

Modeling Subcutaneous Stimulation of Spinal Cord for Neuropathic Pain Treatment

Mieko Hirabayashi, Ladan Naghavian, Sareh Manouchehri

Department of Bioengineering
University of California, San Diego
La Jolla, CA 92093

mieko.hirabayashi@gmail.com, lnaghavian@gmail.com, sareh85@gmail.com

Abstract

Spinal stimulation for neuropathic treatment was modeled by incorporating and inhibitory junctions and a volume conductor into Hodgkin-Huxley and Frankenhaeuser-Huxley axon models. Results indicate successful merging of the models at low resistivity values but also show that these two mechanisms operate independently of one another. At high resistivity values, the model no longer produces reliable results, most likely due to the effect of capacitive and inductive forces that are produced by thick layers of tissue.

1 Introduction

Stimulation of nerves using electrical currents has become one of the most effective therapies for Spinal Cord Injuries (SCI) [1]. Some of the spinal cord injuries cannot be treated with opioid and oral medicines. Two major groups of pain can be developed after spinal cord injuries, nociceptive and neuropathic pains. Neuropathic pain is a chronic pain due to the nerve damage which results from any type of lesion including infectious, neoplastic, or vascular lesions. These damaged nerves can signal other nerves incorrectly which results in dysfunction of the nervous system at various levels ranging from the dorsal horn to the brain [3]. CNS related neuropathic pain is often difficult to relieve. The patients with neuropathic pain usually are offered by various pharmacological treatments but despite having the numerous treatment options, they often respond poorly to standard pain treatments. Thus, physician offer a combination of therapies to their patients which include spinal cord stimulation (SCS). Spinal cord stimulator is an implantable device that generates a low voltage electrical signal and transmits it to the spinal nerve in the dorsal horn. This interrupts the conduction of the pain signal as it travels up to the brain and replaces it with a mild tingling sensation (paresthesia) [2]. The spinal cord stimulation system consisted of a pulse generator, two electrodes which are connected to the stimulator via conducting wires, and a remote control which allows the patient to set the parameters [4]. The patient often undergoes a trial implantation, and the permanent unit will be used only if trial procedure results in 50% of pain relief for the patients [4]. The physiological pathway of SCS has not been fully understood but many studies have shown an increased release of gamma amino butyric acid (GABA) during the stimulation. GABA is the main inhibitory neurotransmitter in the central nervous system and its level is significantly lower in patients with neuropathic pain. The decrease in the GABA level is often accompanied by an increase in the release of glutamine (excitatory neurotransmitter). SCS reduces the pain by balancing the release of GABA and glutamine [5]. It is important for the patients to know that SCS does not eliminate the pain, it reduces the intensity of pain which can vary for different patients. SCS in combination with pharmacological treatments can relieve the pain significantly and help the patient to perform their daily activities [3].

In this project we used computer modeling to simulate the mechanism of SCS and its effect on the conduction of pain signal with respect to electrode placement and GABA inhibition. We used Hodgkin-Huxley and Frankenhaeuser-Huxley model to study the difference between myelinated and non-myelinated axons and how they differ in inhibition of pain signal. Additionally, we studied the effect of electrode placement at different tissue layers using the volume conductor model.

2 Methods

In order to fully understand the effects of the each portion of the model on the output, the Hodgkin-Huxley (HH) and Frankenhaeuser-Huxley (FH) model were implemented in steps. First, both models were expressed as linear point models. Next, these same linear equations were used to implement the axon models into the HH and FH equations to generate models for the myelinated and unmyelinated axons. Finally, the volume conductor was integrated into both the HH and FH models followed by the incorporation of a secondary neuron and an inhibitory junction.

2.1 Hodgkin Huxley and Frankenhaeuser Huxley

The Hodgkin-Huxley (HH) model uses gating variables to describe the probability of an ion channel being open or closed. In the HH model the, gating variable n^4 describes the slow activation of a potassium channel, m^3 describes the fast activation of the sodium channel, and h describes slow activation of the sodium channel. These gating variables' closing and opening rates are given by rate equations α and β equations which are functions of membrane voltage and have coefficients obtained experimentally from an unmyelinated squid axon model [6].

The Frankenhaeuser Huxley model was a reformulation of the HH model that used the same concept of gating variable and rate equations α and β to formulate the model but obtained coefficients for the rate equations from a toad myelinated axon [7]. The fitting of this experimental data resulted in an additional gating variable p , which was termed the non-specific permeability constant. This gating variable was a way to describe the movement of sodium in the previous node that would affect the next node's membrane potential, allowing for the model to account for the effect of myelination between nodes and salutatory conduction [7,8]. Equations describing these equations can be seen in Appendix I.

2.2 The point models

HH is often used to model point neurons. These types of models do not incorporate axon length or various nodes allowing for analysis of simple spiking and interaction between two neurons rather than propagation of an action potential. Thus, to investigate the spiking difference between the FH and HH model, both were first used to generate point models. When creating each of these models, equations describing the currents were linearized using a linear approximation of the Goldman-Hodgkin-Katz equation [9]:

$$J'_s = g_s(V_m - E_s) \quad (1)$$

$$g_s = P_s \left(\frac{[s]_o [s]_i}{[s]_o - [s]_i} \right) \ln \left(\frac{[s]_o}{[s]_i} \right) \quad (2)$$

In order to simulate the interaction between two point neurons, an inhibitory junction was used with both FH and HH (figure 1) point models by introducing an inhibitory current J_{GABA} and GABA voltage dependent variable r described by the equations below:

$$J_{GABA} = g_{GABA} * r * (V - E_{Cl}) \quad (3)$$

$$\frac{dr}{dt} = \alpha_r * \left(\frac{T_{max}}{1 + \exp\left(\frac{V - V_p}{K_p}\right)} \right) * ((1 - r) - \beta_r) * r \quad (4)$$

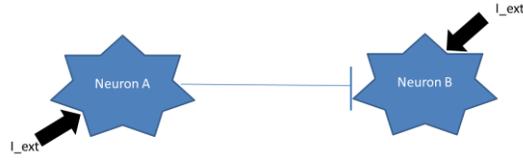


Figure 1: Point Neuron with Inhibitory Junction

2.3 The axon models

Incorporating axons into each model was done by adding nodes to the voltage equations along with the axonal area (given by $\pi L \Delta x$), and the axoplasm resistance r_s [10] a common method used to implement an axon in FH and HH models.

$$\frac{dV_n}{dt} = \frac{1}{c_m} \left(\frac{1}{r_s \pi L \Delta x} (V_{n-1} - 2V_n + V_{n+1} - (J_{Na} + J_K + J_L + J_p)_n) \right) \quad (5)$$

2.4 Volume conductor

The volume conductor is a means of describing the propagation of an external current through various types of tissue. The simplest form of this model is to consider one type of homogeneous tissue with a resistivity of ρ_{tissue} . An electrode placed a vertical distance z and horizontal distance x would thus provide additional voltage to each node V_e (as seen in figure 2) is given by the equation:

$$V_e = \frac{\rho}{4\pi} * (z^2 + x^2)^{-\frac{1}{2}} * I_{ext} \quad (6) \quad [10]$$

This changes the axon model equation to:

$$\frac{dV_n}{dt} = \frac{1}{c_m} * \frac{1}{r_s \pi L \Delta x} \left(V_{n-1} - 2V_n + V_{n+1} + V_{e,n-1} - 2V_{e,n} + V_{e,n+1} - (J_{Na} + J_K + J_L + J_p)_n \right) \quad (7)$$

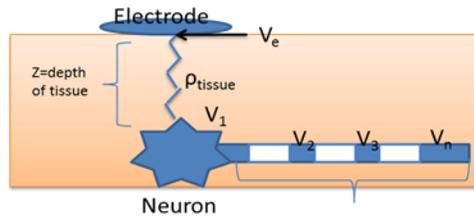


Figure 2: Volume Conductor

2.5 Inhibitory junction

In order to create an inhibitory junction (shown in figure 3), an addition neuron with nodes was added to the model. In this model, the first neuron can be described by equation (7). In order to add an inhibitory current between the two neurons, J_{GABA} , described in equations 3 and 4, was added to the second neuron.

$$\frac{dV2_n}{dt} = \frac{1}{c_m} \left(\frac{1}{r_s \pi L \Delta x} (V2_{n-1} - 2V2_n + V2_{n+1} + (J_{Na} + J_K + J_L + J_p + J_{GABA})_n) \right) \quad (8)$$

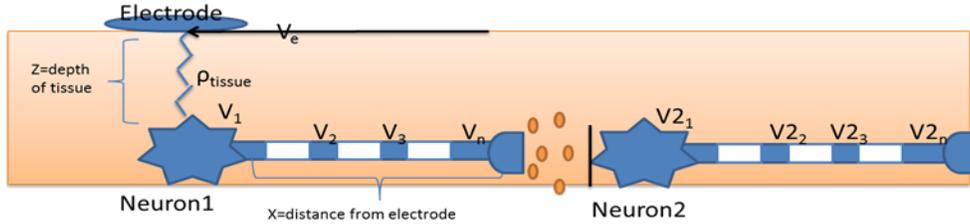


Figure 3: Inhibitory Junction between two myelinated neurons stimulated by a transcutaneous electrode

3 Results

3.1 The point models

In the Hodgkin-Huxley point model the action potential is initiated around the membrane potential (-60mV) spikes up to 40 mV and returns to the resting membrane potential before another action potential occurs. In the FH model for the point, it can be seen that the action potential was fired around the membrane potential, but the spikes do not return to the membrane potential. This discrepancy is due to the permeability constant p , in which it takes to the account the additional Na ions generated from previous nodes. Since this model does not have any nodes, the p gating variable simulates the release of sodium at a later time after the initial spike has already started to return to membrane potential (n , m , h gates are closing). This release of sodium keeps the relative membrane potential positive at subsequent spikes.

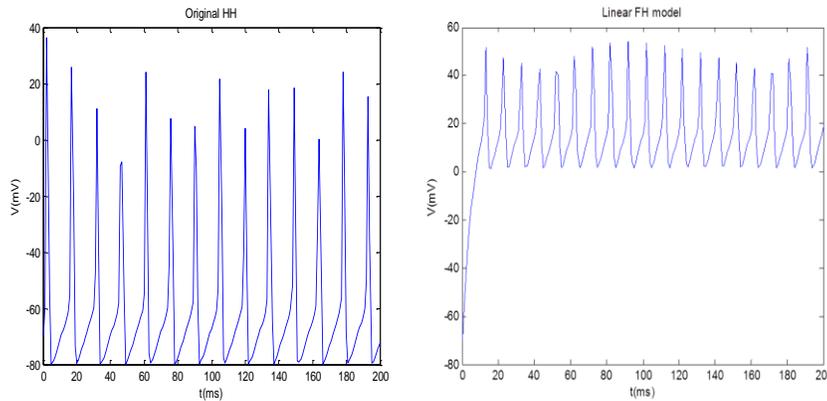


Figure 4: HH and FH Point Models

In Figure 5 the effect of GABA was investigated. Here, it can be seen that release of GABA inhibited the action potentials from spiking in both HH and FH model. In HH and FH model the conductivity of GABA (g_{GABA}), which is related to the relative concentrations of GABA, was raised to 10 and 0.45 respectively in order for the action potential to be inhibited. In HH model the gating variables m and n are more powerful in J_k and J_{Na} , since they have the order of 3 and 4 respectively. This requires more GABA to be released in order for the action potential to be inhibited. In FH model however, the power of m and n is 2 and the permeability constant p of the excess sodium ions from previous nodes is effective after the gating variable, thereby inhibiting the action potential require less release of GABA.

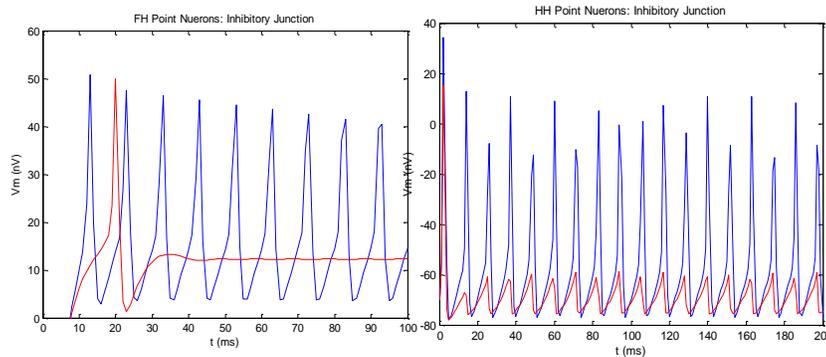


Figure 5: HH and FH Point Neurons with Inhibitory Junction

3.2 The Axon Models

In this part the axon model and the effect of myelination in propagation of action potential were investigated. As it can be seen in figure 6a, in the HH model, the action potential is propagated through the axon from node to node. Furthermore, due to the high order of the Na^+ related gating variables, m and h , the initial node continues to spike but does not propagate after the initial spike. The FH model also shows propagation along the axon, however due to the myelination, the action potential propagated faster from node to node. Also the oscillatory behavior that is shown in the FH model is due to the excess sodium ions (p gating variable) from the previous nodes, which cause a little spike.

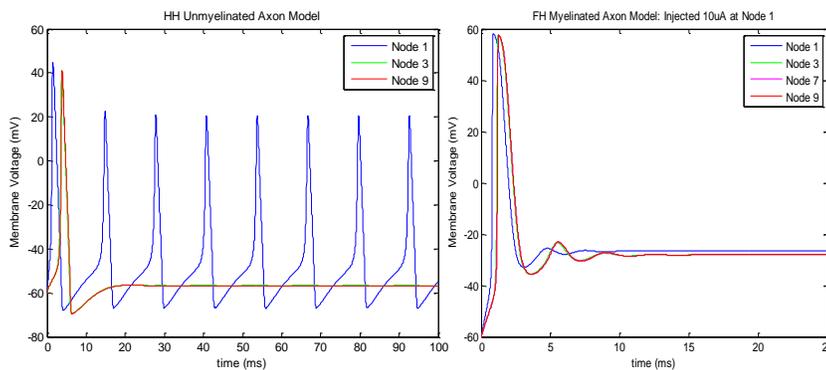


Figure 6: The (a) Unmyelinated Axon and the (b) Myelinated Axon

3.3 Volume Conductor

By incorporating the volume conductor, the action potential is propagated from node to node in both HH and FH model. As it can be seen in the FH model the propagation is faster than HH model and that is due to the myelination. Also the oscillatory behavior that it can be seen in node 3 and 7 of the FH model is due to the influx of excess sodium ions from previous nodes. The effects of different tissues were investigated by changing the resistivity of the tissues such as fat and muscle, and the similar results were observed.

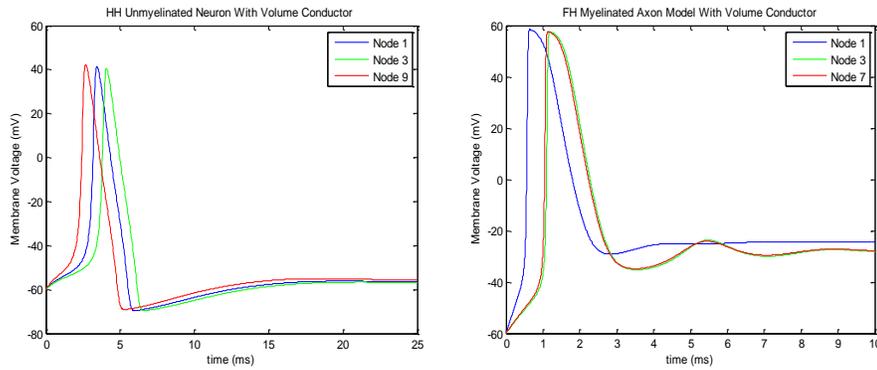


Figure 7: The (a) Unmyelinated Axon and (b) Myelinated Axon stimulated through muscle tissue

3.4 Inhibitory

By incorporating the effect of GABA released from the first neuron after stimulation, the action potential is inhibited in second neuron as it is illustrated in figure 8. It can be seen that in HH model the propagation stops after node 3 and the action potential is inhibited by firing to lower membrane potential values. In FH model however the effect of release of GABA is more dramatic, since the membrane potential is around the resting potential at negative values. Though these results incorporate the volume conductor, the axon, and GABA, the effect of GABA is similar to what was seen in the point model in section 3.1.

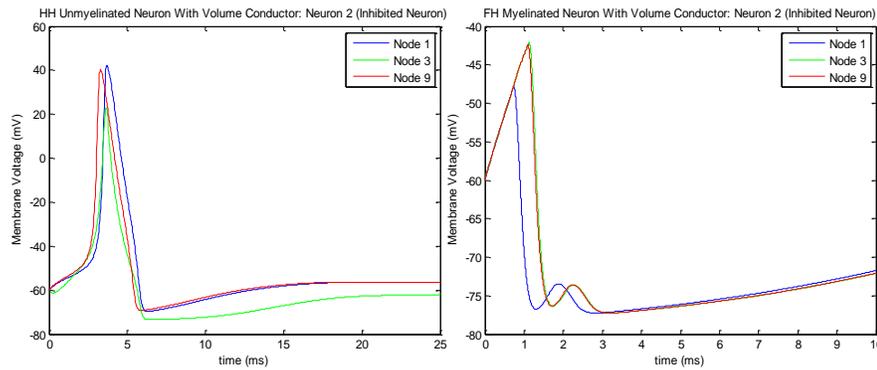


Figure 8: Inhibited Neuron using (a) HH axon model and (b) FH model with a stimulus through muscle tissue.

4 Conclusions

In this paper, the commonly used Hodgkin-Huxley and Frankenhaeuser-Huxley models were investigated and implemented into a multidimensional model that incorporated neuron excitation through tissue using the volume conductor model, nodal propagation down myelinated and unmyelinated axons, and GABA induced inhibition. This study showed that these portions can be incorporated together to model spinal cord stimulation, allowing for simulation of various types of electrodes, subcutaneously implanted, that could be used to stimulate specific portions of the spinal cord. However, when resistivity in the volume conductor becomes too high, the model no longer accurately represents the physical phenomena, and thus, in order to use this model to simulate a transcutaneous stimulation, a more detailed model for the volume conductor that takes into account conductance and inductance of each tissue layer, would be needed.

Acknowledgments

A MATLAB code from <http://people.ece.cornell.edu> by Amie Adams was modified to generate axon models.

References

- [1] Creasey, G. H., C. H. Ho, R. J. Triolo, D. R. Gater, A. F. DiMarco, K. M. Bogie, and M. W. Keith. "Clinical applications of electrical stimulation after spinal cord injury." *The journal of spinal cord medicine* 27, no. 4 (2004): 365.
- [2] Medtronic For Healthcare Professionals, "Spinal Cord Stimulation." Last modified 2013. Accessed December 3, 2013. <http://professional.medtronic.com/pt/neuro>
- [3] Vranken, Jan H. "Mechanisms and treatment of neuropathic pain." *Central Nervous System Agents in Medicinal Chemistry* 9, no. 1 (2009): 71-78.
- [4] American Association of Neurological Surgeons, "Spinal Cord Stimulation ." Last modified October 2008. Accessed December 5, 2013. <http://www.aans.org/PatientInformation/ConditionsandTreatments/SpinalCordStimulation.aspx>.
- [5] Meyerson, Bjorn A., and Bengt Linderoth. "Mechanisms of spinal cord stimulation in neuropathic pain." *Neurological research* 22, no. 3 (2000): 285.
- [6] Gert, Cauwenberghs. University of California, San Diego, "BENG 260 Lecture/Week 1." Last modified September 27, 2013. <http://www.isn.ucsd.edu/classes/beng260/lectures/week1.pdf>.
- [7] Poznanski, Roman R. "Analytical Solution of the Frankenhaeuser-Huxley Equations I: Minimal Model for Backpropagation of Action Potentials in Sparsely Excitable Dendrites" *Journal of Integrative Neuroscience* 3, no. 03 (2004): 267-299. [8] Tuckwell, Henry C. *Introduction to Theoretical Neurobiology: Volume 2, Nonlinear and Stochastic Theories*. Cambridge: Cambridge University Press, 2005.
- [9] Gert, Cauwenberghs. University of California, San Diego, "BENG 260 HW/Week 1." Last modified October 4, 2013. http://www.isn.ucsd.edu/classes/beng260/complab/week1/hw1_2013.pdf
- [10] Rattay, Frank. "Analysis of models for external stimulation of axons." *Biomedical Engineering, IEEE Transactions on* 10 (1986): 974-977.

Appendix I: Equations

Hodgkin-Huxley modeling:

$$\frac{dV}{dt} = \frac{1}{C_m} * (J_{ext} - J_{Na} - J_K - J_L)$$

Rate Equations

$$\alpha_m = \left(0.1 * \frac{V + 45}{1 - \exp\left(-\frac{V + 45}{10}\right)} \right)$$

$$\beta_m = \left(4 * \exp\left(-\frac{V + 70}{18}\right) \right)$$

$$\alpha_h = 0.07 * \exp\left(-\frac{V + 70}{20}\right)$$

$$\beta_h = \frac{1}{1 + \exp\left(\frac{-(V + 40)}{10}\right)}$$

$$\alpha_n = 0.01 * \frac{V + 60}{1 - \exp\left(\frac{-(V + 60)}{10}\right)}$$

$$\beta_n = 0.125 * \exp\left(-\frac{V + 70}{80}\right)$$

Current Equation:

$$J_{Na} = g_{Na} * m^3 * h * (V - E_{Na})$$

$$J_K = g_K * n^4 * (V - E_K)$$

$$J_L = g_L * (V - E_L)$$

Gating Variables:

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

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

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

$$\frac{dV}{dt} = \frac{1}{C_m} * (J_{ext} - J_{Na} - J_K - J_L - J_P)$$

$$J_{Na} = \bar{P}_{Na} h m^2 \left(\frac{EF^2}{RT} \right) \frac{[Na]_o - [Na]_i \exp(EF/RT)}{(1 - \exp(EF/RT))}$$

$$J_K = \bar{P}_K n^2 \left(\frac{EF^2}{RT} \right) \frac{[K]_o - [K]_i \exp(EF/RT)}{(1 - \exp(EF/RT))}$$

$$J_P = \bar{P}_{Na} P^2 \left(\frac{EF^2}{RT} \right) \frac{[Na]_o - [Na]_i \exp(EF/RT)}{(1 - \exp(EF/RT))}$$

$$J_L = g_L * (V - E_L)$$

Linearization:

$$J'_s = g_s(V_m - E_s)$$

$$g_s = P_s \left(\frac{[S]_o [S]_i}{[S]_o - [S]_i} \right) \ln \left(\frac{[S]_o}{[S]_i} \right)$$

Frankenhaeuser -Huxley modeling:

Rate Equations:

$$\alpha_m = \frac{A_1(V - B_1)}{1 - \exp\left(\frac{B_1 - V}{C_1}\right)}$$

$$\beta_m = \frac{A_2(B_2 - V)}{1 - \exp\left(\frac{V - B_2}{C_2}\right)}$$

$$\alpha_h = \frac{A_3(B_3 - V)}{1 - \exp\left(\frac{V - B_3}{C_3}\right)}$$

$$\beta_h = \frac{A_4}{1 + \exp\left(\frac{B_4 - V}{C_4}\right)}$$

$$\alpha_n = \frac{A_5(V - B_5)}{1 - \exp\left(\frac{B_5 - V}{C_5}\right)}$$

$$\beta_n = \frac{A_6(B_6 - V)}{1 - \exp\left(\frac{V - B_6}{C_6}\right)}$$

$$\alpha_p = \frac{A_7(V - B_7)}{1 - \exp\left(\frac{B_7 - V}{C_7}\right)}$$

$$\beta_p = \frac{A_8(B_8 - V)}{1 - \exp\left(\frac{V - B_8}{C_8}\right)}$$

Volume Conductor:

$$\frac{dV_n}{dt} = \frac{1}{c_m} \left(\frac{1}{r_s \pi L \Delta x} (V_{n-1} - 2V_n + V_{n+1} + V_{e,n-1} - 2V_{e,n} + V_{e,n+1} - (J_{Na} + J_K + J_L + J_p)_n) \right)$$

$$V_e = \frac{\rho_e}{4\pi} * (z^2 + x^2)^{-\frac{1}{2}} * I_{ext}(t)$$

GABA:

$$\frac{dV2_n}{dt} = \frac{1}{c_m} \left(\frac{1}{r_s \pi L \Delta x} (V2_{n-1} - 2V2_n + V2_{n+1} + (J_{Na} + J_K + J_L + J_p + J_{GABA})_n) \right)$$

$$J_{GABA} = g_{GABA} * r * (V - E_{Cl})$$

$$\frac{dr}{dt} = \alpha_r * \left(\frac{T_{max}}{1 + \exp\left(-\frac{V - V_p}{K_p}\right)} \right) * ((1 - r) - \beta_r) * r$$

Appendix II: FH Point Model

```
close all;clear all; clc
% Constants
C_m = 2.0;% membrane capacitance, in uF/cm^2
V_t=26.7
%Ion concentrations mM
C_o_Na=114
C_i_Na=13
C_o_K=2.5
C_i_K=12

Ri=110 %ohmcm axoplasam resistance
E_L = 0.026%leak current potential mV
E_r= -70 %rest potential mV
E_Na = V_t*log(C_o_Na/C_i_Na) % Nernst reversal potentials, in mV
E_K = V_t*log(C_o_K/C_i_K)
E_P = V_t*log(C_o_Na/C_i_Na)
%permeabilities
P_Na=8; %cm/ms
P_p=0.54; %cm/ms
P_K=1.2; %cm/ms
%g values
g_K=P_K*((C_o_K*C_i_K)/(C_o_K-C_i_K))*log(C_o_K/C_i_K);
g_Na=P_Na*((C_o_Na*C_i_Na)/(C_o_Na-C_i_Na))*log(C_o_Na/C_i_Na)
g_L=0.3;
g_P=P_p*((C_o_Na*C_i_Na)/(C_o_Na-C_i_Na))*log(C_o_Na/C_i_Na);
%FH Constants
A1=0.36;%1/ms
B1=22; %mV
C1=3; %mV
A2=0.4; B2=13; C2=20;
A3=0.1; B3=-10; C3=6;
A4=4.5; B4=45; C4=10;
A5=0.02; B5=35; C5=10;
A6=0.05; B6=10; C6=10;
A7=0.006; B7=40; C7=10;
A8=0.09; B8=-25; C8=20;
%
% %gating variables
alpha_m= @(V) (A1*(V-B1))/(1-exp((B1-V)/C1));
beta_m= @(V) (A2*(B2-V))/(1-exp((V-B2)/C2));
alpha_h= @(V) (A3*(B3-V))/(1-exp((V-B3)/C3));
beta_h= @(V) A4/(1+exp((B4-V)/C4));
alpha_n= @(V) (A5*(V-B5))/(1-exp((B5-V)/C5));
beta_n= @(V) (A6*(B6-V))/(1-exp((V-B6)/C6));
alpha_p= @(V) (A7*(V-B7))/(1-exp((B7-V)/C7));
beta_p= @(V) (A8*(B8-V))/(1-exp((V-B8)/C8))

alpha_r= 5; %m/M*ms
beta_r = 0.18;
T_max= 1.5; %mM
K_p=5; %mV
V_p=7; %mV
E_Cl=-80;
T_max= 1; %mM
```

```

K_p=5; %mV
V_p=7; %mV
conc_T= @(V) T_max./(1+exp(-(V-V_p)./K_p));
g_GABA= .045
% Membrane currents (in uA/cm^2) HH

J_Na = @(V,m,h) g_Na .* m.^2*h*(V - E_Na);
J_K = @(V,n) g_K .* n.^2 .* (V - E_K);
J_L = @(V) g_L .* (V - E_L);
J_P = @(V,p) p^2*g_P*(V-E_Na);
J_GABA= @(V,r) g_GABA*r*(V-E_Cl);
J_ext=0;

%Initial Conditions
m0=0.0005;
h0=0.8249;
n0=0.0268;
p0=0.0049;
V0=-70;

%time vector s
t_start = 0;
t_stop =100;
t_step = 1;

%differential gating equations
dmdt=@(V,m) alpha_m(V)*(1-m)-beta_m(V)*m;
dhdt=@(V,h) alpha_h(V)*(1-h)-beta_h(V)*h;
dndt=@(V,n) alpha_n(V)*(1-n)-beta_n(V)*n;
dpdt=@(V,p) alpha_p(V)*(1-p)-beta_p(V)*p;
dVdt=@(V,m,h,n,p,r) ((J_ext-J_Na(V,m,h)-J_K(V,n)-J_L(V)-
J_P(V,p)))/C_m;
dmdt2=@(V,m) alpha_m(V)*(1-m)-beta_m(V)*m;
dhdt2=@(V,h) alpha_h(V)*(1-h)-beta_h(V)*h;
dndt2=@(V,n) alpha_n(V)*(1-n)-beta_n(V)*n;
dpdt2=@(V,p) alpha_p(V)*(1-p)-beta_p(V)*p;
drdt2=@(V,r) alpha_r.*(T_max./(1+exp(-(V-V_p)./K_p))).*(1-r)-
beta_r.*r;
dVdt2=@(V,m,h,n,p,r) (J_ext-J_Na(V,m,h)-J_K(V,n)-J_L(V)-J_P(V,p)-
J_GABA(V,r))/C_m;

IC=[V0 m0 h0 n0 p0 V0 m0 h0 n0 p0 8.7e-7];

dxdt = @(t, x) ...
[ ...
dVdt(x(1,:),x(2,:),x(3,:),x(4,:),x(5,:)); ...
dmdt(x(1,:),x(2,:)); ...
dhdt(x(1,:),x(3,:)); ...
dndt(x(1,:),x(4,:)); ...
dpdt(x(1,:),x(5,:));...
dVdt2(x(6,:),x(7,:),x(8,:),x(9,:),x(10,:), x(11,:)); ...
dmdt2(x(6,:),x(7,:)); ...
dhdt2(x(6,:),x(8,:)); ...
dndt2(x(6,:),x(9,:)); ...
dpdt2(x(6,:),x(10,:));...
drdt2(x(1,:),x(11,:));];

```

```

[t, x] = ode23(dxdt, t_start:t_step:t_stop, IC);

V = x(:,1); % the first column is the V values
m = x(:,2); % the second column is the m values
h = x(:,3); % the second column is the h values
n = x(:,4); % the second column is the h values
p = x(:,5); % the second column is the p values
V2 = x(:,6); % the first column is the V values
m2 = x(:,7); % the second column is the m values
h2 = x(:,8); % the second column is the h values
n2 = x(:,9); % the second column is the h values
p2 = x(:,10); % the second column is the p values
r = x(:,10); % the second column is the p values
figure()
plot(t,V)
hold on
plot(t,V2, 'r')
ylim([0 60])
xlabel('t (ms)')
ylabel('Vm (nV)')
title('FH Point Nuerons: Inhibitory Junction')

hold off
figure()
plot( t,r)

```

Appendix III: FH Model with Volume Conductor and an Inhibitory Junction

```
function main()
DT = 0.00025;
TMAX = 10;
t = 0:DT:TMAX;

VREST = -60;

GNA = 254.87;    % mS/cm^2
GK = 5.94;      % mS/cm^2
GLEAK = 0.3;    % mS/cm^2 0.3
GP = 17.2040;

ENA = 57.9724;  % mV
EK = -41.8820;  % mV
ELEAK = 0.0260; % mV
EP = 57.9724;

C_M = 1.0;      % uF/cm^2

g_GABA= 10.5;
E_Cl=-80;
T_max= 1; %mM
K_p=5; %mV
V_p=7; %mV
alpha_r= 5; %m/M*ms
beta_r = 0.18;

%set the axon properties
specR = 90; % ohm cm
dia = .05; % cm 0.05
nodeDist = 0.05; %cm
Raxon = specR*nodeDist/(pi*(dia^2)/4) ; % R=specR*length/area
gAxon = 1000/Raxon ; %invert and convert to milliS
nNodes = 4 ;
nTime = length(t) ;
```

```

INJECT_LIST = 0;
z=0.1 %cm thickness to electrode
I_ext=10 %current applied through electrode
rho_e=200

for j=1:nNodes
x=j*nodeDist;
    ve(j)=rho_e*I_ext*(1/(4*pi*(x^2+z^2)^(1/2)));
end

clear v m h n % clear old variables
v = ones(nTime, nNodes)*VREST; % initial membrane voltage
m = ones(nTime,nNodes).*alpha_m(v-VREST)./(alpha_m(v-VREST)+beta_m(v-VREST));
%initial (steady-state) m
h = ones(nTime,nNodes).*alpha_h(v-VREST)./(alpha_h(v-VREST)+beta_h(v-VREST));
%initial (steady-state) h
n = ones(nTime,nNodes).*alpha_n(v-VREST)./(alpha_n(v-VREST)+beta_n(v-VREST));
%initial (steady-state) n
p = ones(nTime,nNodes).*alpha_p(v-VREST)./(alpha_p(v-VREST)+beta_p(v-VREST));
v2 = ones(nTime, nNodes)*VREST; % initial membrane voltage
m2 = ones(nTime,nNodes).*alpha_m(v-VREST)./(alpha_m(v-VREST)+beta_m(v-VREST));
%initial (steady-state) m
h2 = ones(nTime,nNodes).*alpha_h(v-VREST)./(alpha_h(v-VREST)+beta_h(v-VREST));
%initial (steady-state) h
n2 = ones(nTime,nNodes).*alpha_n(v-VREST)./(alpha_n(v-VREST)+beta_n(v-VREST));
%initial (steady-state) n
p2 = ones(nTime,nNodes).*alpha_p(v-VREST)./(alpha_p(v-VREST)+beta_p(v-VREST));
r2= ones(nTime,nNodes).*8.7e-7;
for i=2:length(t)
    I = [INJECT_LIST, zeros(1,nNodes-1)] ;
    M = m(i-1,:); % get values from last time step
    H = h(i-1,:); % (hopefully this substitution makes
    N = n(i-1,:); % the following code a bit easier to read)
    V = v(i-1,:);
    P = p(i-1,:);

    M2 = m2(i-1); % get values from last time step
    H2 = h2(i-1); % (hopefully this substitution makes
    N2 = n2(i-1); % the following code a bit easier to read)
    P2 = p2(i-1,:);
    R2 = r2(i-1,:);
    V2 = v2(i-1, :);

```

```

gNa = GNA * M.^2 .* H ;
gK = GK * N.^2;
gP = GP * P.^2;

gNa2 = GNA * M2.^2 .* H2 ;
gK2 = GK * N2.^2;
gP2 = GP * P2.^2;

%

mdot = alpha_m(V-VREST).*(1-M) - beta_m(V-VREST).*M;
hdot = alpha_h(V-VREST).*(1-H) - beta_h(V-VREST).*H;
ndot = alpha_n(V-VREST).*(1-N) - beta_n(V-VREST).*N;
pdot = alpha_p(V-VREST).*(1-P) - beta_m(V-VREST).*P;
vdot = (I - gNa.*(V-ENA) - gK.*(V-EK) -gP.*(V-ENA)- GLEAK.*(V-ELEAK))/C_M;

mdot2 = alpha_m(V2-VREST).*(1-M2) - beta_m(V2-VREST).*M2;
hdot2 = alpha_h(V2-VREST).*(1-H2) - beta_h(V2-VREST).*H2;
ndot2 = alpha_n(V2-VREST).*(1-N2) - beta_n(V2-VREST).*N2;
pdot2 = alpha_p(V2-VREST).*(1-P2) - beta_m(V2-VREST).*P2;
rdot2= alpha_r.*(T_max./(1+exp(-(V-V_p)/K_p))).*(1-R2)-beta_r.*R2;
vdot2 = (-g_GABA.*R2.*(V2-E_Cl) - gNa2.*(V2-ENA) - gK2.*(V2-EK) - GLEAK.*(V2-
ELEAK)-gP.*(V-ENA))/C_M;

for j=2:nNodes-1
    vdot(j) = vdot(j) + gAxon * (V(j-1)-V(j)-V(j)+V(j+1)+ve(j-1)-ve(j)-ve(j)+ve(j+1)) ;
    vdot2(j) = vdot2(j) + gAxon * (V2(j-1)-V2(j)-V2(j)+V2(j+1)) ;
end
vdot(1) = vdot(1) - gAxon * (V(2)-V(1)+ve(2)-ve(1)) ;
vdot(nNodes) = vdot(nNodes) + gAxon * (V(nNodes-1)-V(nNodes)) ;

vdot2(1) = vdot2(1) - gAxon * (V2(2)-V2(1)) ;
vdot2(nNodes) = vdot2(nNodes) + gAxon * (V2(nNodes-1)-V2(nNodes)) ;

```

```

m(i,:) = m(i-1,:) + mdot*DT;           % Euler integration
h(i,:) = h(i-1,:) + hdot*DT;
n(i,:) = n(i-1,:) + ndot*DT;
v(i,:) = v(i-1,:) + vdot*DT;
p(i,:) = p(i-1,:) + pdot*DT;
m2(i,:) = m2(i-1,:) + mdot2*DT;       % Euler integration
h2(i,:) = h2(i-1,:) + hdot2*DT;
n2(i,:) = n2(i-1,:) + ndot2*DT;
v2(i,:) = v2(i-1,:) + vdot2*DT;
p2(i,:) = p2(i-1,:) + pdot2*DT;
r2(i,:) = r2(i-1,:) + rdot2*DT;
end

```

```

figure(1);clf;
cmap = jet(nNodes);
%plot(t,v)
for j=1:nNodes
% subplot(nNodes,1,j);
plot(t,v(:,j),'color',cmap(j,:));
hold on
ylabel('v (mV)')
xlabel('t (ms)')
title('FH Myelinated Axon for 10 nodes')
%xlabel('t (msec)'); ylabel('v (mV)')
set(gca,'ylim',[-60 60])

```

```

end
figure(2)
for j=1:nNodes
% subplot(nNodes,1,j);
plot(t,v2(:,j),'color',cmap(j,:));
hold on
ylabel('v (mV)')
xlabel('t (ms)')
title('FH Myelinated Axon for 10 nodes')
%xlabel('t (msec)'); ylabel('v (mV)')
set(gca,'ylim',[-60 60])

```

```

end
figure(3)
plot(t, r2)

%get speed by finding peaks for each trace
for j=1:nNodes
    [vPeak(j),index(j)] = max(v(:,j));
end
% m/sec = (cm/100)/(mSec/1000)
10*nodeDist./(t(index(end-2))-t(index(end-3)))

% % gating variable functions
function am=alpha_m(v)
A1=0.36;% 1/ms
B1=22; % mV
C1=3; % mV
am=zeros(size(v));
idx = find(v~=B1);          % HANDLE NORMAL (NON-SINGULAR)
CASE
if(~isempty(idx))
    am(idx)= (A1.*(v(idx)-B1))./(1-exp((B1-v(idx))./C1));
end
idx = find(v == B1);        % HANDLE SINGULARITY AT V=25.0
if(~isempty(idx))
    am(idx)= (A1)./(exp((B1-v(idx))./C1));
end
function bm=beta_m(v)
A2=0.4; B2=13; C2=20;
bm= zeros(size(v));
idx = find(v~=B2);          % HANDLE NORMAL (NON-SINGULAR)
CASE
if(~isempty(idx))
    bm(idx)=(A2.*(B2-v(idx)))./(1-exp((v(idx)-B2)./C2));
end
idx = find(v == B2);        % HANDLE SINGULARITY AT V=25.0

```

```

if(~isempty(idx))
bm(idx)=(A2)./(exp((v(idx)-B2)./C2));
end
function ah= alpha_h(v)
A3=0.1; B3=-10; C3=6;
ah= zeros(size(v));
idx = find(v~=B3); % HANDLE NORMAL (NON-SINGULAR)
CASE
if(~isempty(idx))
ah(idx)= (A3*(B3-v(idx)))./(1-exp((v(idx)-B3)./C3));
end
idx = find(v == B3); % HANDLE SINGULARITY AT V=25.0
if(~isempty(idx))
ah(idx)= (A3)./(exp((v(idx)-B3)./C3));
end
function bh=beta_h(v)
A4=4.5; B4=45; C4=10;
bh= zeros(size(v));
bh= A4./(1+exp((B4-v)./C4));
function an=alpha_n(v)
A5=0.02; B5=35; C5=10;
an=zeros(size(v));
idx = find(v~=B5); % HANDLE NORMAL (NON-SINGULAR)
CASE
if(~isempty(idx))
an(idx)= (A5.*(v(idx)-B5))./(1-exp((B5-v(idx))./C5));
end
idx = find(v == B5); % HANDLE SINGULARITY AT V=25.0
if(~isempty(idx))
an(idx)= (A5)./(1-exp((B5-v(idx))./C5));
end
function bn=beta_n(v)
A6=0.05; B6=10; C6=10;
bn=zeros(size(v));
idx = find(v~=B6); % HANDLE NORMAL (NON-SINGULAR)
CASE
if(~isempty(idx))
bn(idx)=(A6.*(B6-v(idx)))./(1-exp((v(idx)-B6)./C6));
end

```

```

idx = find(v == B6);          % HANDLE SINGULARITY AT V=25.0
if(~isempty(idx))
    bn(idx)=(A6)./(exp((v(idx)-B6)./C6));
end
function ap=alpha_p(v)
A7=0.006; B7=40; C7=10;
ap=zeros(size(v));
idx = find(v~=B7);          % HANDLE NORMAL (NON-SINGULAR)
CASE
if(~isempty(idx))
    ap(idx)=(A7*(v(idx)-B7))./(1-exp((B7-v(idx))./C7));
end
idx = find(v == B7);        % HANDLE SINGULARITY AT V=25.0
if(~isempty(idx))
    ap(idx)=(A7)./(exp((B7-v(idx))./C7));
end
function bp=beta_p(v)
A8=0.09; B8=-25; C8=20;
bp=zeros(size(v));
idx = find(v~=B8);          % HANDLE NORMAL (NON-SINGULAR)
CASE
if(~isempty(idx))
    bp(idx)=(A8*(B8-v(idx)))./(1-exp((v(idx)-B8)./C8));
end
idx = find(v == B8);        % HANDLE SINGULARITY AT V=25.0
if(~isempty(idx))
    bp(idx) = (A8)./(exp((v(idx)-B8)./C8));
end

```