Methods for Demonstrating Model Equivalence Using Parameter Reduction in Neuronal Models

Aidan Coletta

July 2025

1 Abstract

In the world of computational neuroscience, few individuals have been more influential than Alan Hodgkin and Andrew Huxley. In 1939, prior to the outbreak of World War II, their work resulted in the first known instance of an intracellular recording of the action potential of an in vitro neuron. After publishing their findings in Nature, they worked for another decade in an effort to understand the behavior they observed. Using a technique known as voltage clamping, they were able to control the voltage across the neuronal membrane, allowing them to accurately measure the voltage across a squid giant axon and measure the response of the neuron to different input currents. As a result, in 1952, they published their most cited paper (for which they would later recieve the Nobel prize) in which they first derived a model to predict the action potentials they observed. Known as the Hodgkin/Huxley model, it uses four differential equations and four state variables to describe the change in voltage across the neuronal membrane. At the time of publication, their work represented the most accurate mathematical description of neuronal behavior. Subsequently, the world of computational neuroscience has worked for decades to build upon this discovery. Owing to the computational intensity of the HH model, sister models were derived that reduced the initial 4 dimensional model to 3 and 2 dimensional models. While these models are often very good at predicting the membrane potential and matching the spike trains of the HH model, there are important mathematical differences that are seldom investigated in subsequent literature. However, there is ultimately no general method for determining when a given model can be reduced. In this paper, we introduce various neuronal models, including the HH model and the Adaptive Integrate and Fire model (Gerstner, 2005), and investigate how Active Subspaces can be used to systematically eliminate parameters and explicitly demonstrate that two models are mathematically equivalent.

2 Introduction

2.1 Description of a Biological Neuron

While there are many different types of neurons in the body, they generally function in the same manner at a cellular level. Fundamentally, a neuron receives signals from other neurons in the form of electrical pulses using its dendrites. This signal is processed in the soma, located within the cell body of the neuron. If the signal has a particular strength and duration (specific to that particular neuron) this causes the neuron to fire. It subsequently sends an electrical pulse through its axon, which is then distributed throughout the network at the axon terminal.

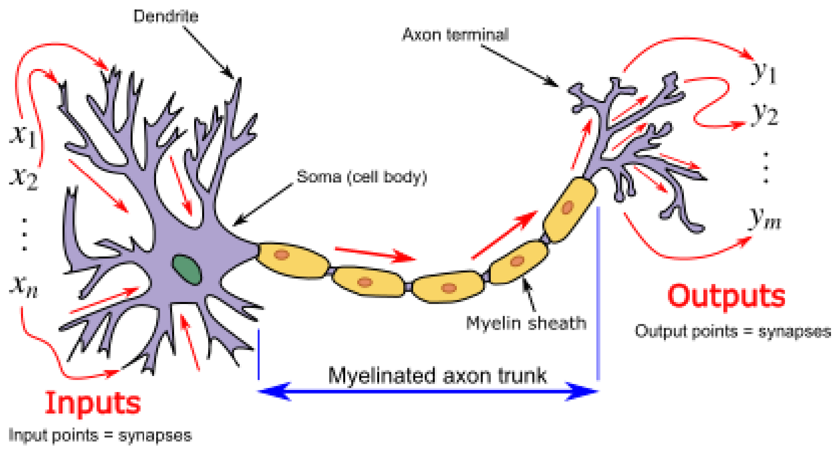

Figure 2.2.1 Simplified Neuronal Model (wikipedia)

In this manner, while a neuron is highly complex from a biological standpoint, it is easy to understand from a mathematical perspective. It receives some small input from its environment, and if that input crosses a given threshold value, a voltage spike is produced, which is followed by a brief refractory period.

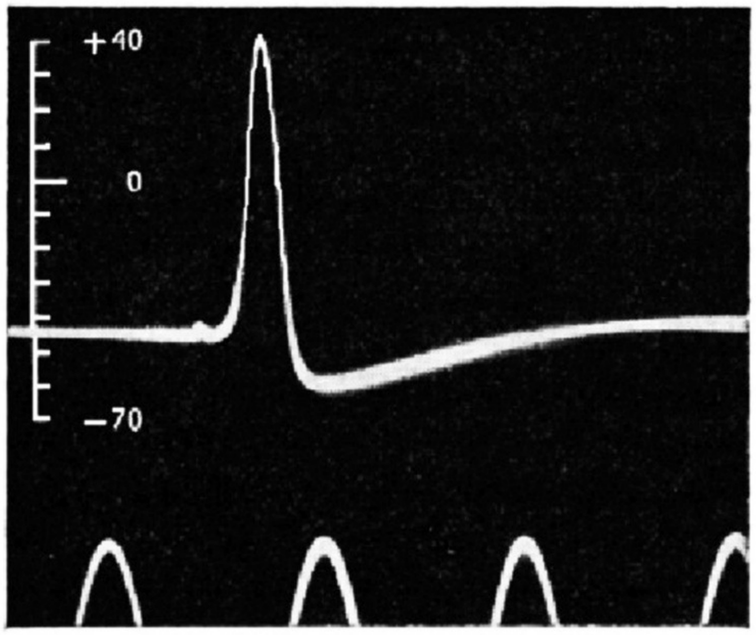

Figure 2.2.2 First intracellular recording of membrane action potentials in the squid giant axon as first measured by Hodgkin/Huxley in 1952

2.2 The LIF Neuron

The Leaky Integrate and Fire model, first introduced by Luis Lapicque in 1907 is in many ways the simplest mathematical description of the action potential of a neuron. Fundamentally, it is a one dimensional dynamical system equipped with a reset condition. Solving the system using Euler integration results in a discontinuous curve meant to represent the membrane action potential.

The dynamics of the LIF model are governed by the following equation:

$$C \cdot \frac{dv}{dt} = v_r - v + I_{ext} \qquad (2.2.1)$$

where I_{ext} is the external (in this case, constant) input current, v_r is the resting potential (mV) of the neuron, and C is the capacitance. Using Euler integration, we approximate a solution to the above equation given a discrete set of time steps by:

$$dv_i = (v_r - v + I_{ext})/C \qquad (2.2.2)$$

$$v_i = v_{i-1} + dv_i \cdot dt \qquad (2.2.3)$$

where dt is the length of each time step. Ultimately, we say that a spike in the neuron has occurred if the voltage across the membrane has surpassed a given threshold, $v_i \geq v_T$, and set v_i to some constant we call the reset potential, v_R. Thus, the relevant constants in this model are $v_r, v_R, v_T, C, I_{ext}$.

This system allows us to predict the membrane potential (mV) as a function of time. However, it is often the case that a neuron in a system of connected neurons is subjected to a 'noisy' input, arising from the firing of adjacent neurons. This 'noisy' input is modeled by a random variable $-1 \leq \zeta \leq 1$ with a gaussian probability distribution, multiplied by a volume constant, μ. If we revise the above model to include noise, the Euler integration process looks like:

$$dv_i = (v_r - v + I_{ext})/C \qquad (2.2.4)$$

$$v_i = v_{i-1} + dv_i \cdot dt + \sqrt{dt} \cdot \zeta_i \cdot \mu \qquad (2.2.5)$$

This allows us to model the firing rate, of the neuron when it is subjected to a noisy input. With input values as follows, we graph the firing rate of the neuron as a function of time alongside the membrane potential using python.

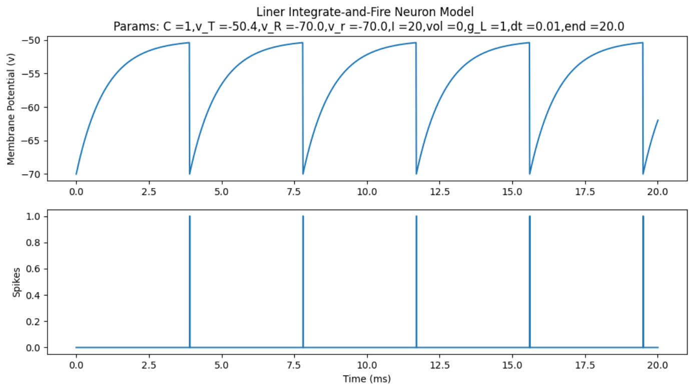

Figure 2.2.1 Spiking LIF model (no noise)

Plotting the system to include noise with $\mu = 5$ yields:

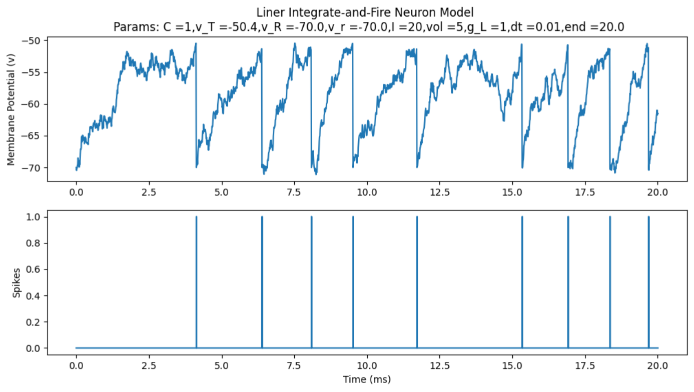

Figure 2.2.2 Spiking LIF model (with noise)

The Leaky Integrate and Fire model, while highly unsophisticated, has its place in computational neuroscience literature. While widely regarded as a toy model, the low number of parameters makes it relatively easy to code, and modeling behavior is highly inexpensive from a computational standpoint. For this reason, it is often used to model large networks of neurons, where it is necessary to compute the voltage trajectories for hundreds, or sometimes thousands of individual neurons simultaneously. However, the lack of parameters makes it impossible to model certain behaviors that can be accessed using other models, which we will discuss in subsequent sections.

2.3 EIF and QIF Models

While the leaky integrate and fire neuron is sufficient for many applications, it does not produce the spiking behavior characteristic of biological neurons. As seen in the plots above, the membrane potential increases ever more slowly (owing to the negative second derivative one obtains by differentiating equation 2.2.1) which is inconsistent with the voltage trace seen in figure 2.2.1. For this reason, alternative models were created to produce this spiking behavior while maintaining the computational simplicity characteristic of the LIF.

Ultimately, exponential/quadratic integrate and fire neurons are coded in much the same manner, to allow for constant input, noisy input, or a combination of both. The governing equations respectively are:

$$C \cdot \frac{dv}{dt} = -g_L(v - v_r) + f(v) + I_{ext} \qquad (2.3.1)$$

Where for the EIF:

$$f(v) = g_L \triangle T \exp(\frac{v - v_T}{\triangle T}) \qquad (2.3.2)$$

while for the QIF, we have that:

$$f(v) = (v - v_r)^2 \qquad (2.3.3)$$

In the EIF model, the constant $\triangle T$ is a time constant that changes the size of the effect of the exponential term in the EIF model. At low values of $\triangle T$, this equation is identical to the one that governs the dynamics of the LIF model. The quadratic integrate and fire model has the same number of parameters as the LIF, however the values of these parameters differ from that of the LIF significantly in some cases.

The parameter values are given in the figures below:

EIF:

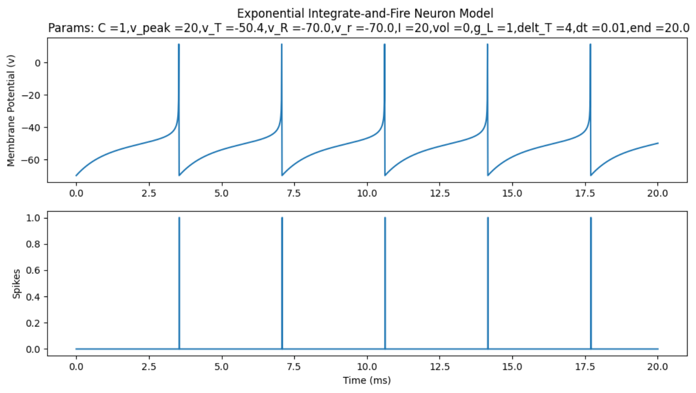

Figure 2.3.1 Spiking EIF model (no noise)

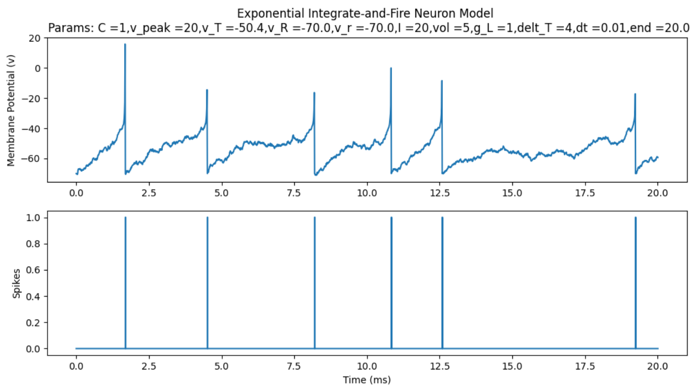

Figure 2.3.2 Spiking EIF model (with noise, $\mu = 5$)
QIF:

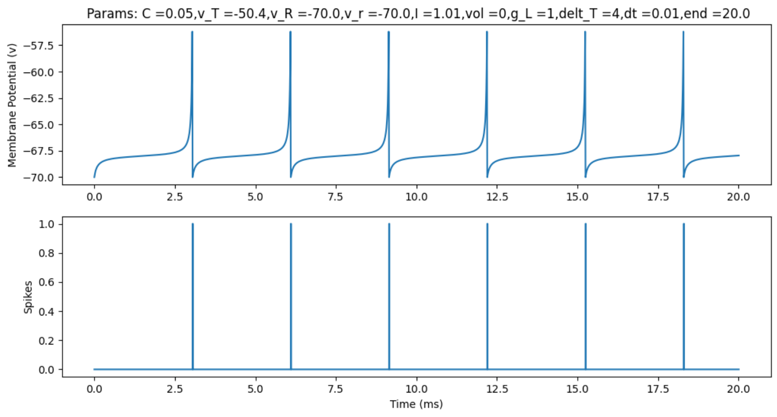

Figure 2.3.3 Spiking QIF model (no noise)
(With Noise)

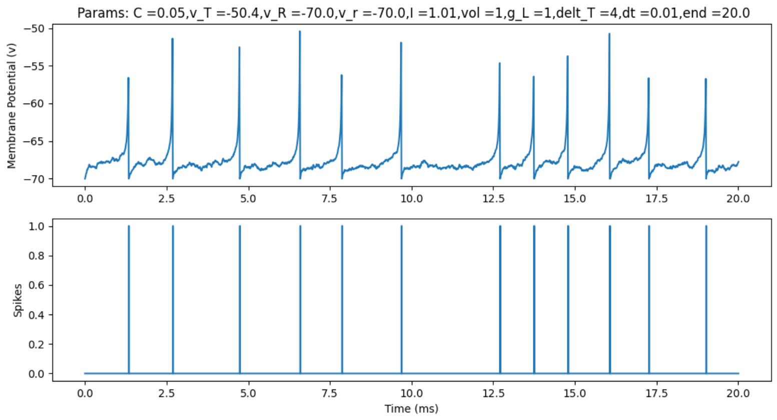

Figure 2.3.1 Spiking QIF model (with noise, $\mu = 1$)

2.4 Adaptive EIF

In the course of our research, we studied the paper that introduced an adapted version of the Exponential Integrate and Fire neuron (Gerschner 2003). This model was initially fit to the HH model in the above work, however it is derived from the models introduced in the above two sections. While the governing equations look much the same, there are some important differences. For instance, in the case of the above models, one can check that a constant input current high enough to cause the neuron to spike will cause it to fire indefinately at a constant frequency.

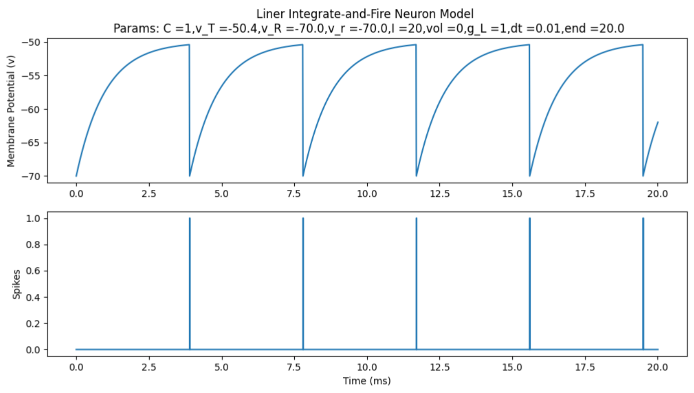

Figure 2.2.1

However, it has been determined through experimentation that a single neuron, subject to a high enough constant input current, will fire at a constant rate for some period of time, and then subsequently begin to fire unpredictably before returning to its resting potential. In other words, a constant input voltage is ignored by the neuron after a certain period of time. This is consistent with other biological observations and intuitive reasoning; our nervous systems react strongly to changes in our environment, because any change in our environment indicates the presence of a potential threat. However, any external input in ones environment, if constant, will be ignored by the nervous system after some period of time so long as said input does not appear to be hurting you.

Ultimately, the adapted EIF was developed in order to account for this behavior. It is a two dimensional system composed of two differential equations, given below:

$$\tau \frac{d\omega}{dt} = \alpha(v - v_r) - \omega \qquad (2.4.1)$$

$$C\frac{dv}{dt} = -g_L(v - v_r) + g_L \cdot \triangle T \cdot e^{(\frac{v-v_T}{\triangle T})} - \omega + I_{ext} \qquad (2.4.2)$$

where α is an adaptation constant and τ is the time constant associated with the adaptation current ω. Using Euler integration, we approximate a solution to these equations for a given a set of inputs and a discrete set of time steps by:

$$dw_i = (\alpha(v_{i-1} - v_r) - \omega_{i-1})/\tau \qquad (2.4.3)$$

$$\omega_i = \omega_{i-1} + d\omega_i \cdot dt \qquad (2.4.4)$$

$$dv_i = -g_L(v_{i-1} - v_r) + g_L \cdot \triangle T \cdot e^{(\frac{v_{i-1}-v_T}{\triangle T})} - \omega_{i-1} + I_{ext} \qquad (2.4.5)$$

$$v_i = v_{i-1} + dv_i \cdot dt + \sqrt{dt} \cdot \mu \cdot \zeta_i \qquad (2.4.6)$$

In our code, we make the stipulation that if $v_i \geq v_T$, we say a spike has occurred, and set $v_i = v_R$. Note above that if $\mu = 0$, and I_{ext} is large enough to cause a spike, the system exhibits the behavior described above for a neuron subjected to constant input. That is to say, it fires periodically for a while before firing aperiodically, and then returning to its resting state.

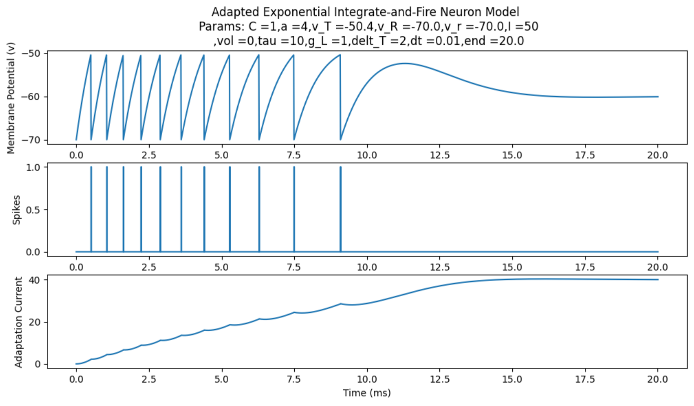

Figure 2.4.2 Adaptive EIF model (no noise)

Any nonzero value of μ introduces noisy input into the system. In this case, the neuron will spike aperiodically for some length of time, and depending on the volume of the noise, may or may not return to a resting state.

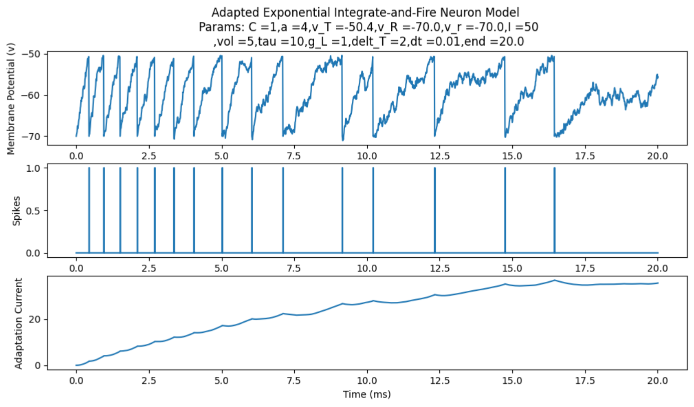

Figure 2.4.3 Adaptive EIF model (with noise, $\mu = 5$)

2.5 The Hodgkin Huxley Model

A main goal of the project is to demonstrate model equivalence between the HH model and the aEIF model. In 2005, Wulfram Gerstner showed that this was possible by fitting his adapted version of the EIF to the HH model. In his paper, he was able to get the spike times of the models to match given the same noisy input. He further claims that in principle, this adapted model may be fit to any repetitive spiking neuronal model, such as the Wang/Buzaki model (shown below). Ultimately, we seek a method of understanding the apparent relations between the parameters of the aEIF model and the HH model that make them fundamentally equivalent. Below, we begin by describing the dynamics that govern the HH model, and the various similarities it shares with the models above.

The Hodgkin Huxley model of a neuron is a 4 dimensional system governed a shortlist of differential equations. In vivo, a neuron is influenced by many different inputs. As described above, it receives noisy input from other neurons, and this is modeled using the normally distributed random variable ζ. It then receives some constant input current, I_{ext} and these currents both contribute to the membrane potential of the neuron given by v. However, voltage across the membrane is also affected by other inputs, most notably voltage gated potassium/sodium channels. Inputs across these channels is modeled by the variables n, m and h. In particular, the voltage gated potassium channel is modeled by potassium activation variable n, while the voltage gated sodium channel is modeled by the sodium activation/inactivation variables m, h respectively. The dynamics of each gating variable are determined by the same equation:

$$\frac{dx}{dt} = \alpha_x(v) \cdot (1 - x) - \beta_x(v) \cdot x \qquad (2.5.1)$$

for $x = n, m, h$ where α_x, β_x are adaptation constants determined by the current membrane potential at any given time. Unfortunately, these constants (really functions of the membrane potential) are all distinct for each variable, and are given below:

$$\alpha_n(v) = \frac{0.01(v + 55)}{1 - e^{-0.1(v+55)}} \qquad (2.5.2)$$

$$\beta_n(v) = 0.125e^{(-0.0125(v+65))} \qquad (2.5.3)$$

$$\alpha_m(v) = \frac{0.1(v + 40)}{1 - e^{-0.1(v+40)}} \qquad (2.5.4)$$

$$\beta_m(v) = 4e^{(-0.0556(v+65))} \qquad (2.5.5)$$

$$\alpha_h(v) = 0.07e^{(-0.05(v+65))} \qquad (2.5.6)$$

$$\beta_h(v) = \frac{1}{1 + e^{-0.1(v+35)}} \qquad (2.5.7)$$

Finally, the membrane potential is given by the following differential equation:

$$C\frac{dv}{dt} = I_{ext} - g_{Na}m^3h(v - E_{Na})) - g_K n^4(v - E_K) - g_L(v - E_L) \qquad (2.5.8)$$

Ultimately, when updating the value of the membrane potential for discrete time steps (using Euler integration) we add:

$$v_i = v_{i-1} + dv_i \cdot dt + \sqrt(dt) \cdot \mu \cdot \zeta_i \qquad (2.5.9)$$

to allow for noisy input. The plot of the membrane potential for an HH neuron with noisy input is given below.

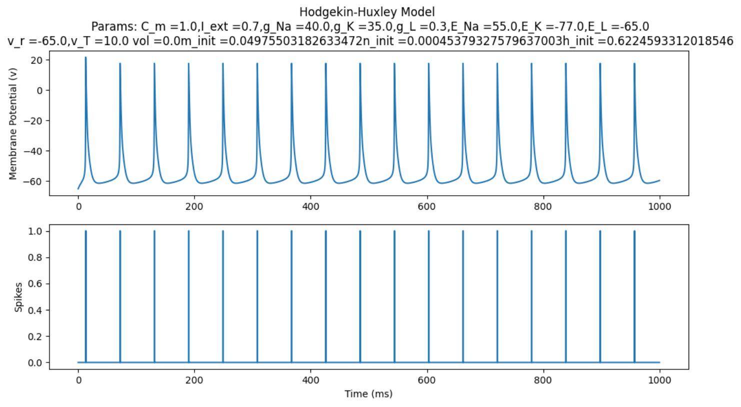

Figure 2.5.1 Hodgkin/Huxley model, simulated for 1000 ms (without noise)

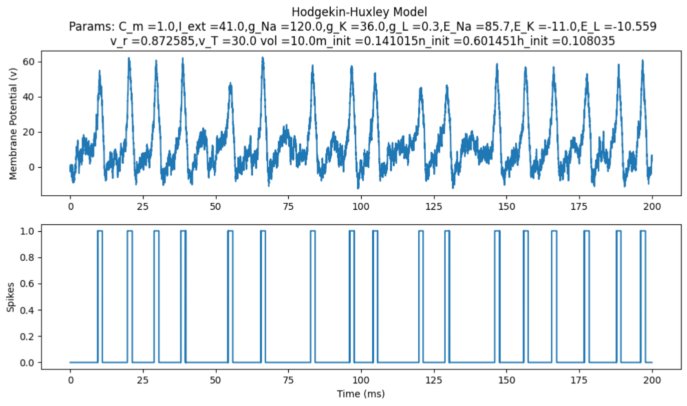

Figure 2.5.2 Hodgkin/Huxley model, simulated for 200 ms (with noise, $\mu = 10$). Note that the parameter values in the plots above are taken different. The parameter values in figure 2.5.1 were taken from (Dunworth) while the parameter values in figure 2.5.2 were taken directly from Gerstner's textbook. When noise is removed from the system, the resemblance to figure 2.1.1 can be more easily identified, as seen in the plot below.

(ultimately, I need to regenerate these plots so that I am using the same parameters for the HH model throughout)

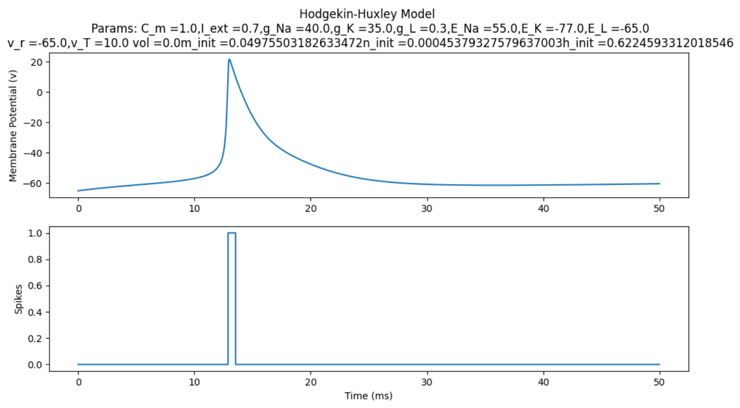

Figure 2.5.3 Hodgkin/Huxley model, simulated for 50 ms (without noise)

One of the many upsides to using this model is its relative (compared to other, newer models) simplicity combined with its capability to model complex behavior. For example, many pufferfish species such as Lagocephalis contain a potent neurotoxin known as tetrodetoxin, which is emmitted during a stinging event. This neurotoxin works by causing the deactivation of sodium channels, which inhibits neurons from firing. In the victim, it can lead to paralysis or death. The effect of the neurotoxin can be demonstrated using the HH model. Note that in figure 2.5.1, the sodium leak capacitence, given by the variable g_K, is a constant. In the context of the pufferfish, we change g_K to a time dependent variable, for instance:

$$g_K(t) = g_K(1 - \frac{t}{r}) \qquad (2.5.10)$$

where the variable r represents the speed at which the neurotoxin takes effect in the individual neuron. The result of this change is seen in the plot below, where a neuron that would typically continue to fire stops firing.

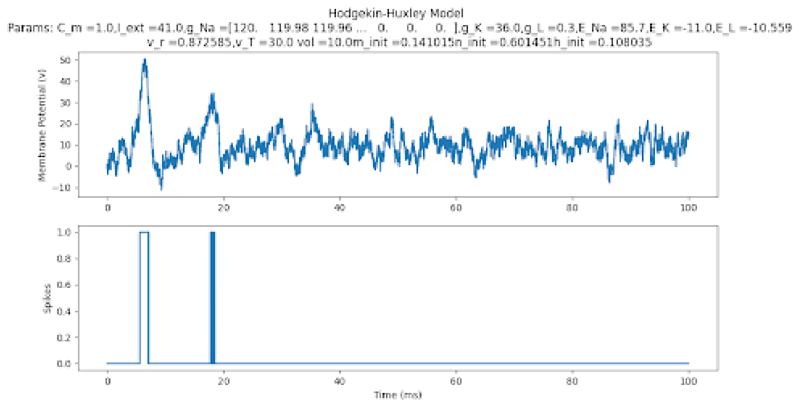

Figure 2.5.4 Hodgkin/Huxley model (noisy input) with $g_K(t) = g_K(1 - \frac{t}{10})$

While the HH model is capable of modeling complex behaviors as seen above, it is unfortunately very computationally expensive to do so. With four time variant inputs, each with their own dynamics, this system is not only very inefficient to work with at a large network scale, but parameter values can be sensitive to small changes, making it difficult to code. For this reason, it is often useful to eliminate as many insensitive parameters and variables as possible. Below, we show a direct example of a relatively simple variable reduction.

3 Methods

3.1 Model Similarity

Given that many of the plots above that depict the membrane potential as a function of time appear different for different models, it is important to demonstrate model similarity. The approaches demonstrated below are frequently used by researchers to show that a given reduction of an existing model is indeed valid. While the previous simulations may convince some, a more principled approach is to show that the different models match in a statistical sense. Ideally, one would like to be able to match spike trains between models exactly, though this goal is more complicated than it may initially appear. To begin, we can match the models at the level of the first marginal statistic, the firing rate.

Several different measures were used to evaluate model similarity. While action potential curves may appear different, these measures are used to show that the models are qualitatively similar. One of the most informative values given a neuronal model and an input current is the firing rate, which tells you how quickly the neuron is spiking. This quantity is simply obtained by counting the number of spikes in a time interval, and dividing by the length of that time interval. The plot below was obtained by calculating the firing rate for each model as a function of a constant input current.

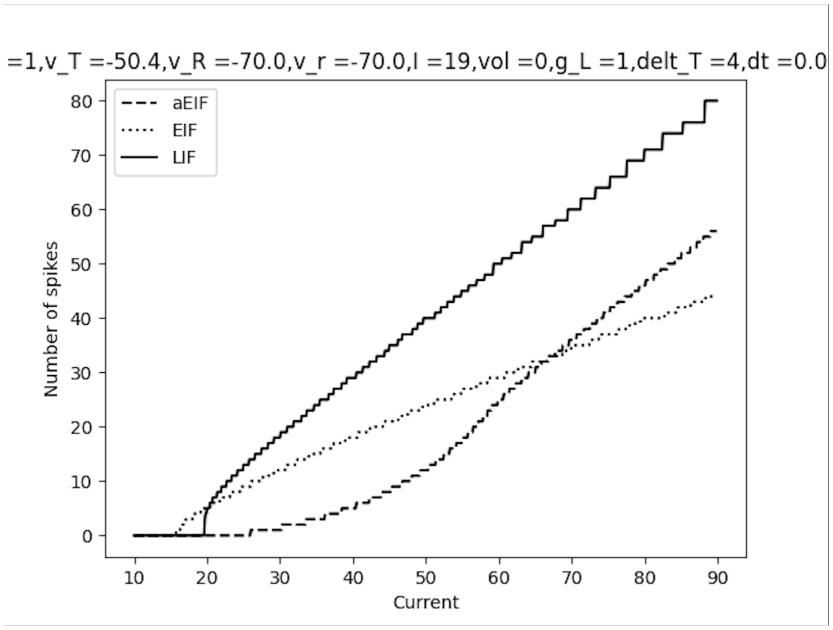

Figure 3.1.1: Plot of the firing rate curve for the aEIF, EIF, and LIF models, generated by calculating the firing rate for static input current ranging from 0 to 90 (units) for 90 different values.

Noting the shape of the aEIF curve, consider the plot below of the firing rate curve for the Hodgkin Huxley model with the parameter values in Gerstner's textbook:

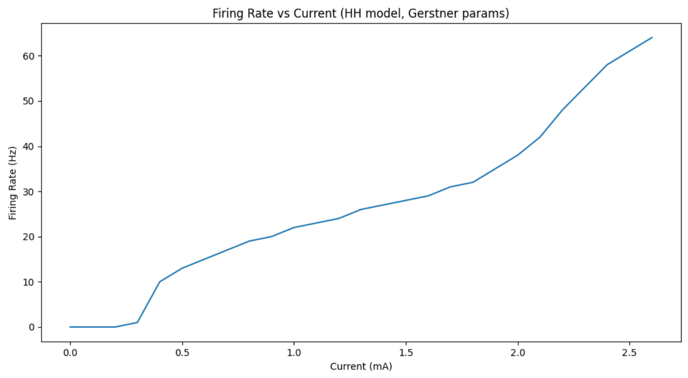

Figure 3.1.2: Firing rate for the HH model, generated by calculating the firing rate for static input current ranging from 0 to 2.5 (units) for 50 different values.

Here, we observe that the firing rate curve for these two models is practically identical. However, these plots could not be overlayed due to a scaling issue: the parameter values in Gerstner's textbook differ from those used to fit the aEIF model, and thus, the input current is an order of magnitude off from where it should be. Correcting this is unfortunately not as simple as changing all parameter values by an order of magnitude, as parameter fitting can be quite challenging, as we will see subsequently. However, the shape of these curves indicate that change in the neuronal response to different input currents is consistent across models.

If the neuronal models are in fact, describing the same thing, one should be able to input the same noise into each model, and obtain the same result each time. Using the parameter values for the HH model above, we applied the parameter fitting techniques used in Gerstner 2005 show proof of concept that the parameters of the aEIF model can indeed be fit to the HH model.

First, we note that in both cases, equations 2.5.8 and 2.4.2 for the HH and aEIF models respectively can be written:

$$C\frac{dv}{dt} = I_{ext} - F(v,n,m,h) \qquad (3.1.1)$$

$$C\frac{dv}{dt} = I_{ext} - w - f(v) \qquad (3.1.2)$$

Hence, if we can fit the parameters of the aEIF model to match $f(v)$ to the shape of the curve $F(v,n,m,h)$, it makes sense to say that we have found a valid approximation of the HH system. First, in order to understand the dynamics of the function f, we graph it for different values of $\triangle T$. Changing will ultimately alter the shape of each spike. Note that in the limit as $\triangle T \to 0$ we find that:

$$g_L \cdot \triangle T \cdot e^{(\frac{v-v_T}{\triangle T})} \to 0 \qquad (3.1.3)$$

$$\Rightarrow -g_L(v-v_r) + g_L \cdot \triangle T \cdot e^{(\frac{v-v_T}{\triangle T})} \to -g_L(v-r_r) \qquad (3.1.4)$$

so that the aEIF model resembles the LIF model (with adaptation) in the limit. The plots of $f(v)$ for different values of $\triangle T$ are given below.

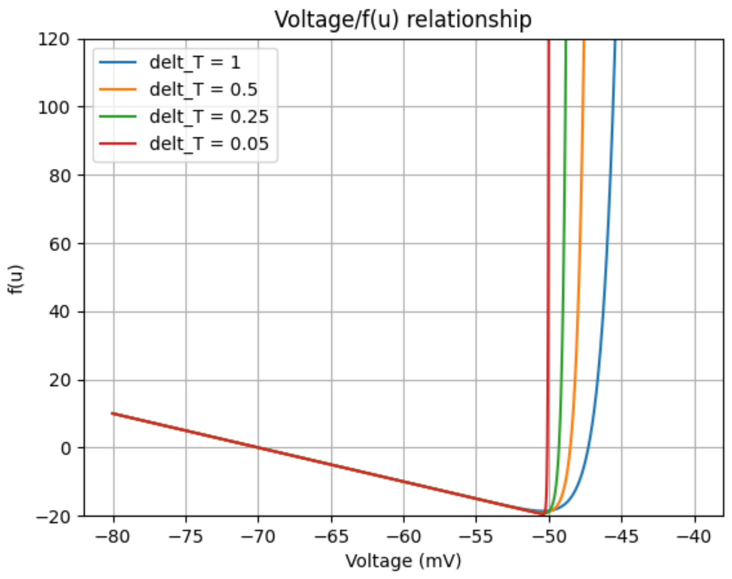

Figure 3.1.3: Plot of the $f(v)$ curve for values of v ranging from -80 to -40mV.

Note that for smaller values of $\triangle T$, the exponential relationship between the membrane potential and the function $f(v, w)$ increases at a faster rate for values of $v \geq -50$. Thus, we can, in principle, use an exponential fitting algorithm to fit these curves to that of $F(v, n, m, h)$.

Note that above, we have plotted $f(v)$ not as a function of time, but as a function of v. However, in practice, F is a function with multi-dimensional input. Thus, holding the values of n, m and h constant and simply increasing the value of v linearly does not give us very much information about the system as a whole. In practice, v is a variable with time dependent dynamics, and it does not increase linearly in either system. Nonetheless, we seek a method of plotting $f(v)$ and $F(v, n, m, h)$ alongside one another so that a fit can be achieved. In order to do this, we treat F and f as time dependent variables that evolve with each system respectively, and plot F and f alongside each other as functions of time.

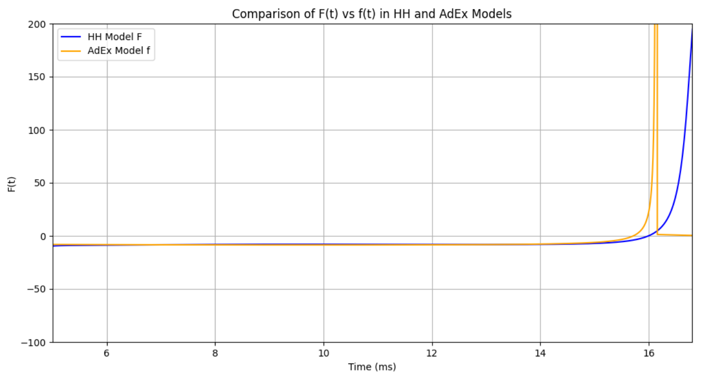

Figure 3.1.4: Plot of $F(v, n, m, h)$ and $f(v)$ as the aEIF and HH models evolve over time from 4 to 16ms.

Subsequently, we defined an exponential function with the equation:

$$y(t) = abe^{\frac{r(t)}{b}} + c \qquad (3.1.5)$$

Note here the resemblance of the above function to $f(v)$. Since $F(t)$ is plotted as a function of time, the input parameter for the exponential function needed to also be the time parameter in order to scipy's optimization function to work. However, $f(v)$ is a function of v not of time, although v changes exponentially over time. Therefore, we note that the function $f(v)$ sees a spike at approximately the time that the aEIF model sees a spike (i.e., when $v \geq v_T$). At that time, v increases by approximately 20mV over a period of one millisecond. Therefore, we make the approximation:

$$v - v_T = r(t) = 20(t - 14) \qquad (3.1.6)$$

since the first spike in the $f(v)$ occurs at approximately 14 milliseconds. With this approximation, we were able to use scipy's exponential fitting algorithm to fit the curve to the data points above cooresponding to $F(t)$.

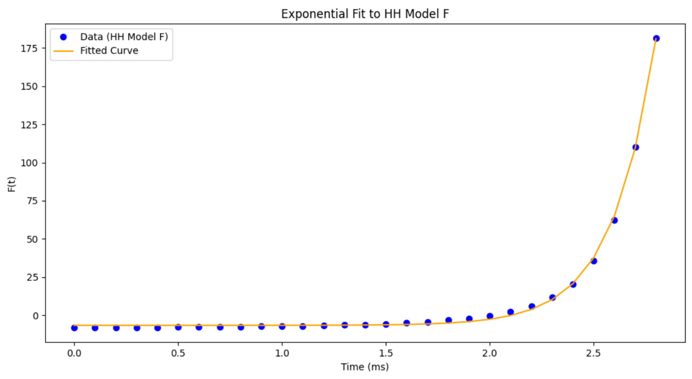

Figure 3.1.5: Plot of $y(t)$ fit to the data points cooresponding to $F(v, n, m, h)$ from figure (3.1.4).

The fitted parameter values of a, b and c were given by $a = 6.2$, $b = 4.1$, and $c = -6.5$. Inputting the fitted a and b parameters in for g_L and $\triangle T$ did not quite work, most likely due to the fact that $f(v)$ has a linear component as well as an exponential component. However, after increasing the value of g_L to 25, making no changes to the other variables, we were able to achieve the following result:

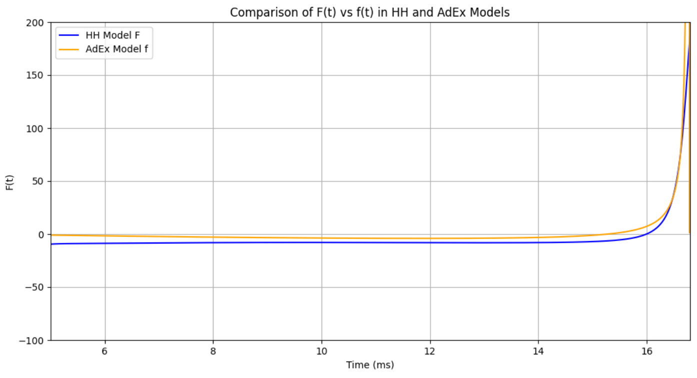

Figure 3.1.6: Plot of $F(v, n, m, h)$ and $f(v)$ as the aEIF and HH models evolve over time from 4 to 16ms using the new parameter values of g_L and $\triangle T$ found above.

We subsequently plotted the voltage trace of the aEIF and HH models together with these parameter values for an input current of $10\mu a^2$. Afteradjusting the starting voltage for the aEIF model to -40 to account for the longer burn in period of the aEIF model, suprisingly, this happened:

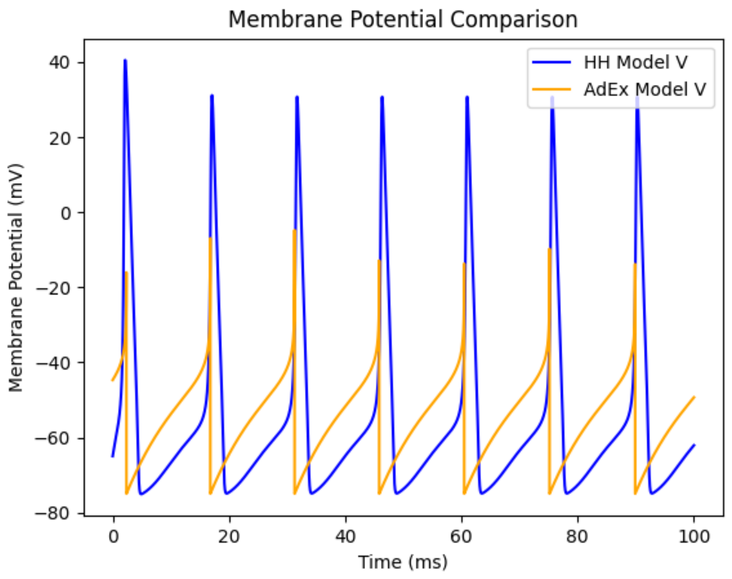

Figure 3.1.7: Plot of the HH and aEIF voltage traces, simulated for 100ms, subject to the same static input current.

Plotting both systems with noise, we were able to match the voltage trace of the aEIF model to the voltage trace of the HH model more or less spike for spike:

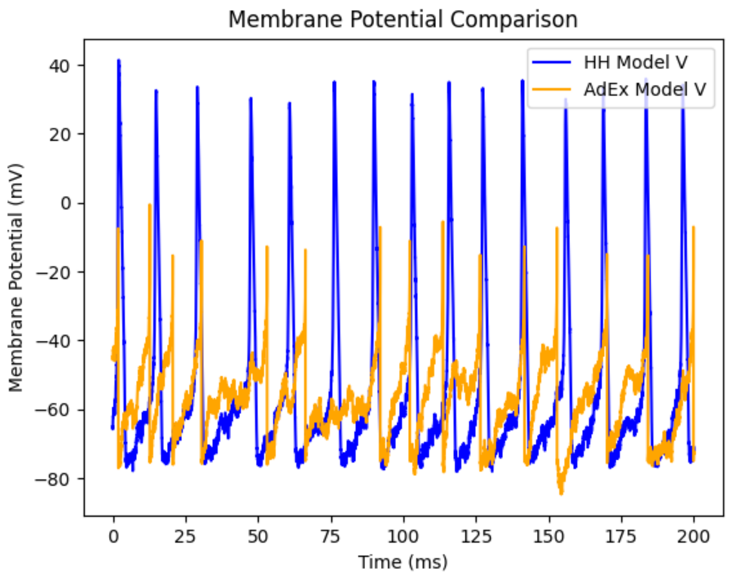

Figure 3.1.7: Plot of the HH and aEIF voltage traces, simulated for 200ms, subject to the same noisy input current.

Note here that the aEIF model occasionally misses some spikes that occur in the HH model, spikes when it should not, or the timing of certain spikes is slightly off. Gerstner noted the presence of these 'missed spikes' in his paper, although there is only one missed spike and one extra spike in the above plot after simulating the models for a period of 200ms.

3.2 Parameter Reduction

We showed above that the parameters of the aEIF model can be fit to match the behavior of the HH model. Mathematically, it is important to understand why we can do this. First, in considering equation 2.5.8 of the HH model, one should note that unlike the aEIF model, the HH model need not be reset at a given value of the input current. This is due to the fact that the sodium and potassium gating activation variables inhibit the growth of the membrane potential at a certain threshold value, and unlike the aEIF model, there is no positive term in the equation for $C\frac{dv}{dt}$. Hence, we may write equation 2.5.8 in the following manner:

$$C\frac{dv}{dt} = I_{ext} - g_L(v - E_L) - (I_{Na} + I_K) \qquad (2.5.8)$$

Note that for the parameter values of the HH model given in figure 2.5.1, since the reset potential $v_r = -65$, the term $(v - E_K)$ is positive, while the term $(v - E_{Na})$ may be either positive or negative. This means that sodium activation may be excitatory or inhibitory depending on the current state of the membrane potential. Therefore, one can conceive of it having a positive component as well as a negative component. Taking note of these features, we consider again equation 2.4.2:

$$C\frac{dv}{dt} = -g_L(v - v_r) + g_L \cdot \triangle T \cdot e^{(\frac{v-v_T}{\triangle T})} - \omega + I_{ext} \qquad (2.4.2)$$

At least in principle, based on the above similarities, one may be inclined to make the approximation:

$$(I_{Na} + I_K) = g_L \cdot \triangle T \cdot e^{(\frac{v-v_T}{\triangle T})} - \omega \qquad (3.2.1)$$

It should be noted that such an approximation is completely invalid unless one has access to the correct parameter values, as demonstrated above. However, it shows how one could theoretically reduce the dimension of the system using fewer variables. For a more concrete and straightforward example of parameter reduction, we turn to a 3-dimensional adaptation of the HH model caled the Wang/Busaki model.

The Wang-Busáki model is a refinement of the HH model that takes the 4 dimensional dynamical system proposed by HH and reduces it to a 3 dimensional model. It does this by taking the gating variable m for sodium activation and reducing it to a function of the membrane potential v. It uses the same general equation for the membrane potential:

$$C_m\frac{dV}{dt} = -(I_{Na} + I_K + I_L) + I_{ext} \qquad (3.2.2)$$

However, the Wang-Busáki model makes the following change; note that in the HH model:

$$I_{Na}(v) = g_{Na}m^3h(v - E_{Na}) \qquad (3.2.3)$$

While in the WB model:

$$I_{Na} = g_{Na}m_\infty(v)^3h(v - E_{Na}) \qquad (3.2.4)$$

where:

$$m_\infty(v) = \frac{\alpha_m(v)}{\alpha_m(v) + \beta_m(v)} \qquad (3.2.5)$$

As it turns out, this sort of reduction is common. Potassium and sodium gating variables alike can be reduced into functions of voltage, as voltage affects these quantities in a sub linear manner, and can be modeled using a rational function. The gating variable m simply responds fastest to a change in the membrane potential, and so it makes sense to reduce the model using this variable first. To demonstrate how m is affected by a change in voltage, a plot of the membrane potential for the HH model is given alongside a plot of the gating variable.

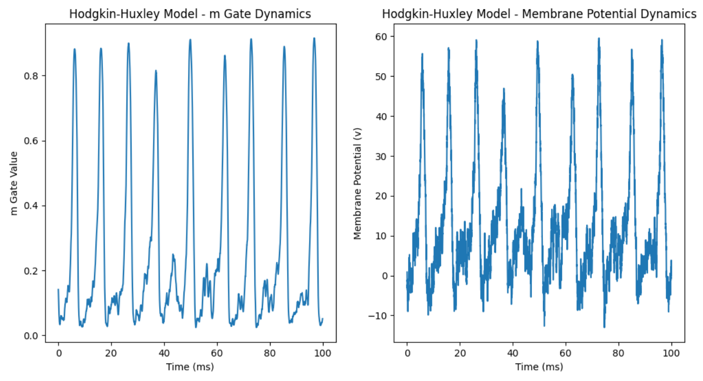

Figure 3.2.1: Plot of the voltage trace of the HH model simulated for 100ms and subjected to noisy input alongside a plot of the gating variable m.

To show that the approximation of the gating variable $m = m_\infty(v)$ is valid, we plot the trajectories of m and $m_\infty(v)$ as the HH system evolves over the course of 200ms:

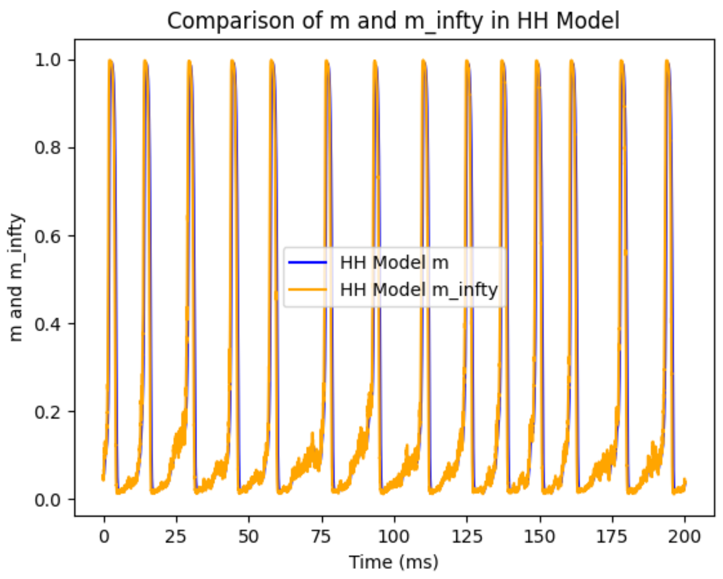

Figure 3.2.2: Plot the gating variable m and $m_\infty(v)$ in the HH model as the system is simulated for 200ms, subjected to noisy input.

An example plot of a WB neuron, subjected to noisy input, is shown below alongside the plot of the HH neuron subjected to the same input:

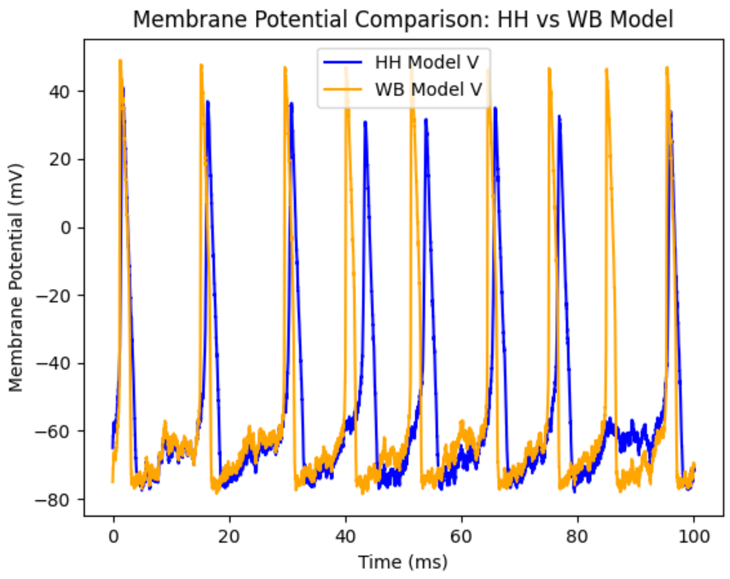

Figure 3.2.2: Plot the voltage traces of the HH and WB models subjected to the same noisy input and simulated for 100ms.

As with the plot of the aEIF model, the WB model occasionally misses spikes as well. In this case, we have not followed the procedures to fit the parameters of the WB model to those of the HH model, as that would go beyond the scope of what we are trying to demonstrate here. The point is that in some cases, you can reduce the dimension of these models, and after correctly fitting the parameters, one can obtain a system of equations that exhibits the same behavior when subjected to the same input. We finally seek a general method for determining when there is a relation on two or more model variables that can be exploited, and if so, what the nature of that relationship may be. We proceed to do that using active subspaces.

3.3 Active Subspaces:

The aim of this project is to show that certain models can be reduced into less complicated models with less dimensions. There are many frameworks and methods to which to approach this problem, including the MBAM method. However, while the method itself is very robust in generality, it suffers from being highly computationally inefficient. Therefore, we seek other more refined numerical methods for reducing the models above. As it turns out, there is this really nice, linear-algebra-adjacent way to think about all of this called active subspaces. The method of *Active Subspaces* seeks to reduce the number of parameters needed to approximate a quantity of interest. An active subspace is a region in parameter space throughout which the output of the model is highly sensitive to changes in initial parameter values. An inactive subspace is a region where you see mostly the same output, regardless of your initial input. Geometrically, this method is essentially the linear algebra equivalent of the manifold boundary approximation method. It finds sensitive and insensitive directions in parameter space, and these directions define a plane, which is a local approximation to an arbitrary manifold. Fundamentally, it speaks to the *identifiability* of the model (which parameter combinations can be uniquely identified by the output of a given model).

Consider a system of ODE's with the following form:

$$x' = w(x, t, \theta) \qquad (3.3.1)$$

$$y = v(x, \theta) \qquad (3.3.2)$$

Here, t is time, w, v are functions, θ is the input vector, x is the unobserved state vector, y is the observed output vector.

Let f denote a map from the space of model parameters to the model output: $f : \theta \in^n \to^m$.

A model is said to be identifiable if it can be uniquely determined from model output. A model is identifiable if all of its parameters are identifiable. If a parameter cannot be solved for, it may be an insensitive parameter, or it could be part of an identifiable parameter combination. That is to say, either the parameter simply does not matter very much in the context of the qualitative behavior of the model, or there may be some relation whereby that parameter can be expressed in terms of the other parameters in the model.

For example, consider the system below.

$$y = (m_1 + m_2)x + b \qquad (3.3.3)$$

Here, with (x, y) pairs as the observed output, b is identifiable, and while (m_1+m_2) is an identifiable quantity, the parameters m_1, m_2 are not individually.

In this case, the parameter b exhibits *global gdentifiability.* A parameter θ_i in θ is structurally globally identifiable if for almost all values of the parameter θ_i^*, the observation of an output value $y = y^*$ uniquely determines the value of $\theta_i = \theta_i^*$ if only one value of θ_i could have resulted in the observation that $y = y^*$. A parameter is said to be locally identifiable if there are a finite number of parameter values that generate the observed output. Note here that the quantify $(m_1 + m_2)$ is also structurally globally identifiable, since any value of the output y specifies a unique line in 2 dimensions expressing the possible values of m_1 and m_2

One way to determine how to reduce your parameter space is to fix them one at a time and solve for the remaining parameters until you can identify the sloppy ones. This is computationally expensive. However, one could also attempt to reparameterize the model in terms of its identifiable parameter combinations. This is called *identifiable reparameterization.* For example, in the model above, one could set $m = (m_1 + m_2)$ so that now the model has a set of outputs (x, y) identifiable from m, b. In order to determine when this can be done, we use something called the Fischer Information Matrix (FIM).

3.4 FIM:

Suppose a system of equations gives m observed quantities and n parameters, with θ^* representing any given set of parameter values, and $f(\theta^*)$ representing the model output cooresponding that set of parameter values.

We consider a matrix with entries

$$F_{i,j}(f:\theta) = [\sum_{k=1}^{m}(\frac{\partial f_k}{\partial \theta_i})(\frac{\partial f_k}{\partial \theta_j})]_{i,j} \qquad (3.4.1)$$

Note here that:

$$F(f:\theta) = Df(\theta) * (Df(\theta))^T \qquad (3.4.2)$$

This is called the sensitivity Fischer Information Matrix (which we denote sFIM in subsequent sections). Note that it is a real symmetric matrix, and is therefore diagonalizable. Finding the eigenvalue decomposition of this matrix and the cooresponding eigenvalues can help to identify sensitive and insensitive directions in parameter space. In spirit, the FIM can be understood as the square of the derivitive matrix of the output in terms of each of the parameter values. Therefore, each entry gives you information about which linear directions in parameter space lead to the largest change in the output. It is said that the FIM is only capable of determining linear identifiable parameter combinations, however, we propose a method to find nonlinear identifiable combinations in subsequent sections.

4 Results:

4.1 The FIM in a Toy Model:

We used the FIM in two settings to determine if it is capable of finding any sort of relation between the variables/parameters in the model, and if so, what the nature of that relation might be. In constructing a function $f(\theta)$ for a given set of parameter values, we seek some output that adequately describes the behavior of the model for which it is possible to take the derivative in terms of each parameter. Ultimately, we chose the coefficients of the fast fourier transform for our output. In order to demonstrate that this output is capable of finding identifiable parameter combinations using the FIM, we first implemented it on a toy model. Consider the following dynamical system given by:

$$\frac{dx}{dt} = (m_1 + m_2)y \qquad (4.1.1)$$

$$\frac{dy}{dt} = x + k \qquad (4.1.2)$$

with parameters m_1, m_2, k, x_0, y_0 where x_0, y_0 are the initial values of x and y respectively. Plotting the system over time yeilds:

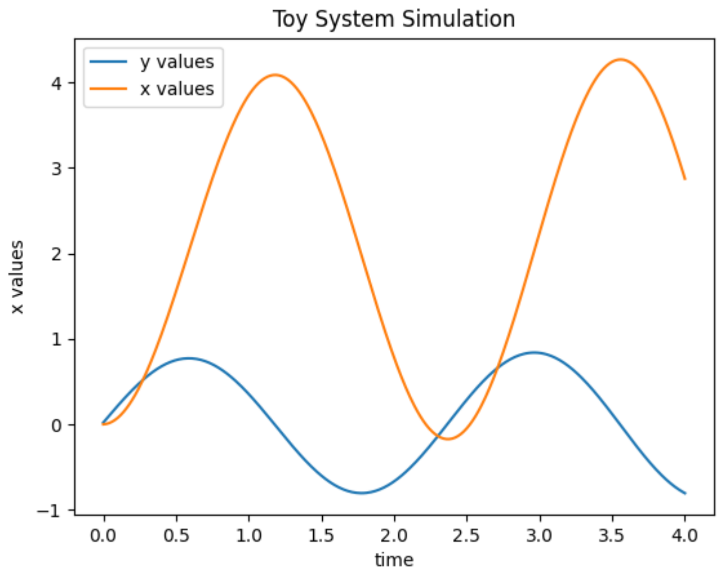

Figure 4.1.1: Plot of the trace of x and y as the system is simulated over 4s.

Subsequently, we let $\theta = (m_1, m_2, k, x_0, y_0$, and we define the function:

$$f(\theta) = [X_j = |\sum_{n=1}^{N} x_n e^{-i\frac{2\pi kn}{N}}|]_{j=1}^{10} \qquad (4.1.3)$$

where the X_j is the magnitude of the complex number given by the j^{th} pair of coefficients (which outputs a vector of length 10 for the first 10 harmonics) cooresponding to the FFT of the y curve simulated for 4ms, or $N = \frac{4}{0.01} = 400$ time steps. We then take the FIM of the output function $f(\theta^*)$ for a given set of parameter values θ^* by approximating the derivative using secant approximation and calculating:

$$F(f : \theta^*) = Df(\theta^*) \cdot (Df(\theta^*)^T) \qquad (4.1.4)$$

We then use the linear algebra 'eigh' function of numpy to get the eigenvalues and eigenvectors cooresponding to the FIM matrix. At the values of the parameters described above, we use bargraphs to illustrate the different directions in parameter space indicated by each eigenvector.

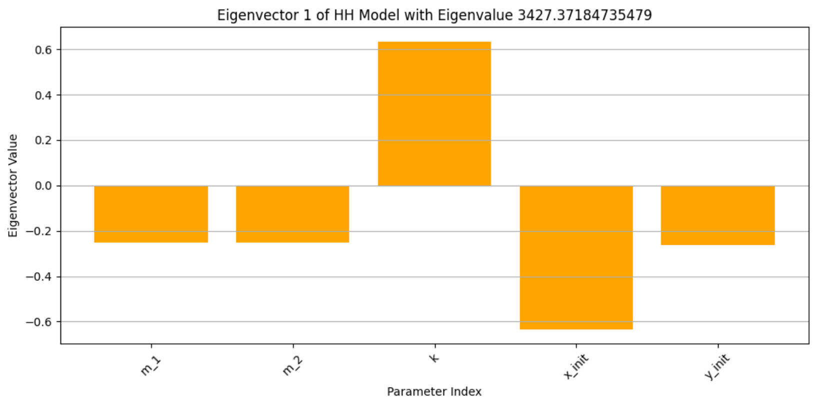

Figure 4.1.2

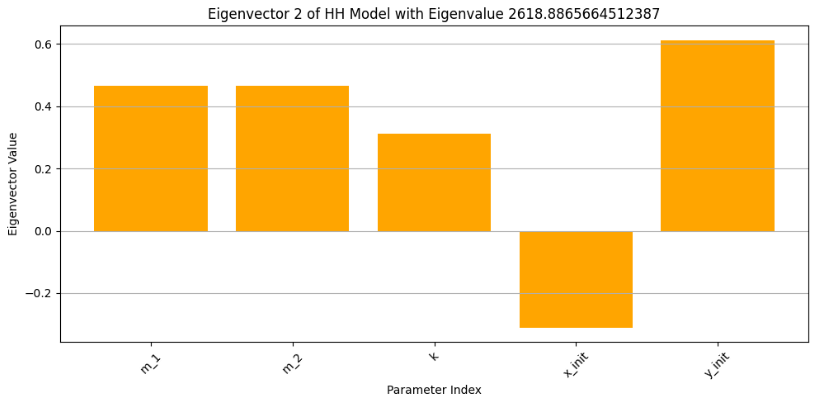

Figure 4.1.3

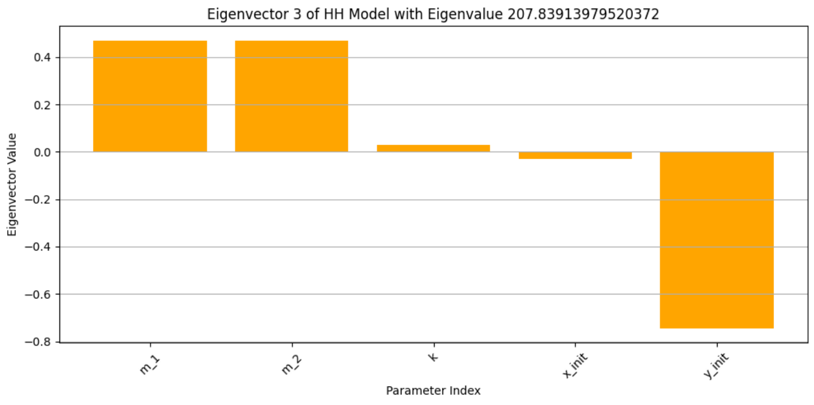

Figure 4.1.4

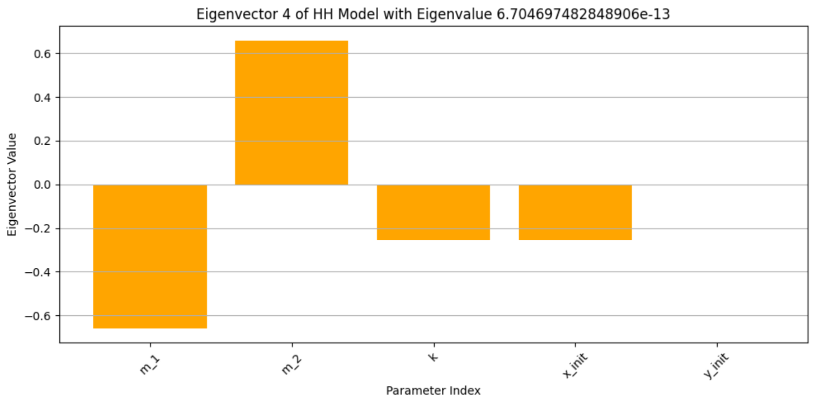

Figure 4.1.5

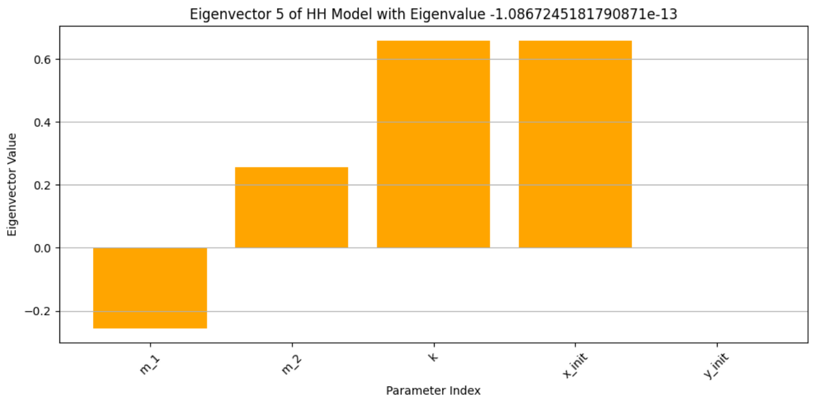

Figure 4.1.6

We now demonstrate how this data can be used to uncover the identifiable parameter combination $(m_1 + m_2)$. Note that the first 3 eigenvectors describe sensitive directions in parameter space. Therefore, adding these eigenvevtors to the input parameter vector θ^* will yeild some nonzeo change in the output of the function f. Alternatively, the final 2 eigenvectors coorespond to insensitive directions in parameter space. Adding these vectors to the input parameter vector θ^* should theoretically yield no change in the output of the model. Therefore, we can calculate the initial FIM, and take, for instance, a small step in the direction of the 4^{th} eigenvector (which should yeild no change in the model) and recalculate the FIM at the new parameter values. Iterating this process and keeping track of the parameter values will yeild a curve in parameter space on the surface of output manifold cooresponding to zero change. Hence, we can graph these parameters as functions of one another in order to determine a relation on the variables themselves.

Beginning at the parameter values above, we apply this method to the model, and plot the subsequent values of m_1 and m_2, which yeilds the following curve:

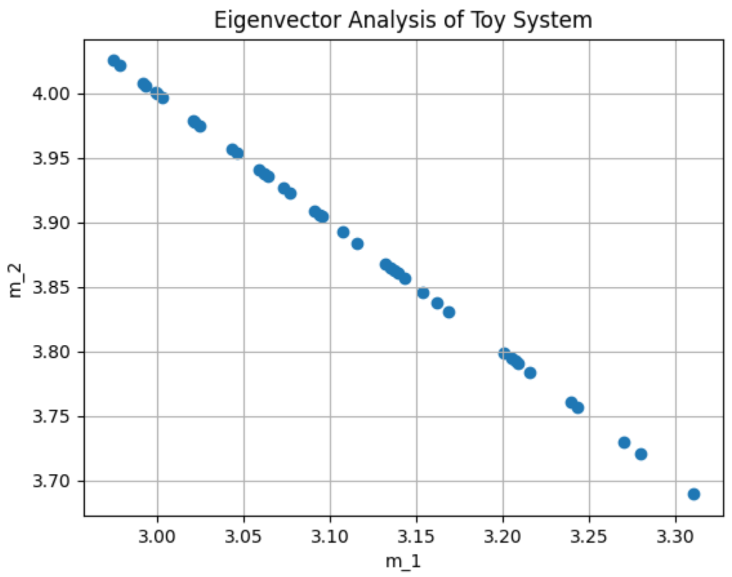

Figure 4.1.7: Scatter plot of the values of m_1 and m_2 generated using the method described above.

Note that the line of best fit for the above set of data points cooresponds to the line parameterized by:

$$r(t) = t[-1, 1] + [m_1^0, m_2^0] \qquad (4.1.5)$$

Noting that this line cooresponds to zero change in the output of the model, we deduce that the direction perpendicular to this curve, $[1, 1]$ cooresponds to some nonzero change in the output of the model. Hence, we can reduce the number of parameters in the system by letting $\alpha = m_1 + m_2$, which uncovers precisely the parameter reduction that we were seeking.

4.2 The FIM in the HH model using FFT output:

With sufficient proof that the above method can be used to find identifiable parameter combinations in the context of a similar (yet simpler) model, we proceed by applying this method to the Hodgekin/Huxley model. First, for a given set of parameter values:

$$\theta = (C_m,\ g_{Na},\ g_K,\ g_L,\ E_{Na},\ E_K,\ E_L) \qquad (4.2.1)$$

we consider the system given by:

$$f(\theta) = [X_k = |\sum_{n=1}^{N} x_n e^{-i\frac{2\pi kn}{N}}|]_{k=0}^{K} \qquad (4.2.2)$$

where X_k is the k^{th} coefficient of the fast forier transform (computed for $K = 50$ harmonics) cooresponding to the voltage trace of the HH model simulated for 25ms, or $N = 2500$ time steps. The following plot shows the voltage trace of the HH model overlayed with the resutling FFT approximation:

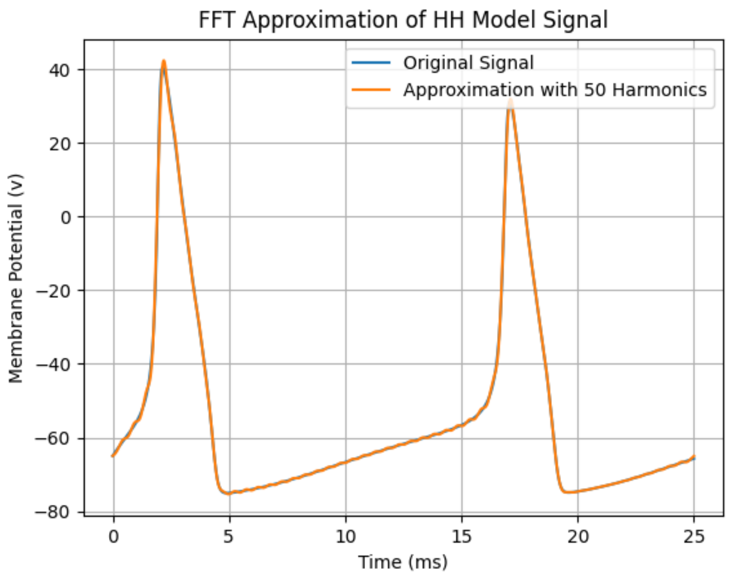

Figure 2.4.1: Plot of the voltage trace of the HH model simulated for 25ms and subjected to static input current with the cooresponding FFT approximation.

We used secant approximation of the derivative for a point θ^* in parameter space to find the matrix:

$$F_{i,j}(f:\theta) = [\sum_{k=1}^{m}(\frac{\partial f_k}{\partial \theta_i})(\frac{\partial f_k}{\partial \theta_j})]_{i,j} = Df(\theta^*) \cdot (Df(\theta^*)^T) \qquad (4.2.3)$$

In this case, θ^* is given by the same parameter values for the HH model used in the previous section. After applying the same eigenvector estimation package used above, we were able to get the following eigenvectors and their cooresponding eigenvalues:

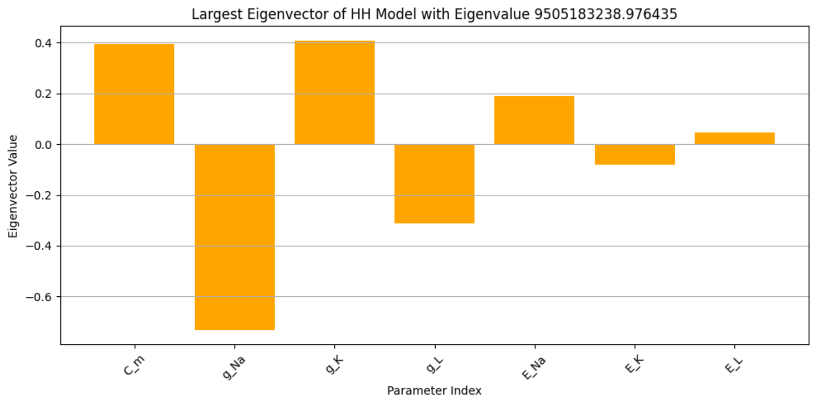

Figure 2.4.2

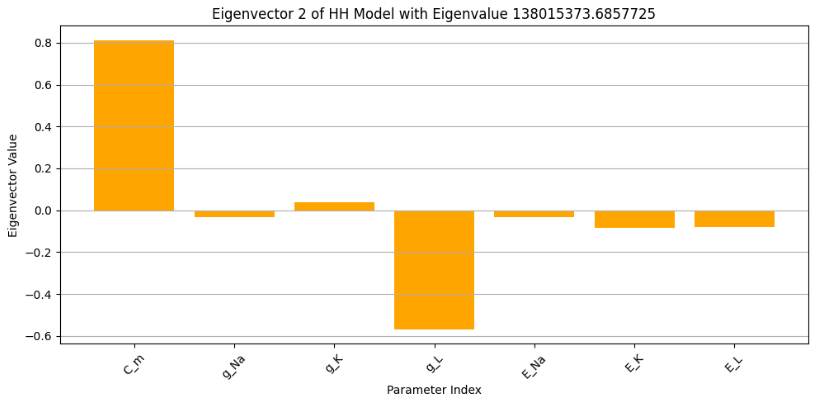

Figure 2.4.3

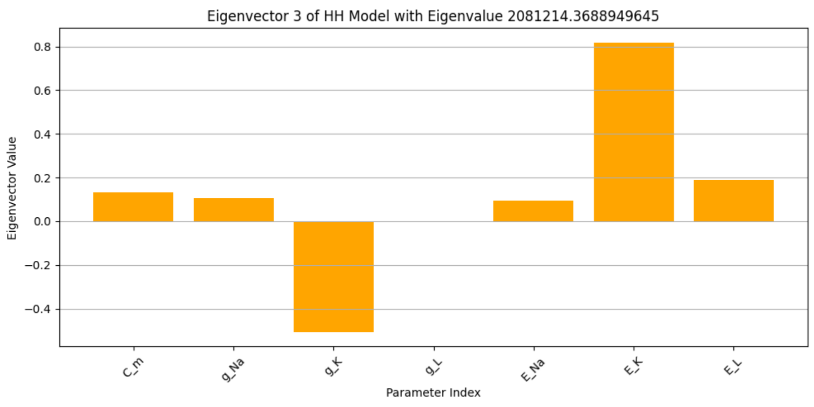

Figure 2.4.4

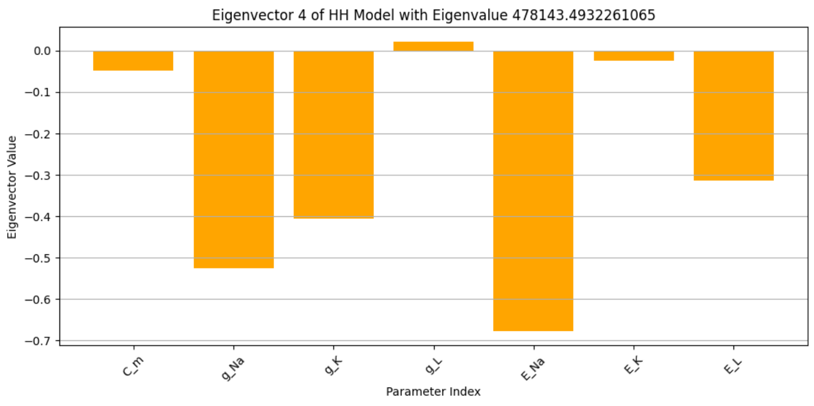

Figure 2.4.5

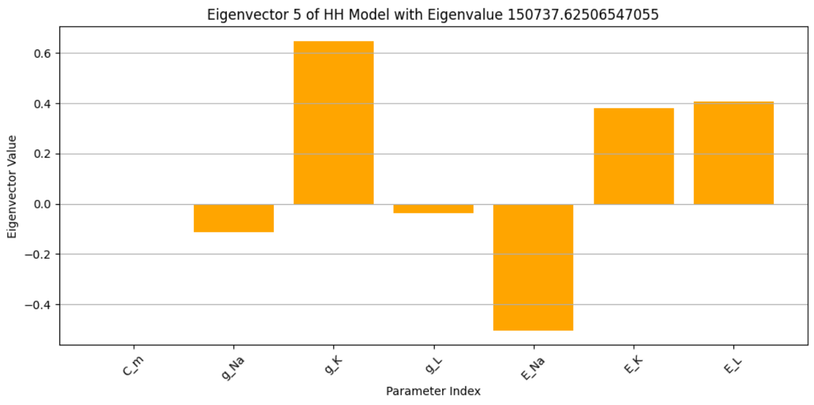

Figure 2.4.6

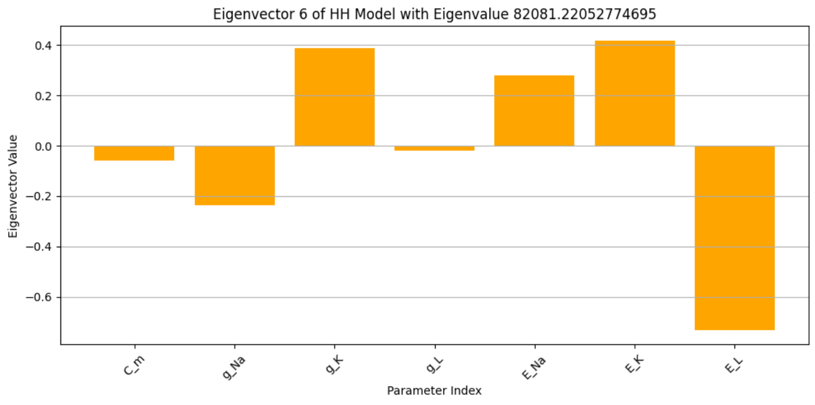

Figure 2.4.7

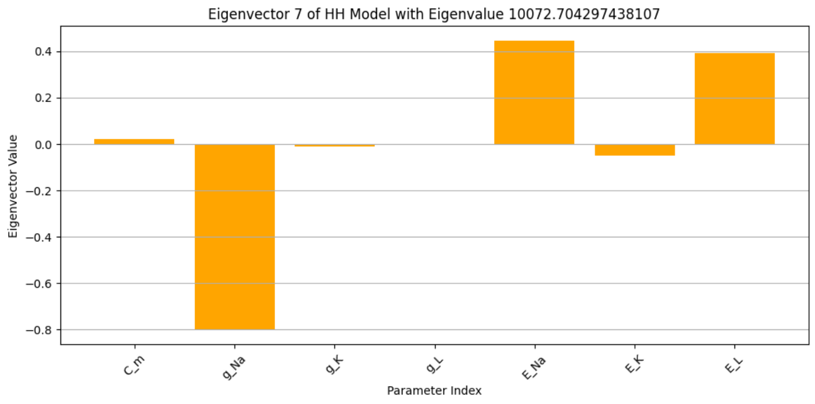

Figure 2.4.8

Note here that the eigenvalues associated with the above eigenvectors all appear to be rather large. However, qualitatively, it must be noted that the FIM matrix has values ranging from 10^6 to 10^{10}. This means the difference between the larges value in the matrix and the smallest value in the matrix is 4 orders of magnitude. Therefore, it is not obvious whether the eigenvalues cooresponding to eigenvectors 6 and 7 are qualitatively large or small. However, when plotting the trajectories for the HH model with parameter sets θ^* and $\theta' = \theta^* + E_{\lambda_7}$, we find that there is almost no discernable difference between the plots:

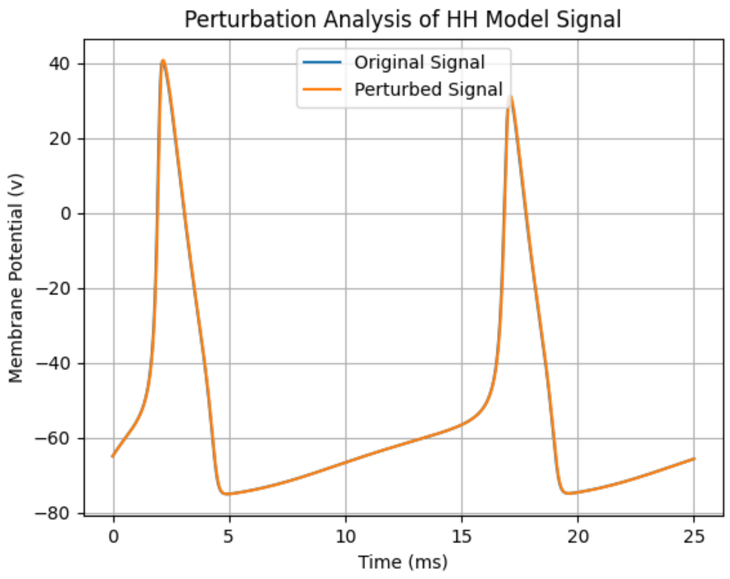

Figure 2.4.9: Plot of the HH model simulated for 25ms and subjected to noisy input for parameter sets θ^* and θ'.

While if we plot the same HH voltage trajectory alongside the voltage trajectory cooresponding to $\theta'' = \theta^* + E_{\lambda_0}$, we get:

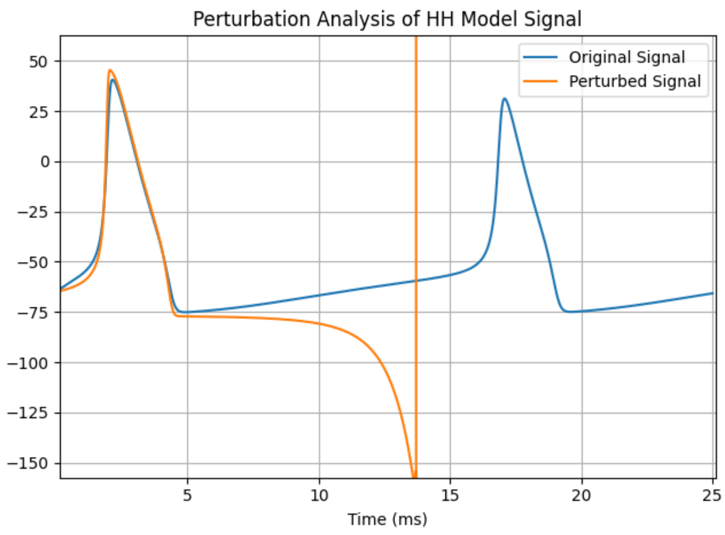

Figure 2.4.10: Plot of the HH model simulated for 25ms and subjected to noisy input for parameter sets θ^* and θ''.

This suggests that eigenvectors 5, 6 and 7 do indeed coorespond to insensitive directions in parameter space, even though the size of the eigenvectors would seem to suggest otherwise. Therefore, it may make sense to re-examine the above eigenvectors to look for potential relationships between variables. Note that in equation 2.5.8:

$$C\frac{dv}{dt} = I_{ext} - g_{Na}m^3h(v - E_{Na})) - g_K n^4(v - E_K) - g_L(v - E_L)) \qquad (2.5.8)$$

the size of g_{Na} and g_K control the size of the effect of the sodium and potassium gating variables on the voltage trace. Hence, it makes intuitive sense that increasing one while decreasing the other would lead to little change in the output of the model, as seen in eigenvectors 5 and 6. Additionally, it should be noted that there seems to be some relationship between C_m and g_{Na} as well as C_m and g_K illustrated in eigenvectors 7 and 6 respectively. Note that in the above equation, a change in the size of the capacitence C_m leads to a change in the speed at which v increases or decreases. In practice, it is a time parameter that will either stretch or shink the trace of the voltage horizontally by some amount. Recall that the gating variables m and h either increase or decrease $\frac{dv}{dt}$ depending on whether $v < E_{Na}$ or $v > E_{Na}$ respectively. Therefore a relationship between C_m and g_{Na} would hint at a potential relationship between v and m, although certainly, some different approach is necessary to confirm the existence of that relationship and the nature of it.

Ultimately, makes sense that the above approach would not necessarily give us much information about how to approximate gating variables using v. Of course, the above approach does not use these gating variables as inputs, and so it is difficult to see any relationship between them especially given the number of parameters involved. Therefore, we devise a new method for uncovering this relationship using the FIM.

4.3 The FIM in the HH model using gating variables as parameters:

In the above approach, we took the input of the funciton f to be:

$$\theta = (C_m,\ g_{Na},\ g_K,\ g_L,\ E_{Na},\ E_K,\ E_L) \qquad (4.2.1)$$

Conventionally, this is the type of input that one would normally use for parameter reduction. However, while we would ultimately like to reduce the number of parameters used in the model, we must first reduce the dimension of the model; we must find a way to relate one or more of the gating variables to the membrane potential v. In order to do this, we instead construct a function f that takes the gating variables themselves as the input, and return an output that says something about the change in the membrane potential. Therefore, we propose the following function:

$$f(\theta) = v + \frac{dv}{dt} * dt \qquad (4.3.1)$$

$$\theta = (n, m, h, v) \qquad (4.3.2)$$

where $\frac{dv}{dt}$ is given above, and $dt = 0.01$. In other words, we simply take the input to be the values of n, m, h, and v at a particular time, and take the output to be the new value of v after using Euler integration to evolve the system one time step. We subsequently used secant approximation to calculate the FIM for this function:

$$F(f : \theta^*) = Df(\theta^*) \cdot (Df(\theta^*)^T) \qquad (4.3.3)$$

where we take θ^* to be:

$$\theta^* = [n_0, m_0, h_0, v_0] \qquad (4.3.4)$$

$$v_0 = v_r = -65, \qquad x_0 = \frac{\alpha_x(v_0)}{\beta_x(v_0) + \alpha_x(v_0)} \qquad (4.3.5)$$

In other words, we take the input to be the initialized values of the gating variables so that the input is simply the state of the HH model at $t = 0$. We then calculated the FIM for an array (of size 10000) of points surrounding these initial values, and added these With the same eigenvalue/eigenvector approximation package used above, we calculated the following eigenvectors cooresponding to this input:

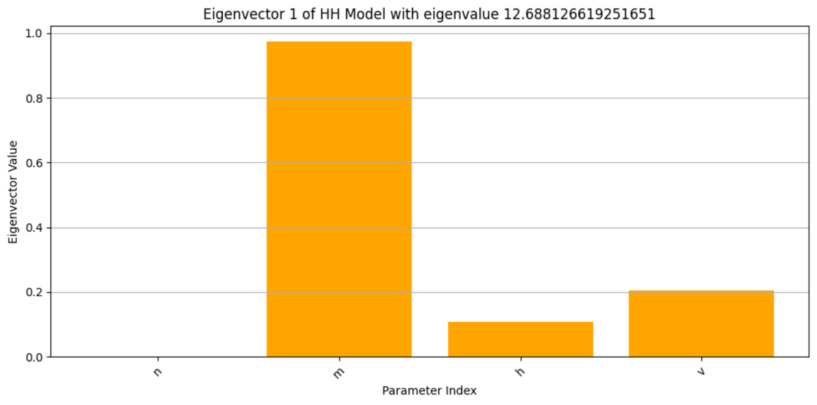

Figure 4.3.1

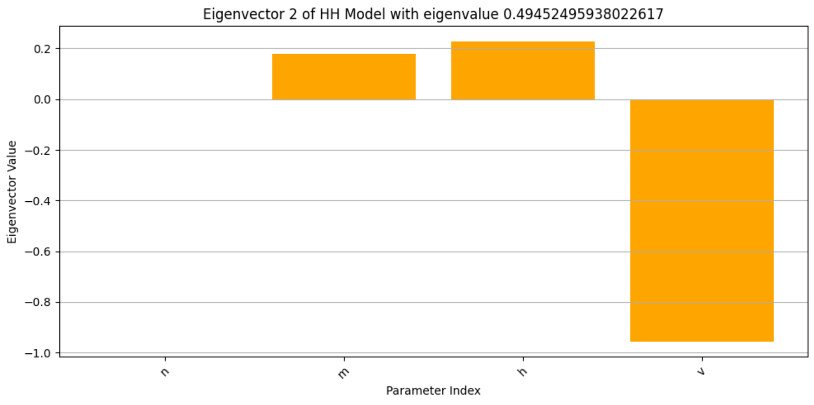

Figure 4.3.2

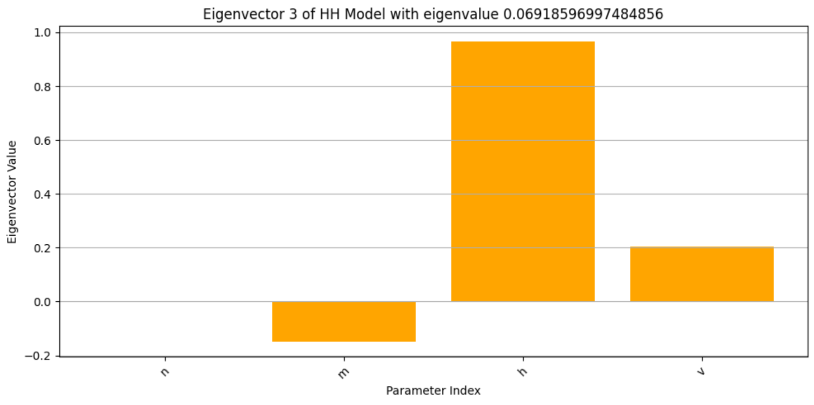

Figure 4.3.3

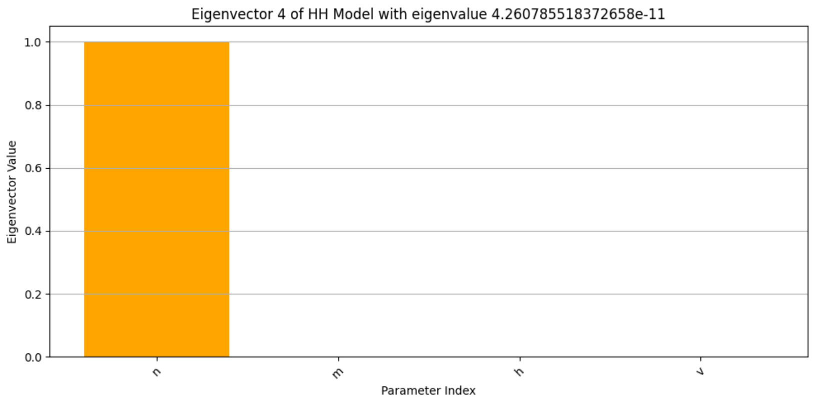

Figure 4.3.4

According to the above plots, there is one (perhaps two or three) sensitive directions in parameter space and at least two insensitive directions in parameter space. In examining the first eigenvector, we find that a large positive change in v combined with a large positive change in m coorespond to the greatest direction of change. In examining eigenvectors 2 and 3 we find that a if there is a positive change in either m or v and a negative change in the opposite variable, this leads to almost no change. The final eigenvector simply states that changing n does not change the membrane potential qualitatively, at least initially. This is consistent with the fact that n is the slowest and least impactful gating variable in the model. Below, we plot the evolution of the HH model for all four gating variables simultaneously in order to illustrate this:

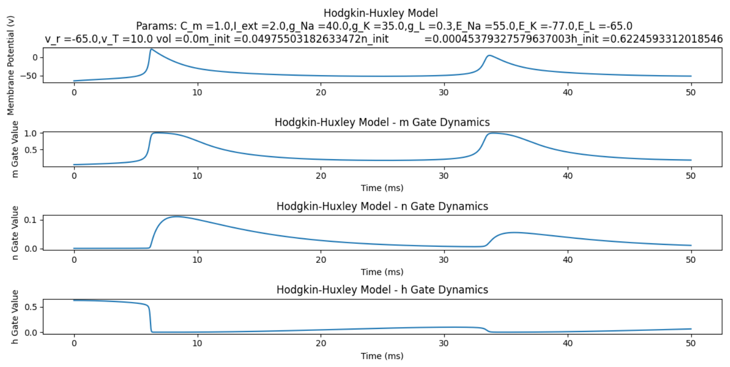

Figure 4.3.5: Plot of v alongside the gating variables in the HH model, simulated for 50ms and subjected to static input.

In considering v and m in the first 3 eigenvectors, we note that this is remarkably similar to what we observed in the toy model, where either increasing or decreasing m_1 and m_2 by the same amount leads to a large change in the output, while increasing one and decreasing the other by the same amount will lead to negligible change. This again suggests that there is some exploitable relationship between m and v.

In order to uncover the nature of this relationship, we use the technique demonstrated in the toy model, where we compute the value of the eigenvectors for the FIM at θ^*, and then instead of adding an eigenvector cooresponding to an insensitive direction in parameter space, we initially made the choice to add the eigenvector with the largest eigenvalue. This is due to the fact that v and m are the gating variables that have the greatest impact on the state of the model, and so any relationship between them will be most prevalent in the first eigenvector. Adding this eigenvector to the initial parameter vector gives us a new set of parameters, $\theta^* + E_\lambda 0$, and we recompute the FIM at this new point. Ultimately, this process is iterated forward, and track the development of the variables v and m as we go. In using this method, we were able to obtain the following data:

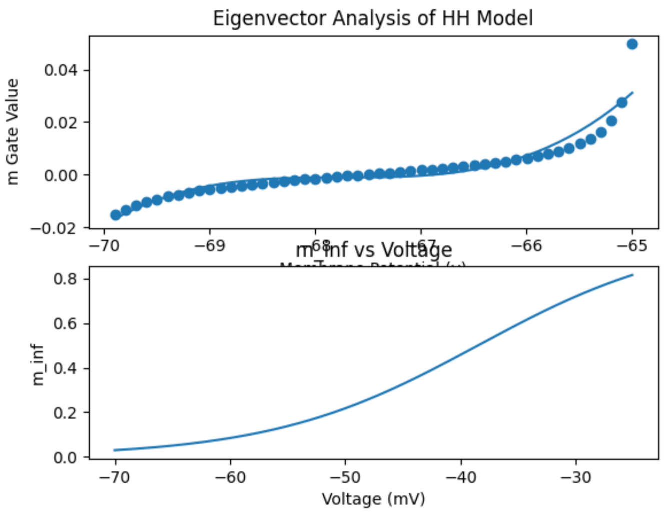

Figure 4.3.6: Scatter plot of the data points cooresponding to values of m and v obtained using the method above alondside a plot of the $m_\infty(v)$ curve for values of v ranging from -65 to -20mV

Here, we see that after appropriate rescaling, the best fit line for the data appears to qualitatively resemble the inverse of the $m_\infty(v)$ relation (in the second plot). In order to further analyze the potential relationship here, we chose to itterate using the second eigenvector instead. This yeilds the following plot:

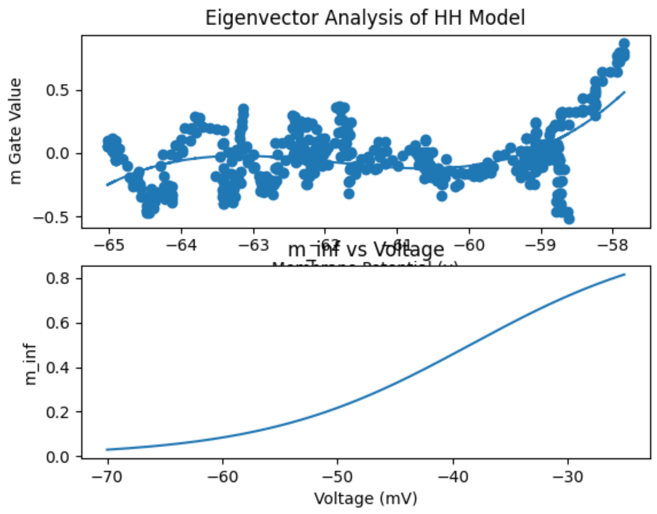

Figure 4.3.7: Scatter plot of the data points cooresponding to values of m

and v obtained using the method above alondside a plot of the $m_\infty(v)$ curve for values of v ranging from -65 to -20mV

In analyzing this plot, we find that while there is significantly more noise, the graph of $m_\infty(v)$, after appropriate rescaling, appears to be almost identical to the line of best fit for the data.

5 Conclusion:

In this paper, we have introduced several neuronal models, and explored their dynamics in several circumstances. We then demonstrated that the adapted version of the aEIF model can be fit to match the behavior of more complex models. Subsequently, after showing that the Hodgekin/Huxley model can be reduced to a 3, and even a 2 dimensional dynamical system, we used active subspaces to demonstrate how these reductions could be found using the FIM matrix. While the above approach by no means constitutes a general method for parameter reduction in higher dimensional models, these results do indeed demonstrate that active subspaces and the FIM can be used in concert with one another to give researchers an idea or when a model can be reduced, and what the nature of the reduction might be. While the explicit reduction $m = m_\infty(v)$ was not replicated, one could conceivably do this by using an exponential fitting algorithm with input parameters cooresponding to the various constants in α_m, β_m to get a best fit line for the data points in figure 4.3.7, austensibly recovering $m_\infty(v)$. Unfortunately, there were scaling issues with these plots that prevented us from achieving such a fit. A likely cause here would be the fact that the gating variables themselves are time dependent, and do not increase linearly with time. Therefore, incrementing them by a small fixed step size leads to a shear in the data, causing it to look noisy and mishapen. However, we used to toy model to show that this is a valid approach, at least in circumstances where the relation on the parameters can be approximated linearly.

6 References:

(yet to be formatted) Hodgkin, A. L., Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. The Journal of physiology, 117(4), 500–544. https://doi.org/10.1113/jphysiol.1952.sp004764

Gerstner, Adaptive Exponential Integrate-and-Fire Model as an Effective Description of Neuronal Activity, 2005

Gerstner, Neruonal Dynamics, textbook

Brouwer, Eisenberg, The underlying connections between identifiability, active subspaces, and parameter space dimension reduction

Fourcard, Hansel, How Spike Generation Mechanisms Determine the Neuronal Response to Fluctuating Inputs Mark K. Transtrum, Peng Qiu, Model Reduction by Manifold Boundaries

Alain Destexhe,Diego Contreras, andMircea Steriade, Mechanisms Underlying the Synchronizing Action of Corticothalamic Feedback Through Inhibition of Thalamic Relay Cells

Walch, Eisenberg, Parameter identifiability and identifiable combinations in generalized Hodgkin–Huxley models

Simeone Marino, Ian B Hogue, Christian J Ray, Denise E KirschnerA methodology for performing global uncertainty and sensitivity analysis in systems biology