1 Single Non-Spiking Neuron

1.1 Single Non-Spiking Neuron Mathematical Model

The governing equation for a single non-spiking neuron with a leak current, n incoming synapses, sodium channels, and an applied current is

$$C_m \dot{V} = I_{leak} + I_{syn} + I_{Na} + I_{app}, \tag{1}$$

where

$$\begin{aligned} I_{leak} &= G_m(E_r - V), & (2)\\ I_{syn} &= \sum_{i=1}^{n} G_{s,i}(E_{s,i} - V), & (3)\\ I_{Na} &= G_{Na} m_\infty h (E_{Na} - V), & (4) \end{aligned}$$

and I_{app} is some known function of time. V, G_m, and C_m are the membrane voltage, conductance, and capacitance of the neuron of interest, respectively. E_r, $E_{s,i}$, and E_{Na} are the reversal potentials of the neuron of interest, the incoming synapses, and the sodium channels, respectively. G_{Na}, m, h are the sodium channel conductance, activation parameter, and deactivation parameter, respectively. $G_{s,i}$ is the conductance of the ith synapse.

Note that the sodium channel deactivation parameter, h, is a second dynamical variable for the neuron, with the governing differential equation

$$\dot{h} = \frac{h_\infty - h}{\tau_h}. \tag{5}$$

The time constant associated with the sodium channel deactivation parameter, τ_h, is given by

$$\tau_h = \tau_{h,max} h_\infty \sqrt{A_h e^{S_h(V - E_h)}}, \tag{6}$$

where $\tau_{h,max}$ is the maximum sodium channel deactivation parameter time constant. The steady-state activation and deactivation parameters, m_∞ and h_∞, respectively, are given by

$$\begin{aligned} m_\infty &= \frac{1}{1 + A_m e^{S_m(V - E_m)}}, & (7)\\ h_\infty &= \frac{1}{1 + A_h e^{S_h(V - E_h)}}. & (8) \end{aligned}$$

Finally, the synaptic conductances are given by

$$G_{s,i} = g_{i,max} \min\left(\max\left(\frac{V_{pre,i} - E_{lo,i}}{E_{hi,i} - E_{lo,i}}, 0 \right), 1 \right) \tag{9}$$

where $g_{i,max}$, $E_{hi,i}$, and $E_{lo,i}$ are the maximum conductance, the voltage limit, and the voltage threshold, of synapse i, respectively. $V_{pre,i}$ is the membrane voltage of pre-synaptic neuron i.

Now that we have fully defined our single neuron system, we want to collapse all of this information into a system of two, non-linear, non-homogeneous, ordinary differential equations. We begin by substituting each of our current definitions into our membrane voltage equation to find

$$\dot{V} = \frac{G_m}{C_m}(E_r - V) + \sum_{i=1}^{n} \frac{G_{s,i}}{C_m}(E_{s,i} - V) + \frac{G_{Na}}{C_m} m_\infty h(E_{Na} - V) + \frac{I_{app}}{C_m}. \tag{10}$$

Substituting Equation (9) into Equation (10) (i.e., substituting our synaptic conductances into our membrane voltage differential equation), we get

$$\dot{V} = \frac{G_m}{C_m}(E_r - V) + \sum_{i=1}^{n} \frac{g_{i,max}}{C_m} \min\left(\max\left(\frac{V_{pre,i} - E_{lo,i}}{E_{hi,i} - E_{lo,i}}, 0\right), 1\right)(E_{s,i} - V) + \frac{G_{Na}}{C_m} m_\infty h(E_{Na} - V) + \frac{I_{app}}{C_m}. \tag{11}$$

Substituting Equation (7) into Equation (11) (i.e., substituting our sodium channel activation parameter into our membrane voltage differential equation), we have

$$\begin{aligned} \dot{V} = \frac{G_m}{C_m}(E_r - V) + \sum_{i=1}^{n} \frac{g_{i,max}}{C_m} \min\left(\max\left(\frac{V_{pre,i} - E_{lo,i}}{E_{hi,i} - E_{lo,i}}, 0\right), 1\right)(E_{s,i} - V) \\ + \frac{G_{Na}}{C_m}\left(\frac{1}{1 + A_m e^{S_m(V - E_m)}}\right) h(E_{Na} - V) + \frac{I_{app}}{C_m}. \end{aligned} \tag{12}$$

Substituting Equation (8) into Equation (5) (i.e., substituting our steady state sodium channel deactivation parameter equation into our sodium channel activation differential equation), we have

$$\dot{h} = \frac{1}{\tau_h\left(1 + A_h e^{S_h(V - E_h)}\right)} - \frac{h}{\tau_h}. \tag{13}$$

Substituting Equation (6) into Equation (13) (i.e., substituting our sodium channel deactivation parameter time constant equation into our sodium channel activation differential equation), we have

$$\dot{h} = \frac{1 - h\left(1 + A_h e^{S_h(V - E_h)}\right)}{\tau_{h,max} h_\infty \sqrt{A_h e^{S_h(V - E_h)}}\left(1 + A_h e^{S_h(V - E_h)}\right)}. \tag{14}$$

So we write the system of two, non-linear, non-homogeneous, ordinary differential equations as

$$\begin{aligned}\dot{V} &= \frac{G_m}{C_m}(E_r - V) + \sum_{i=1}^{n} \frac{g_{i,max}}{C_m} \min\left(\max\left(\frac{V_{pre,i} - E_{lo,i}}{E_{hi,i} - E_{lo,i}}, 0\right), 1\right)(E_{s,i} - V) \\ &+ \frac{G_{Na}}{C_m}\left(\frac{1}{1 + A_m e^{S_m(V - E_m)}}\right) h(E_{Na} - V) + \frac{I_{app}}{C_m},\end{aligned} \tag{15}$$

$$\dot{h} = \frac{1 - h\left(1 + A_h e^{S_h(V - E_h)}\right)}{\tau_{h,max} h_\infty \sqrt{A_h e^{S_h(V - E_h)}}\left(1 + A_h e^{S_h(V - E_h)}\right)}. \tag{16}$$

Note that, since this model considers only a single neuron, the pre-synaptic voltages are constants. As such, they contribute a constant current to the neuron of interest of a magnitude whose value depends on the properties of the synaptic connections.

1.2 Example: Single Non-Spiking Neuron with Leak & Applied Currents

Now that we have created a mathematical model for a single non-spiking neuron, we can consider an illustrative special case to verify that this model agrees with our expectations. Consider the most simple neuron model – the non-spiking neuron with only a leak and applied current. Since we are considering only the leak and applied current of the neuron, the synaptic currents and sodium channel current will be ignored. In this case, our governing equation becomes

$$\dot{V} = \frac{G_m}{C_m}(E_r - V) + \frac{I_{app}}{C_m}. \tag{17}$$

Note that, since we are not considering sodium channel currents, the dynamical variable h no longer appears in the governing equation, so we only have a single ordinary differential equation. Furthermore, since there are no synaptic inputs, the system is now linear with respect to the neuron's membrane voltage V. Such a system can easily be solved with standard ODE techniques. While this problem is quite simple, solving the full equation is much more complicated due to the non-linearity. As such, we will use the standard Runge-Kutta numerical methods as implemented in Matlab's ode45() equation solver to solve this and all other ODEs discussed herein. The membrane voltage of a single non-spiking neuron with only leak and applied currents yields the result shown below.

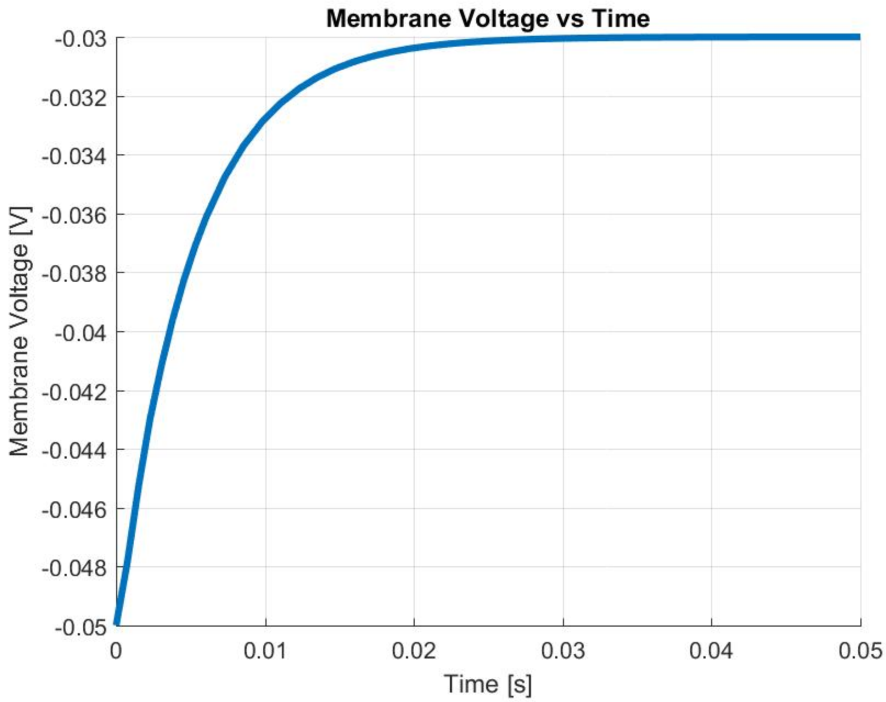

Figure 1: Non-Spiking Neuron Membrane Voltage with Leak and Applied Current (Step Input).

Note that the form of this solution agrees with existing knowledge of neuron behavior. When a step input current is applied to the neuron, the neuron membrane voltage increase from its resting potential to a new equilibrium value. The time constant associated with this increase, as well as the new steady state value, depend on the specific membrane properties that are chosen and the magnitude of the input current. Values were chosen here to produce physically realistic results.

While we did not include synaptic currents in this example, their inclusion does not effect the form of the result. As mentioned above, the synaptic currents for our single neuron model are constant, since the pre-synaptic voltages do not change over time. Since we used a step input applied current in the above example, the synaptic currents would simply raise or lower the steady state membrane voltage of our neuron of interest.

Realistically, the membrane voltages of the pre-synaptic neurons change over time. We will next consider how to incorporate this feature into our above single neuron mathematical model to expand it to describe a network of neurons.

2 Network of Non-Spiking Neurons

2.1 Non-Spiking Neural Network (NSNN) Mathematical Model

Now that we have established the system of two differential equations that govern the membrane voltage of a single neuron with a leak current, n synaptic inputs, sodium channels, and an applied current, we can expand this system to a network of neurons. We will switch the interpretation of n to now indicate the number of neurons in the network (not necessarily the number of synaptic inputs to any specific neuron). Since the voltage of any single neuron depends on the pre-synaptic voltage of all of the connected neurons, and we require two differential equations to describe the voltage of a single neuron when all of the pre-synaptic voltages are known, we will have a total of $2n$ differential equations in our system.

Assuming that neuron i is connected to neuron j, $\forall i, j \in \{1, \dots, n\}$, we can rewrite Equations (15) and (16) (i.e., our system of ODEs for a single neuron) to apply to a network of neurons. Doing so, we have

$$\begin{aligned}\dot{V}_j = {} & \frac{G_{m,j}}{C_{m,j}}(E_{r,j} - V_j) + \sum_{i=1}^{n} \frac{g_{i,j,max}}{C_{m,j}} \min\left(\max\left(\frac{V_i - E_{lo,i,j}}{E_{hi,i,j} - E_{lo,i,j}}, 0\right), 1\right)(E_{s,i,j} - V_j) \\ & + \frac{G_{Na,j}}{C_{m,j}}\left(\frac{1}{1 + A_{m,j}e^{S_{m,j}(V_j - E_{m,j})}}\right) h_j(E_{Na_j} - V_j) + \frac{I_{app,j}}{C_{m,j}},\end{aligned} \tag{18}$$

$$\dot{h}_j = \frac{1 - h_j\left(1 + A_{h,j}e^{S_{h,j}(V_j - E_{h,j})}\right)}{\tau_{h,max,j} h_{\infty,j}\sqrt{A_{h,j}e^{S_{h,j}(V_j - E_{h,j})}}\left(1 + A_{h,j}e^{S_{h,j}(V_j - E_{h,j})}\right)}, \tag{19}$$

$\forall j \in \{1, \dots, n\}$. Note that none of the expressions have changed significantly from the single neuron case. They are simply indexed such that each neuron has its own parameters and each neuron is connected to every other neuron in the network (including itself) with a maximum synaptic conductance of $g_{i,j,max}$. In this model, neurons can be disconnected from one another by choosing a maximum synaptic conductance of $g_{i,j,max} = 0$ for some $i, j \in \{1, \dots, n\}$. Inhibitory synapses are assigned negative synaptic conductances, while exictatory synapses are assigned positive synaptic conductances. Each neuron also has its own applied current $I_{app,j}$ which may set to zero to indicate that there is no applied current or set to any desired function of time. Similarly, each neuron has there own sodium channel conductance $G_{Na,j}$ which may be set to zero to turn off sodium channels for that neuron. In this way, we have a fully connected network of non-spiking neurons that have their own leak current properties, may be connected to any other neuron in the network, have optional sodium channels, and have the ability to interpret an applied current.

While much less concise, it may be illustrative to expand the summations and write out several terms in the sequence. We will therefore also write the above system in the following form

$$\begin{aligned}
\dot{V}_1 &= \frac{G_{m,1}}{C_{m,1}}(E_{r,1} - V_1) \\
&+ \frac{g_{1,1,max}}{C_{m,1}} \min\left(\max\left(\frac{V_1 - E_{lo,1,1}}{E_{hi,1,1} - E_{lo,1,1}}, 0\right), 1\right)(E_{s,1,1} - V_1) \\
&+ \frac{g_{2,1,max}}{C_{m,1}} \min\left(\max\left(\frac{V_2 - E_{lo,2,1}}{E_{hi,2,1} - E_{lo,2,1}}, 0\right), 1\right)(E_{s,2,1} - V_1) \\
&+ \cdots + \frac{g_{n,1,max}}{C_{m,1}} \min\left(\max\left(\frac{V_n - E_{lo,n,1}}{E_{hi,n,1} - E_{lo,n,1}}, 0\right), 1\right)(E_{s,n,1} - V_1) \\
&+ \frac{G_{Na,1}}{C_{m,1}}\left(\frac{1}{1 + A_{m,1}e^{S_{m,1}(V_1 - E_{m,1})}}\right) h_1(E_{Na_1} - V_1) + \frac{I_{app,1}}{C_{m,1}}, \qquad (20)
\end{aligned}$$

$$\dot{h}_1 = \frac{1 - h_1\left(1 + A_{h,1}e^{S_{h,1}(V_1 - E_{h,1})}\right)}{\tau_{h,max,1} h_{\infty,1}\sqrt{A_{h,1}e^{S_{h,1}(V_1 - E_{h,1})}}\left(1 + A_{h,1}e^{S_{h,1}(V_1 - E_{h,1})}\right)}, \qquad (21)$$

$$\begin{aligned}
\dot{V}_2 &= \frac{G_{m,2}}{C_{m,2}}(E_{r,2} - V_2) \\
&+ \frac{g_{1,2,max}}{C_{m,2}} \min\left(\max\left(\frac{V_1 - E_{lo,1,2}}{E_{hi,1,2} - E_{lo,1,2}}, 0\right), 1\right)(E_{s,1,2} - V_2) \\
&+ \frac{g_{2,2,max}}{C_{m,2}} \min\left(\max\left(\frac{V_2 - E_{lo,2,2}}{E_{hi,2,2} - E_{lo,2,2}}, 0\right), 1\right)(E_{s,2,2} - V_2) \\
&+ \cdots + \frac{g_{n,2,max}}{C_{m,2}} \min\left(\max\left(\frac{V_n - E_{lo,n,2}}{E_{hi,n,2} - E_{lo,n,2}}, 0\right), 1\right)(E_{s,n,2} - V_2) \\
&+ \frac{G_{Na,2}}{C_{m,2}}\left(\frac{1}{1 + A_{m,2}e^{S_{m,2}(V_2 - E_{m,2})}}\right) h_2(E_{Na_2} - V_2) + \frac{I_{app,2}}{C_{m,2}}, \qquad (22)
\end{aligned}$$

$$\dot{h}_2 = \frac{1 - h_2\left(1 + A_{h,2}e^{S_{h,2}(V_1 - E_{h,2})}\right)}{\tau_{h,max,2} h_{\infty,2}\sqrt{A_{h,2}e^{S_{h,2}(V_2 - E_{h,2})}}\left(1 + A_{h,2}e^{S_{h,2}(V_2 - E_{h,2})}\right)}, \qquad (23)$$

$$\vdots$$

$$\begin{aligned}
\dot{V}_n &= \frac{G_{m,n}}{C_{m,n}}(E_{r,n} - V_n) \\
&+ \frac{g_{1,n,max}}{C_{m,n}} \min\left(\max\left(\frac{V_1 - E_{lo,1,n}}{E_{hi,1,n} - E_{lo,1,n}}, 0\right), 1\right)(E_{s,1,n} - V_n) \\
&+ \frac{g_{2,n,max}}{C_{m,n}} \min\left(\max\left(\frac{V_2 - E_{lo,2,n}}{E_{hi,2,n} - E_{lo,2,n}}, 0\right), 1\right)(E_{s,2,n} - V_n) \\
&+ \cdots + \frac{g_{n,n,max}}{C_{m,n}} \min\left(\max\left(\frac{V_n - E_{lo,n,n}}{E_{hi,n,n} - E_{lo,n,n}}, 0\right), 1\right)(E_{s,n,n} - V_n) \\
&+ \frac{G_{Na,n}}{C_{m,n}}\left(\frac{1}{1 + A_{m,n}e^{S_{m,n}(V_n - E_{m,n})}}\right) h_n(E_{Na_n} - V_n) + \frac{I_{app,n}}{C_{m,n}}, \qquad (24)
\end{aligned}$$

$$\dot{h}_n = \frac{1 - h_n\left(1 + A_{h,n}e^{S_{h,n}(V_n - E_{h,n})}\right)}{\tau_{h,max,n} h_{\infty,n}\sqrt{A_{h,n}e^{S_{h,n}(V_n - E_{h,n})}}\left(1 + A_{h,n}e^{S_{h,n}(V_n - E_{h,n})}\right)}. \qquad (25)$$

$$(26)$$

2.2 Example 1: NSNN with Ex. Synapses (No Na Channels)

To verify the accuracy of our mathematical model for a network of non-spiking neurons, we will again apply our model to an illustrative example. In the previous example, we left off both the synaptic connections and sodium channels to focus on the single neuron model. Since we now seek to test the efficacy of our model for networks, we will again neglect sodium channels but this time incorporate synaptic connections for a simple network. The network that will be considered can be seen below.

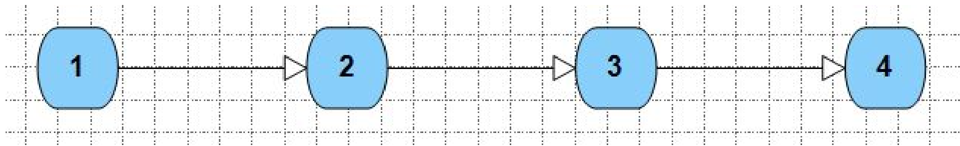

Figure 2: Simple Animatlab Non-spiking Neural Network with Excitatory Synapses.

Since we will ignore sodium channels, our mathematical model can be simplified as shown below.

$$\dot{V}_j = \frac{G_{m,j}}{C_{m,j}}(E_{r,j} - V_j) + \sum_{i=1}^{n} \frac{g_{i,j,max}}{C_{m,j}} \min\left(\max\left(\frac{V_i - E_{lo,i,j}}{E_{hi,i,j} - E_{lo,i,j}}, 0\right), 1\right)(E_{s,i,j} - V_j) + \frac{I_{app,j}}{C_{m,j}}. \quad (27)$$

Since there are four neurons in our network and each are connected sequentially, we can write the above system out as

$$\dot{V}_1 = \frac{G_{m,1}}{C_{m,1}}(E_{r,1} - V_1) + \frac{I_{app,1}}{C_{m,1}}, \quad (28)$$

$$\dot{V}_2 = \frac{G_{m,2}}{C_{m,2}}(E_{r,2} - V_2) + \frac{g_{1,2,max}}{C_{m,2}} \min\left(\max\left(\frac{V_1 - E_{lo,1,2}}{E_{hi,1,2} - E_{lo,1,2}}, 0\right), 1\right)(E_{s,1,2} - V_2) + \frac{I_{app,2}}{C_{m,2}}, \quad (29)$$

$$\dot{V}_3 = \frac{G_{m,3}}{C_{m,3}}(E_{r,3} - V_3) + \frac{g_{2,3,max}}{C_{m,2}} \min\left(\max\left(\frac{V_2 - E_{lo,2,3}}{E_{hi,2,3} - E_{lo,2,3}}, 0\right), 1\right)(E_{s,2,3} - V_3) + \frac{I_{app,3}}{C_{m,3}}, \quad (30)$$

$$\dot{V}_4 = \frac{G_{m,4}}{C_{m,4}}(E_{r,4} - V_4) + \frac{g_{3,4,max}}{C_{m,4}} \min\left(\max\left(\frac{V_4 - E_{lo,3,4}}{E_{hi,3,4} - E_{lo,3,4}}, 0\right), 1\right)(E_{s,3,4} - V_4) + \frac{I_{app,4}}{C_{m,4}}. \quad (31)$$

Note that Equation (28) has fewer terms than the others in the system due to the fact that the first neuron has no synaptic inputs in this example. Also, Equations (29)-(31) are nonlinear when

$$0 < \frac{V_i - E_{lo,i,j}}{E_{hi,i,j} - E_{lo,i,j}} < 1 \quad (32)$$

for the appropriate $i, j \in \{1, \dots, n\}$. This would significantly complicate the solution method if we were using an analytical approach. However, this is no problem for our numerical solution method. Setting the network input functions such that

$$I_{app,1} = (10 * 10^{-9})u_s(t), \;\; I_{app,2} = 0, \;\; I_{app,3} = 0, \;\; \& \;\; I_{app,4} = 0, \quad (33)$$

where $u_s(t)$ is the unit step function, we can solve the system to find the behavior shown below.

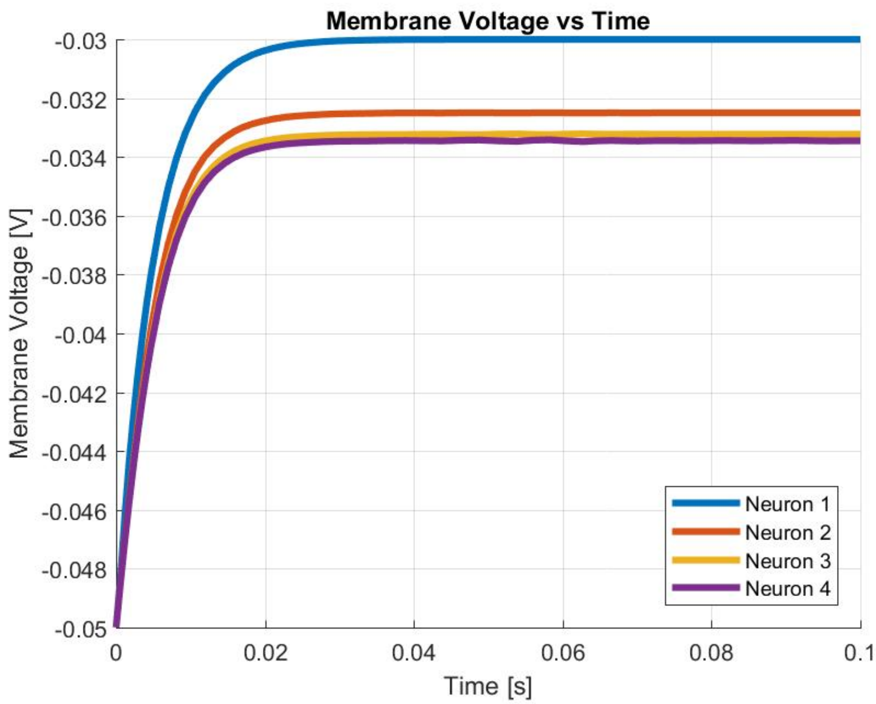

Figure 3: Non-Spiking Neural Network Membrane Voltages (Step Input, No Sodium Channels).

Note that all four of the neurons behave in a similar manner as our single neuron simulation. All four neurons increase rapidly to steady state values of differing magnitude. Their time constants and steady state membrane voltage values are determined by their synaptic properties, as well as their order in the network. We can verify this result by running the same network in Animatlab. This results in the plot shown below.

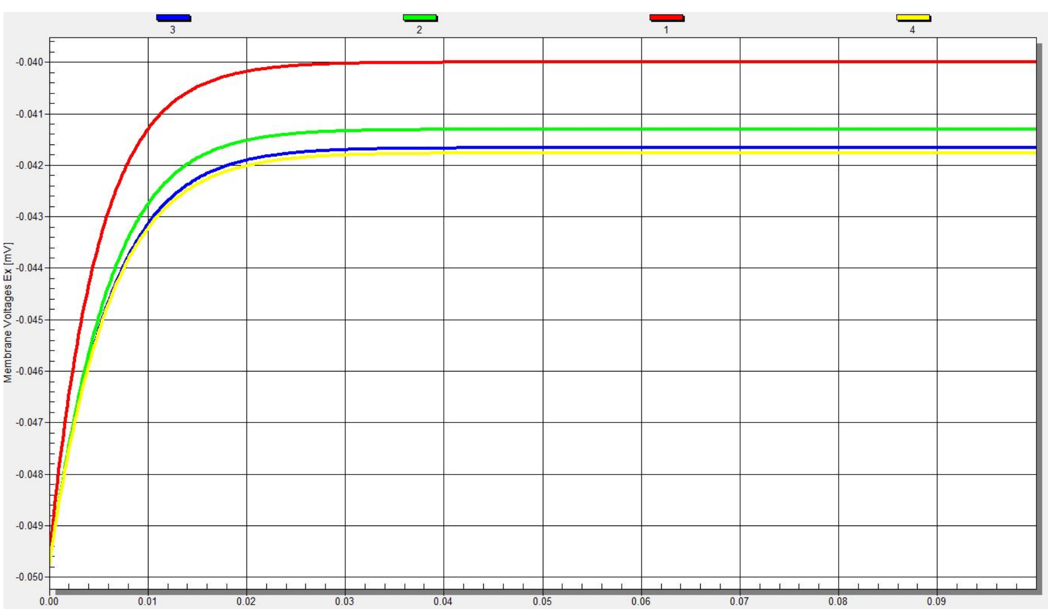

Figure 4: NS Neural Network Membrane Voltages in Animatlab (Step Input, No Sodium Channels).

2.3 Example 2: NSNN with Ex. & In. Synapses (No Na Channels)

We now repeat the previous example with one alteration – the synapse between neurons 2 and 3 is now inhibitory. Our model should capture the behavior of the network regardless as to the type of synapses we have, so we include an inhibitory synapse to demonstrate this capability. Now the network being considered is that shown below.

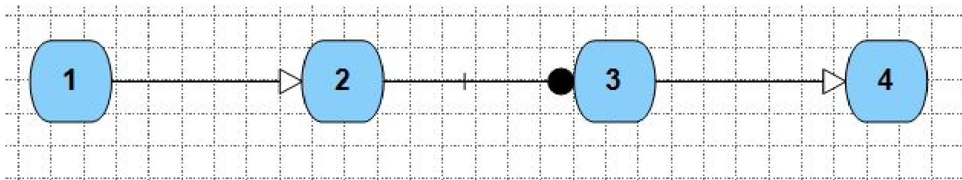

Figure 5: Simple Animatlab Non-spiking Neural Network with Excitatory & Inhibitory Synapses.

Our neural network model produces membrane voltages as shown in the figure below.

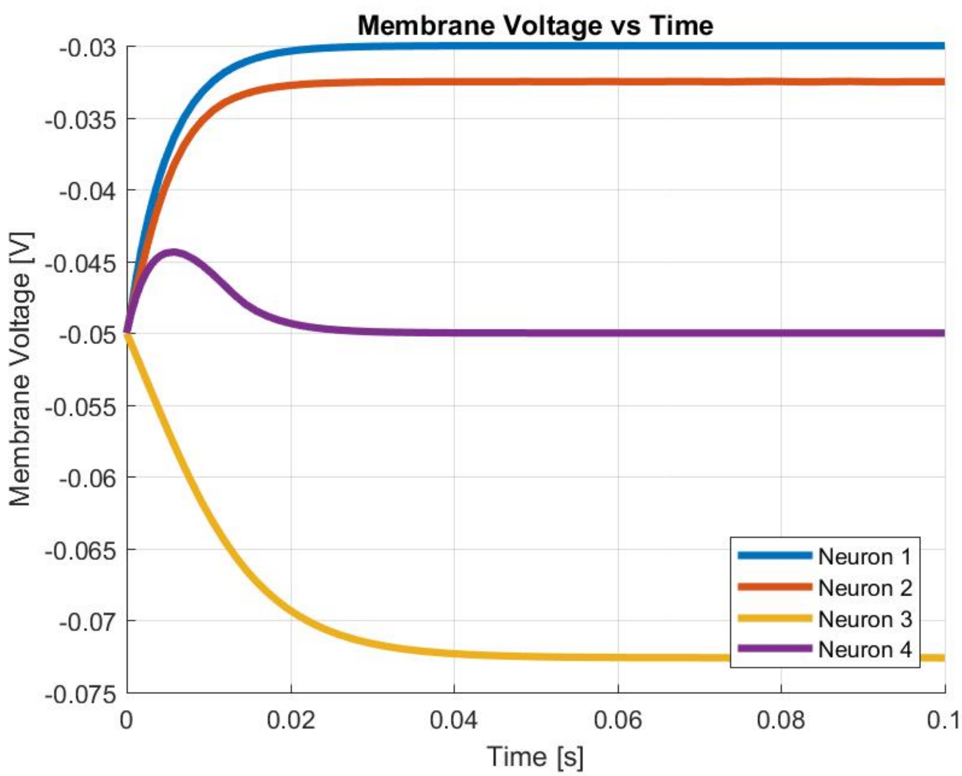

Figure 6: NSNN Membrane Voltages with Ex. & In. Synapses (Step Input, No Sodium Channels).

Note that both neurons 1 and 2 respond in a fashion similar to the previous example. This makes sense because their input connections have not been altered. Neuron 3, however, now decreases until it approaches a new steady state membrane voltage below its resting potential. This result agrees with our intuition, because the synapse between neuron's 2 and 3 is inhibitory. Finally, neuron 4 experiences a temporary increase in membrane voltage before returning to its resting potential. The transient behavior of neuron 4 is due to the initial synaptic current from neuron 3 (due to the starting membrane voltages and the synaptic threshold value). However, since neuron 3 is being inhibited, its voltage quickly drops below the synaptic threshold for the connection from neuron 3 to neuron 4, causing their to be no current supplied to neuron 4. Hence, neuron 4 returns to its initial value as it approaches steady state conditions.

Simulating this neural network in Animatlab, we get similar behavior that confirms the efficacy of our model.

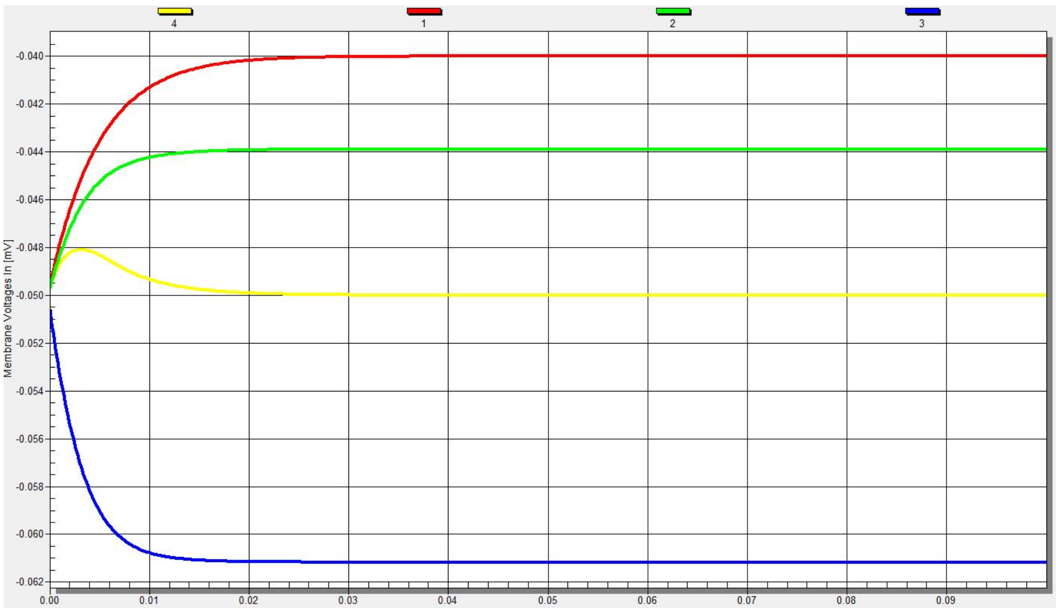

Figure 7: NSNN MV in Animatlab with Ex. & In Synapses (Step Input, No Na Channels).

2.4 Example 3: NSNN with Ex. Synapses & Na Channels

Now we apply our complete NSNN mathematical model with sodium channels (Equations (18) and (19)) to our neural network shown in Figure 2. The results of this simulation can be seen below.

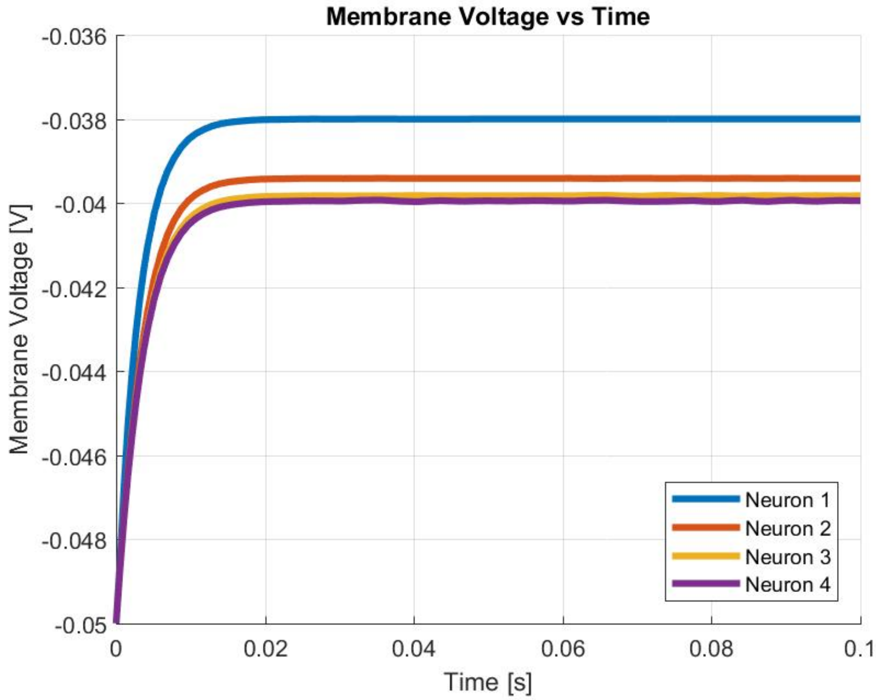

Figure 8: NSNN MV with Ex. Synapses & Na Channels (Step Input).

Note that this plot looks nearly identical to that generated from our simulation without sodium channels. The primary difference is that the steady state values of each of the neurons have been lowered (i.e., they are closer to their resting potential). The Animatlab simulation shown below yields similar results.

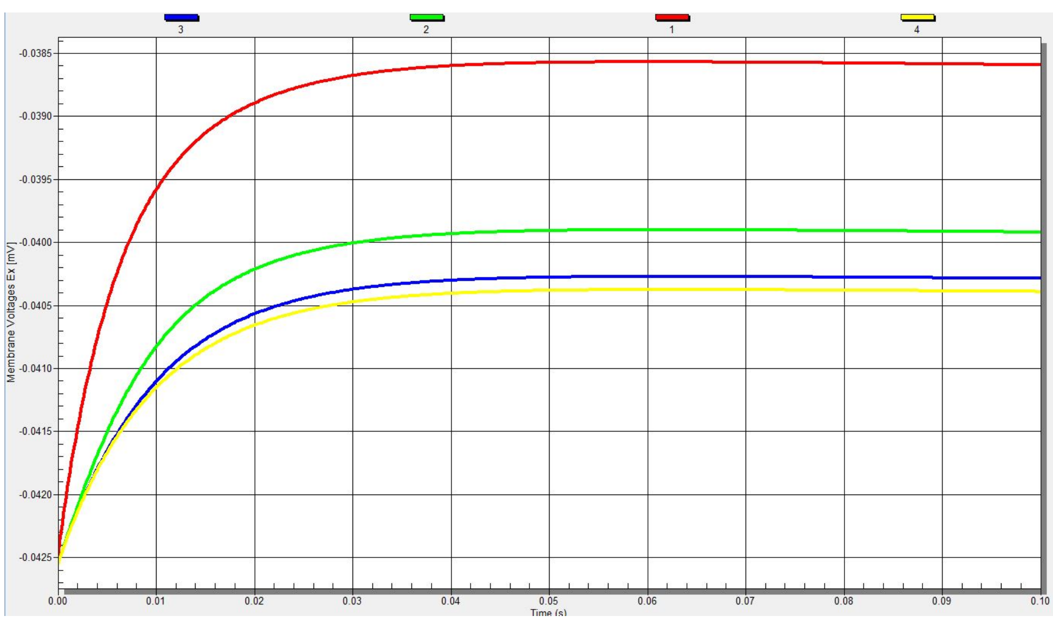

Figure 9: NSNN MV in Animatlab with Ex. Synapses & Na Channels (Step Input).

2.5 Example 4: NSNN with Ex./In. Synapses & Na Channels

Now we apply our complete NSNN mathematical model with sodium channels (Equations (18) and (19)) to our neural network shown in Figure 5. The results of this simulation can be seen below.

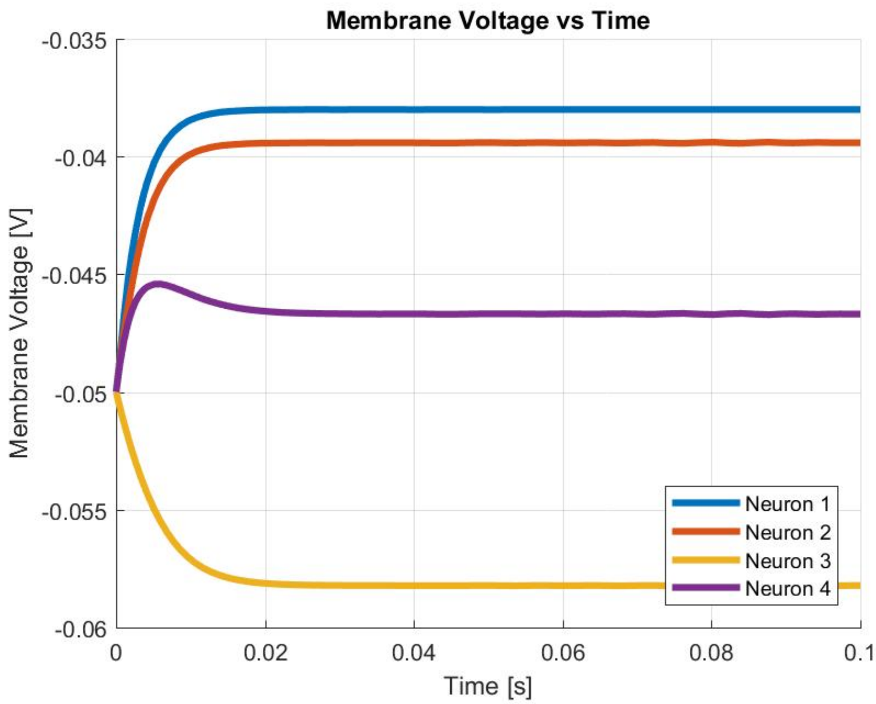

Figure 10: NSNN MV with Ex./In Synapses & Na Channels (Step Input).

Note that this plot looks nearly identical to that generated from our simulation without sodium channels. The primary difference is that the steady state values of each of the neurons have been shifted closer to their resting potential. Also, neuron 4's steady state membrane voltage is no longer at its resting potential. The Animatlab simulation shown below yields different results that I have yet to explain.

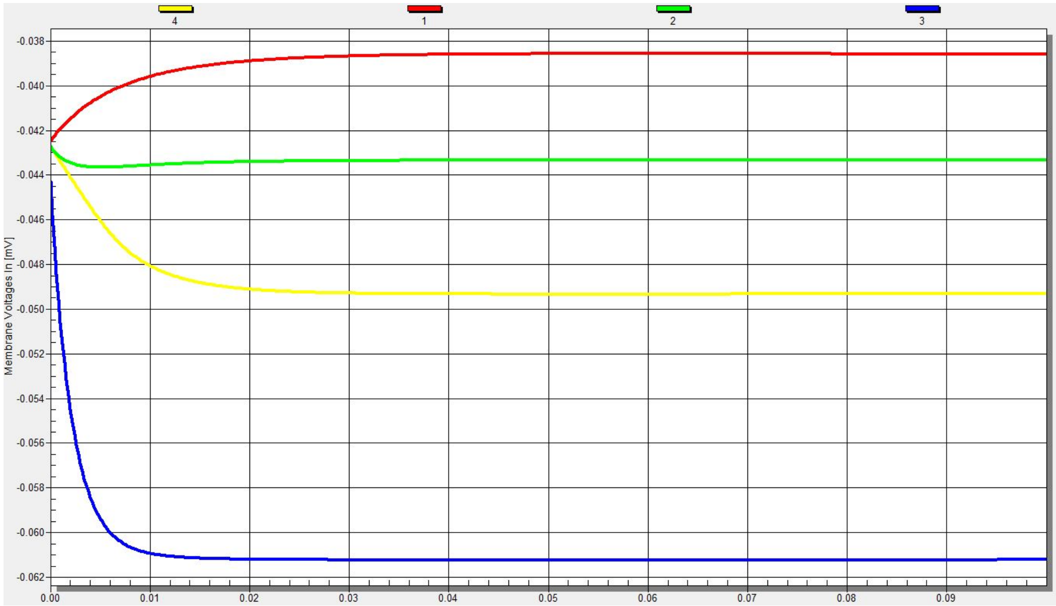

Figure 11: NSNN MV in Animatlab with Ex./In. Synapses & Na Channels (Step Input).

2.6 Example 4: CPG of NSNs

Now that we have described a NSNN with mixed excitatory & inhibitory synapses and sodium channels, we can fully model a CPG network. The CPG network we seek to describe is that shown below.

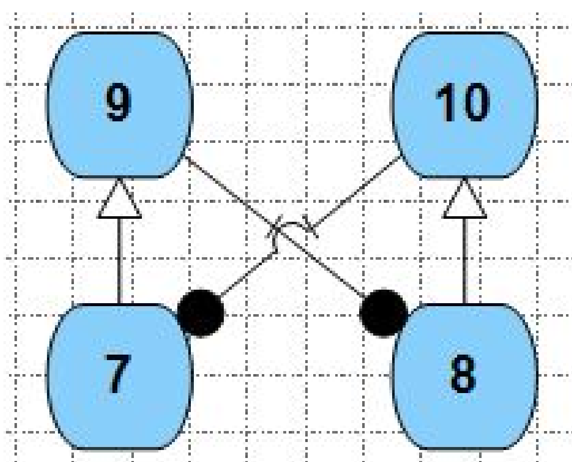

Figure 12: NS CPG Model.

For our CPG, Neurons 7 and 8 are half-centers, while Neurons 9 and 10 are interneurons. The half-center neurons have sodium channels, while the interneurons have no sodium channels. As such, we can simplify our general NSNN model shown in Equations (18) and (19) to apply to this specific case as shown below.

$$
\begin{aligned}
\dot{V}_1 = & \frac{G_{m,1}}{C_{m,1}}(E_{r,1} - V_1) \\
& + \frac{g_{4,1,max}}{C_{m,1}} \min\left(\max\left(\frac{V_4 - E_{lo,4,1}}{E_{hi,4,1} - E_{lo,4,1}}, 0\right), 1\right)(E_{s,4,1} - V_1) \\
& + \frac{G_{Na,1}}{C_{m,1}}\left(\frac{1}{1 + A_{m,1}e^{S_{m,1}(V_1 - E_{m,1})}}\right) h_1(E_{Na_1} - V_1) + \frac{I_{app,1}}{C_{m,1}},
\end{aligned} \tag{34}
$$

$$
\dot{h}_1 = \frac{1 - h_1\left(1 + A_{h,1}e^{S_{h,1}(V_1 - E_{h,1})}\right)}{\tau_{h,max,1} h_{\infty,1}\sqrt{A_{h,1}e^{S_{h,1}(V_1 - E_{h,1})}}\left(1 + A_{h,1}e^{S_{h,1}(V_1 - E_{h,1})}\right)}, \tag{35}
$$

$$
\begin{aligned}
\dot{V}_2 = & \frac{G_{m,2}}{C_{m,2}}(E_{r,2} - V_2) \\
& + \frac{g_{1,2,max}}{C_{m,2}} \min\left(\max\left(\frac{V_1 - E_{lo,1,2}}{E_{hi,1,2} - E_{lo,1,2}}, 0\right), 1\right)(E_{s,1,2} - V_2) \\
& + \frac{G_{Na,2}}{C_{m,2}}\left(\frac{1}{1 + A_{m,2}e^{S_{m,2}(V_2 - E_{m,2})}}\right) h_2(E_{Na_2} - V_2) + \frac{I_{app,2}}{C_{m,2}},
\end{aligned} \tag{36}
$$

$$
\begin{aligned}
\dot{V}_3 = & \frac{G_{m,3}}{C_{m,3}}(E_{r,3} - V_3) \\
& + \frac{g_{2,3,max}}{C_{m,3}} \min\left(\max\left(\frac{V_3 - E_{lo,2,3}}{E_{hi,2,3} - E_{lo,2,3}}, 0\right), 1\right)(E_{s,2,3} - V_3) \\
& + \frac{G_{Na,3}}{C_{m,3}}\left(\frac{1}{1 + A_{m,3}e^{S_{m,3}(V_3 - E_{m,3})}}\right) h_3(E_{Na_3} - V_3) + \frac{I_{app,3}}{C_{m,3}},
\end{aligned} \tag{37}
$$

$$
\dot{h}_3 = \frac{1 - h_3\left(1 + A_{h,3}e^{S_{h,3}(V_3 - E_{h,3})}\right)}{\tau_{h,max,3} h_{\infty,3}\sqrt{A_{h,3}e^{S_{h,3}(V_3 - E_{h,3})}}\left(1 + A_{h,3}e^{S_{h,3}(V_3 - E_{h,3})}\right)}. \tag{38}
$$

$$
\begin{aligned}
\dot{V}_4 = & \frac{G_{m,4}}{C_{m,4}}(E_{r,4} - V_4) \\
& + \frac{g_{3,4,max}}{C_{m,4}} \min\left(\max\left(\frac{V_4 - E_{lo,3,4}}{E_{hi,3,4} - E_{lo,3,4}}, 0\right), 1\right)(E_{s,3,4} - V_4) \\
& + \frac{I_{app,4}}{C_{m,4}},
\end{aligned} \tag{39}
$$

$$
\tag{40}
$$

3 Neuron - DC Motor Interface

3.1 Neuron - DC Motor Interface Mathematical Model

Now that we have mathematically described the behavior of both single neurons and networks of neurons, we will consider the interface between a single non-spiking neuron and a DC motor. Since we want to use our neural network to control a physical system, it is important that we are able to determine how the properties of this interface affect the response of the physical system. In general, the electronics associated with a DC motor has the transfer function

$$\frac{I}{V} = \frac{1}{L_m s + R_m} \equiv \text{Motor Electronics Transfer Function.} \tag{41}$$

Similarly, the motor and load mechanics have a transfer function

$$\frac{\omega}{T} = \frac{s}{J_{eq}s^2 + D_{eq}s + K_{eq}} \equiv \text{Motor \& Load Mechanics Transfer Function,} \tag{42}$$

where J_{eq}, D_{eq}, and K_{eq} are equivalent moments of inertia, damping ratios, and spring constants defined by

$$J_{eq} = J_m + \left(\frac{N_1}{N_2}\right)^2 J_{load}, \tag{43}$$

$$D_{eq} = D_m + \left(\frac{N_1}{N_2}\right)^2 D_{load}, \tag{44}$$

$$K_{eq} = K_m + \left(\frac{N_1}{N_2}\right)^2 K_{load}. \tag{45}$$

With a motor torque constant of K_t, a back-emf constant of K_{emf}, an amplifier gain of K_{amp}, and a position encoder gain of K_{pos}, the complete motor transfer function is

$$G_{motor}(s) = \frac{K_{amp}K_tK_{pos}}{L_mJ_{eq}s^3 + (L_mD_{eq} + R_mJ_{eq})s^2 + (L_mR_{eq} + R_mD_{eq} + K_tK_{emf})s + R_mR_{eq}}. \tag{46}$$

Considering a neuron with only an applied current (i.e., no incoming synapses or persistent sodium channels), the transfer function relating input current and the neuron's membrane voltage above its resting potential is

$$\frac{U_n(s)}{I_{app}(s)} = \frac{1}{C_n s + G_n}. \tag{47}$$

So the actual membrane voltage is related to the applied current by

$$V_n(s) = \frac{I(s)}{C_n s + G_n} + E_r. \tag{48}$$

Finally, we need to reconcile the feedback from the encoder, which is likely in units of voltage, with the current input to the adapter neuron. As such, we will represent the desired DC motor angle with a voltage, V_{in}, and then convert the error, V_{err}, into a current by using a gain as a conversion factor, K_n. With this in mind, the complete open loop transfer function for the neuron-DC motor system is

$$G(s) = \frac{k_6 s + k_7}{k_1 s^4 + k_2 s^3 + k_3 s^2 + k_4 s + k_5} \tag{49}$$

where

$$\begin{aligned}
k_1 &= C_n L_m J_{eq}, && (50)\\
k_2 &= C_n(L_m D_{eq} + R_m J_{eq}) + G_n L_m J_{eq}, && (51)\\
k_3 &= C_n(L_m K_{eq} + R_m D_{eq} + K_t K_{emf}) + G_n(L_m D_{eq} + R_m J_{eq}), && (52)\\
k_4 &= C_n R_m K_{eq} + G_n(L_m K_{eq} + R_m D_{eq} + K_t K_{emf}), && (53)\\
k_5 &= G_n R_m K_{eq}, && (54)\\
k_6 &= E_r C_n K_{amp} K_t K_{pos}, && (55)\\
k_7 &= (E_r G_n + K_n) K_{amp} K_t K_{pos}. && (56)
\end{aligned}$$

4 Nullclines & Equilibria of NSNN Model

While we can solve our NSNN mathematical model using numerical methods, this procedure is time consuming if we want to run many different simulations with different neural parameters. Instead, we can gain insight into the behavior of the system by considering the nullclines and equilibria of the system.

5 Questions

1. What transfer function / state space model is being used to generate the frequency response shown in Figure 3? Is it just Equation (38)? If so, then the only section of the network being considered is that in the boxed diagram, and I feel confident that I can replicate this result (despite the fact that when I attempt to, my results look different...).

2. If Figure 3 is just based on Equation (38), then what was the point of deriving Equation (35)? Equation (38) could have been arrived at without Equation (35). Most of my confusion with this section comes from the interpretation of Equation (35). Why are the conductances in the denominator DC and those in the numerator AC?

3. If I understand Figure 3, then I see that this could be used to set the neuron-servo synapse properties. However, this approach would need to be repeated for DC motors or your artificial muscles to use these types of actuators. I have already done this for a DC motor on paper with what I believe are reasonable results, but I have not typed them.

4. Why is the servo-motor assumed to use P control? Is this a property of the servo motor or a design choice? If it is a design choice, why not use PID control?

5. The neuron equations are non-linear once you include a network of neurons (due to the synapse equation). They become even more non-linear once you add the sodium channels. This makes it difficult to solve for the nullclines and equilibria by hand.

6. How are the nullclines/equilibria being computed? Are you using a numerical root finding method, such as Newton's method or Secant method to deal with the nonlinearity? It almost seems like the paper is suggesting that the equations are linear...

7. Do you have any particular strategy for determining the quantity of equilibria you expect to find? i.e., in order to use a numerical root finding method, you need to have an idea of how many you expect to find before you start so that you can define your stopping conditions.

8. To expand on the above, there appear to be either 0, 1, or 2 equilibria, which may or may not be stable. How do you know which result you will get in each regime?

9. Are solutions to the linearized equations near the equilirbia points useful?

10. I understand the interpretation of Figure 5. Clearly δ is a bifurcation parameter. I conceptually understand the behavior of the system in the $\delta < 0$ and $\delta \in (0, 5)$ regimes. However, I do not conceptually understand the $\delta > 5$ regime. What is happening conceptually in this regime? Why is there only one stable equilibrium?

11. I understand the CPG speed descending command, but what about the intended rotation descending command? If the CPG is constantly oscillating, how would one be able to specify a specific intended rotation at any given time? Wouldn't the system merely keep oscillating? I could see maybe changing the point about which the system is oscillating, but not specifying joint angles.