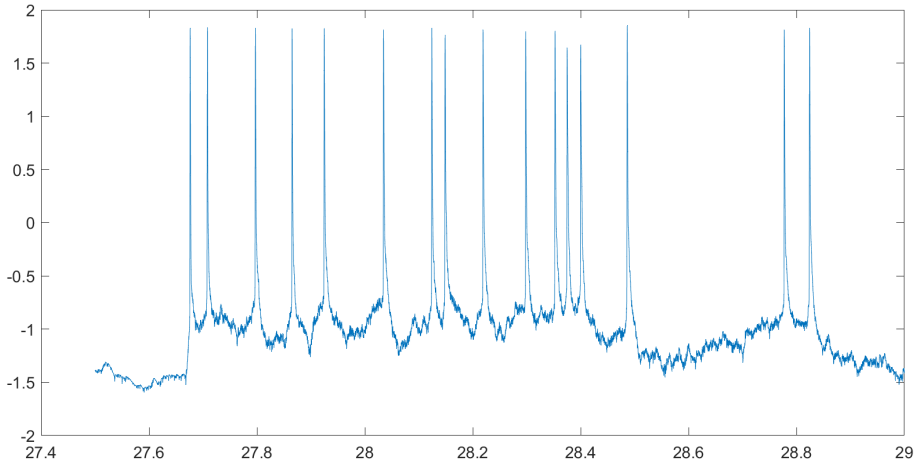


**Figure 1.2:** Snippet of data from the 2009 Quantitative Single-Neuron Model Competition [17].

## 1.1 Quantitative Single-Neuron Model Competition

Motivation for this project comes from a competition held in 2009 developed by researchers in Lausanne, Switzerland. In the competition, individuals were tasked with three challenges. Each of these challenges pertained to modeling the spiking behavior of L5 pyramidal neurons. L5 pyramidal neurons are the most populous neuron in the excitatory part of the brain and reach all six layers of the cortex [45]. In fact, approximately 2/3 of all neurons in the mammalian cerebral cortex are made up of pyramidal neurons [5]. Competitors were given data from an experiment where an input current was passed through the cell membrane of a mouse neuron at varying intervals and voltage levels. A snippet of the voltage data that the competitors were given is shown in Figure 1.2

The goal of the competition was to formulate the most optimized and accurate mathematical model such that it could capture when the actual neuron spiked as well as predict spiking behavior beyond the given data. The same fluctuating input current was injected into a live neuron via an electrode and into a model neuron [17]. The performance of the model was measured as a percentage of the correctly predicted spikes from the model to the live neuron [16].

In the competition, only threshold single neuron models were submitted with the winner of 2 out of 3 challenges gaining an accuracy of 59.6% of the 13 repetitions of the injected current. How well would a dynamical model do in this situation? The competition developers speculated that the omission of detailed, dynamical model submissions was due to their complex nature and tuning constraints [16]. Through the course of this project, we analyze the H-H model inside of a stochastic framework with hopes that this mathematical model can accurately depict the spiking behavior of a neuron with respect to time. For potential future work, this framework with proper parameter tuning can be tested against a real dataset similar to that of this competition to see if the voltage spiking behavior is captured.

# Chapter 2

# Mathematical Background: Ordinary and Stochastic Differential Equations

In literature and practice, we use models to simulate the behavior of a biological neuron. By definition, a neuron has an input, internal dynamics, and output functions [22]. In this paper, we look at dynamical models due to their complexity and acceptance in literature. More specifically, the H-H model will be studied for its wide use in understanding electrophysiological neuron dynamics [7, 25, 22]. There exists a substantial amount research on deterministic forms of a single neuron model, most consisting of systems of ODEs to describe the behaviors of neuron states [7, 15, 28, 34, 27]. SDEs extend from ODEs by introducing noise terms in efforts to better depict neuron activity [9, 20, 41, 40, 54]. In the following sections, we provide background of deterministic and stochastic models and discuss the methods used to find their numerical solutions. We demonstrate the performance of these methods through a working example of a damped mass-spring harmonic oscillator.

## 2.1 Deterministic Models

Deterministic models are mathematical models that contain no random components [38]. All predictions in these models are solely based on initial conditions and defined parameter inputs into a set of equations [19]. A system of ODEs, or a set of equations which represent the derivative of an unknown function based on a set of real variables, is a form of a deterministic mathematical model [14]. ODEs help explain physical phenomena and require careful construction to accurately depict the dynamics of a variable of interest. ODEs appear in many application areas, including geophysics [38, 12, 11, 4], weather forecasting [14, 12, 24, 3], biology [7, 2, 1, 53, 24], business [32, 43], chemistry [24, 1], and epidemiology [37, 1].

### 2.1.1 Numerical Methods for ODEs

We commonly solve and visualize the dynamics of a deterministic system using numerical methods. Numerical methods are especially helpful when our ODE system is highly complex and nonlinear, since a closed form solution is not usually possible to find. Different approaches to solve a nonlinear system of ODEs include Euler's method, Heun's method–which modifies Euler's method [10]–and the Runge-Kutta methods in their explicit and implicit forms [55]. In this work, we use Euler's method to approximate solutions to ODEs. We apply Euler's method for its simplicity and our ability to choose small step sizes for integration. Euler's method can be derived from the first two terms of the Taylor

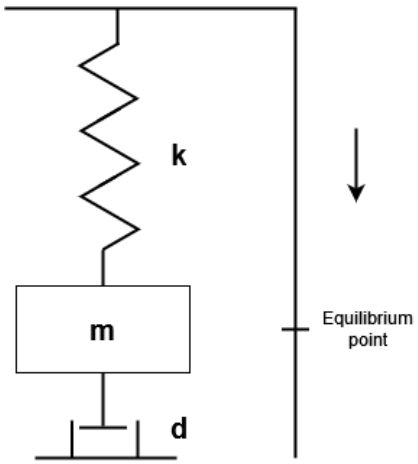


**Figure 2.1:** Damped Mass-Spring Oscillator. Here, the mass $m$ is at rest (equilibrium) with $m > 0$, the system is unforced ($F(t) = 0$), the damping coefficient $d$ is restricted by $d \geq 1$, and the spring constant $k$ is any real value satisfying $k > 0$.

series expansion [14, 31]. In other words, given an equation of the form:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}), \qquad \mathbf{x}(t_0) = \mathbf{x}_0 \tag{2.1}$$

Euler's method to solve our ODE for $\mathbf{x}$ is:

$$\mathbf{x}_{j+1} = \mathbf{x}_j + f(\mathbf{x}_j)\Delta t. \tag{2.2}$$

Euler's method uses a uniform incremental step size ($\Delta t$) for a fixed number of iterations. Here, $\mathbf{x}_j$ represents a $k \times 1$ vector, where $k$ is the number of variables or states in the system. Each $j$ denotes a time index, $j = 0, ..., T$, where we approximate $\mathbf{x}_j \approx \mathbf{x}(t_j)$. Euler's method allows us to capture the dynamics of variables in a deterministic system of equations.

### 2.1.2 Example: Mass-Spring System

To put Euler's method into practice, we'll consider the homogeneous spring-mass oscillator defined in equation (2.3):

$$mp''(t) + dp'(t) + kp(t) = 0. \tag{2.3}$$

We define $m > 0$ as the mass, $d \geq 1$ as the damping coefficient, $k > 0$ as the spring constant, and $p(t)$ denotes the position of the mass at a given time $t$. We depict the damped spring-mass oscillator in Figure 2.1.

We can rewrite this equation as a first order linear system:

$$\frac{dp}{dt} = v \tag{2.4}$$

$$\frac{dv}{dt} = -\frac{k}{m}p - \frac{d}{m}v \tag{2.5}$$

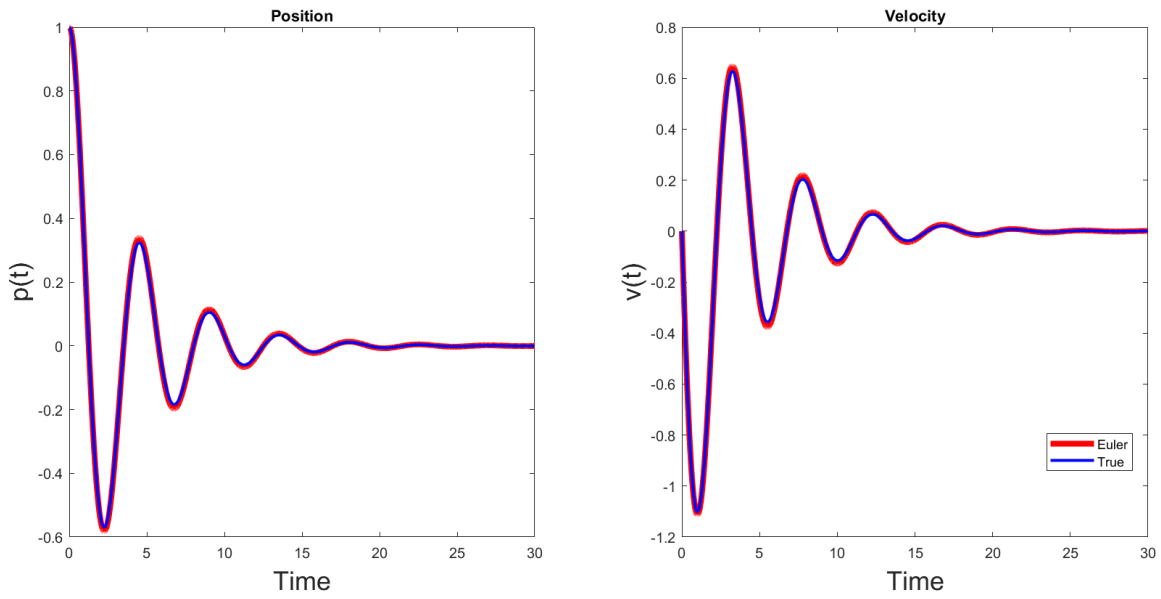


**Figure 2.2:** Euler's method to approximate the dynamics of the damped mass-spring oscillator. Here, $\Delta t = 0.01$, initial conditions for $p$ and $v$ are 0 and 1 respectively. The "true" solution is computed using the MATLAB built-in solver ode15s [46].

where $v(t) = p'(t)$ is the velocity. Let's consider the following damped, unforced system:

$$\frac{dp}{dt} = v \tag{2.6}$$

$$\frac{dv}{dt} = -2p - \frac{1}{2}v \tag{2.7}$$

Here we have a 8 slug mass attached to a spring stretched by a spring constant 16 lb/ft and a damping constant of 4 lb-sec/ft. We set the initial conditions $p(0) = 1$ and $v(0) = 0$. Using Euler's method, we can visualize the solutions $p(t)$ and $v(t)$, as seen in Figure 2.2.

The "true" solution uses a MATLAB (The MathWorks, Inc., Natick, MA) built-in ODE solver ode15s [46]. We use ode15s to compute the "true" solution as a benchmark for comparing it with the solution obtained from a numerical method. In this application, we visualize position and velocity as a function of time over a preset time interval $[0, 30]$. Our step size, $\Delta t$, is defined as 0.01, giving us 3000 time steps. By defining a small $\Delta t$, we observe continuous position and velocity curves. Euler's method for a numerical solution of ODEs sets the baseline for investigating numerical solutions of SDEs discussed in the next section.

## 2.2 Stochastic Models

Non-deterministic models, in contrast to deterministic, incorporate at least one random component. In the literature, there are different ways to extend the deterministic modeling framework to stochastic systems. Swietlicka et al. [49] include additive noise through Markov kinetic schemes inside a compartmental model or neural network, where a neural network in [49] is defined as the dendritic structure of the neuron in the H-H model. Gill [19] uses a network model derived from random graphs to incorporate stochasticity in modeling infectious disease transmission. Other works focus on Monte Carlo

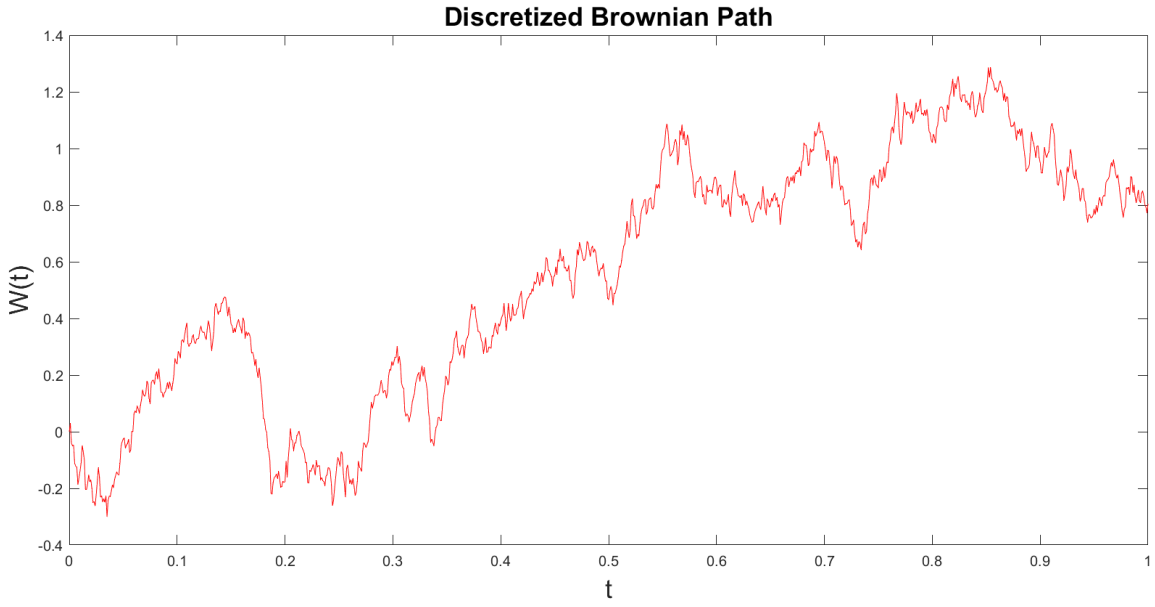


**Figure 2.3:** A singular discretized Brownian path over time interval $[0, 1]$. Each time step is 0.001 time units.

adaptations using discrete- or continuous-time Markov chains [24], stochastic difference equations [2], and SDEs [2, 6, 9, 32, 20, 41, 43, 48]–the stochastic framework studied in this project.

### 2.2.1 Stochastic Processes and Brownian Motion

In all these cases, the variables of interest are stochastic processes. A stochastic process is a collection of one or more random variables that are both dependent on time and defined on a sample space [2, 24, 18]. The sample space for a random variable $X$ is all the possible values $X$ can take. When both time and the random variables of interest are continuous, the stochastic realization $X$ becomes the solution to an SDE.

The stochastic process that is a defining feature of SDEs is Brownian motion. Brownian motion originated from work conducted by botanist Robert Brown in 1827 who observed random, erratic behavior of pollen grains. Albert Einstein, with his studies on gold particles in solution [2], and Norbert Wiener [24] are well-known scientists who provided contributions to the area. Brownian motion, often referred to as a standard Wiener process and denoted as $W(t)$, over a fixed time interval has three key properties [24]:

- $W(0) = 0$
- For two time points $s$ and $t$, $s < t$ over a fixed interval, $W(t) - W(s) \sim \mathcal{N}(0, t - s)$.
- For time points $s, t, u$ and $v$, with $s \leq t \leq u \leq v$ over a fixed time interval, increments $W(t) - W(s)$ and $W(v) - W(u)$ are independent.

Figure 2.3 shows a singular discretized Brownian path, taking discrete time points of our continuous time interval. Despite its jagged nature, the expectation of Brownian motion is smooth across multiple discretized Brownian paths, as seen in Figure 2.4.

When incorporating Brownian motion into a stochastic framework, the jagged behavior of Brownian paths can pose a problem. Brownian paths are not directly differentiable. Thus the notation $dW(t)$

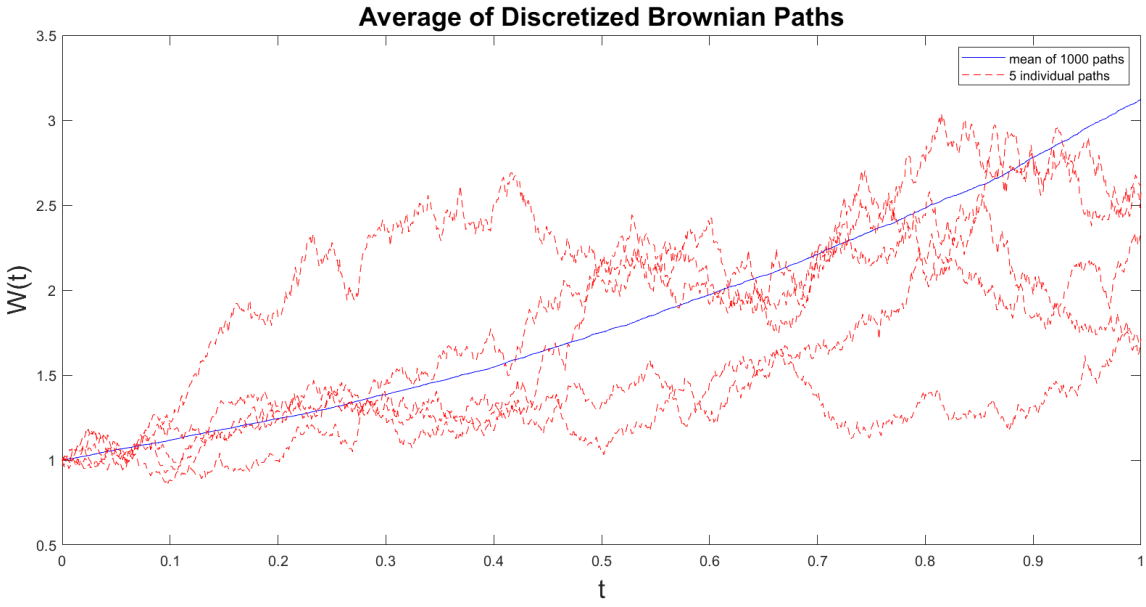


**Figure 2.4:** Five individual discretized Brownian paths over time interval $[0, 1]$, with $\Delta t = 0.001$ and the average of 1000 paths. Note that the average appears smooth. Adapted from [23].

does not have a practical meaning but is used for notation and referred to as Gaussian white noise [24]. This notation is important when defining SDEs.

### 2.2.2 SDEs

Differential equations are solved by integration as defined by the Fundamental Theorem of Calculus. However, to integrate stochastically, we have to perform integration with respect to Brownian motion [24, 18]. Given a function $g$, there are two common ways to integrate through use of the Riemann sum definition–the Itô integral and the Stratonovich integral. The Itô integral takes the Riemann sum integral approximation of the left-hand endpoints for our function $g$. To approximate this integral in the stochastic case, or integration with respect to Brownian motion, we get the following summand for $N$ being the number of time steps in our approximation:

$$\sum_{j=0}^{N-1} g(t_j)(W(t_{j+1}) - W(t_j)) \tag{2.8}$$

On the other hand, the Stratonovich integral takes the Riemann sum integral approximation based on midpoints. Therefore, the Stratonovich stochastic integral results in a midpoint Riemann sum formulation:

$$\sum_{j=0}^{N-1} g(\frac{1}{2}t_j + t_{j+1})(W(t_{j+1}) - W(t_j)) \tag{2.9}$$

Due to its usage in literature and its usefulness in determining the behavior of solutions [24]–an objective of this project–we use the Itô integral to derive the form of our SDEs.

With working definitions of Brownian motion and Itô integration established, we now formally define a system of SDEs. We recall the form of an ODE system from equation (2.1). A system of SDEs

in Itô form is given by:

$$d\mathbf{X}(t) = f(\mathbf{X}(t))dt + g(\mathbf{X}(t))d\mathbf{W}(t) \tag{2.10}$$

where $\mathbf{X}(t)$ is a stochastic process and our solution to the system. The function $f$ is commonly referred to as the drift coefficient, representing the direction and magnitude of the drift of the process dependent on time. In many cases, and in the applications used in this report, the drift coefficient is the deterministic form of our system of equations. The function $g$ is the diffusion coefficient which scales the fluctuations of random noise added to our system, and $\mathbf{W}(t)$ is a Brownian motion, vectorized by the number of states in our model. It is important to note that $dt$ and $d\mathbf{W}(t)$ have no practical differential meaning and serve simply as notation to extend from ODEs.

### 2.2.3 Numerical Methods for SDEs

Computing the dynamics of a stochastic system require additional considerations compared to its deterministic form. The solution to equation (2.10) can not always be derived analytically depending on the way $f$ and $g$ are defined, in addition to the fact that Brownian motion is not differentiable. An extension of Euler's method, Euler-Maruyama (or EM for short), is a popular and applicable method for SDEs. The EM method is a numerical approximation of solutions to SDEs named after Leonhard Euler and Gisiro Maruyama. The method extends from Euler's method for ODEs, given that our SDEs use the Itô integral form [24]. Other numerical methods used to find SDE solutions include Milstein's method which uses a truncation of Taylor expansions [13, 6], second order Runge-Kutta methods [6, 39], and rooted trees [6]. For its natural extension from Euler's method of ODEs and its usage in literature, we investigate the EM method further in this report. We recall the form for an SDE from equation (2.10). For a given step size $\Delta t$ and time index $j$ for $j = 0, ..., T$, the EM method approximates the solution of $\mathbf{X}(t_j)$ by $\mathbf{X}_j$ through the following equation:

$$\mathbf{X}_{j+1} = \mathbf{X}_j + f(\mathbf{X}_j)\Delta t + g(\mathbf{X}_j)\Delta \mathbf{W}_j \tag{2.11}$$

Each $\Delta \mathbf{W}_j$ is a Brownian path increment defined below:

$$\Delta \mathbf{W}_j := \mathbf{W}(t_{j+1}) - \mathbf{W}(t_j)$$

Given $k$ states in our model, we can define $W_j^i$, for $i = 1, ..., k$, as independent Brownian motion following $\mathcal{N}(0, \Delta t)$. In the literature, it has been shown that the EM approximation $\mathbf{X}_j \approx \mathbf{X}(t_j)$ improves as we increase the number of time indices and decrease our step size, leading to ideas of weak and strong convergence [24, 32, 13, 26]. Rather than prove this rigorously, we perform simulations where we increase the number of EM approximations and take their average in the next section.

### 2.2.4 Example: Mass-Spring System

We recall the damped mass-spring oscillator system in equations (2.4)–(2.5). We then extend the system to SDEs of the form:

$$dp = \Big[v\Big]dt + \sigma_p dW^1(t) \tag{2.12}$$

$$dv = \Big[-\frac{k}{m}p - \frac{d}{m}v\Big]dt + \sigma_v dW^2(t) \tag{2.13}$$

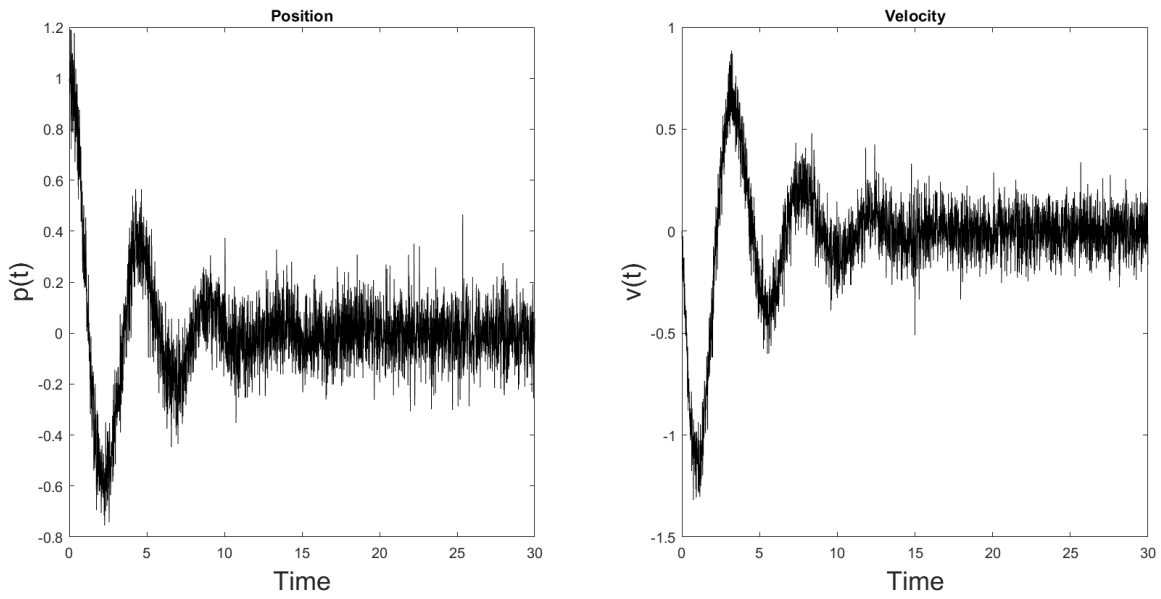


**Figure 2.5:** EM for damped mass-spring oscillator SDE system for a singular Brownian path. We define $\sigma_p = \sigma_v = 1$.

We denote the diffusion coefficient, or scale on our Brownian motion, on position and velocity in (2.13) as $\sigma_p$ and $\sigma_v$ respectively. We can replace $m$, $d$, and $k$ with the defined constants from the example in Section 2.1.2 to get the following system:

$$dp = vdt + \sigma_p dW^1(t) \tag{2.14}$$

$$dv = \Big[ -2p - \frac{1}{2}v \Big] dt + \sigma_v dW^2(t) \tag{2.15}$$

Let's define $\sigma_p = \sigma_v = 1$. By applying the EM method for a step size of 0.01 and 3000 time instances, we observe the dynamics of the position and velocity seen in Figure 2.5. These plots depict some jagged behavior, but as we increase the number of Brownian paths and take the average of our EM approximations, we see improved behavior of our two states. Figures 2.6, 2.7, 2.8, and 2.9 show this result. In these figures, we plot the results of our mass-spring SDEs solved via EM to compare with the results of the ODE system solved using Euler's method. As we increase the number of SDE simulations, we see the average approaches the value of our solution to the deterministic mass-spring system.

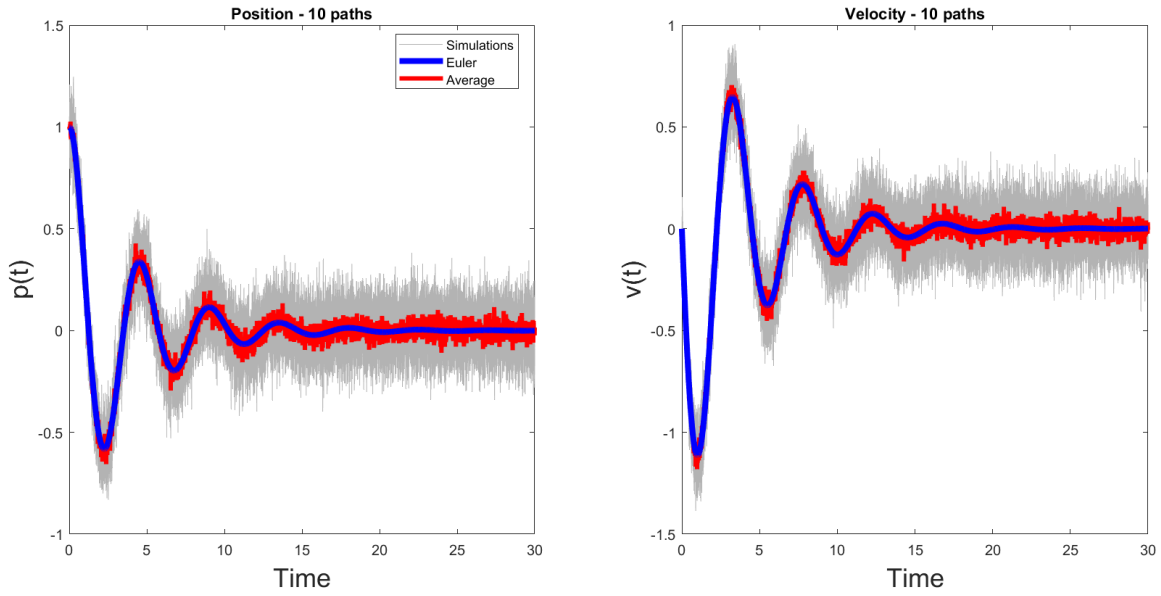


**Figure 2.6:** EM for damped mass-spring oscillator SDE system for $n = 10$ Brownian paths. We define $\sigma_p = \sigma_v = 1$. Comparing Euler's solution of the ODE system to the 10 EM solutions of the SDE system, we see that the average of the 10 EM solutions approaches Euler's solution. However, we still observe jagged behavior.

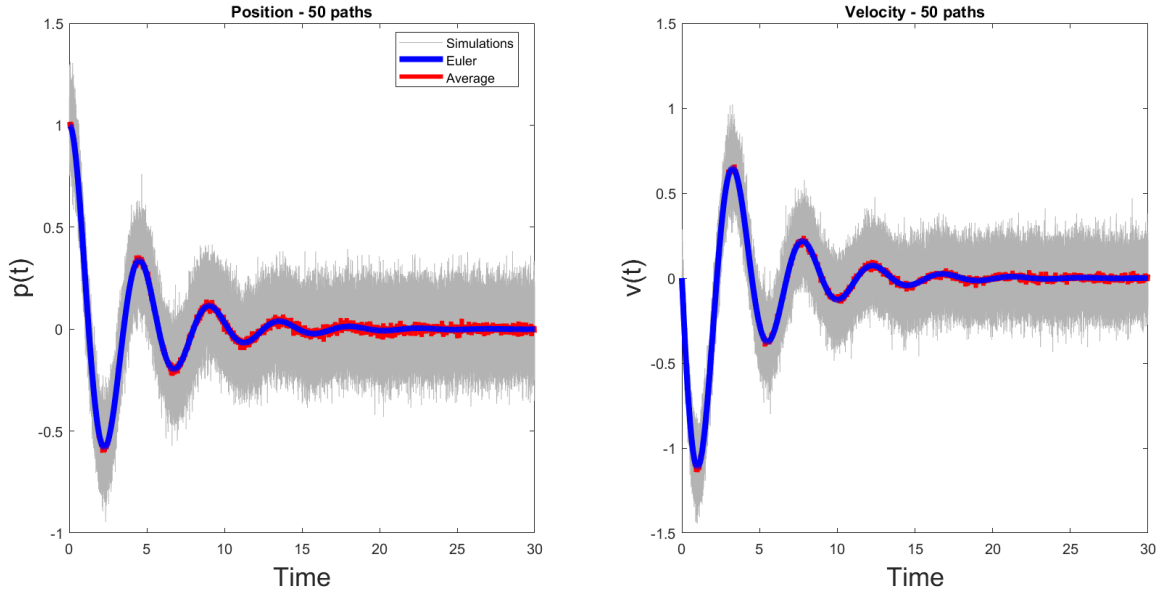


**Figure 2.7:** EM for damped mass-spring oscillator SDE system for $n = 50$ Brownian paths. We define $\sigma_p = \sigma_v = 1$. Comparing Euler's solution of the ODE system to the 50 EM solutions of the SDE system, we see that the average of the 50 EM solutions is much closer to that of Euler's solution. The average of 50 SDE simulations is closer to Euler's solution compared to 10 SDE simulations.

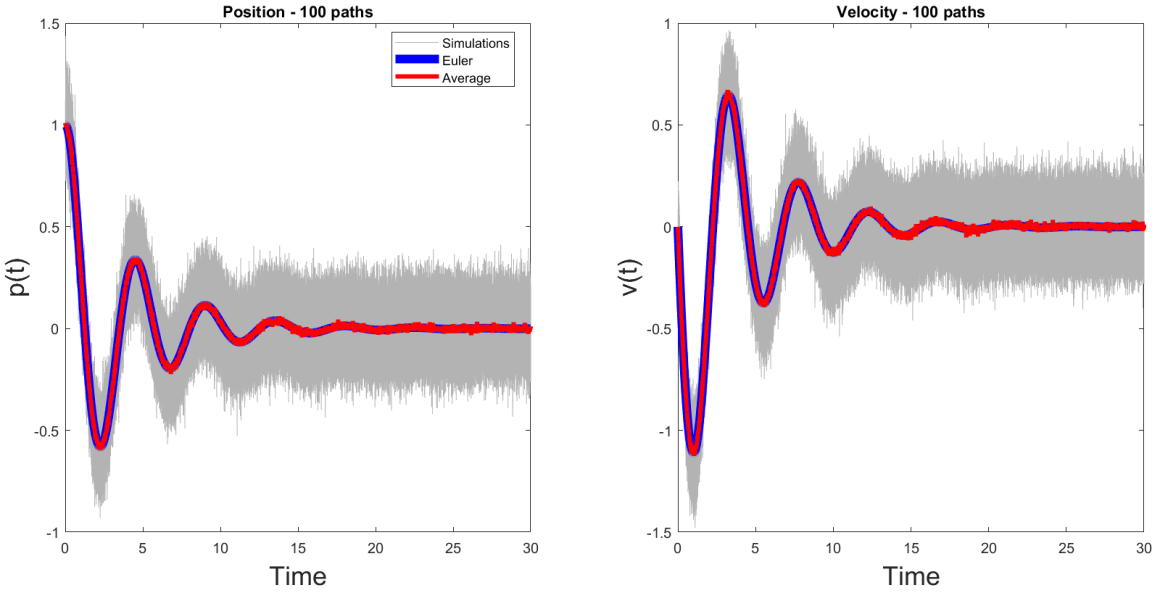


**Figure 2.8:** EM for damped mass-spring oscillator SDE system for $n = 100$ Brownian paths. We define $\sigma_p = \sigma_v = 1$. Comparing Euler's solution of the ODE system to the 100 EM solutions of the SDE system, we see that the average of the 100 EM solutions is closer to Euler's solution than that of 50 SDE simulations.

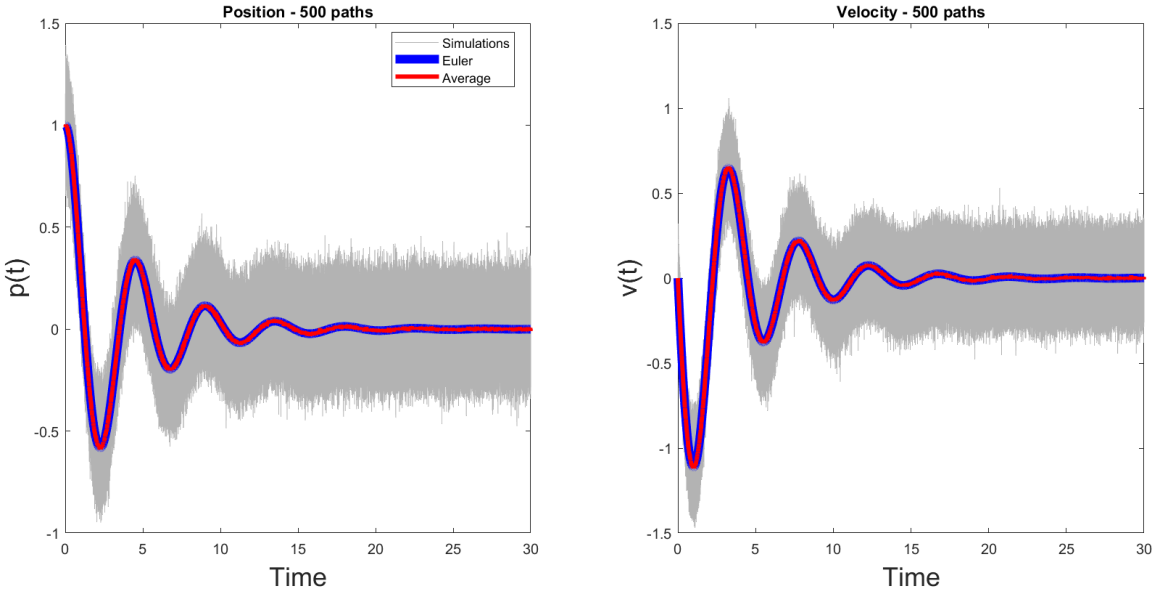


**Figure 2.9:** EM for damped mass-spring oscillator SDE system for $n = 500$ Brownian paths. We define $\sigma_p = \sigma_v = 1$. Comparing Euler's solution of the ODE system to the 500 EM solutions of the SDE system, we see that the average of the 500 EM solutions is closer to Euler's solution than that of 100 SDE simulations.

# Chapter 3

# Hodgkin-Huxley Model: Deterministic and Stochastic Forms

Biological neurons are inherently electrically excitable. In response to an electrical current being passed through the cell membrane, the neuron fires electric signals, referred to as action potential spikes. There have been numerous efforts throughout the literature that have attempted to mathematically optimize and predict the spiking behavior of neurons through modeling [24, 22, 27, 36, 34, 28]. Many of these models extend from the work by Alan Hodgkin and Andrew Huxley in 1952 [25, 41, 9]. In this chapter, we present an overview of the H-H equations and its extensions to SDEs. We apply Euler's method and the EM method in efforts to depict the dynamics of the H-H system in its ODE and SDE forms.

## 3.1 Experiments of Hodgkin and Huxley

The work of Alan Hodgkin and Andrew Huxley is a substantial contribution to the field of electrophysiology [44, 7]. Through experiments, Hodgkin and Huxley aimed to understand how voltage-gated ion channels cause an action potential event [51]. These experiments included studies of cell membrane permeability and the relationship between ion conductance and membrane potential [20]. The experiments suggested that sodium and potassium ion currents were products of conductances dependent on time with constant membrane and equilibrium potentials [25]. These conclusions were drawn from their depiction of a membrane as a simple electrical circuit [44, 7, 25]. Components of the circuit display how the resistance of an electrically charged ion channel is a function of both time and membrane potential [53]. Upon their observations and manual derivations, Hodgkin and Huxley divided up the total membrane current to summarize ionic currents as a system of relations between conductances and potentials [25]. These relations form a coupled system of differential equations we denote as the H-H deterministic model.

## 3.2 Deterministic Form of H-H

The H-H deterministic model is arguably one of the most dynamical, complex and precise models to describe the behavior of neuron states [22, 20, 15]. It is a four-dimensional model defined as the rate of change in the membrane potential (voltage $V$) that is proportional to the input ($I$) minus current output from the passing of sodium ($Na$) and potassium ($K$) ions through the cell membrane. We can

**Table 3.1:** Constant parameter values used in the H-H model.

| Parameter | Description | Value | Units |
|---|---|---|---|
| $C$ | Membrane capacity | 1.0 | $\mu F/cm^2$ |
| $V_{Na}$ | Sodium voltage | 115 | $mV$ |
| $V_K$ | Potassium voltage | $-12$ | $mV$ |
| $V_L$ | Leakage voltage | 10.6 | $mV$ |
| $\overline{g}_{Na}$ | Sodium gate constant | 120 | $m.mho/cm^2$ |
| $\overline{g}_K$ | Potassium gate constant | 36 | $m.mho/cm^2$ |
| $\overline{g}_L$ | Leakage gate constant | 0.3 | $m.mho/cm^2$ |

represent this relationship mathematically by equation (3.1):

$$C\frac{dV}{dt} = I - [g_{Na}(V - V_{Na}) + g_K(V - V_K) + g_L(V - V_L)] \tag{3.1}$$

where $C$ is the membrane conductance, $I$ is the applied current, $g_{Na}, g_K, g_L$ are the sodium, potassium and leak conductances, and $V_{Na}, V_K, V_L$ are the reserved potentials for sodium, potassium and leak, respectively [20, 40, 22]. These constants are summarized in Table 3.1. Equation (3.1) summarizes the deterministic form for the H-H model [40]. Equations (3.2) and (3.3) express the sodium and potassium conductances:

$$g_{Na} = m^3 h \bar{g}_{Na} \tag{3.2}$$

$$g_K = n^4 \bar{g}_K \tag{3.3}$$

Here, $\overline{g}_{Na}$ and $\overline{g}_K$ are the peak sodium and potassium conductances, and $m$, $h$, and $n$ are the voltage gating variables satisfying the differential equations (3.4):

$$\frac{dx}{dt} = \alpha_x(V)(1 - x) - \beta_x(V)x, \quad x = m,\ h,\ n \tag{3.4}$$

We refer to visualizations of the potassium and sodium Markov networks in the literature [22, 40, 47].

Each $m$, $h$ and $n$ can be thought of as the proportion of sodium or potassium gates open (for $m$, $n$) or closed ($h$) respectively at a given time in the ion channel [42, 7]. Governing transition rate functions define the proportion of activation and inactivation gates open at a given time. We define the rate functions ($\alpha_x$, $\beta_x$) in Table 3.2 for $x = m$, $h$, and $n$. Pairing equation (3.1) with (3.4) for each gating variable, we arrive at a coupled system of ODEs describing the dynamics of the four variables ($V$, $m$, $h$, and $n$) which forms the H-H model:

$$\frac{dV}{dt} = (I - (I_{Na} + I_K + I_L))/C \tag{3.5}$$

$$\frac{dm}{dt} = \alpha_m(V)(1 - m) - \beta_m(V)m \tag{3.6}$$

$$\frac{dh}{dt} = \alpha_h(V)(1 - h) - \beta_h(V)h \tag{3.7}$$

$$\frac{dn}{dt} = \alpha_n(V)(1 - n) - \beta_n(V)n \tag{3.8}$$

We summarize constants and parameters in Table 3.3.

**Table 3.2:** Equations for the ion channel rate functions.

$$\alpha_m(V) = \frac{0.1(25-V)}{\exp\left(\frac{25-V}{10}-1\right)} \qquad \beta_m(V) = 4\exp\left(\frac{-V}{18}\right)$$

$$\alpha_h(V) = 0.7\exp\left(\frac{-V}{20}\right) \qquad \beta_h(V) = 1/\exp\left(\frac{30-V}{10}-1\right)$$

$$\alpha_n(V) = \frac{0.1(10-V)}{\exp\left(\frac{10-V}{10}-1\right)} \qquad \beta_n(V) = 0.125\exp\left(\frac{-V}{80}\right)$$

**Table 3.3:** Components of the H-H system.

| | |
|---|---|
| $I_{Na} + I_K + I_L$ | ionic current density |
| $I$ | total membrane current |
| $C$ | membrane capacity |
| $\alpha_x, \beta_x$ | rates of ion flow (in/out) |
| $V(t)$ | voltage (membrane potential) |
| $m(t), h(t), n(t)$ | gating variables |

## 3.3 Numerical Solution to the Deterministic Form of H-H

To find a solution to the deterministic form of the H-H system, we apply Euler's method. We recall the H-H system of ODEs from equations (3.5), (3.6), (3.7), (3.8). Euler's method requires an initial condition at time 0. We define the initial conditions for voltage and the gating variables as $V(0) = -65mV$, $m(0) = 0.0529$, $h(0) = 0.5961$, and $n(0) = 0.3177$ [42]. Here $m(t)$, $h(t)$ and $n(t)$ are dimensionless proportions taking values between 0 and 1. With the given constants for the H-H system defined in Section 3.2, $\Delta t = 0.01$, and 3000 time indices, we obtain Euler's approximation to the H-H system in Figure 3.1. We observe continuous curves of the measurable membrane potential and the unobserved gating variables. This is due to our choice of a small step size to improve our Euler's approximation. However, when stimulating a neuron in real-life, our voltage observations are noisy, as shown with the competition dataset in Figure 1.2. A numerical solution to a deterministic system does not account for noise and may be a poor estimate of the true neuron dynamics. To resolve this, we consider the SDE formulation of H-H described in the next section.

## 3.4 SDE Form of H-H

In the literature, we observe different schematic approaches to incorporate stochasticity into the H-H model. Authors of [22] propose that transforming the $\alpha_x$ and $\beta_x$ rates into stochastic rates results in a stochastic model. This idea involves rewriting our conductance functions $g_{Na}$ and $g_K$ with stochastic components. Different simulations and visualizations (such as a bifurcation diagram) validate the stochasticity and robustness of the model [40, 22]. Researchers in [40] and [20] present full systems of SDEs for each ion channel and respective Gaussian noise. Those in [41, 20, 9] modify the H-H neuron

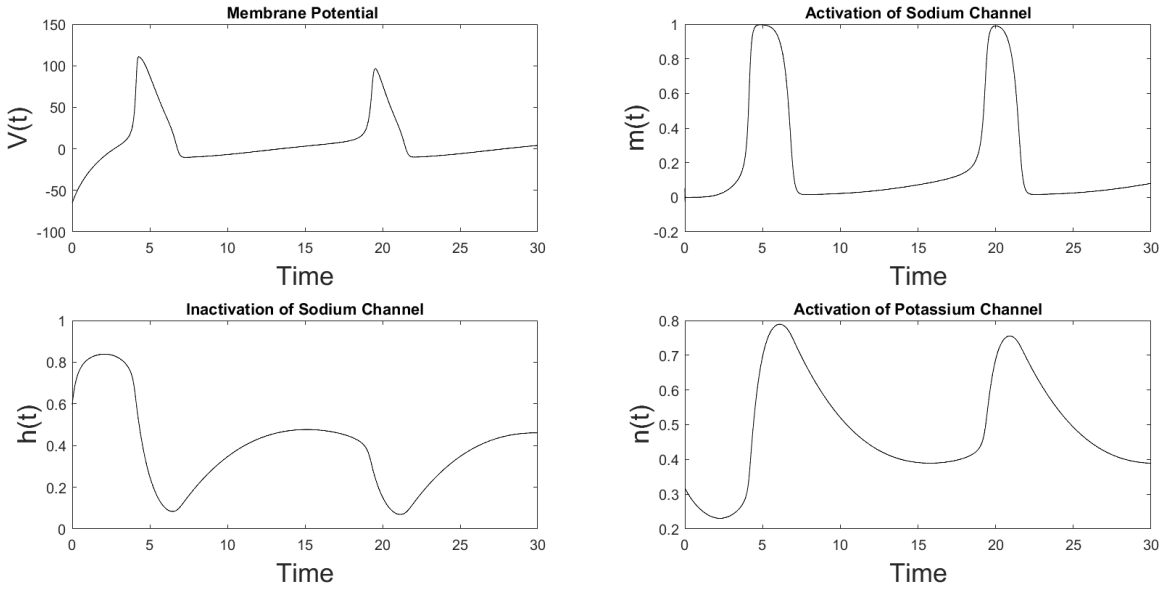


**Figure 3.1:** Euler's method to approximate the dynamics of the deterministic H-H system. Here, $\Delta t = 0.01$, initial conditions for each variable are $V(0) = -65mV, m(0) = 0.0529, h(0) = 0.5961$ and $n(0) = 0.3177$. Note that $m(t)$, $h(t)$ and $n(t)$ are dimensionless proportions.

model via SDEs for each of the activation and inactivation conductances.

The stochastic model for H-H combines the behaviors of the stable and unstable limit cycles for the neuron (firing or quiescent states) that we use for the state estimation problem. We recall the form of an SDE system from (2.10). For the H-H model, $\mathbf{X}(t)$ denotes a stochastic process for the $4 \times 1$ vector of membrane potential $V$, states of the activation and inactivation gates of the sodium ion channel $m$ and $h$, and the activation of the potassium channel $n$ [22]. We define the drift coefficient $f$ using the deterministic ODE form of the H-H model in equations (3.5)–(3.8), and $\mathbf{W}(t)$ is a multivariate Brownian motion for each state in the H-H model as a function of time. We therefore arrive at the following form for our SDE H-H system:

$$dV = [(I - (I_{Na} + I_K + I_L))/C]dt + \sigma_V dW^1(t) \tag{3.9}$$

$$dm = [\alpha_m(V)(1-m) - \beta_m(V)m]dt + \sigma_m dW^2(t) \tag{3.10}$$

$$dh = [\alpha_h(V)(1-h) - \beta_h(V)h]dt + \sigma_h dW^3(t) \tag{3.11}$$

$$dn = [\alpha_n(V)(1-n) - \beta_n(V)n]dt + \sigma_n dW^4(t) \tag{3.12}$$

We denote $\sigma_V, \sigma_m, \sigma_h$ and $\sigma_n$ as the diffusion coefficients on each of the four states of the H-H model.

## 3.5 Numerical Solution to the SDE Form of H-H

To estimate the dynamics of the SDE form of the H-H model, we apply the EM method. For our diffusion coefficient $g$, we define a diagonal matrix of constants to scale the Brownian motion term of our SDE. We define the function $g$ as a constant function for simplicity, since our objective is to see if we can capture the dynamics of the H-H variables by incorporating noise into a SDE framework. We leave other definitions of $g$ for potential future work outside of this project. We choose different

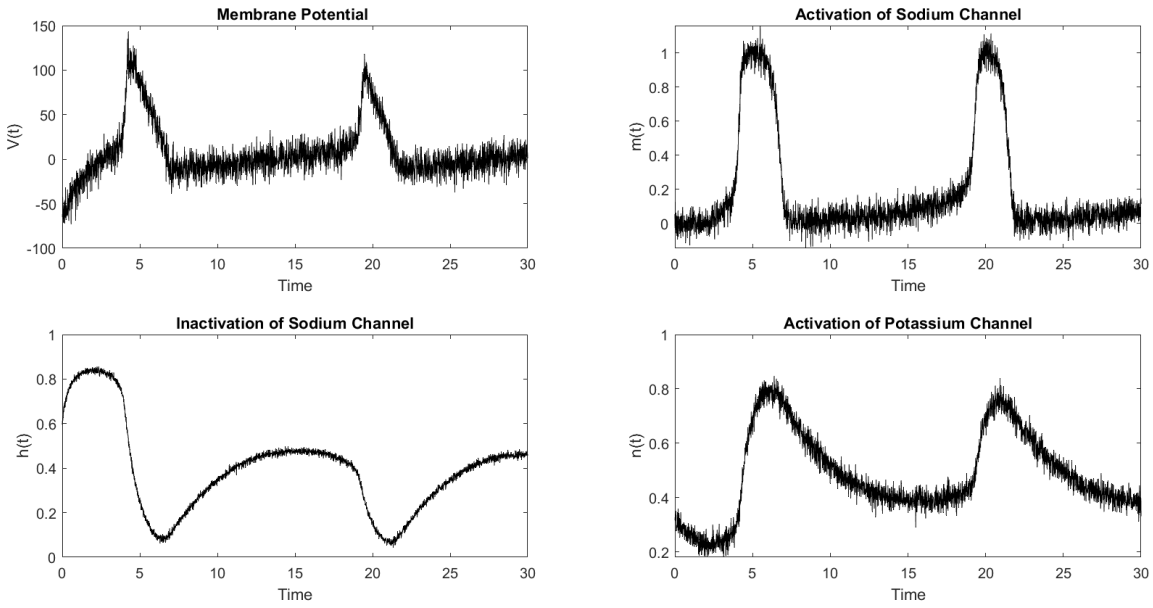


**Figure 3.2:** EM numerical approximation for SDE form of the H-H model. Here, a singular Brownian path is depicted where our diffusion coefficients on each of the H-H variables are $\sigma_V = 10$, $\sigma_m = 0.7$, $\sigma_h = 0.3$, and $\sigma_n = 0.5$. We define $\Delta t = 0.01$ for 3000 time indices.

constants for each of $V$, $m$, $h$, and $n$ dependent on the possible range of values each of our states can take. Figures 3.3, 3.5, 3.7, 3.9, 3.4, 3.6 3.8, and 3.10 use the values $\sigma_V = 10$, $\sigma_m = 0.7$, $\sigma_h = 0.3$, and $\sigma_n = 0.5$.

Figure 3.2 depicts a singular Brownian path. Compared to Euler's approximation to the deterministic form of H-H, we still can capture the overall shape or dynamics of our variables. However, as shown in Figures 3.3, 3.5, 3.7, 3.9, 3.4, 3.6 3.8, and 3.10, when we increase the number of Brownian paths, we see that the average of our numerical solutions to the SDE H-H system approaches the numerical solution to the ODE H-H system. We expect this observation from Brownian motion's Gaussian assumption with mean 0. We choose $n = 10$, 50, 100 and 500 Brownian paths in our simulations to compare and contrast the amount of stochasticity shown in our dynamics. With $n = 10$ paths, our average numerical approximation to the SDE H-H system still demonstrates very stochastic estimates as a function of time. With $n = 500$ paths, in comparison, the average is very close to our numerical solution to the ODE H-H system. As we increase the number of approximations, or Brownian paths, the extent of the stochasticity in our dynamics decreases. By taking an average of multiple approximations to our SDE H-H system, we gain a better understanding of the underlying dynamics of a noisy system under bounds of uncertainty. That is, with random noise introduced, we still capture dynamics of our measured voltage and unmeasured gating variables.

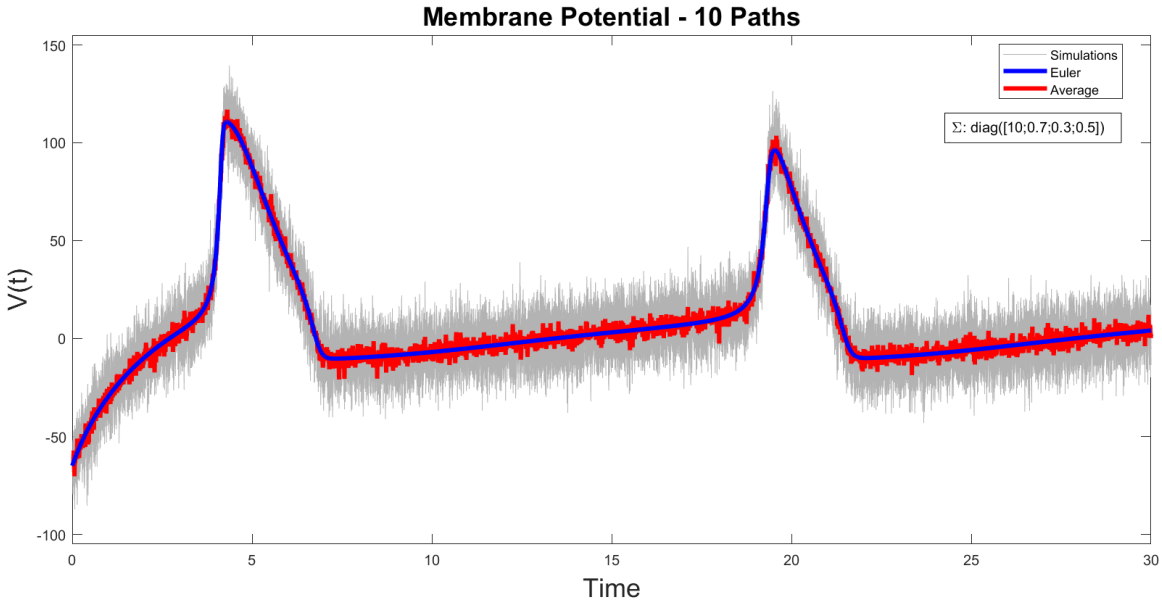


**Figure 3.3:** EM numerical average approximation of the membrane potential for the H-H SDEs with $n = 10$ Brownian paths. We define $\sigma_V = 10$ and $\Delta t = 0.01$ for 3000 time indices.

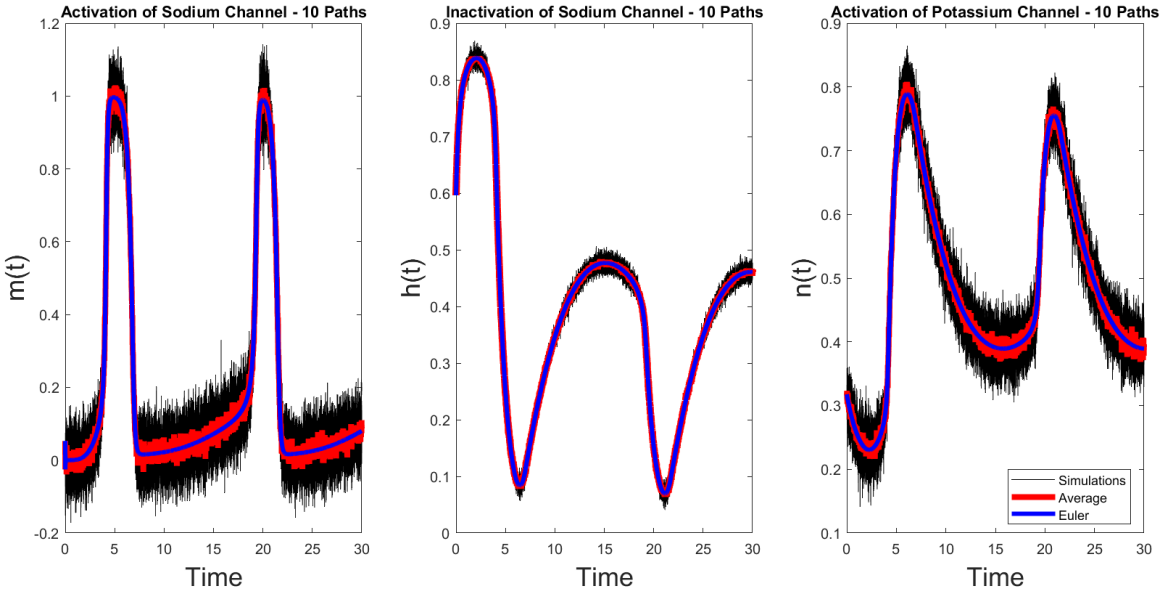


**Figure 3.4:** EM numerical average approximation of the gating variables for the H-H SDEs with $n = 10$ Brownian paths. We define $\sigma_m = 0.7$, $\sigma_h = 0.3$, $\sigma_n = 0.5$ and $\Delta t = 0.01$ for 3000 time indices.

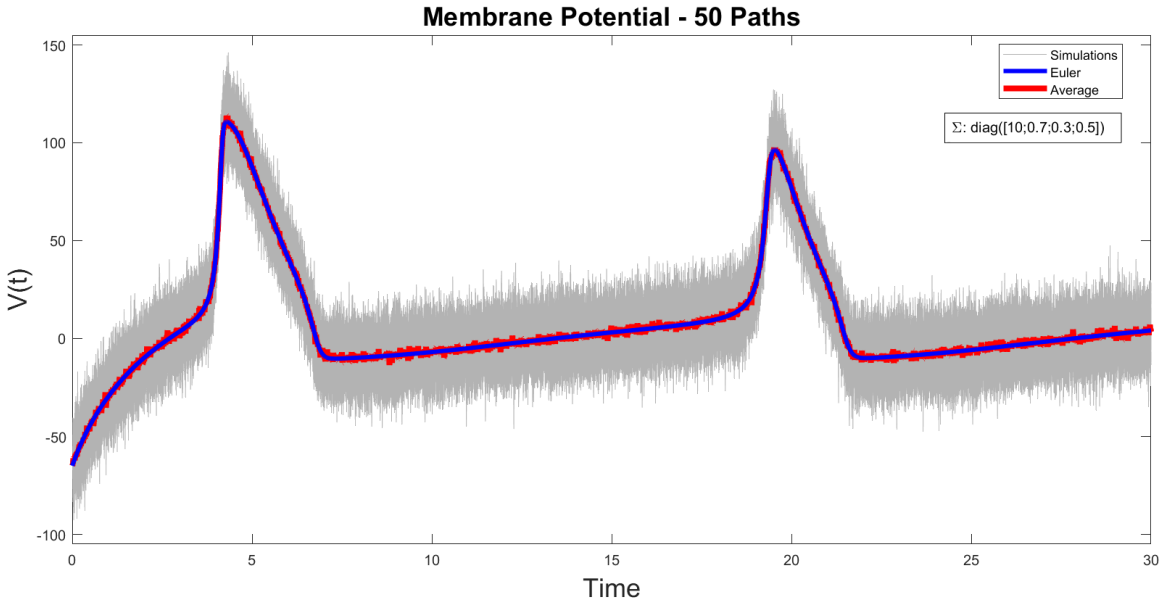


**Figure 3.5:** EM numerical average approximation of the membrane potential for the H-H SDEs with $n = 50$ Brownian paths. We define $\sigma_V = 10$ and $\Delta t = 0.01$ for 3000 time indices.

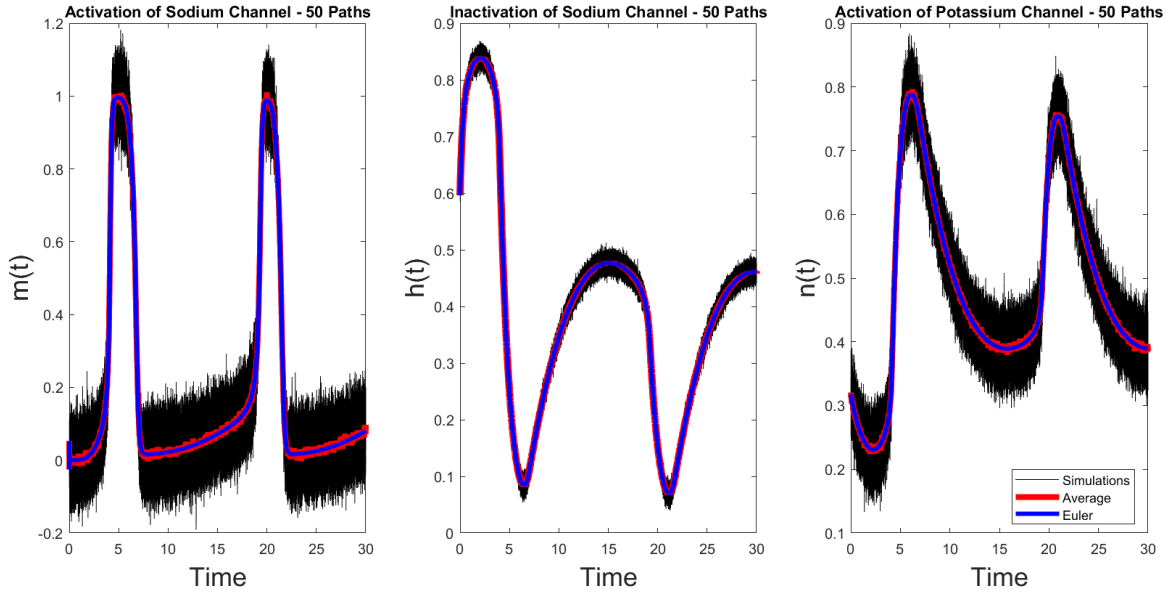


**Figure 3.6:** EM numerical average approximation of the gating variables for the H-H SDEs with $n = 50$ Brownian paths. We define $\sigma_m = 0.7$, $\sigma_h = 0.3$, $\sigma_n = 0.5$ and $\Delta t = 0.01$ for 3000 time indices.

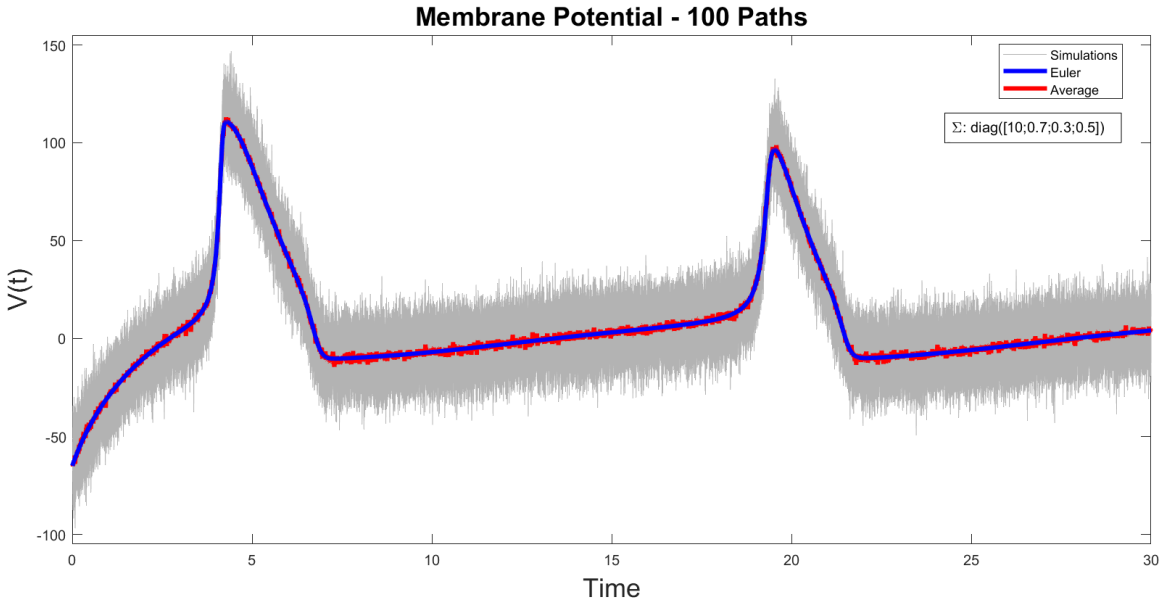


**Figure 3.7:** EM numerical average approximation of the membrane potential for the H-H SDEs with $n = 100$ Brownian paths. We define $\sigma_V = 10$ and $\Delta t = 0.01$ for 3000 time indices.

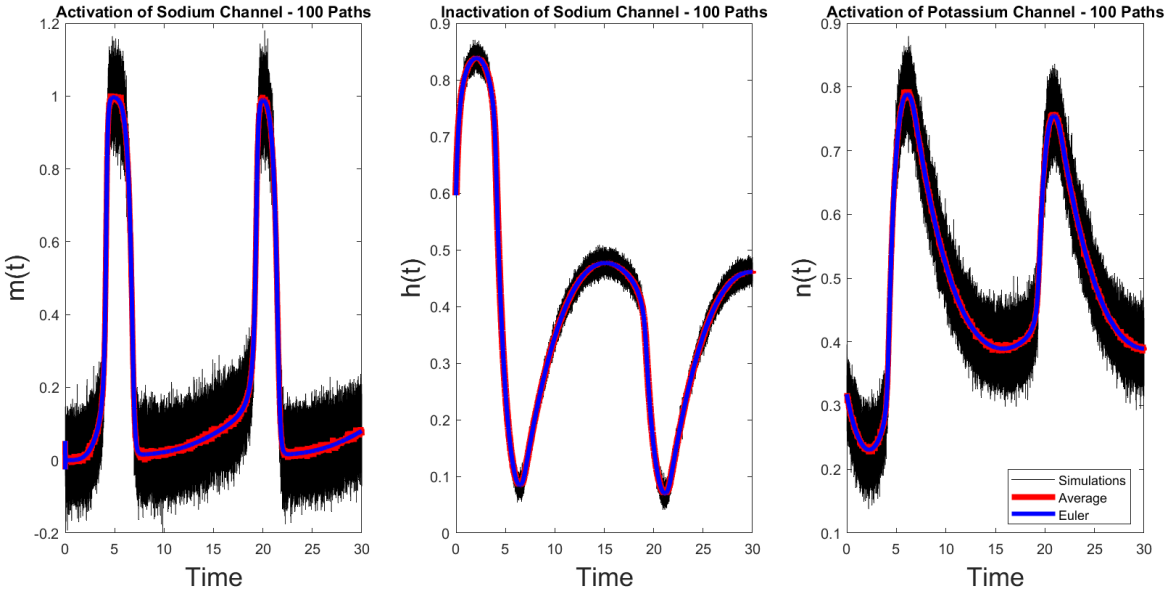


**Figure 3.8:** EM numerical average approximation of the gating variables for the H-H SDEs with $n = 100$ Brownian paths. We define $\sigma_m = 0.7$, $\sigma_h = 0.3$, $\sigma_n = 0.5$ and $\Delta t = 0.01$ for 3000 time indices.

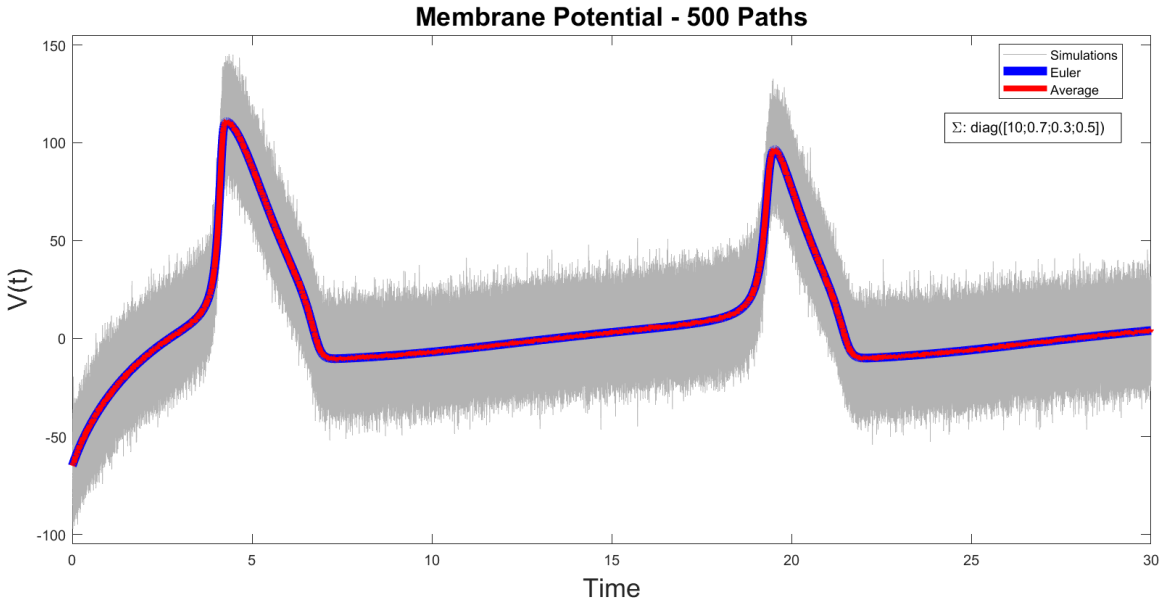


**Figure 3.9:** EM numerical average approximation of the membrane potential for the H-H SDEs with $n = 500$ Brownian paths. We define $\sigma_V = 10$ and $\Delta t = 0.01$ for 3000 time indices.

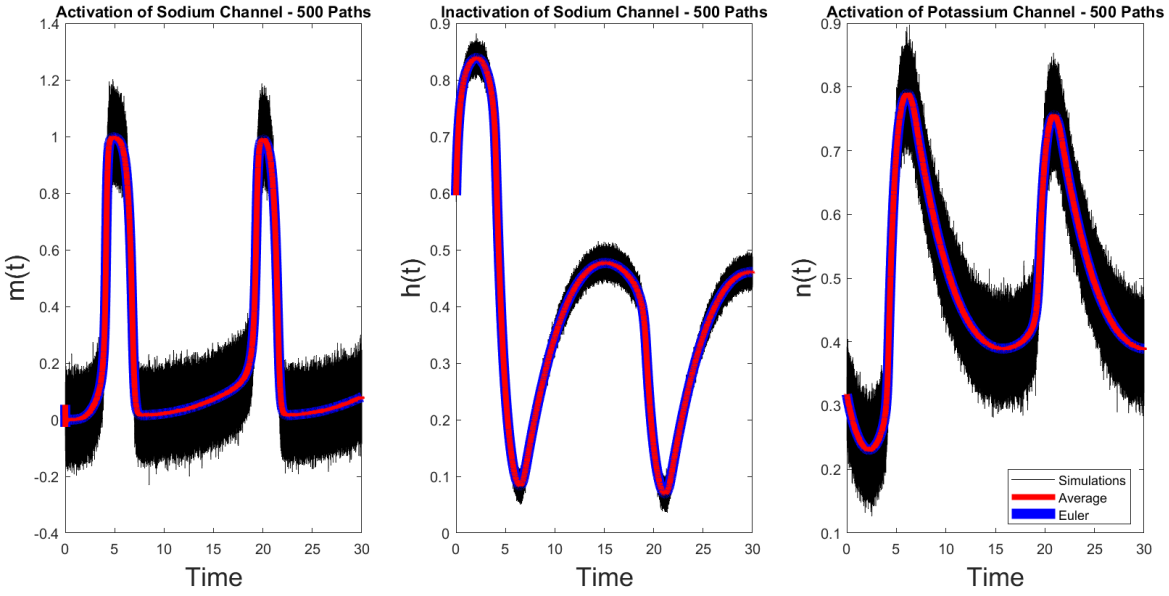


**Figure 3.10:** EM numerical average approximation of the gating variables for the H-H SDEs with $n = 500$ Brownian paths. We define $\sigma_m = 0.7$, $\sigma_h = 0.3$, $\sigma_n = 0.5$ and $\Delta t = 0.01$ for 3000 time indices.

# Chapter 4

# Kalman Filtering Approaches

We use Euler's and the EM numerical methods in this project to capture the dynamics of the H-H neuron model, as described in Chapter 3. But what if we have data? Can we utilize data assimilation to better predict the states in our system that we can't collect data on? The literature uses Kalman filtering techniques for the state estimation inverse problem within some bounds of uncertainty. Specifically, in this work we apply the EnKF. To our knowledge, little research has been published using EnKF to compare the tracking of states between the deterministic and stochastic H-H model. In the following sections, we provide a brief history of Kalman filtering, describing the classic Kalman filter, EnKF, and an application to our working damped mass-spring oscillator example.

## 4.1 History of Kalman Filtering

Kalman filtering methods stem back to early applications in geophysical data assimilation [4]. These geophysical applications, such as numerical weather predictions and oceanography, used numerical methods that combined high-dimensional matrices of systems that were sophisticated yet costly. Decades later, Monte Carlo algorithms were further researched to model these systems [8], leading into the development Kalman filtering approaches. The literature accredits Kalman filtering to one of its developers, Rudolf Kalman–an electrical engineer and mathematician receiving the National Medal of Science for his work on the approach [37]. The first publication of Kalman filter emerged in the early 1960s for its usage by NASA in the Apollo missions [37]. In the early 1990s, linear quadratic estimation algorithms such as Kalman filtering appeared into other applications in the literature [3, 4]. Benefits of using ensemble filtering methods (i.e., Kalman filtering and its modifications) include simplicity, ease, and low cost of implementation for large datasets compared to variational systems [4]. The literature credits Evensen (1994) to introducing the EnKF, which moved from computing Jacobian matrices in linearized model approximations to using ensemble statistics [12, 11]. Usage of ensemble filtering techniques continues to gain popularity in Bayesian estimation and neuron modeling [50, 48].

## 4.2 Classic Kalman Filter

The usage of Kalman filtering and its variations appear in brain imaging research, inertial navigation, time-varying analysis, and other engineering applications [50, 7]. Kalman filtering aims to estimate the current state of a system of equations having only incomplete and/or noisy information. The classic Kalman filter algorithm is recursive, stemming from Markov model (memoryless) properties

and consists of both state evolution and observation equations:

$$X_{j+1} = FX_j + V_{j+1}, \quad j = 0, 1, 2, ... \tag{4.1}$$

$$Y_{j+1} = GX_{j+1} + W_{j+1}, \tag{4.2}$$

where $F$ in (4.1) and $G$ in (4.2) are known matrices, often referred to as propagation models and projection matrices, respectively. $X_{j+1}$ and $Y_{j+1}$ are random variables that denote the model states and observations of the system, and $V_{j+1}$ and $W_{j+1}$ are the noise processes introduced to the system, independent of state vector $X_j$ at given time $j$.

The classic Kalman Filter is a special situation where both the state evolution equation (4.1) and observation equation (4.2) are linear and Gaussian. That is, we can find an explicit matrix form of $F$ that keeps the equation linear and our noise terms $V_{j+1}$ and $W_{j+1}$ both follow a normal distribution and are mutually independent. We define the innovation process $V_{j+1}$ and observation noise $W_{j+1}$ below:

$$V_{j+1} \sim \mathcal{N}(0, C) \tag{4.3}$$

$$W_{j+1} \sim \mathcal{N}(0, D) \tag{4.4}$$

where $C$ and $D$ are covariance matrices we select by choices of the variance coefficient $\sigma^2$ ($C = \sigma_C^2 I$ and $D = \sigma_D^2$, where $I$ is the corresponding identity matrix).

### 4.2.1 Algorithm

From the prediction step of variables $X_{j+1}$ and $Y_{j+1}$ in (4.1) and (4.2), we define the following three steps of the classic Kalman filter algorithm.

#### Prediction Step

We define $\overline{\mathbf{x}}_{j|j}$ as the state prediction mean at a given time index $j$ and the state covariance as $\Gamma_{j|j}$ such that $\mathbf{x}_{j+1} \sim \mathcal{N}(\overline{\mathbf{x}}_{j|j}, \Gamma_{j|j})$. Here $\overline{\mathbf{x}}_{j|j}$ is a vector denoting the mean of $k$ states in our model, $k \geq 1$. We update the prior mean $\overline{\mathbf{x}}_{j+1|j}$ and covariance $\Gamma_{j+1|j}$ through the following equations (with $C$ defined previously):

$$\overline{\mathbf{x}}_{j+1|j} = F\overline{\mathbf{x}}_{j|j} \tag{4.5}$$

$$\Gamma_{j+1|j} = F\Gamma_{j|j}F^T + C \tag{4.6}$$

#### Analysis/Observation Step

We correct our prediction by updating the posterior mean $\overline{\mathbf{x}}_{j+1|j+1}$ and covariance $\Gamma_{j+1|j+1}$ at time $j+1$ upon observing $y_{j+1}$:

$$\overline{\mathbf{x}}_{j+1|j+1} = \overline{\mathbf{x}}_{j+1|j} + K_{j+1}(y_{j+1} - G\overline{\mathbf{x}}_{j+1|j}) \tag{4.7}$$

$$\Gamma_{j+1|j+1} = (I - K_{j+1}G)\Gamma_{j+1|j} \tag{4.8}$$

The Kalman gain $K_{j+1}$ helps to determine what weight is given to the state mean and covariance in the posterior update. We define the Kalman gain as below, where $D$ keeps the same previous definition:

$$K_{j+1} = \Gamma_{j+1|j}G^T(G\Gamma_{j+1|j}G^T + D)^{-1} \tag{4.9}$$

We repeat for a specified number $T$ of iterations such that $j < T$.

### 4.2.2 Example: Mass-Spring System

To demonstrate and verify the effectiveness of the classic Kalman filter, we once again consider the damped spring-mass oscillator. We recall the coupled spring-mass system from equations (2.4)–(2.5). Let $\mathbf{x}(t) = \begin{bmatrix} p(t) \\ v(t) \end{bmatrix} \in \mathbb{R}^2$, where $p(t)$ is the position of the spring and $v(t)$ is velocity with respect to time $t$. Writing the equation for $d\mathbf{x}/dt$ in matrix form leads to the following matrix on the right-hand side: $\begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}$ which we will denote as $A$. We can then write the ODE system as

$$\frac{d\mathbf{x}}{dt} = A\mathbf{x}. \tag{4.10}$$

To apply the classic Kalman filter, we apply an Euler's method discretization to (4.10) so that we can write $F$ in our Kalman filter equations as follows:

$$F = I - \Delta t A \tag{4.11}$$

where $I$ is a $2 \times 2$ identity matrix (since $A$ is $2 \times 2$ and we have two variables $p$ and $v$ contained in $\mathbf{x}$). We now find our state evolution and observation equations:

$$X_{j+1} = FX_j + V_{j+1} = (I - \Delta t A)X_j + V_{j+1} \tag{4.12}$$

$$Y_{j+1} = GX_{j+1} + W_{j+1} = [1 \quad 0]X_{j+1} + W_{j+1} \tag{4.13}$$

where $G$ is a projection matrix for the measurable variable $p$.

Following the steps of the algorithm, we produce graphs for $p(t)$ and $v(t)$ as shown in Figures 4.1, 4.2, 4.3, and 4.4. These figures vary based on how we select matrices $C$ and $D$. For very small process and measurement noise, our estimates are very close to the estimates of that shown by the deterministic damped mass-spring oscillator. We expect this since usage of little noise in the prediction step to compute the prior mean does not change substantially in the update step. We see, however, when we introduce more noise–especially in the definition of $D$ in our update step, the posterior mean estimates of the position and velocity are more noisy in comparison to the "true" solution obtained using MATLAB's ode15s.

## 4.3 Ensemble Kalman Filter

The classic Kalman filter requires the assumption that our system is linear. What if our system is nonlinear, as is the case with the H-H system of equations? The literature uses different modifications of the Kalman filter depending on the application. These include the EKF [33, 48], the unscented Kalman filter [33], the EnKF [12, 51, 7, 37, 48], and its augmentation [37, 7]. The EKF and the EnKF are the two nonlinear extensions most commonly used in the literature. EnKF, however, has some computational advantages over EKF. For example, EnKF, as its name suggests, computes the posterior mean and covariance matrix using ensemble statistics. This neglects the need to compute the Jacobian matrix required for forward propagation in EKF. In terms of stochasticity, an analytic solution of the Jacobian matrix for a specific system may be complex and not explicit to derive. Due to its ease of implementation and interpretability, we use EnKF in this project. More notably, at the time of this project, little research has been conducted on EnKF for state estimation of neuron models using SDEs.

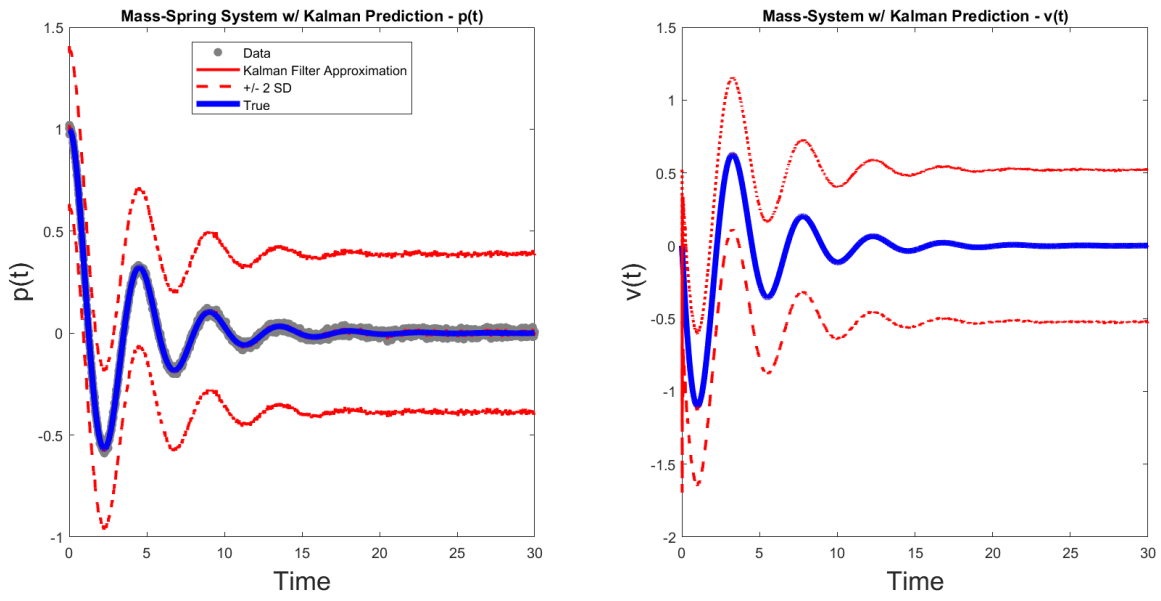


**Figure 4.1:** Classic Kalman filter prediction for damped mass-spring oscillator with $\sigma_C = \sigma_D = 0.01$.

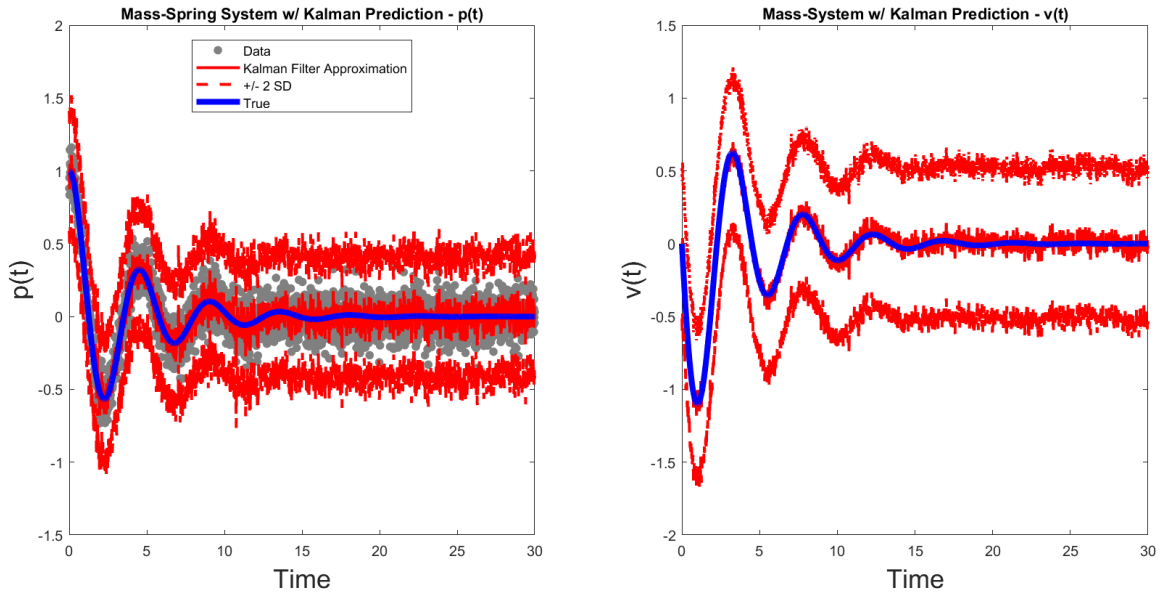


**Figure 4.2:** Classic Kalman filter prediction for damped mass-spring oscillator with $\sigma_C = \sigma_D = 0.1$

### 4.3.1 Algorithm

EnKF solves nonlinear systems through combining ensemble statistics into the classic Kalman filter framework. The EnKF approach:

- Takes an ensemble of prediction state estimates from the prior distribution,
- Computes the prediction ensemble mean and prior covariance matrix, and
- Updates the (posterior) state mean and covariance upon observing data.

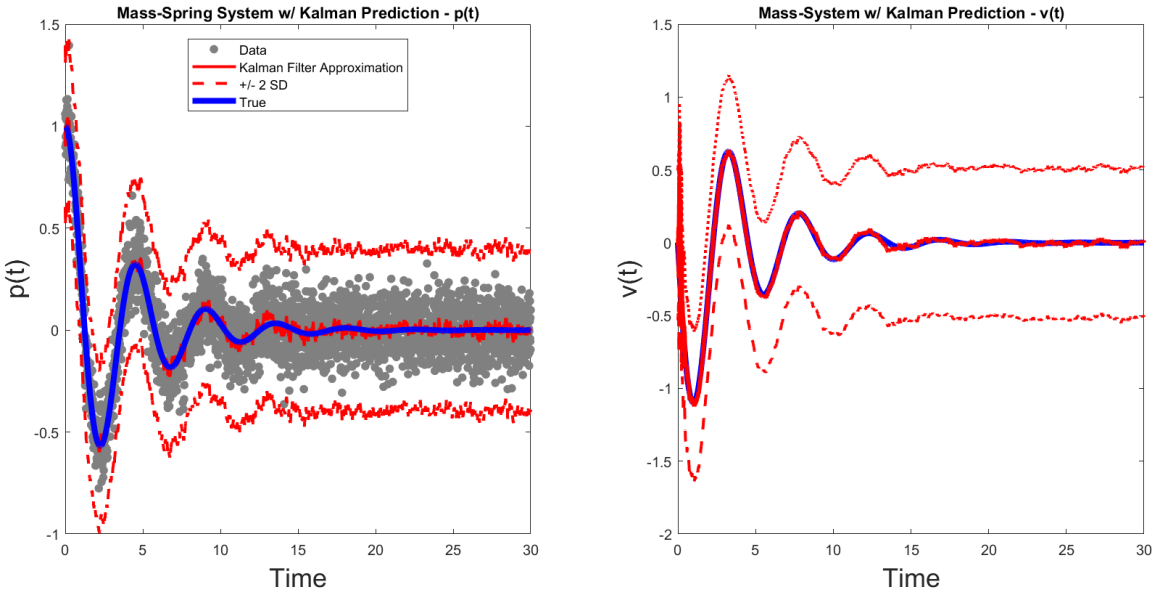


**Figure 4.3:** Classic Kalman filter prediction for damped mass-spring oscillator with $\sigma_C = 0.01$ and $\sigma_D = 0.1$

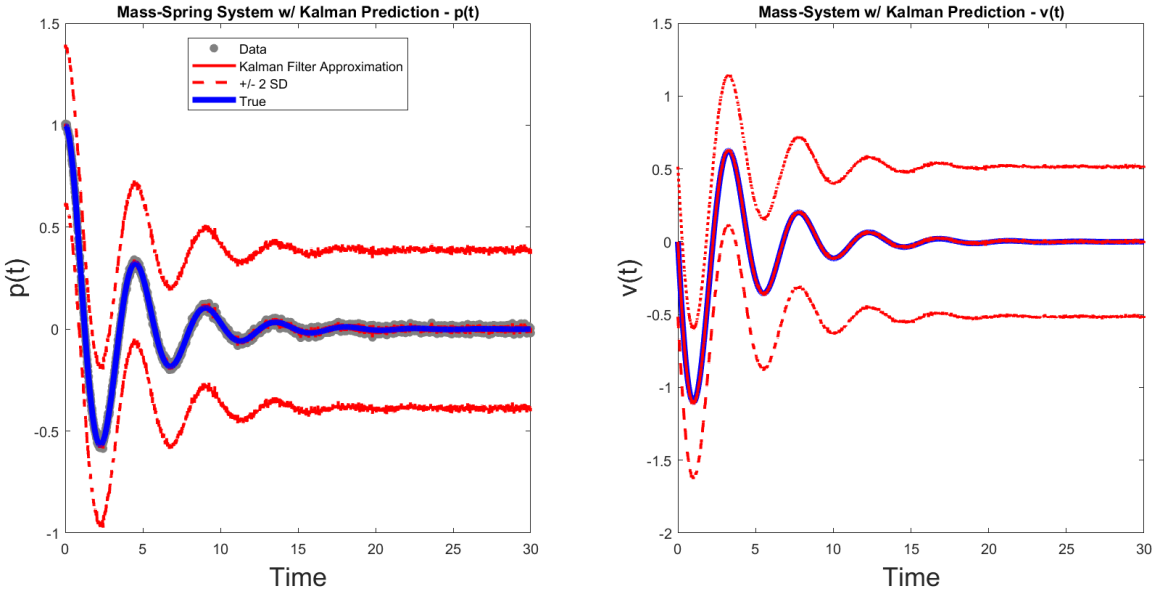


**Figure 4.4:** Classic Kalman filter prediction for damped mass-spring oscillator with $\sigma_C = 0.1$ and $\sigma_D = 0.01$.

One key advantage of the classic Kalman filter lies in its reliance on the system being both linear and Gaussian, leveraging the property that Gaussian distributions maintain their form even under linear transformations. EnKF accommodates the usage of nonlinear systems under a Bayesian framework. Specifically, EnKF uses an ensemble of discrete realizations under the prior distribution to calculate the ensemble mean and covariance with updating equations.

**Prediction Step**

From the current ensemble

$$\{\mathbf{x}_{j|j}^1, \mathbf{x}_{j|j}^2, \ldots, \mathbf{x}_{j|j}^N\}, \tag{4.14}$$

the we define the state prediction ensemble by the following equation:

$$\mathbf{x}_{j+1|j}^n = F(\mathbf{x}_{j|j}^n) + \mathbf{v}_{j+1}^n, \qquad n = 1, 2, ..., N \tag{4.15}$$

where we draw $\mathbf{v}_{j+1}^n \sim \mathcal{N}(0, C)$. We then calculate the prediction ensemble mean and prior covariance matrix:

$$\overline{\mathbf{x}}_{j+1|j} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_{j+1|j}^n \tag{4.16}$$

$$\Gamma_{j+1|j} = \frac{1}{N-1} \sum_{n=1}^{N} \left(\mathbf{x}_{j+1|j}^n - \overline{\mathbf{x}}_{j+1|j}\right) \left(\mathbf{x}_{j+1|j}^n - \overline{\mathbf{x}}_{j+1|j}\right)^T \tag{4.17}$$

**Analysis/Observation Step**

We generate the (simulated) observation ensemble by:

$$y_{j+1}^n = y_{j+1} + w_{j+1}^n, \qquad n = 1, 2, ...N$$

where we independently draw each $w_{j+1}^n \sim \mathcal{N}(0, D)$. For ease of calculations, we define our observation ensemble to be the same size as our state prediction ensemble. We then find the posterior state ensemble:

$$\mathbf{x}_{j+1|j+1}^n = \mathbf{x}_{j+1|j}^n + K_{j+1}(y_{j+1}^n - G\mathbf{x}_{j+1|j}^n), \qquad n = 1, 2, ..., N \tag{4.18}$$

We calculate the Kalman gain matrix $K_{j+1}$ the same as before:

$$K_{j+1|j} = G^T(G\Gamma_{j+1|j}G^T + D)^{-1} \tag{4.19}$$

where we assume a linear observation model. We fine the posterior state mean and covariance from posterior ensemble statistics.

### 4.3.2 Example: Mass-Spring System

Despite our damped mass-spring oscillator system being linear, we still demonstrate the effectiveness of EnKF in estimating the position and velocity states. We recall the system with its predefined constants from Section 2.1.2. Figures 4.5, 4.6, 4.7, and 4.8 show the dynamics and state estimation of the position and velocity for ensemble sizes $N = 10, 50, 100$ and 500. Here we show the situation where our innovation process and observation noise have covariance matrices defined with $\sigma_C$ and $\sigma_D = 0.1$. For smaller ensemble sizes, as shown in Figure 4.5 which takes a prior prediction ensemble of 10 members, we see that our unmeasured state (velocity) has very noisy posterior mean estimates. Between two standard deviations, our uncertainty does not maintain the dynamics expected from that of our true solution. However, as we increase our ensemble size, we get mean estimates of our velocity closer to that of our true solution with some uncertainty. Overall, our measured variable (position) has estimates very close to our true solution. Within our bounds of uncertainty, we observe high confidence (95%) that we capture our position state estimates well upon observing noisy data. From this deterministic setting, we note that how large we define our ensemble size and the scale of which we define our noise are critical to getting appropriate estimates. This becomes especially important when we consider a stochastic framework that is inherently noisy.

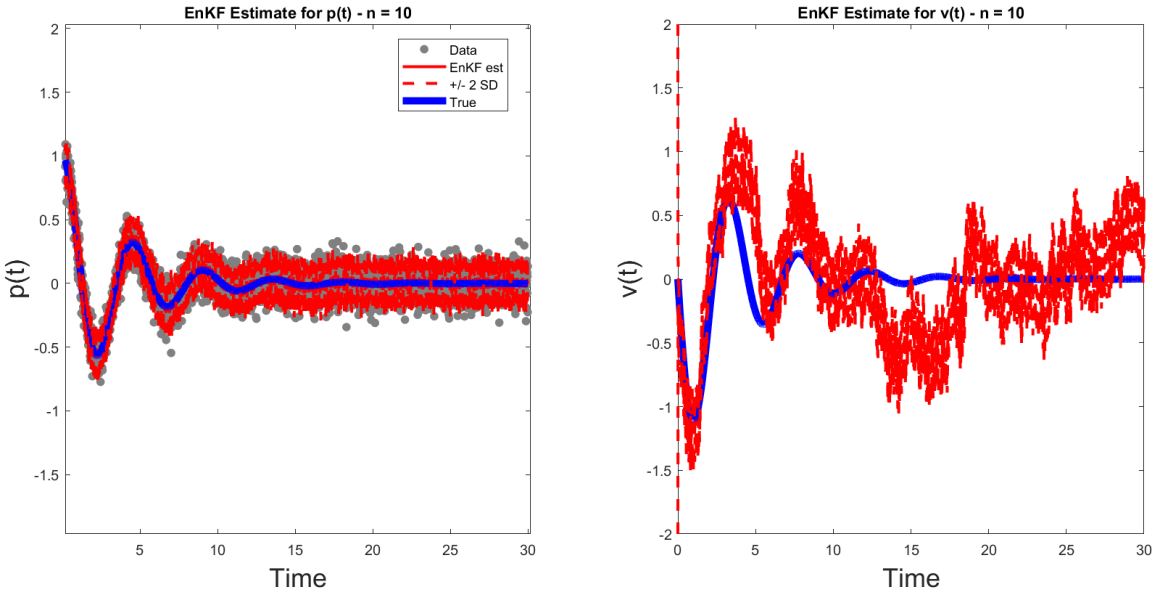


**Figure 4.5:** EnKF estimate for damped mass-spring oscillator with $\sigma_C = \sigma_D = 0.1$, $N = 10$. Our scale on Brownian motion for both $p$ and $v$ is $\sigma_p = \sigma_v = 1$.

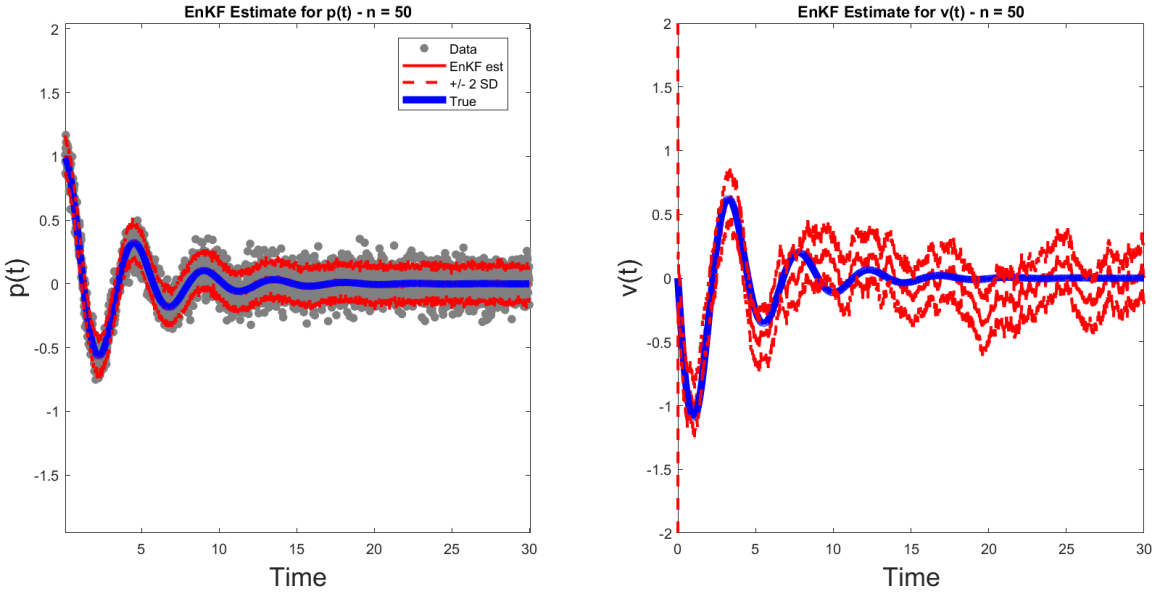


**Figure 4.6:** EnKF estimate for damped mass-spring oscillator with $\sigma_C = \sigma_D = 0.1$, $N = 50$. Our scale on Brownian motion for both $p$ and $v$ is $\sigma_p = \sigma_v = 1$.

## 4.4 EnKF in SDEs

In the stochastic framework, we can extend EnKF to estimate the states of the stochastic process vector $\mathbf{X}(t)$. Since a system of SDEs already has noise incorporated through Brownian motion, we modify the forward prediction step of EnKF to generate the ensemble:

$$\{\mathbf{X}_{j|j}^1, \mathbf{X}_{j|j}^2, ..., \mathbf{X}_{j|j}^N\} \tag{4.20}$$

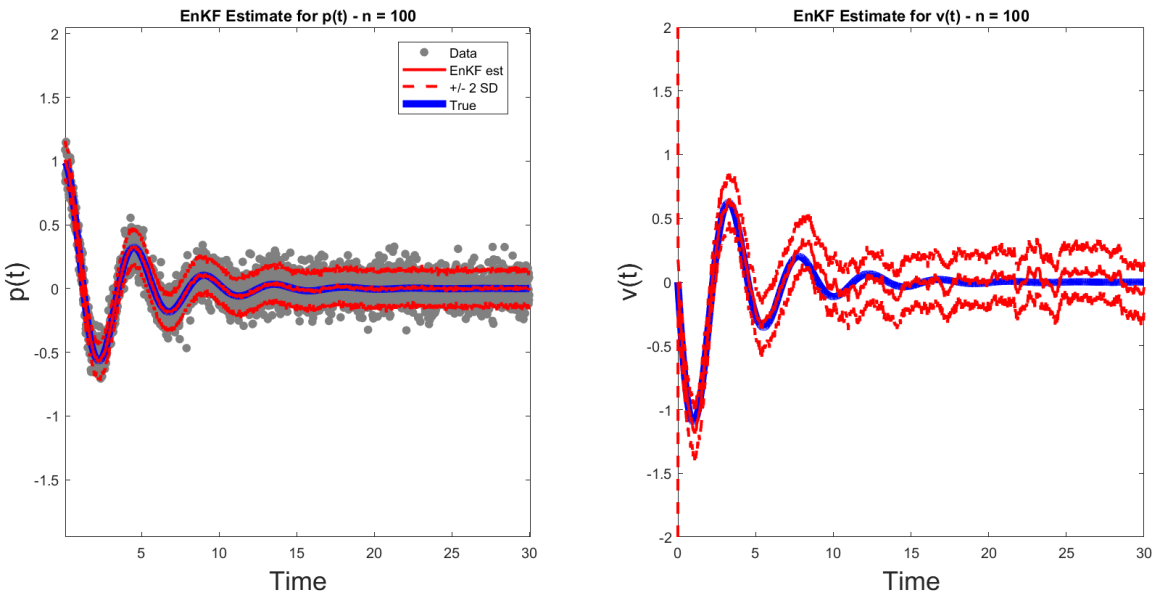


**Figure 4.7:** EnKF estimate for damped mass-spring oscillator with $\sigma_C = \sigma_D = 0.1$, $N = 100$. Our scale on Brownian motion for both $p$ and $v$ is $\sigma_p = \sigma_v = 1$.

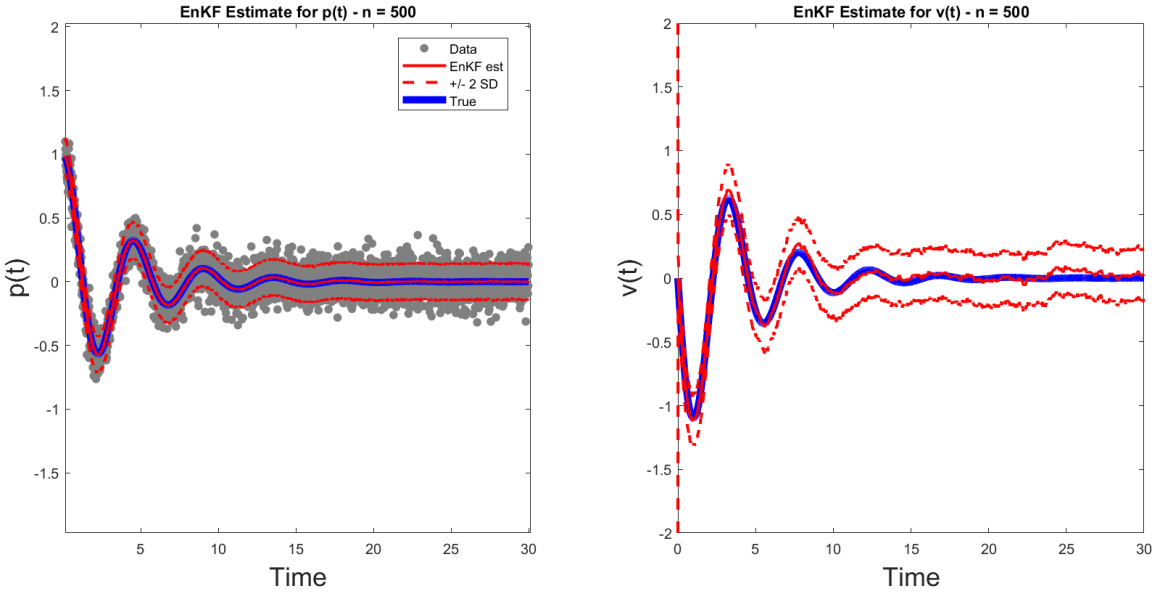


**Figure 4.8:** EnKF estimate for damped mass-spring oscillator with $\sigma_C = \sigma_D = 0.1$, $N = 500$. Our scale on Brownian motion for both $p$ and $v$ is $\sigma_p = \sigma_v = 1$.

We then can generate our state prediction ensemble:

$$\mathbf{X}^n_{j+1|j} = F(\mathbf{X}^n_{j|j}, \mathbf{V}^n_{j+1}), \qquad n = 1, 2, ..., N \tag{4.21}$$

where $F$ follows from the numerical solution to the SDE system–in this project, the EM approximation for $\mathbf{X}(t)$. Since an SDE framework already incorporates noise, we move $\mathbf{V_{j+1}}$ inside our function $F$. The rest of the EnKF algorithm in Section 4.3.1 remains unchanged.

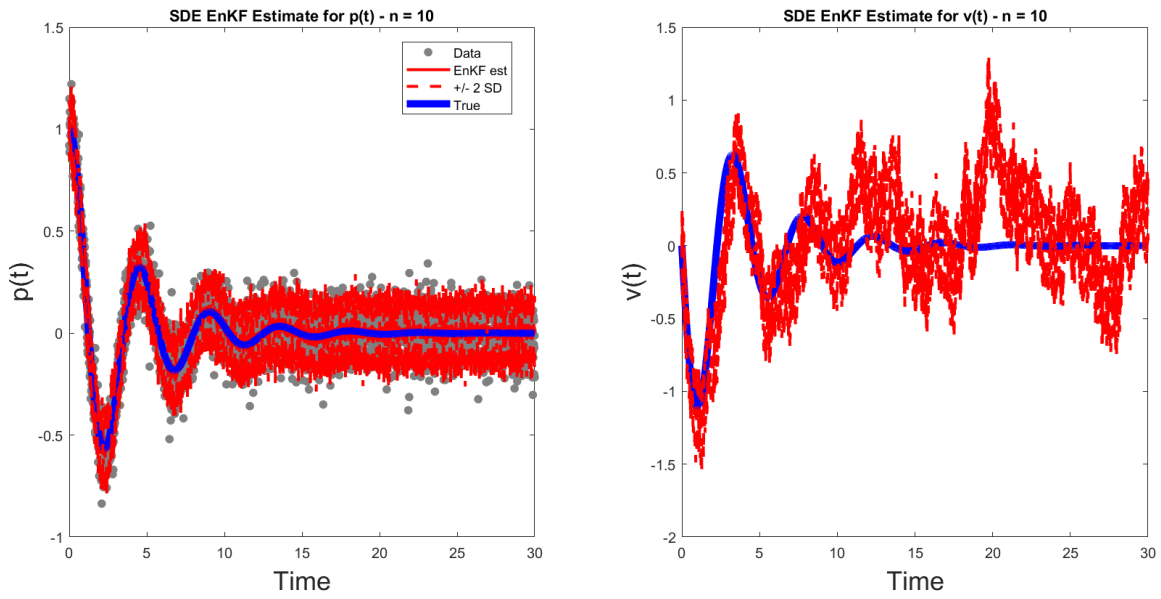


**Figure 4.9:** EnKF estimate for SDE damped mass-spring oscillator with $\sigma_C = \sigma_D = 0.1$, $N = 10$. Our scale on Brownian motion for both $p$ and $v$ is $\sigma_p = \sigma_v = 1$.

### 4.4.1 Example: Mass-Spring System

We recall the stochastic damped mass-spring oscillator in equation (2.12)–(2.13). We define the mass-spring model parameters as $m = 8$, $d = 4$, and $k = 16$ to match our example from Chapter 2. Given initial conditions $p(t) = 1$ and $v(t) = 0$, Figures 4.9, 4.10, 4.11, and 4.12 show the dynamics and state estimation for the mass-spring system of ensemble member sizes $N = 10, 50, 100$ and 500 where $\sigma_C = \sigma_D = 0.1$. We draw similar conclusions to that of our deterministic case. For increasing ensemble sizes, we obtain better estimates. Specifically, the state estimation of velocity is best when we have an ensemble size of $N = 500$. Our state estimation of position closely follows the true solution, meaning upon observing data our posterior state ensemble does not change much. We expect this since our innovation process and observation noise follow Gaussian distributions centered at 0. When we take the average of a larger ensemble, we obtain a solution closer to that of the "true" solution found using ode15s.

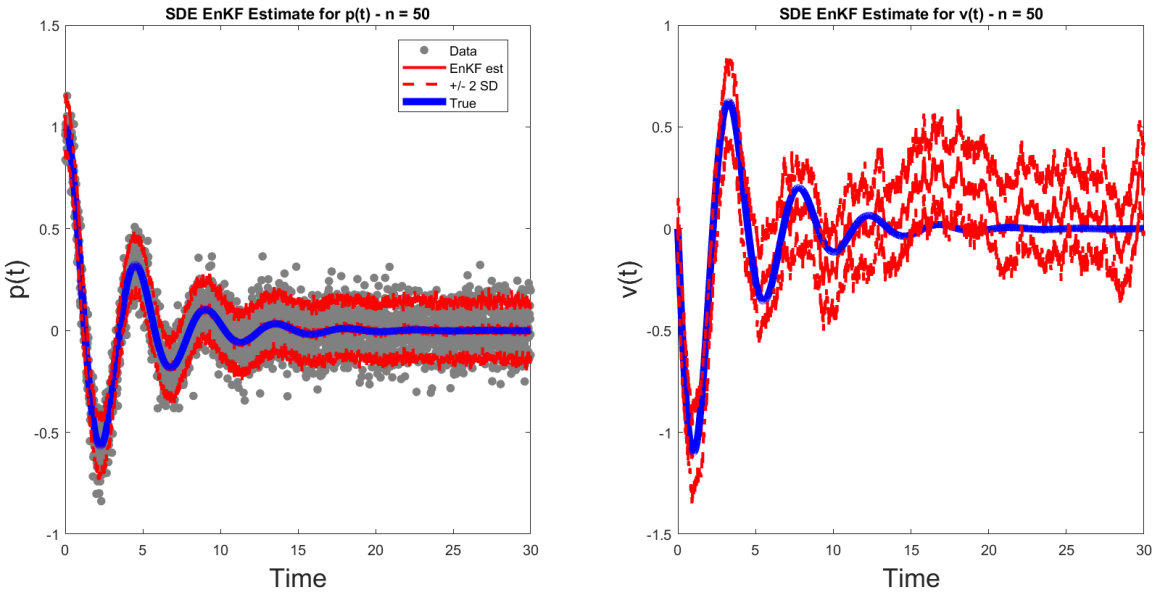


**Figure 4.10:** EnKF estimate for SDE damped mass-spring oscillator with $\sigma_C = \sigma_D = 0.1$, $N = 50$. Our scale on Brownian motion for both $p$ and $v$ is $\sigma_p = \sigma_v = 1$.

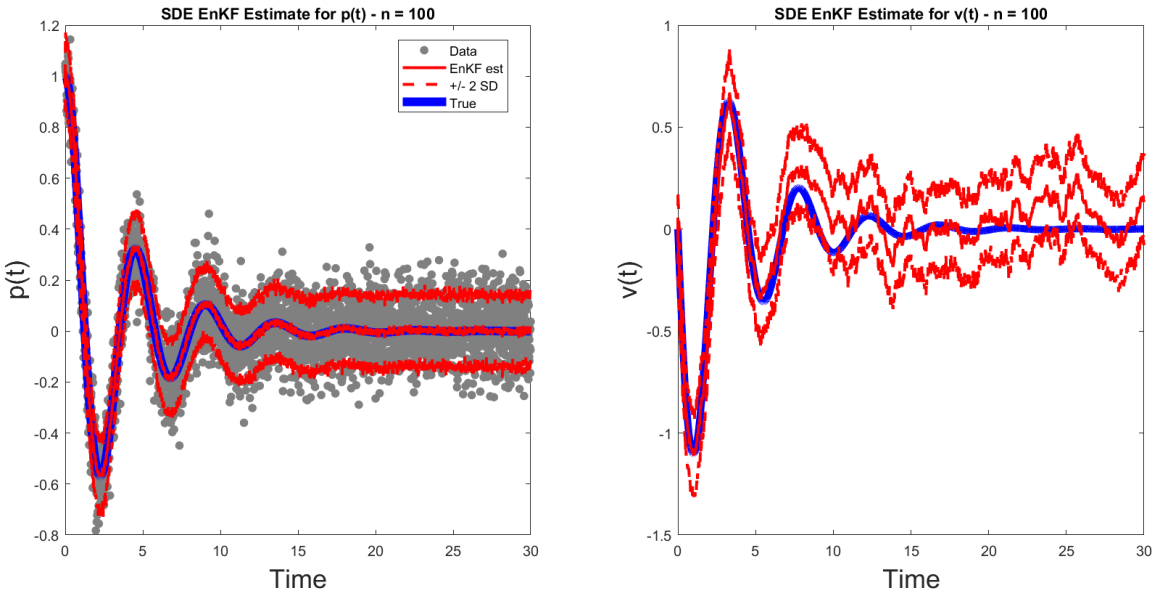


**Figure 4.11:** EnKF estimate for SDE damped mass-spring oscillator with $\sigma_C = \sigma_D = 0.1$, $N = 100$. Our scale on Brownian motion for both $p$ and $v$ is $\sigma_p = \sigma_v = 1$.

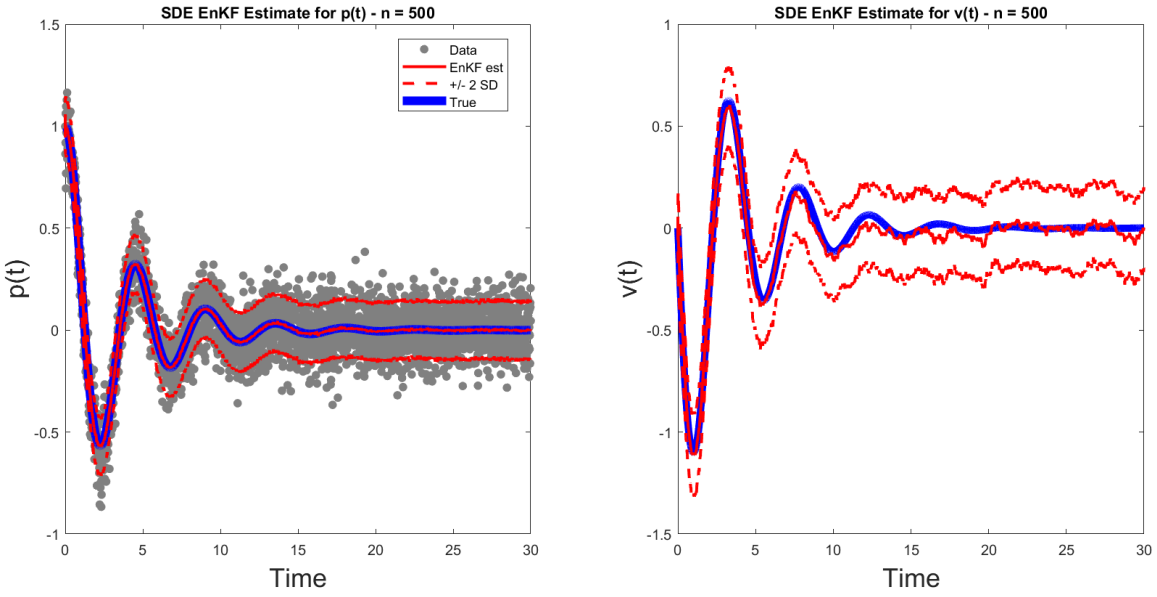


**Figure 4.12:** EnKF estimate for SDE damped mass-spring oscillator with $\sigma_C = \sigma_D = 0.1$, $N = 500$. Our scale on Brownian motion for both $p$ and $v$ is $\sigma_p = \sigma_v = 1$.

# Chapter 5

# State Estimation of the Hodgkin-Huxley System

Kalman filtering techniques allow us to perform state estimation of our dynamic system. State estimation is an inverse problem. An inverse problem refers to a type of mathematical problem where the goal is to determine the states or parameters of a system (input) based on observed data (output). Inverse problems work backwards from observed results to infer properties or estimates of our inputs. In this project, we generate estimates of our states through numerical methods as described in Chapter 2, and update our estimates through data generation using a built-in MATLAB ODE solver. This chapter applies the Kalman filtering approaches defined in Chapter 4 to perform state estimation of our deterministic and stochastic forms of the H-H system.

## 5.1 EnKF for Deterministic H-H Model

As defined in Chapter 4, the EnKF takes an ensemble of prediction estimates from the prior distribution, computes the prediction ensemble mean and covariance matrix, and updates these computations upon observing data. From the system of four H-H ODEs, we generate the prediction state ensemble through Euler's method. We define our step size $\Delta t = 0.01$ to ensure proper integration of our deterministic dynamical system. Analogously to our damped mass-spring oscillator example, we present estimates for ensemble sizes of $N = 10, 50, 100$ and 500 members.

In this chapter, we use the same values of the covariance matrices in the distributions for each noise process respectively for our numerical experiments. Appendix A includes figures where we modify the definitions of $C$ and $D$ to validate and verify our state estimates. We define our process matrix $C$ as the diagonal matrix with the value $(10)^2$ in the first row and column index, $(0.1)^2$ for the remaining indices down the main diagonal, and zeros elsewhere. We set $D = 10^2$. We choose these constants due to the sample space our H-H variables span by model constraints. Proper selection of these values is critical and a limitation to the conclusions drawn. Here $G$ in our EnKF is a projection matrix size $1 \times 4$. This is due to the fact that our observation ensemble consists of simulated observations of voltage. Figures 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, and 5.8 show the state estimates for the membrane potential and gating variables of the deterministic H-H system.

Observing Figures 5.1 and 5.2, we see that we do not track the unmeasured variables $h$ and $n$ well, potentially impart to our selection of $C$. The activation of the sodium channel $m$ seems to have decent estimates within our bounds of uncertainty. We also track our measured variable, voltage, for $N = 10$ prediction ensemble members well such that most of our observed data fall within two standard

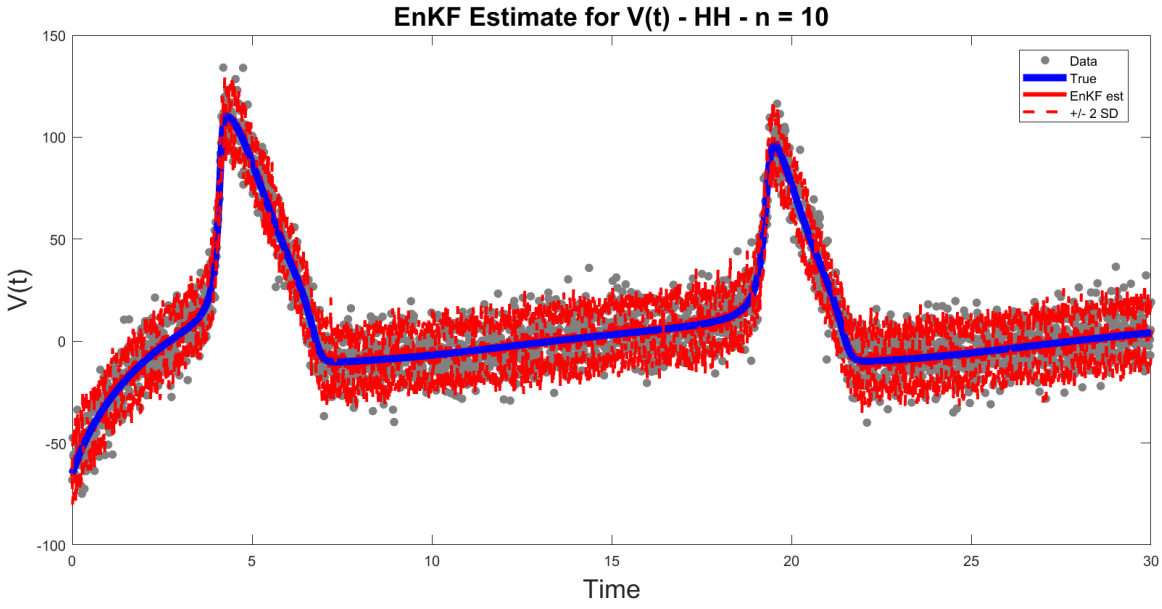


**Figure 5.1:** EnKF estimates for membrane potential with an ensemble size of $N = 10$.

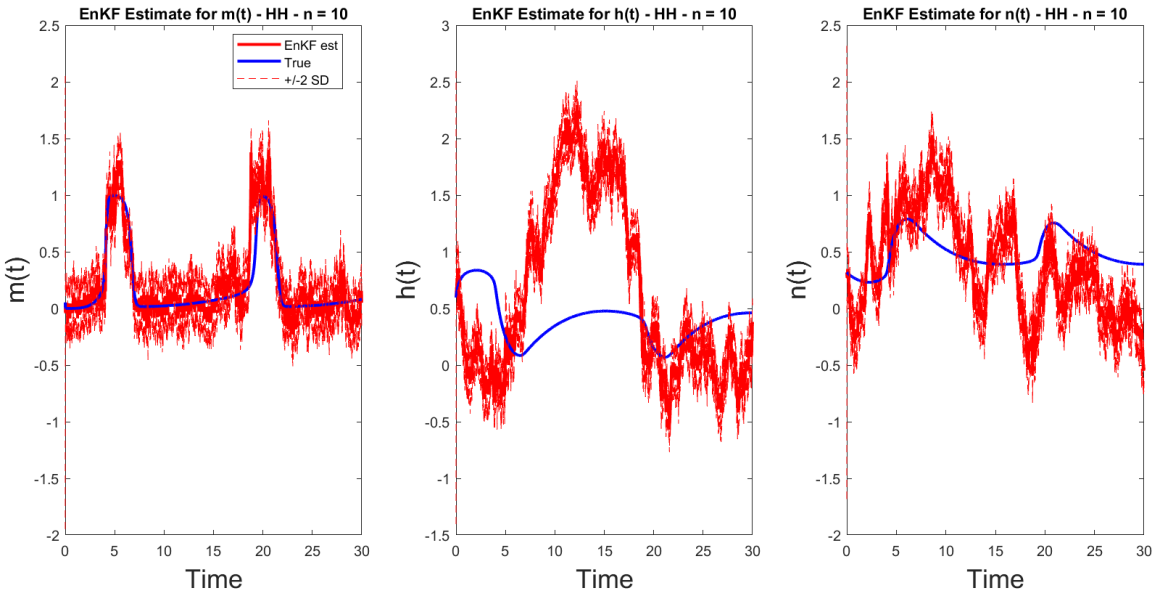


**Figure 5.2:** EnKF estimates for gating variables with an ensemble size of $N = 10$.

deviations of our posterior mean estimates at a given time. However, despite tracking voltage closely to that of our true solution, the estimates still display stochastic behavior and may be the underlying cause as to losing information of our unmeasured variables $h$ and $n$. As we increase our ensemble size, specifically to $N = 100$ or 500 members, we achieve better estimates of our unmeasured variables. With an ensemble size of $N = 500$, we gain estimates that still display stochasticity, but approach much closer to that of the true solution. This result follows from what we investigate with the damped mass-spring oscillator. For large ensemble sizes, our prior prediction ensemble mean is closer to that of our numerical solution to the deterministic system by the construction of our innovation noise.

By increasing the ensemble size, keeping all other parameters constant, we capture better estimates

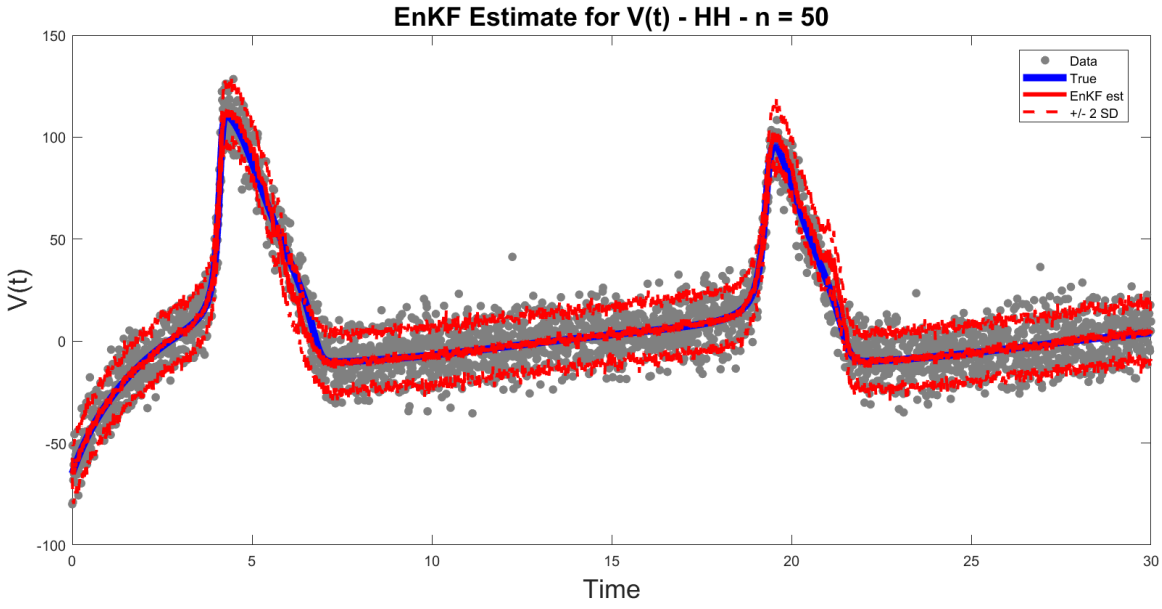


**Figure 5.3:** EnKF estimates for membrane potential with an ensemble size of $N = 50$.

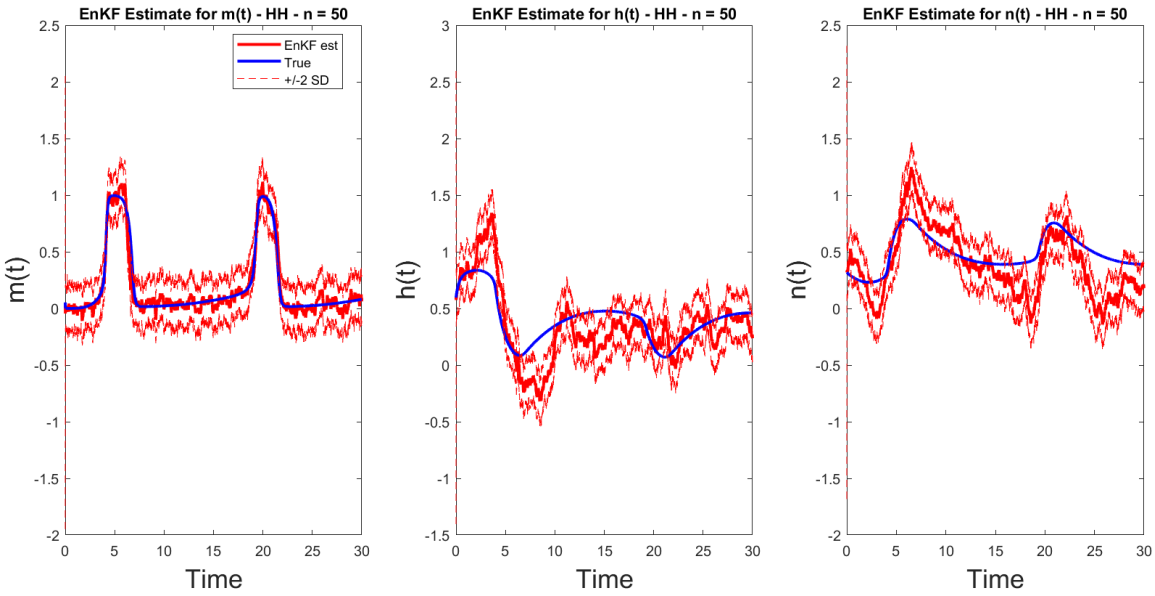


**Figure 5.4:** EnKF estimates for gating variables with an ensemble size of $N = 50$.

and information about our unmeasured variables. However, we find these estimates by having an observation at every time step. In the next section, we estimate states of our stochastic model of H-H and explore how these estimates change if we have less data.

## 5.2 EnKF for SDE H-H Model

We now extend the state estimation problem to the stochastic framework of the H-H model. We use the same initial conditions, time step ($\Delta t$) and constants as in Chapter 3. The diffusion coefficients on

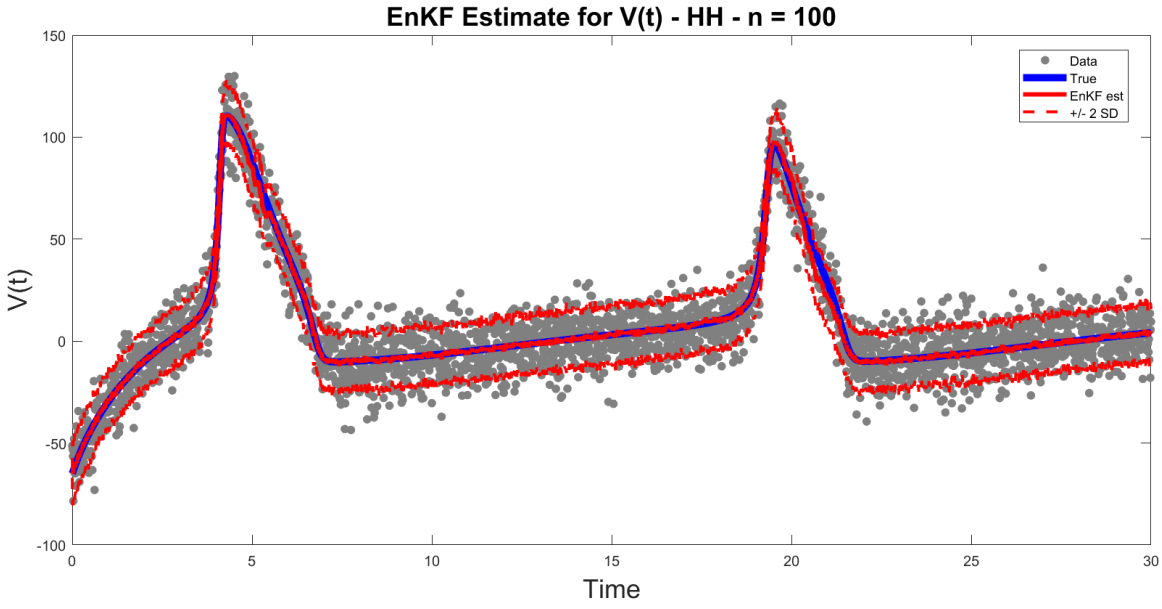


**Figure 5.5:** EnKF estimates for membrane potential with an ensemble size of $N = 100$.

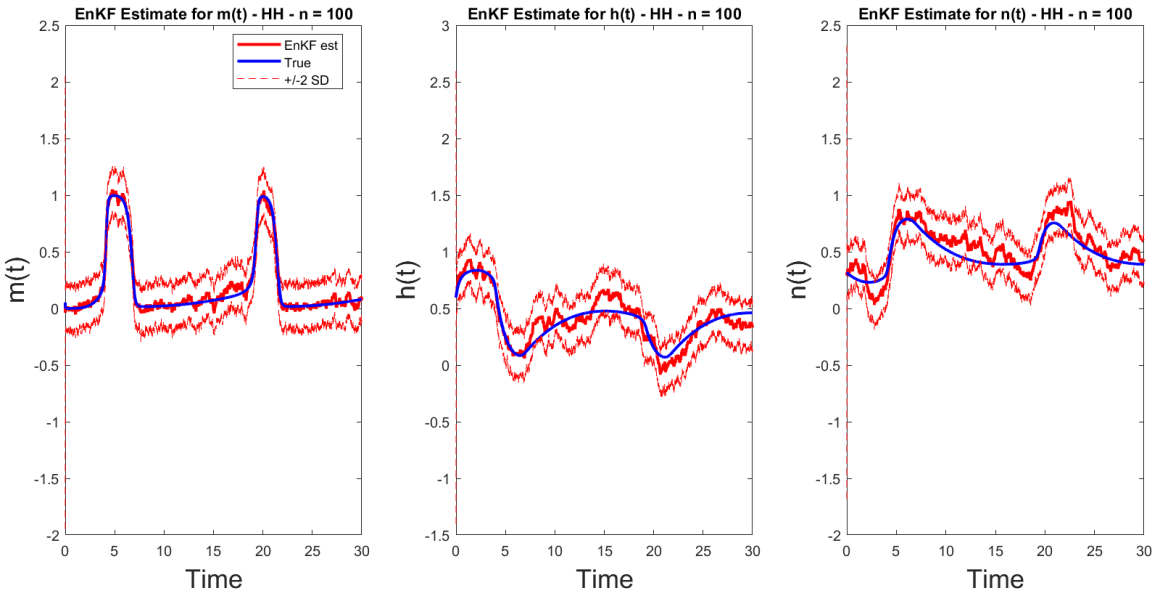


**Figure 5.6:** EnKF estimates for gating variables with an ensemble size of $N = 100$.

our state variables are have the same definitions as that in Chapter 3 ($\sigma_V = 10, \sigma_m = 0.7, \sigma_h = 0.3$ and $\sigma_n = 0.5$). The difference between the state estimation problem in the previous section to the one here via EnKF techniques is how we generate our prediction state ensemble. In the deterministic setting, we use Euler's method to create our prediction ensemble. In our stochastic framework, we use the EM method to generate our prediction ensemble. Likewise to the deterministic state estimation problem, we define the variance matrix $C$ as the diagonal matrix with the value $(10)^2$ in the first row and column index, $(0.1)^2$ for the remaining indices down the main diagonal, and zeros elsewhere. We also keep the same value for our observation noise $D = 10^2$. Figures 5.9, 5.10, 5.11, 5.12, 5.13, 5.14, 5.15, and 5.16 show the state estimates of our membrane potential and gating variables for ensemble sizes of $N = 10$,

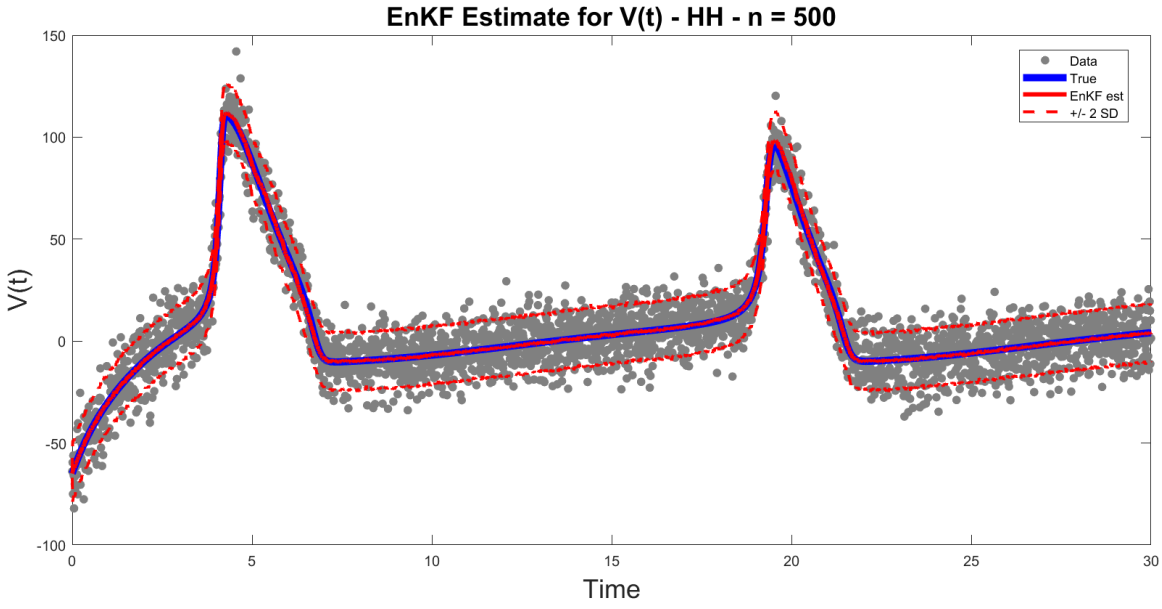


**Figure 5.7:** EnKF estimates for membrane potential with an ensemble size of $N = 500$.

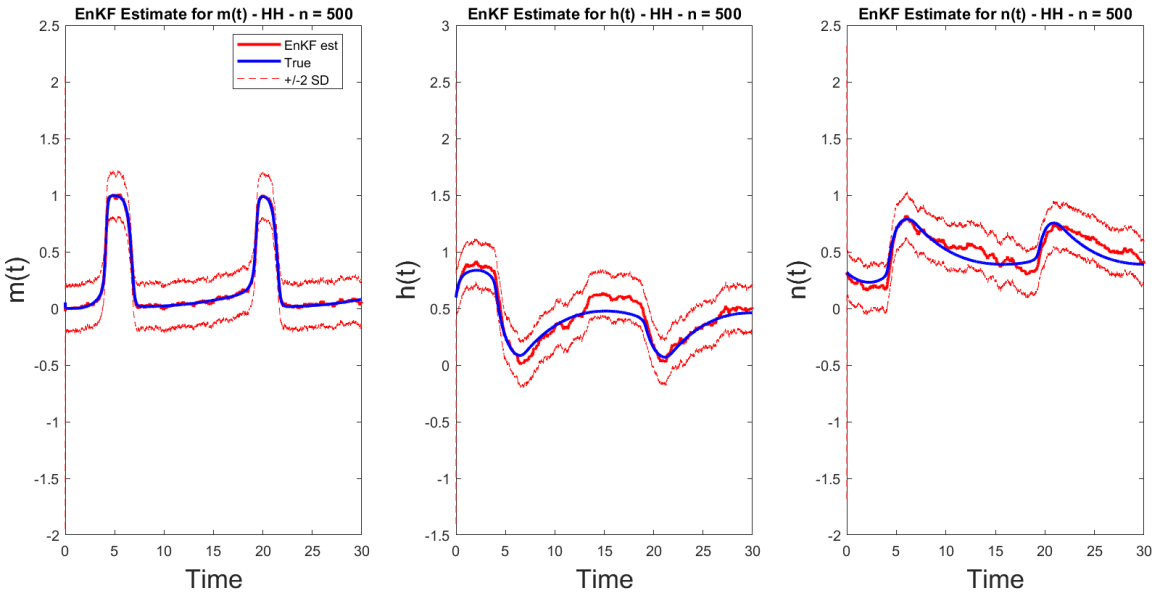


**Figure 5.8:** EnKF estimates for membrane potential with an ensemble size of $N = 500$.

50, 100, and 500. Appendix B includes figures where we modify the definitions of $C$ and $D$ to validate and verify our state estimates within our stochastic framework.

Looking at the figures, we draw similar conclusions of our state estimates to that of our deterministic system. As we increase the size of our ensemble, holding all other parameters constant, we see better tracking of our state estimations of both the measured variable (voltage) and unmeasured variables (gating variables). Specifically looking at an ensemble of size $N = 500$, we track our voltage fairly close to that of the true ode15s solution to the SDE H-H system, within bounds of uncertainty. However, our estimates of the unmeasured variables are more noisy than the estimates in our deterministic setting. The inherent noise in our SDE framework along with how we generate our prediction ensemble are

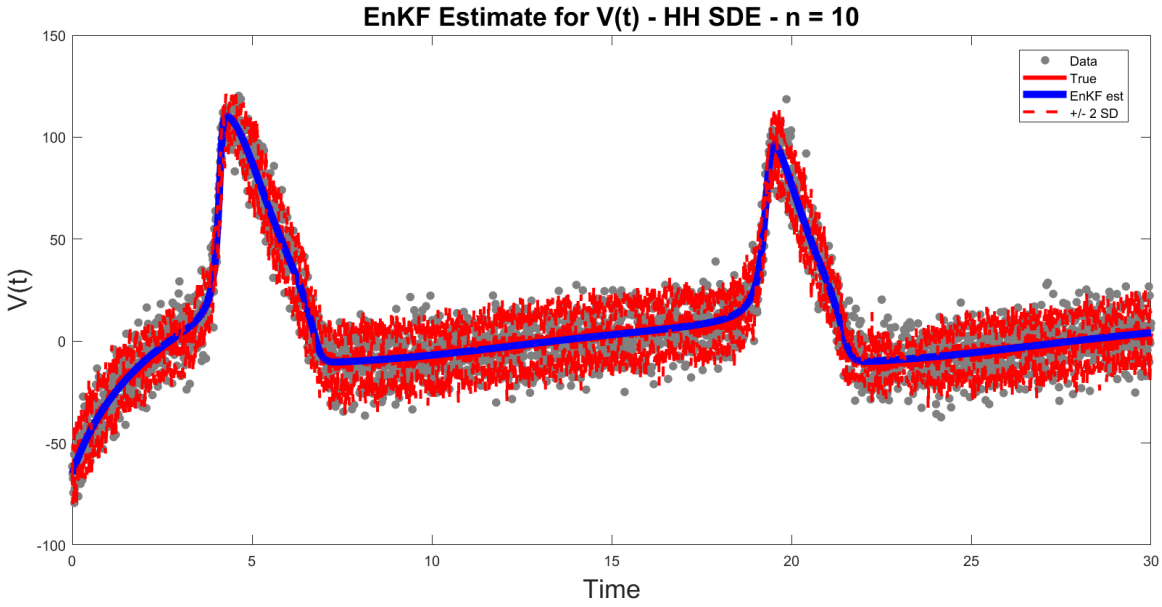


**Figure 5.9:** EnKF estimates for membrane potential with an ensemble size of $N = 10$ in SDE H-H system.

highly stochastic, which in our figures shows to degrade the EnKF state estimates.

In this approach, we are able to observe data at every individual time step. In real life, it may not be possible to have such granular elements. How well does the EnKF perform in estimating the H-H states when we have less data? Figures 5.17, 5.18, 5.19, and 5.20 explore the strength of state estimation when we have 1/100 and 1/10 of the observed data from that of the original approach. For sake of argument, we draw a comparison using a prediction ensemble size of $N = 500$. We generate an observation ensemble at every 10th time step to obtain an observation ensemble 1/100 of the original ensemble, and an observation ensemble at every 100th time step for 1/10 of the original ensemble. When we have 1/100 of the original observed data, we see that the state estimation of the voltage still performs well, with the bounds of uncertainty capturing information of our data. Our unmeasured variables still have noisy estimates, but are comparable to that of the original approach of an ensemble size $N = 500$. We conclude that 1/100 of the original data may be sufficient enough to track estimates of our SDE H-H system. With 1/10 of the observed data, we see a drastic degradation of our state estimates and a slight shift with respect to time. Our unmeasured variables reflect the shift, and also produce noisy estimates. As expected, when we decrease the amount of data available to update our state estimates, we lose some traceability and have lowered performance.

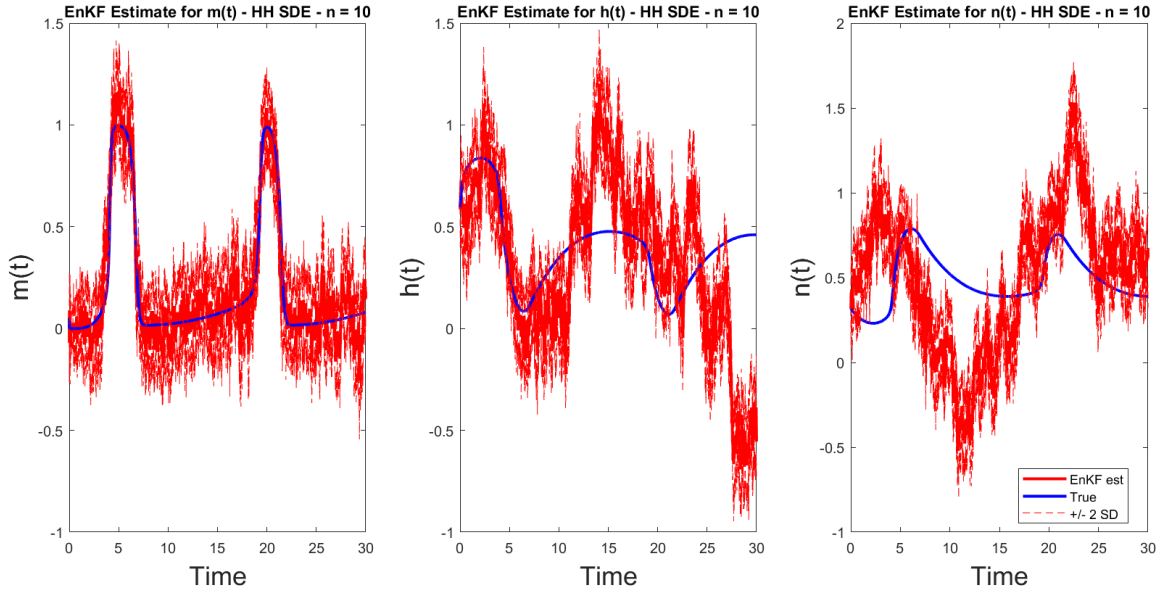


**Figure 5.10:** EnKF estimates for gating variables with an ensemble size of $N = 10$ in SDE H-H system.

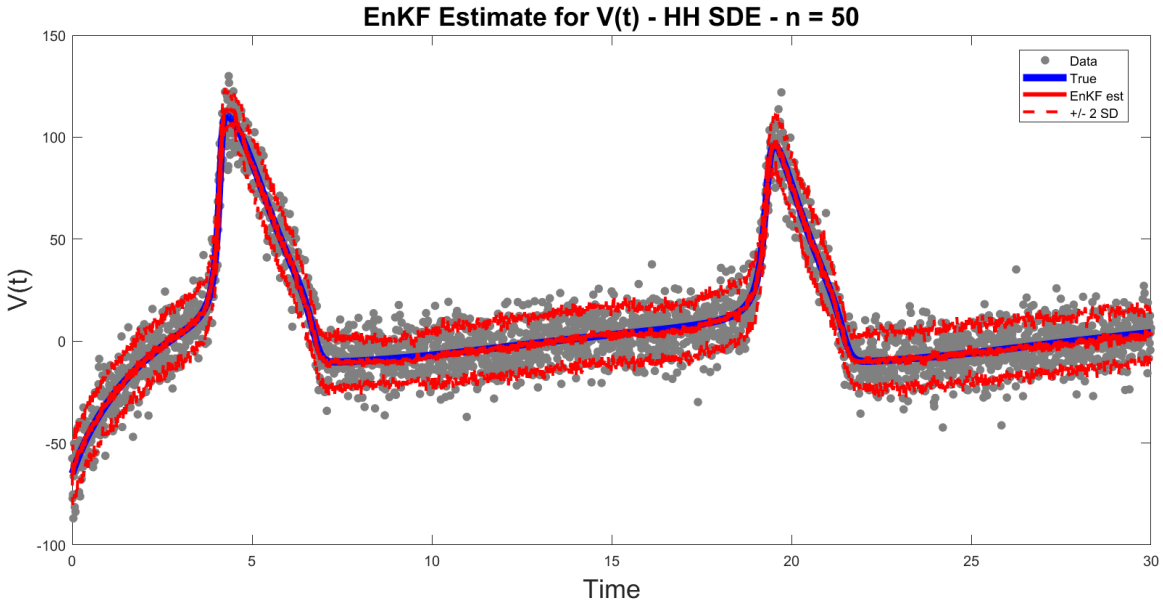


**Figure 5.11:** EnKF estimates for membrane potential with an ensemble size of $N = 50$ in SDE H-H system.

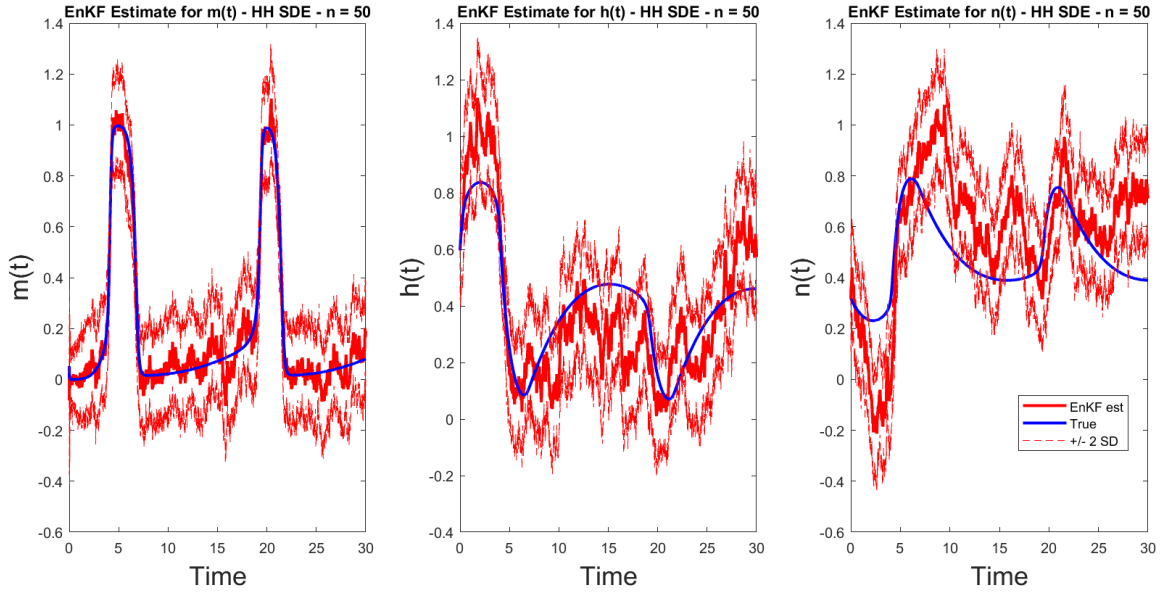


**Figure 5.12:** EnKF estimates for gating variables with an ensemble size of $N = 50$ in SDE H-H system.

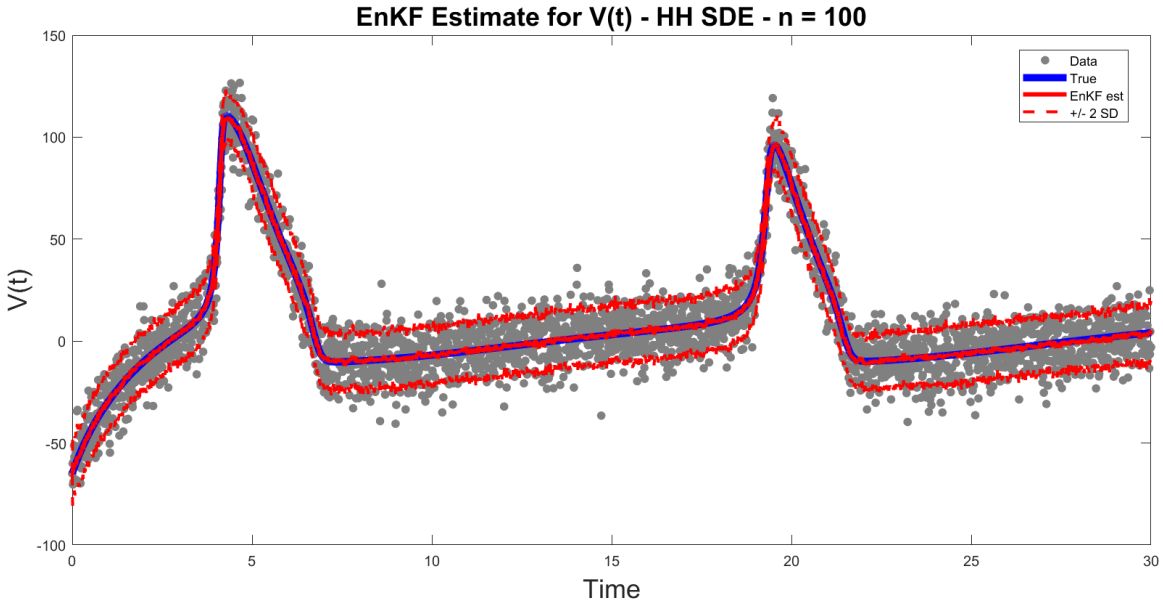


**Figure 5.13:** EnKF estimates for membrane potential with an ensemble size of $N = 100$ in SDE H-H system.

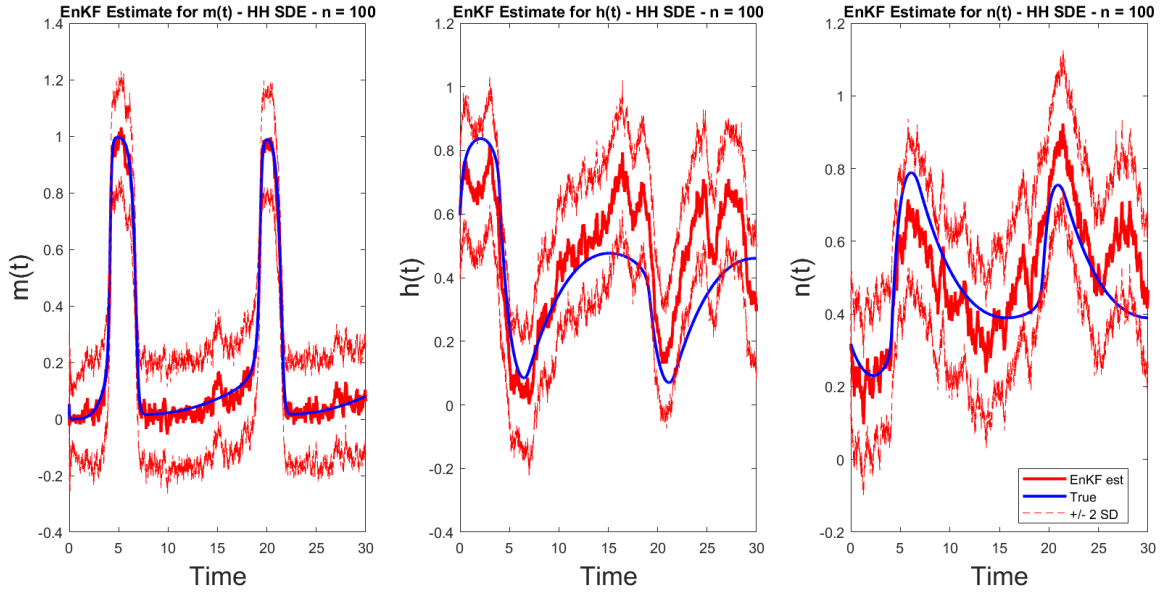


**Figure 5.14:** EnKF estimates for gating variables with an ensemble size of $N = 100$ in SDE H-H system.

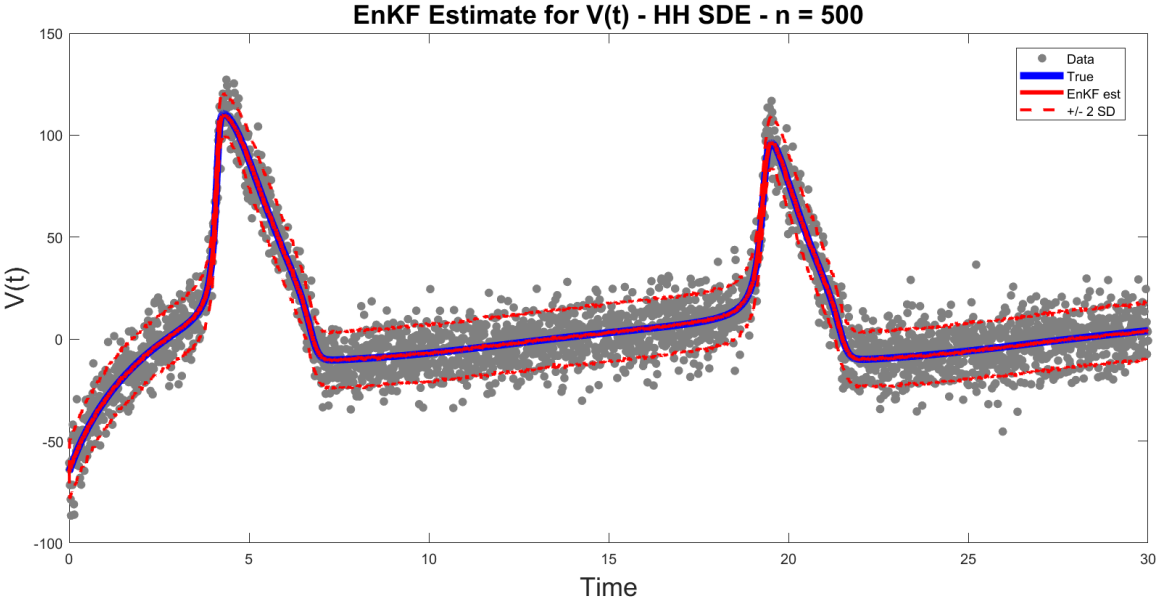


**Figure 5.15:** EnKF estimates for membrane potential with an ensemble size of $N = 500$ in SDE H-H system.

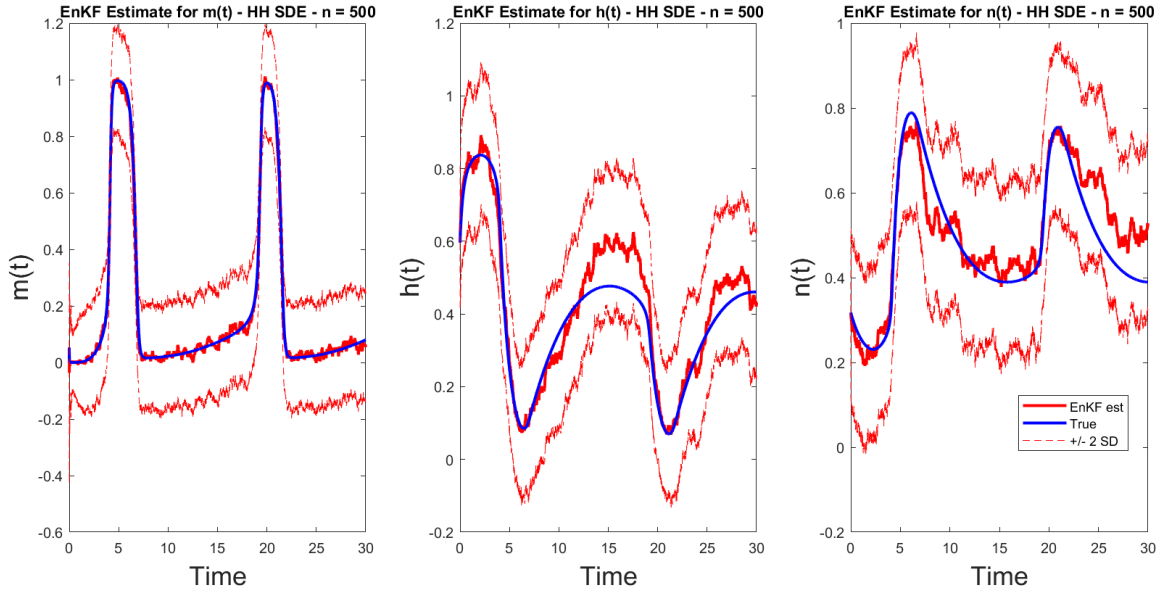


**Figure 5.16:** EnKF estimates for gating variables with an ensemble size of $N = 500$ in SDE H-H system.

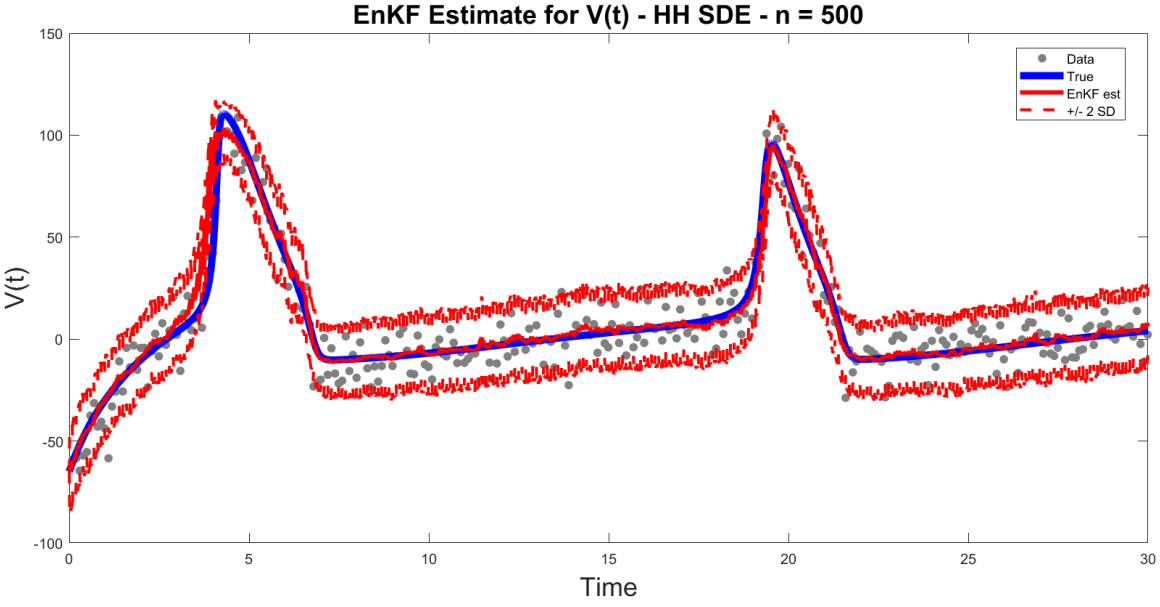


**Figure 5.17:** EnKF estimates for membrane potential with a prediction ensemble size of $N = 500$ and 300 data observations in SDE H-H system.

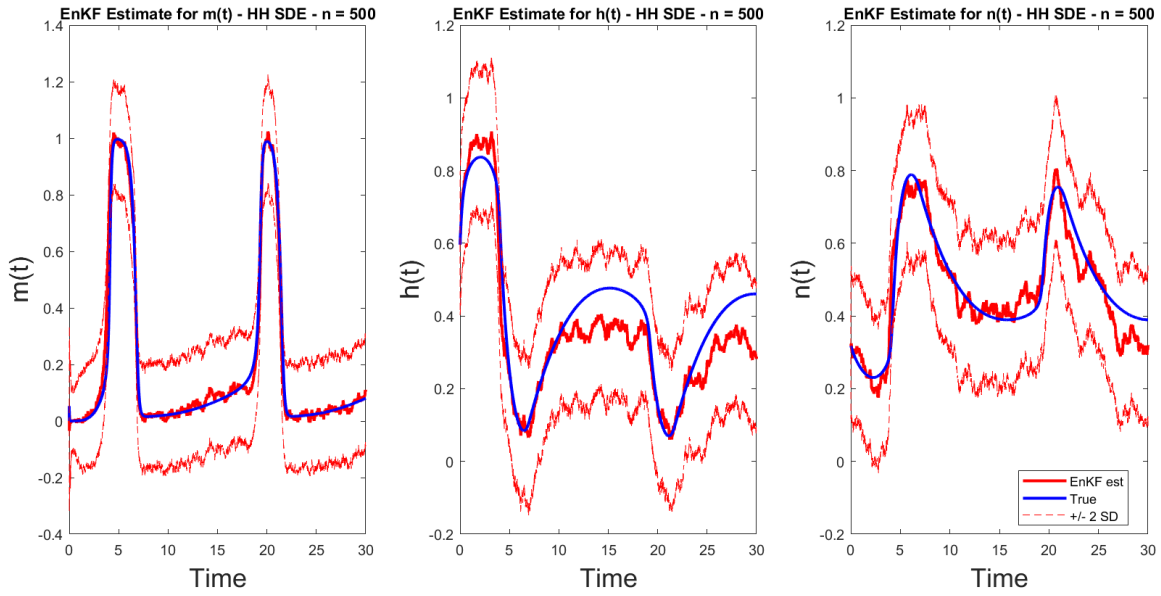


**Figure 5.18:** EnKF estimates for gating variables with a prediction ensemble size of $N = 500$ and 300 data observations in SDE H-H system.

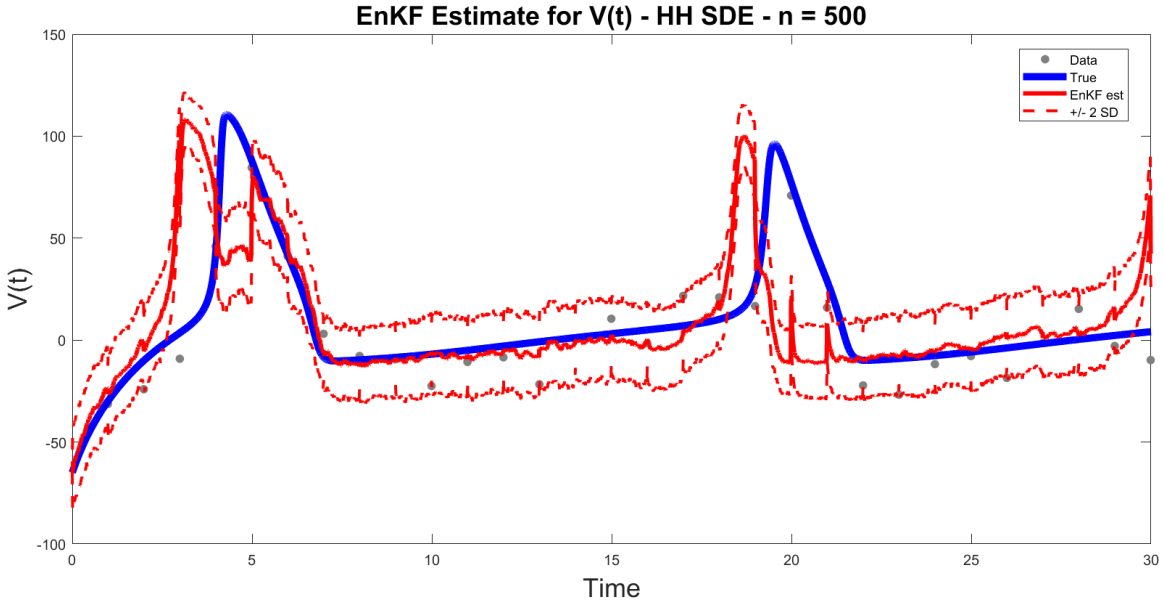


**Figure 5.19:** EnKF estimates for membrane potential with a prediction ensemble size of $N = 500$ and 30 data observations in SDE H-H system.

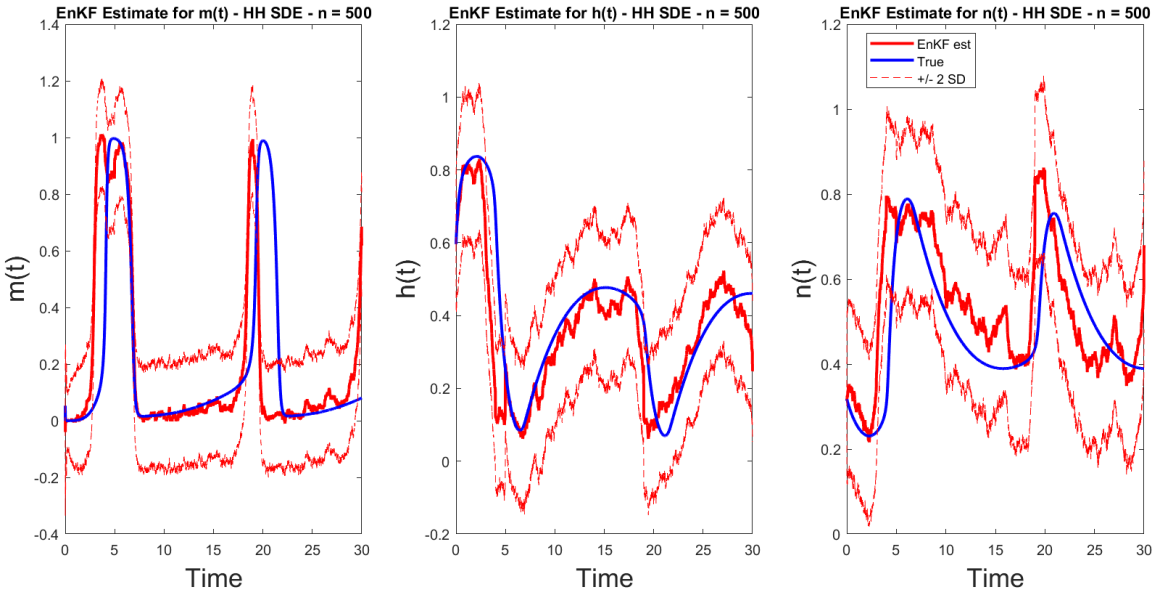


**Figure 5.20:** EnKF estimates for gating variables with a prediction ensemble size of $N = 500$ and 30 data observations in SDE H-H system.

# Chapter 6

# Conclusions and Future Work

In this chapter, we discuss the key results of the numerical experiments performed in this project, starting with the mathematical modeling results found in Chapter 3. We then discuss the conclusions drawn from Chapter 5 in regards to the state estimation problem of our deterministic and stochastic formulation of the H-H neuron model. We conclude by providing insights about future work beyond this project.

## 6.1 Mathematical Modeling

To capture the dynamics of the H-H system under a stochastic framework, we first extended the governing equations of the deterministic H-H system as proposed by Hodgkin and Huxley in 1952 to SDEs. We formulated the SDE system with the definition of the Itô integral. We then upon selection of a constant diffusion coefficient for each of the 4 variables apply the EM method to approximate the solution to our coupled SDEs. We defined our step size for integration to be 0.01 to ensure proper integration of our system. We showed through simulating various numbers of Brownian paths the expected value or average of numerous EM approximations approached that of the numerical solution to the ODE H-H system. This is to be expected, since Brownian motion is Gaussian centered at 0 with independent increments with respect to time. Increasing the number of EM approximations additionally showed improved dynamics of both the membrane potential and the gating variables.

## 6.2 State Estimation

In this work, we employed the EnKF to perform state estimation of the membrane potential and gating variables of the SDE H-H system. We first performed state estimation with EnKF to the deterministic H-H system as a baseline for comparison. We defined our step size to 0.01 and presented estimates for prediction and observation ensemble sizes of $N = 10$, 50, 100, and 500 members. We used Euler's method to form our prediction state ensemble and MATLAB's ode15s to create our observation ensemble. We experimented with different values for $C$ and $D$ with respect to our states, where our innovation process noise $V_{j+1}^n \sim \mathcal{N}(0, C)$ and observation error $w_{j+1}^n \sim \mathcal{N}(0, D)$. We found that larger ensemble sizes gave more accurate predictions of our states, with ensemble sizes of $N = 100$ and 500 members performing best for state estimation of our gating variables. All our ensemble sizes performed well in the state estimation of our membrane potential in part to how we define our noise and the construction of our observation ensemble.

In the stochastic framework, one of the goals of our project was to see if we could capture the dynamics of the unmeasured variables. We extended the EnKF to encompass our SDE framework by redefining how we generate our prediction state ensemble. We use the EM method to generate the prediction state ensemble where we embedded our innovation noise $V_{j+1}^n$ inside our numerical solution. We kept the same step size, $C$ and $D$ definitions, and initial conditions to directly contrast our estimates between the deterministic and stochastic H-H system. We drew similar conclusions to that of the deterministic setting in that as we increase our ensemble size, we gain better estimates of our states. For small ensemble sizes, we saw that we were not able to track the inactivation gates of the sodium channel ($h$) and the activation gates of the potassium channel ($n$) accurately. We well depicted estimates of our voltage, regardless of ensemble size and keeping other parameters constant, within a measure of uncertainty.

We noted that the performance of these estimates was also due to the fact of having a large amount of data to update our posterior state means and covariance. We then aimed to see how the performance would degrade if we only had 1/100 of the data from the original approach, and again where we only had 1/10. This resulted in estimates where we observed data (simulated voltage data) at every 10th time step or 100th time step respectively over the given time interval (300 or 30 data points). We found that with 1/100 of the original data we were still able to capture proper estimates of our voltage with some uncertainty. The estimates of the unmeasured variables were comparable to that of the original approach. When we decreased the number of observations to 30, we found our estimates to degrade substantially. Specifically, we noted that little observed data gave inaccurate estimates of when we expect voltage spiking behavior. This result was carried through to the estimates of our unmeasured variables where we saw highly stochastic results.

## 6.3 Conclusion

Randomness is an important element in mathematical modeling, particularly within the biological space. Through working inside a stochastic framework, this project compared the dynamics of the deterministic H-H neuron model to the model written as a system of SDEs. We leveraged the EM method to find a numerical solution to the SDE H-H system and the EnKF to perform state estimation with the overarching goal of uncovering insights into how stochasticity influences neuron behavior. By employing data assimilation techniques, we were motivated by the prospect of applying these approaches to address inverse problems with real-world data in hopes to understand complex biological systems.

## 6.4 Future Work

There are several avenues that could be taken to expand the scope of this project. All the suggestions we give here further express the goal of capturing the dynamics of the H-H model as represented by a system of SDEs.

Our motivation for this project was to understand in-vivo neuron dynamics. In future work, performing state estimation via the EnKF but having the observation ensemble be taken from actual observed data could provide insight into the effectiveness of this approach applied to the H-H SDE framework. Moreover, due to time constraints of this project, parameter estimation through Kalman filtering is left to future work. To our knowledge, there have been efforts in the literature to perform parameter estimation of the deterministic H-H system [51, 33], but less on the SDE formulation [48].

Additionally, there are several changes we could implement to our model specification that could

further help to validate usage of the EM method and EnKF. For example, in this project we fix the current of our system to a value accepted in the literature. However, there have been efforts in the literature that emphasize estimation through a time-varying current [7]. The competition dataset also applies time-varying currents to enhance the challenge [16, 17]. There are many unknowns in our system that need to either be chosen carefully or estimated as parameters. Upon selection or estimation, we hope that future work can further improve our understanding of the H-H SDE framework.

# References


[1] E. J. Allen, L. J. S. Allen, A. Arciniega, and P. E. Greenwood. Construction of equivalent stochastic differential equation models. *Stochastic Analysis and Applications*, 26:274–297, 2008.

[2] L. J. S. Allen. *An Introduction to Stochastic Processes with Applications to Biology*. CRC Press, 2 edition, 2010.

[3] J. L. Anderson. An ensemble adjustment Kalman filter for data assimilation. *Monthly Weather Review*, 129(12):2884–2903, 2001.

[4] J. L. Anderson. Ensemble Kalman filters for large geophysical applications. *IEEE Control Systems Magazine*, 29(3):66–82, 2009.

[5] J. M. Bekkers. Pyramidal neurons. *Current Biology*, 21(24):R975, 2011.

[6] K. Burrage, P.M. Burrage, and T. Tian. Numerical solutions for strong solutions of stochastic differential equations: an overview. *Proceedings: Mathematical, Physical and Engineering Sciences*, 460(2041):373–402, 2004.

[7] K. Campbell, L. Staugler, and A. Arnold. Estimating time-varying applied current in the Hodgkin-Huxley model. *Applied Sciences*, 10(2):550, 2020.

[8] M. D. Collins and L. Fishman. Efficient navigation of parameter landscapes. *The Journal of the Acoustical Society of America*, 98(3):1637–1644, 1995.

[9] C. E. Dangerfield, D. Kay, and K. Burrage. Modeling ion channel dynamics through reflected stochastic differential equations. *Physical Review E*, 85(5):051907, 2012.

[10] V. Dobrushkin. Mathematica tutorial for the first course. part III: Heun methods. Tutorial notes for Applied Mathematics I, 2022. Accesed via `https://www.cfm.brown.edu/people/dobrush/am33/Mathematica/heun.html`.

[11] G. Evensen. Sequential data assimilation with a non-linear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99(C5):10143–10162, 1994.

[12] G. Evensen. The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53:343–367, 2003.

[13] S. E. Fadugba, B. G. Adegboyegun, and O. T. Ogunbiyi. On the convergence of Euler Maruyama method and Milstein scheme for the solution of stochastic differential equations. *International Journal of Applied Mathematics and Modeling*, 1(1):9–15, 2013.

[14] S. J. Farlow. *An Introduction to Differential Equations and Their Applications*. Dover Publications, 2006.

[15] R. F. Fox and Y. Lu. Emergent collective behavior in large numbers of globally coupled independently stochastic ion channels. *Phys. Rev. E*, 49:3421, 1994.

[16] W. Gersterner and R. Naud. How good are neuron models? *Science*, 326:379–380, 2009.

[17] W. Gerstner, R. Naud, M. Carandini, L. C. Sincich, J. C. Horton, D. L. Adams, and J. R. Economides. About ch-epfl-2009. CRCNS.org, 2009. Accessed via `https://crcns.org/data-sets/challenges/ch-epfl-2009/about-ch-epfl-2009`.

[18] I. I. Gikhman and A. V. Skorokhod. *The Theory of Stochastic Processes III*. Springer, 1979.

[19] N. Gill. Deterministic and stochastic models of infectious disease: circular migrations and HIV transmission dynamics. Technical report, University of Chicago, 2015. Accessed via `https://math.uchicago.edu/~may/REU2015/REUPapers/Gill.pdf`.

[20] J. H. Goldwyn, N. S. Imennov, M. Famulare, and E. Shea-Brown. Stochastic differential equation models for ion channel noise in Hodgkin-Huxley neurons. *Physical Review E*, 83(4 Pt 1):041908, 2011.

[21] A. Gray, D. Greenhalgh, L. Hu, X. Mao, and J. Pan. A stochastic differential equation SIS epidemic model. *SIAM Journal on Applied Mathematics*, 71(3):876–902, 2011.

[22] P. E. Greenwood and L. M. Ward. *Stochastic Neuron Models*. Springer, 2016.

[23] D. J. Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM Review*, 43(3):525–546, 2001.

[24] D. J. Higham and P. E. Kloeden. *An Introduction to the Numerical Simulation of Stochastic Differential Equations*. Society of Industrial and Applied Mathematics, 2021.

[25] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 117:500–544, 1952.

[26] M. Hutzenthaler, A. Jentzen, and P. E. Kloeden. Strong convergence of an explicit numerical method for SDEs with nonglobally Lipschitz continuous coefficients. *The Annals of Applied Probability*, 22(3):1611–1641, 2012.

[27] E. Izhikevich and R. FitzHugh. FitzHigh-Nagumo model. *Scholarpedia Journal*, 1(9):1349, 2006.

[28] E. M. Izhikevich. Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6):1569–1572, 2003.

[29] N. Kasabov. To spike or not to spike: A probabilistic spiking neuron model. *Neural Networks*, 23:16–19, 2010.

[30] M. Katzfuss, J. R. Stroud, and C. K. Wikle. Understanding the Kalman filter. *The American Statistician*, 70(4):350–357, 2016.

[31] A. K. Kaw and E. E. Kalu. Ordinary differential equations: Euler's method. In *Numerical Methods with Applications*, pages 525–534. Autarkaw.com, 2008. https://nm.mathforcollege.com/chapter-08-02-eulers-method/.

[32] P. Kloeden and A. Neuenkirch. Convergence of numerical methods for stochastic differential equations in mathematical finance. In *Recent Developments in Computational Finance: Foundations, Algorithms and Applications*, pages 49–80. World Scientific, 2013.

[33] M. Lankarany, W. P. Zhu, and M. N. S. Swamy. Joint estimation of states and parameters of Hodgkin-Huxley neuronal model using Kalman filtering. *Neurocomputing*, 136:289–299, 2014.

[34] H. Lecar. Morris-Lecar model. *Scholarpedia Journal*, 2(10):1333, 2007.

[35] S. Lee and K. Seunghwan. Parameter dependence of stochastic resonance in stochastic Hodgkin-Huxley neuron. *Physical Review E*, 60:826, 1999.

[36] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[37] L. Mitchell and A. Arnold. Analyzing the effects of observation function selection in ensemble Kalman filtering for epidemic models. *Mathematical Biosciences*, 339:108655, 2021.

[38] I. Overeem. Geological modeling: deterministic and stochastic models. University of Colorado Boulder, 2008. Accessed via https://csdms.colorado.edu/csdms_wiki/images/Lecture_4_objects.pdf.

[39] A. Robler. Second order Runge-Kutta methods for Ito stochastic differential equations. *SIAM Journal on Numerical Analysis*, 47(3):1713–1738, 2009.

[40] P. F. Rowat and P. E. Greenwood. The ISI distrubution of the stochastic Hodgkin-Huxley neuron. *Frontiers in Computational Neuroscience*, 8:111, 2014.

[41] A. Saarinen, M. Linne, and O. Yli-Harja. Modeling single neuron behavior using stochastic differential equations. *Neurocomputing*, 69:1091–1096, 2006.

[42] N. H. Sabah and K. N. Leibovic. Subthreshold oscillatory responses of the Hodgkin-Huxley cable model for the squid giant axon. *Biophysical Journal*, 9(10):1206–1222, 1969.

[43] T. Sauer. Numerical solution of stochastic differential equations in finance. In *Handbook of Computational Finance*, pages 529–550. Springer Berlin Heidelberg, 2012.

[44] C. J. Schwiening. A brief historical perspective: Hodgkin and Huxley. *Journal of Physiology*, 590(11):2571–2575, 2012.

[45] A. S. Shai, C. A. Anastassiou, M. E. Larkum, and C. Koch. Physiology of layer 5 pyramidal neurons in mouse primary visual cortex: coincidence detection through bursting. *PLoS Computational Biology*, 11(3):e1004090, 2015.

[46] L. F. Shampine and M. W. Reichelt. The MATLAB ODE Suite. *SIAM Journal on Scientific Computing*, 18:1–22, 1997.

[47] E. Shneidman, B. Freedman, and I. Segev. Ion channel stochasticity may be critical in determining the reliability and precision of spike timing. *Neural Computation*, 10:1679–1703.

[48] J. Soto and S. Infante. Ensemble Kalman filter and extended Kalman filter for state-parameter dual estimation in mixed effects models defined by a stochastic differential equation. In *Technology, Sustainability and Educational Innovation (TSIE)*, pages 285–300. Springer International Publishing, 2020.

[49] A. Swietlicka, K. Gugala, W. Pedrycz, and A. Rybarczyk. Development of the deterministic and stochastic Markovian model of a dendritic neuron. *Biocybernetics and Biomedical Engineering*, 37:201–216, 2017.

[50] S. Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[51] G. Ullah and S. J. Schiff. Tracking and control of neuronal Hodgkin-Huxley dynamics. *Phys Rev E Stat Nonlin Soft Matter Phys*, 79(4 Pt 1):04091, 2009.

[52] J. A. White, J. T. Rubinstein, and A. R. Kay. Channel noise in neurons. *Trends in Neurosciences*, 23:131–137, 2000.

[53] G. Wulfram, W. M. Kistler, R. Naud, and L. Paninski. *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition and Beyond*. Cambridge University Press, 2014.

[54] M. E. Yamakou, T. D. Tran, L. H. Duc, and J. Jost. The stochastic FitzHugh-Nagumo neuron model in the excitable regime embeds a leaky integrate-and-fire model. *Journal of Mathematical Biology*, 79:509–532, 2019.

[55] L. Zheng and X. Zhang. Numerical methods. In *Modeling and Analysis of Modern Fluid Problems*, pages 361–455. Elsevier Inc, 2017.

# Appendices

# A Further Simulations of EnKF Estimates for the Deterministic H-H System

In this appendix, we present state estimation via the EnKF where we change how our innovation process is defined. Specifically, given that our innovation process $V_{j+1}^n \sim \mathcal{N}(0, C)$, we aim to validate the estimates of $V$, $m$, $h$, and $n$ through changing the definition of matrix $C$. We consider the range of values that the states $V$, $m$, $h$, and $n$ can take from the constraints of the H-H model and show two alternative definitions of matrix $C$ compared to that in Chapter 5.

- Define $C$ as the diagonal matrix consisting of the value 1 in the first row and column index and $(0.1)^2$ for the other indices down the main diagonal. We decrease the amount of noise introduced to prediction state ensemble of $V$.
- Define $C$ as the diagonal matrix consisting of the value $10^2$ in the first row and column index and $(0.01)^2$ for the other indices down the main diagonal. We decrease the amount of noise introduced to prediction state ensemble of the gating variables $m$, $h$, and $n$.

### Decreasing Prediction State Ensemble Noise to $V$

Figures A.1, A.2, A.3, and A.4 show the state estimates of our deterministic H-H system for ensemble sizes $N = 50$ and 500. We keep the same step size of $\Delta t = 0.01$ for consistency. By decreasing the value of the innovation noise applied to our prediction state ensemble of $V$, our estimates of $V$ by the EnKF are not as well tracked as they were in Chapter 5. We note this may be plausible due to not changing the observation noise $D$. Our observation ensemble has more noise, impacting our update step, Kalman gain, and our overall posterior mean estimates for $V$ over time. Our error is small since our posterior covariance is computed using the updated posterior ensemble and posterior mean. As expected, our unmeasured variables exhibit similar estimates to our previous definition of $C$ since nothing is changed in relation to our gating variables.

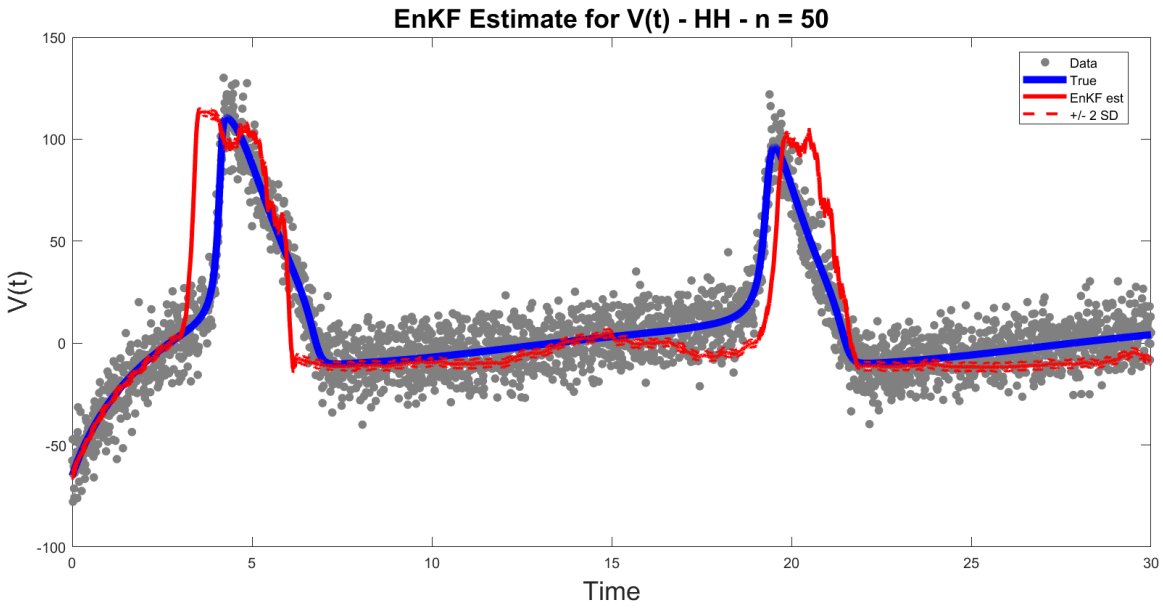


**Figure A.1:** EnKF estimates for membrane potential with an ensemble size of $N = 50$.

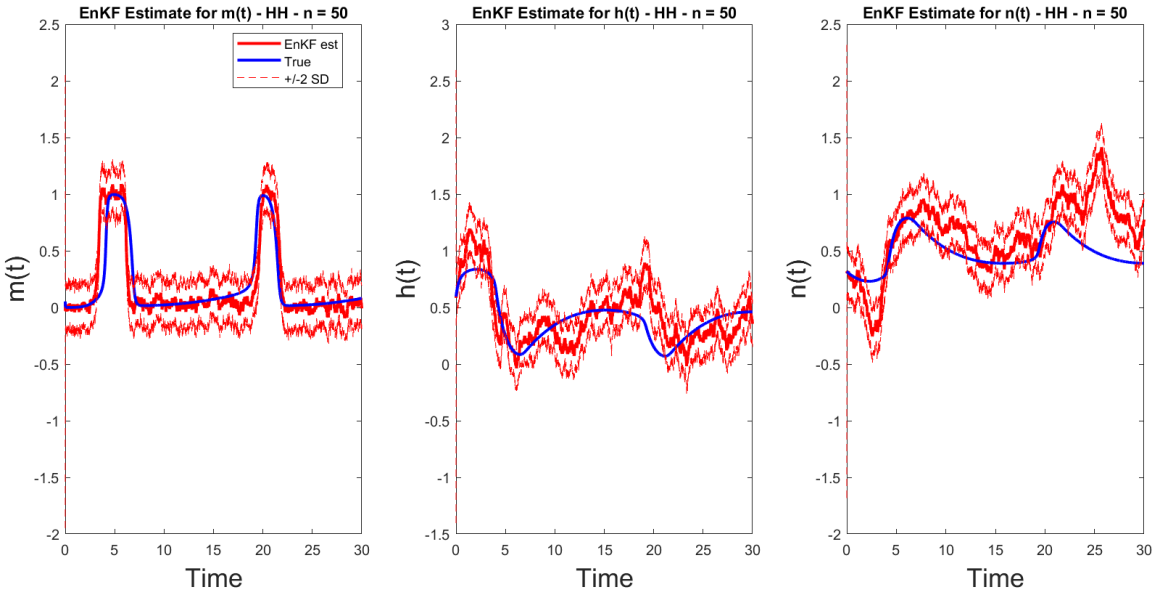


**Figure A.2:** EnKF estimates for gating variables with an ensemble size of $N = 50$.

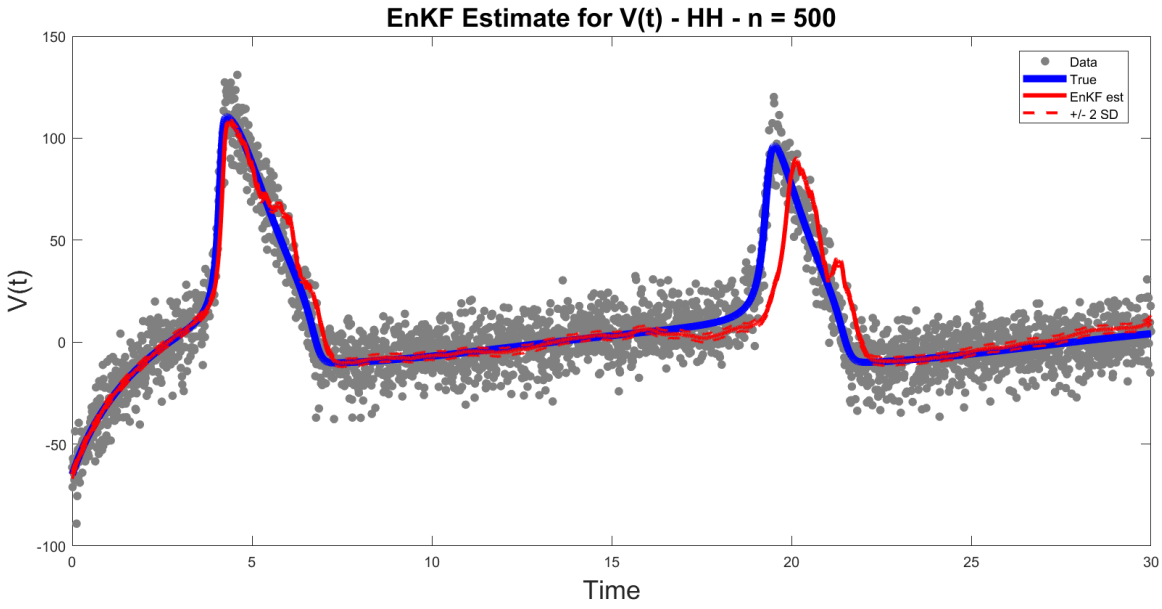


**Figure A.3:** EnKF estimates for membrane potential with an ensemble size of $N = 500$.

### Decreasing Prediction State Ensemble Noise to $m$, $h$, and $n$

Figures A.5, A.6, A.7, and A.8 show the state estimates of our deterministic H-H system for ensemble sizes $N = 50$ and 500. Likewise, we use the same step size $\Delta t = 0.01$. In this case, we change values in our matrix $C$ that impact the estimates of our unmeasured variables. As expected, the state estimation of the membrane potential remains unchanged from the conclusions found in Chapter 5. We track our gating variables, on the other hand, very well with estimates close to that of the true solution. This situation decreases the amount of noise we attribute to the unmeasured variables, so we expect the estimates to improve. It is critical that we define our innovation process well to best capture realistic

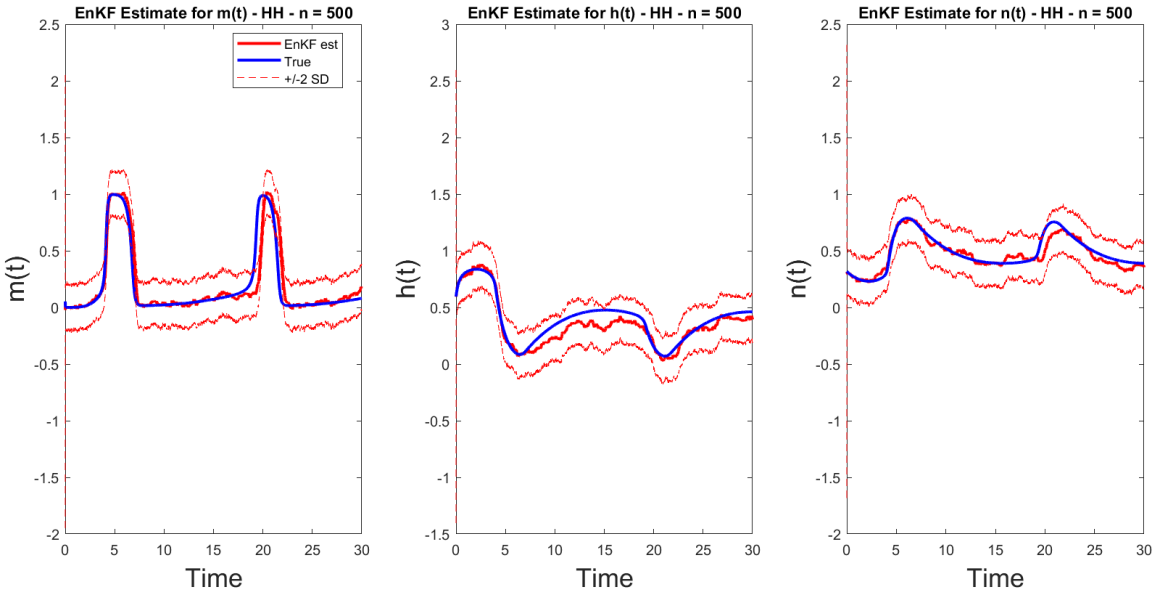


**Figure A.4:** EnKF estimates for gating variables with an ensemble size of $N = 500$.

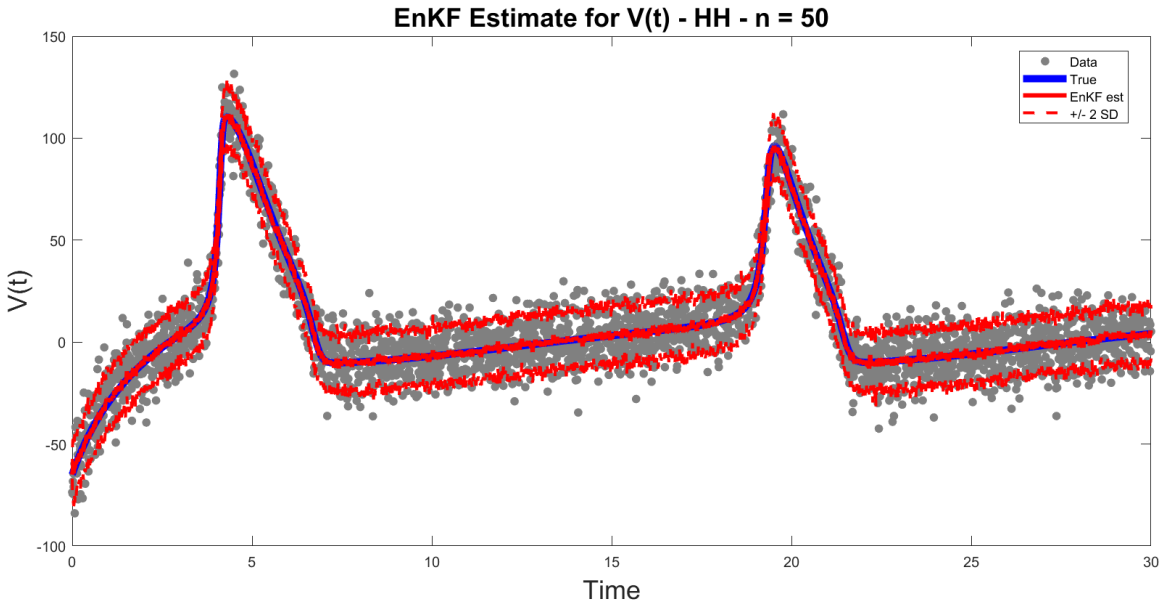


**Figure A.5:** EnKF estimates for membrane potential with an ensemble size of $N = 50$.

estimates. A variance of $(0.01)^2$ is very small in relation to our gating variable estimates, giving values that fall very closely to that of the true solution computed by MATLAB's ode15s.

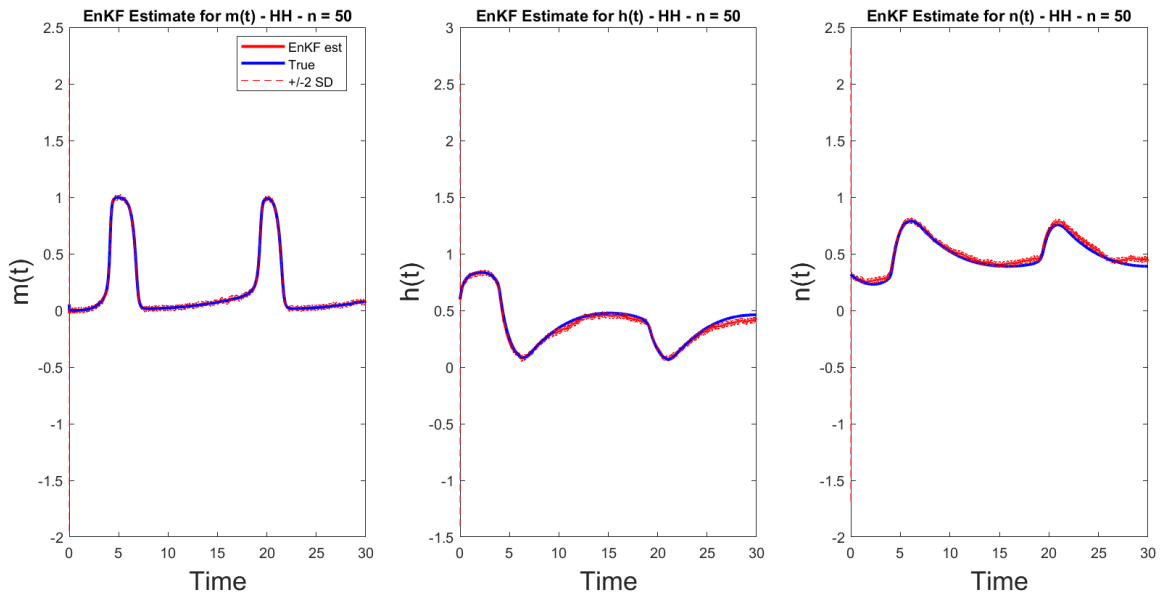


**Figure A.6:** EnKF estimates for gating variables with an ensemble size of $N = 50$.

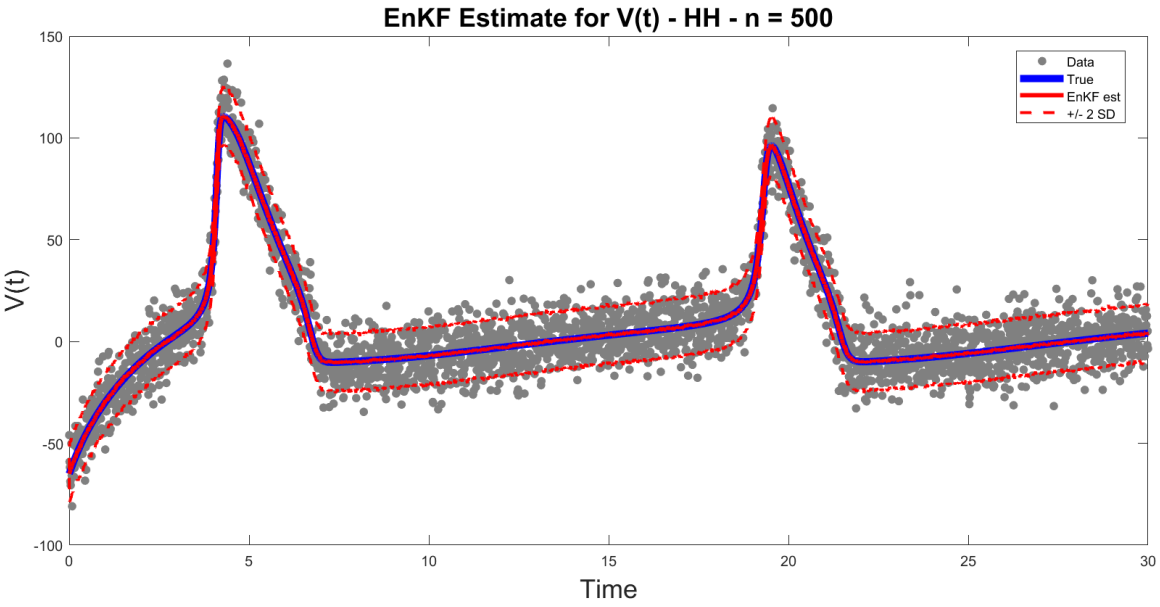


**Figure A.7:** EnKF estimates for membrane potential with an ensemble size of $N = 500$.

# B Further Simulations of EnKF Estimates for the SDE H-H System

Like Appendix A, we present state estimation via the EnKF inside the SDE H-H framework where we change how our innovation process is defined. Similarly, we aim to validate the estimates of $V$, $m$, $h$, and $n$ through how matrix $C$ is defined. We consider the same two situations shown in Appendix A.

- Define $C$ as the diagonal matrix consisting of the value 1 in the first row and column index and $(0.1)^2$ for the other indices down the main diagonal. We decrease the amount of noise introduced

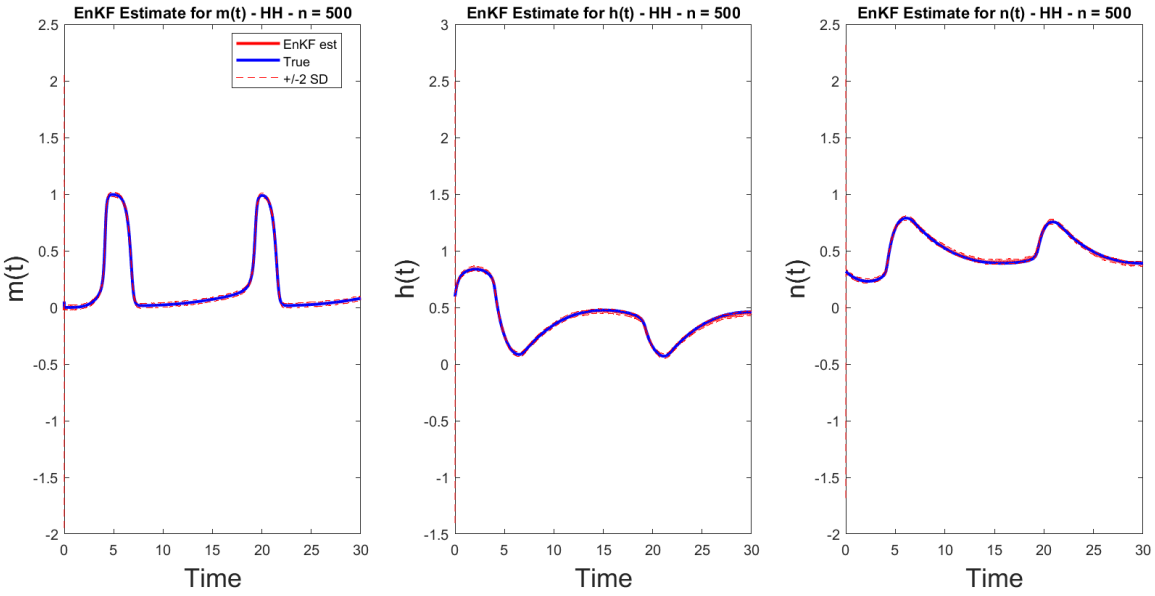


**Figure A.8:** EnKF estimates for gating variables with an ensemble size of $N = 500$.

to prediction state ensemble of $V$.

- Define $C$ as the diagonal matrix consisting of the value $10^2$ in the first row and column index and $(0.01)^2$ for the other indices down the main diagonal. We decrease the amount of noise introduced to prediction state ensemble of the gating variables $m$, $h$, and $n$.

### Decreasing Prediction State Ensemble Noise to $V$

Figures B.1, B.2, B.3, and B.4 show the state estimates of our deterministic H-H system for ensemble sizes $N = 50$ and 500. We keep the same step size of $\Delta t = 0.01$ for consistency. By decreasing the value of the innovation noise applied to our prediction state ensemble of $V$, we observe poor tracking of our estimates of $V$ by the EnKF filter inside the SDE framework compared to the results found in Chapter 5. We note this may be plausible due to not changing the observation noise $D$. Our observation ensemble has more noise, impacting our update step, Kalman gain, and our overall posterior mean estimates for $V$ over time. Our error is small since our posterior covariance is computed using the updated posterior ensemble and posterior mean. As expected, our unmeasured variables exhibit similar estimates as our previous definition of $C$ since nothing is changed in relation to our gating variables. These results are analogous to the results we find in the deterministic setting in Appendix A.

### Decreasing Prediction State Ensemble Noise to $m$, $h$, and $n$

Figures B.5, B.6, B.7, and B.8 show the state estimates of our SDE H-H system for ensemble sizes $N = 50$ and 500. In this case, we change values in our covariance $C$ that impact the estimates of our unmeasured variables. As expected, the state estimation of the membrane potential remains unchanged from the conclusions found in Chapter 5. We track our gating variables, on the other hand, very well with estimates close to that of the true solution. We reason these conclusions similar to that of Appendix A. We attribute less noise to our unmeasured variables so we expect the estimates to improve. A variance of $(0.01)^2$ is very small in relation to our gating variable estimates, giving values that fall very closely to that of the true solution computed by MATLAB's ode15s.

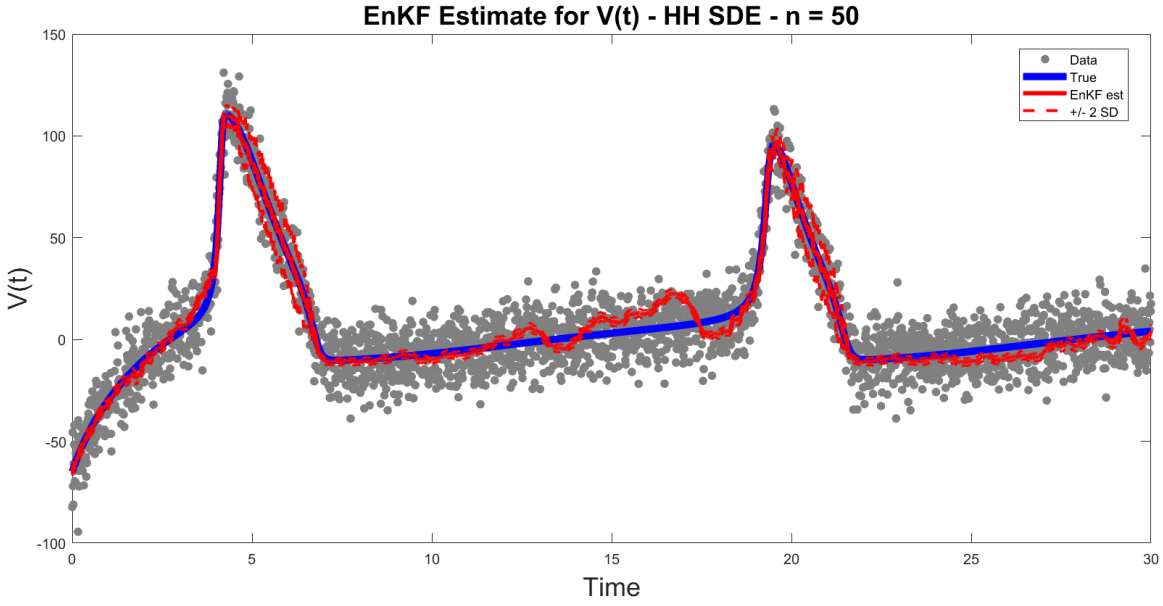


**Figure B.1:** EnKF estimates for membrane potential with an ensemble size of $N = 50$.

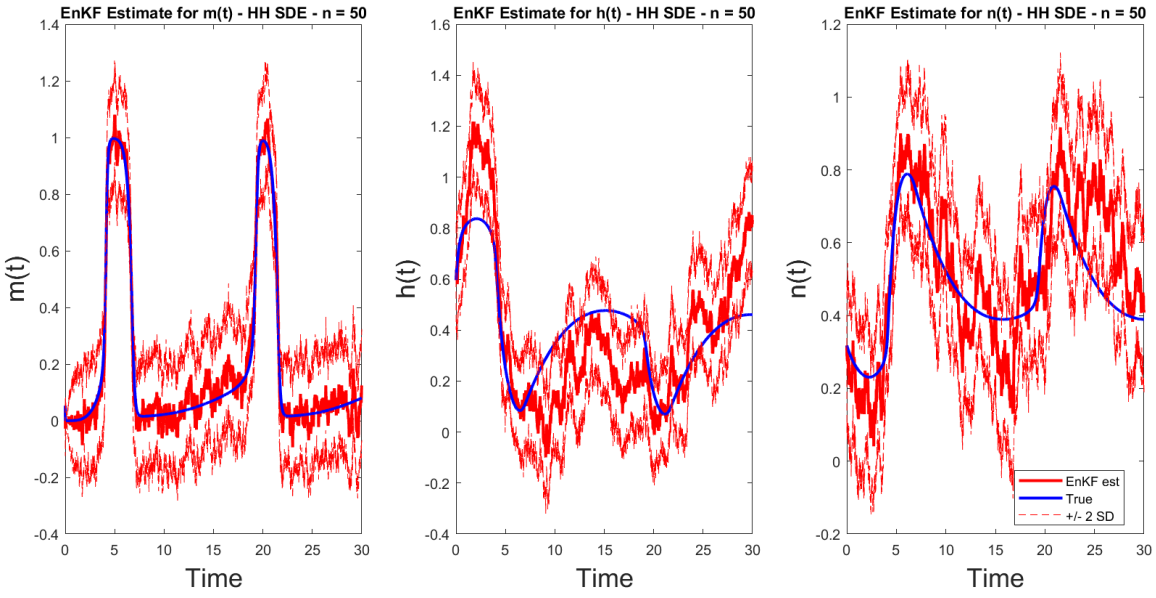


**Figure B.2:** EnKF estimates for gating variables with an ensemble size of $N = 50$.

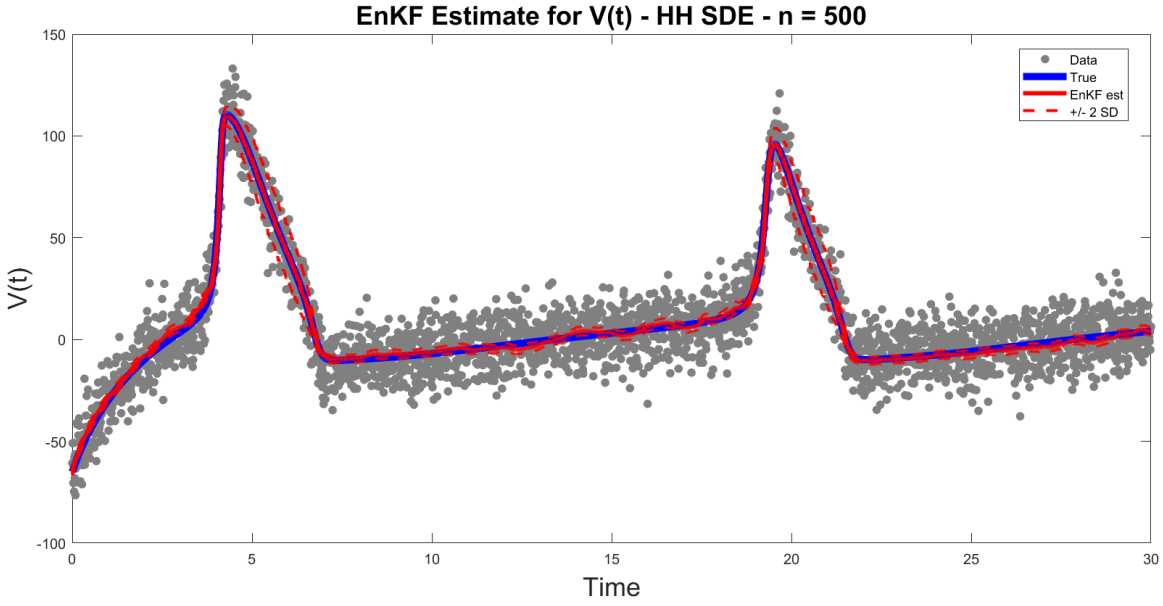


**Figure B.3:** EnKF estimates for membrane potential with an ensemble size of $N = 500$.

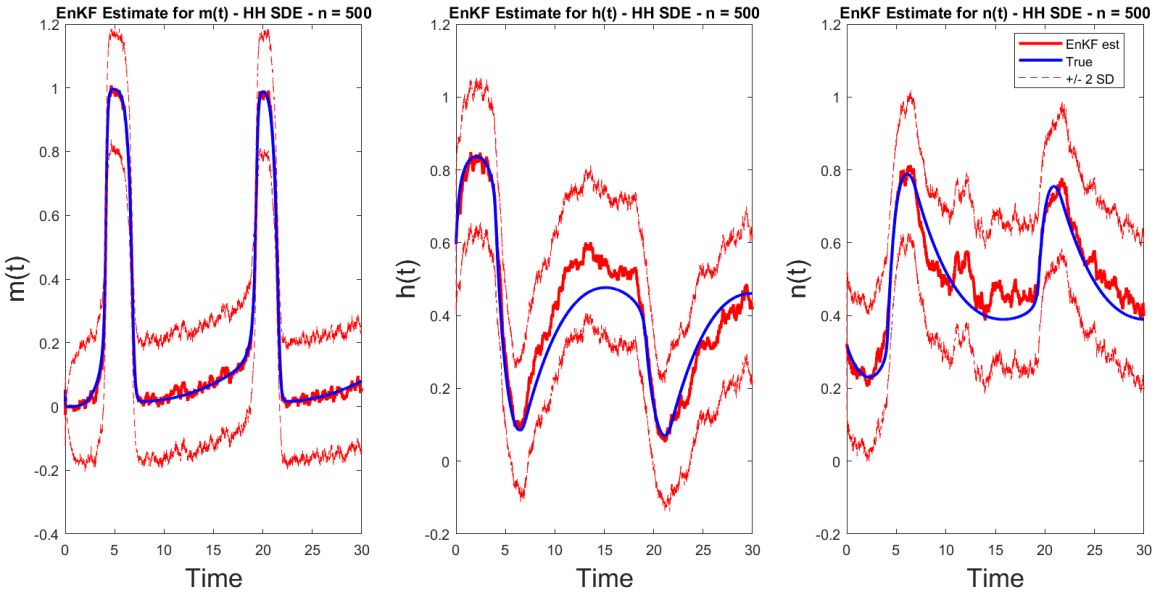


**Figure B.4:** EnKF estimates for gating variables with an ensemble size of $N = 500$.

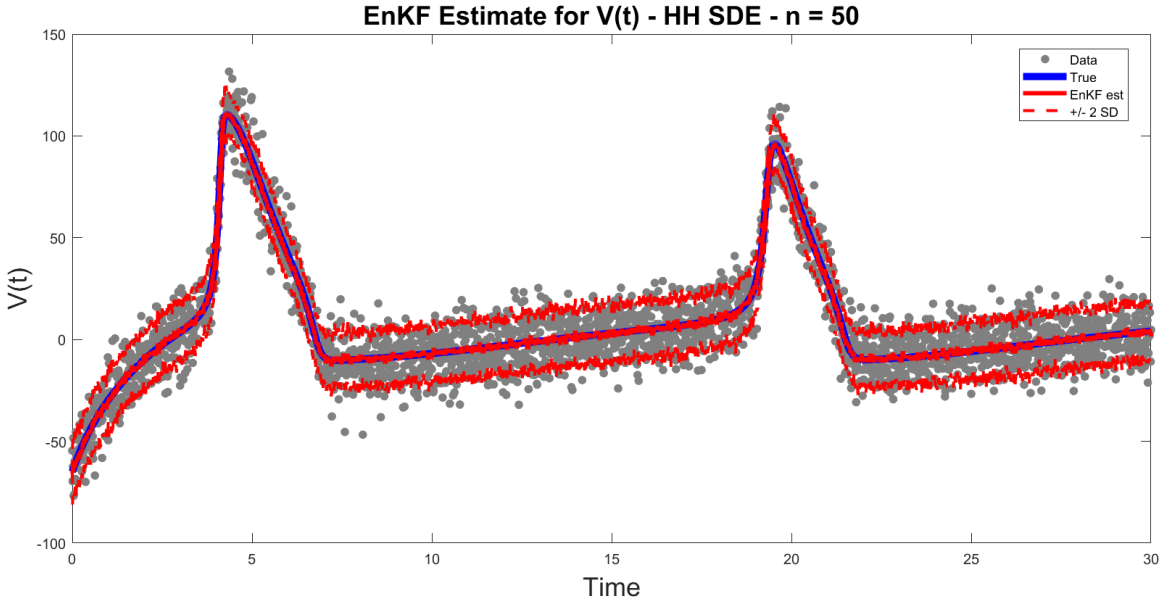


**Figure B.5:** EnKF estimates for membrane potential with an ensemble size of $N = 50$.

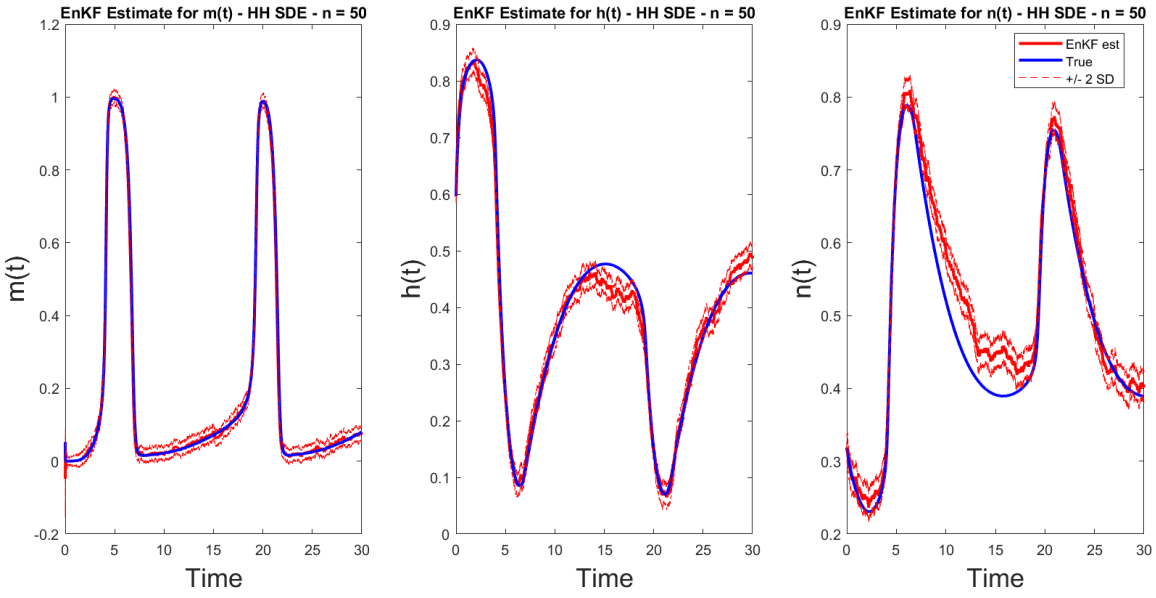


**Figure B.6:** EnKF estimates for gating variables with an ensemble size of $N = 50$.

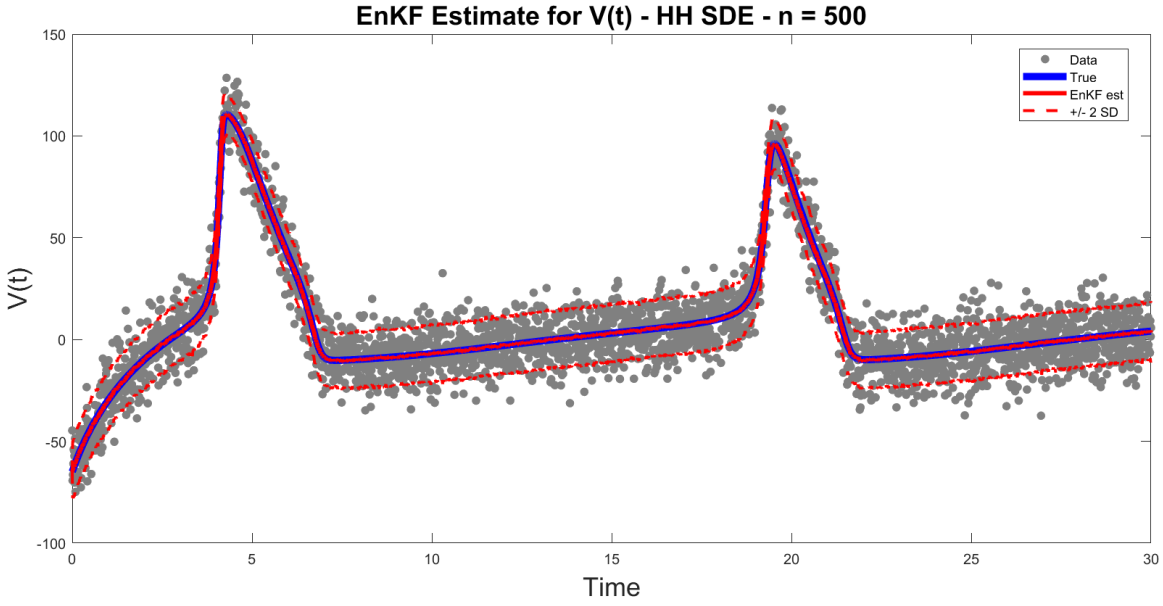


**Figure B.7:** EnKF estimates for membrane potential with an ensemble size of $N = 500$.

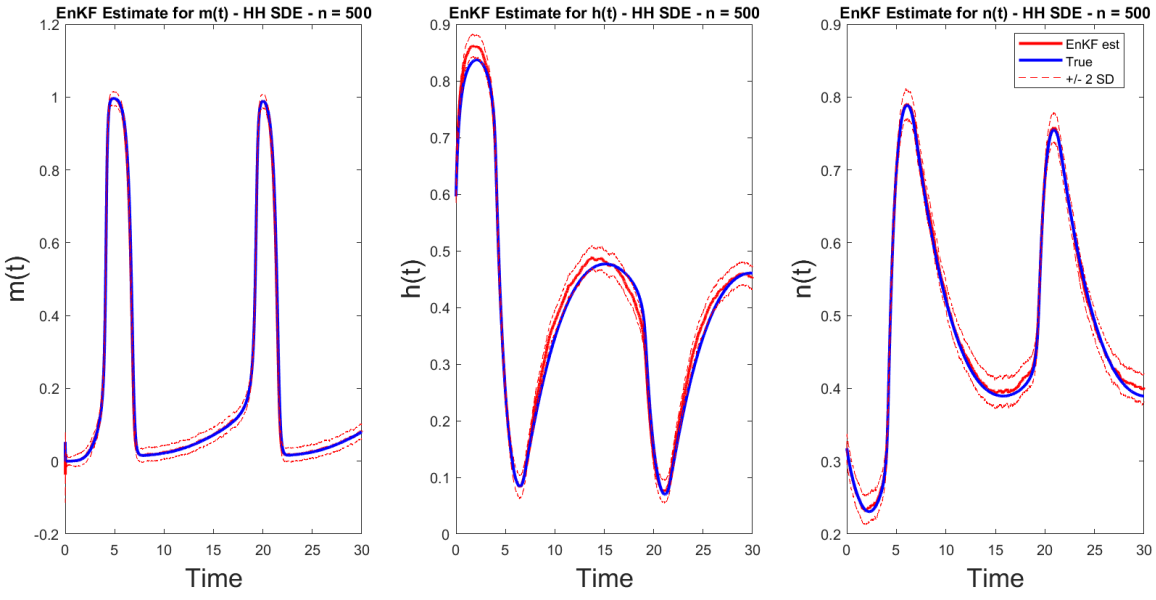


**Figure B.8:** EnKF estimates for gating variables with an ensemble size of $N = 500$.