

NEUROMODULATION: ACTION POTENTIAL MODELING

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirement for the Degree

Master of Science in Biomedical Engineering

by

Vladimir Ruzov

March 2014

© 2014
Vladimir Ruzov
ALL RIGHTS RESERVED

COMMITTEE MEMBERSHIP

TITLE: Neuromodulation: Action Potential Modeling

AUTHOR: Vladimir Ruzov

DATE SUBMITTED: March 2014

COMMITTEE CHAIR: Robert B. Szlavik, Ph.D.
Assistant Professor of Biomedical Engineering

COMMITTEE MEMBER: Lanny Griffin, Ph.D.
Professor of Biomedical Engineering

COMMITTEE MEMBER: David Clague, Ph.D.
Biomedical Engineering Graduate Coordinator

ABSTRACT

Neuromodulation: Action Potential Modeling

Vladimir Ruzov

There have been many different studies performed in order to examine various properties of neurons. One of the most important properties of neurons is an ability to originate and propagate action potential. The action potential is a source of communication between different neural structures located in different anatomical regions. Many different studies use modeling to describe the action potential and its properties. These models mathematically describe physical properties of neurons and analyze and explain biological and electrochemical processes such as action potential initiation and propagation. Therefore, one of the most important functions of neurons is an ability to provide communication between different neural structures located in different anatomical regions. This is achieved by transmitting electrical signals from one part of the body to another. For example, neurons transmit signals from the brain to the motor neurons (efferent neurons) and from body tissues back to the brain (afferent neurons). This communication process is extremely important for a being to function properly.

One of the most valuable studies in neuroscience was conducted by Alan Hodgkin and Andrew Huxley. In their work, Alan Hodgkin and Andrew Huxley used a giant squid axon to create a mathematical model which analyzes and explains the ionic mechanisms underlying the initiation and propagation of action potentials. They received the 1963 Nobel Prize in Physiology/Medicine for their valuable contribution to medical science. The Hodgkin and Huxley model is a mathematical model that describes how the action potential is initiated and how it propagates in a neuron. It is a

set of nonlinear ordinary differential equations that approximates the electrical characteristics of excitable cells such as neurons and cardiomyocytes.

This work focuses on modeling the Hodgkin and Huxley model using MATLAB extension - Simulink. This tool provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. Simulink model is used to describe the mechanisms and underlying processes involved in action potential initiation and propagation.

Keywords: Hodgkin and Huxley model, Axon properties, Action Potentials.

TABLE OF CONTENTS

LIST OF TABLES.....	viii
LIST OF FIGURES.....	ix
Chapter 1 - Background.....	1
1.1 Introduction.....	1
1.2 Neurons.....	4
1.3 Cell Membrane.....	5
1.4 Voltage Gated Channels.....	7
1.5 Ion Pumps.....	9
1.6 Leak Channels.....	11
1.7 Action Potential.....	12
1.8 Definition of the Hodgkin-Huxley Model.....	16
1.9 Voltage Gated Sodium Channel.....	19
1.10 Voltage Gated Potassium Channel.....	21
1.11 Leak Channel.....	23
1.12 Membrane Capacitance.....	24
1.13 Action Potential Propagation.....	24
1.14 Propagation Velocity.....	26
1.15 Time Constant.....	27
1.16 Space Constant.....	30
1.17 Propagation Velocity.....	33
1.18 Action Potential Propagation in Myelinated Fibers.....	35
1.19 Action Potential Propagation: Neuron Properties.....	37
Chapter 2 - Modeling.....	43
2.1 Modeling the Hodgkin and Huxley Model.....	43
2.2 Modeling Voltage Gated Sodium Channel.....	44
2.3 Modeling Voltage Gated Potassium Channel.....	45

2.4 Modeling Leak Channel	46
2.5 Modeling Membrane Capacitance.....	47
2.6 Modeling Periodic Stimulation.....	48
2.7 Modeling Temperature Effect.....	50
Chapter 3 – Simulation Results	54
3.1 Membrane Conductance.....	54
3.2 Gating Variables.....	55
3.3 Ionic Currents	58
3.4 Temperature Effect.....	61
3.5 Current Injection.....	62
3.6 Threshold.....	66
Chapter 4.....	69
4.1 Future Work	69
4.2 Conclusion.....	71
References	73
Appendix.....	76
Appendix A: Simulink Constants and Variables.....	76
Appendix B: Simulink Model Source Code.....	78

LIST OF TABLES

1. Ion concentrations [4].....	10
--------------------------------	----

LIST OF FIGURES

1. Neuron Components [4].....	4
2. Phospholipid Bilayer [8].....	6
3. Voltage gated Sodium Channel [9].....	9
4. Ion Pump [12].....	11
5. Leak channel [13].....	12
6. Ion Flow in Action Potential [7].....	13
7. Schematic diagram for the Hodgkin-Huxley model. [15].....	17
8. Action Potential propagation across a neuron [16].....	25
9. Time Constant [16].....	28
10. Length (Space) Constant.....	32
11. Action potential propagation in myelinated fiber	36
12. Schematics of Action potential propagation [4]	37
13. High Level Schematics of the Hodgkin and Huxley model	43
14. Voltage Gated Sodium channel Simulink model	45
15. Voltage Gated Potassium Channel Simulink model	46
16. Leak Channel Schematics	46
17. Schematics of membrane capacitance.....	47
18. Action Potential.....	48
19. The Hodgkin and Huxley model (External stimulation)	49
20. Action potential – periodic stimulation.....	50
21. Temperature Effect	51
22. Potassium Channel (with temperature effect)	52
23. Sodium Channel (with temperature effect).....	53

24. Sodium and Potassium conductance and their sum	54
25. Gating variables (n , m and h) during action potential activation.....	56
26. Membrane Action Potential [21]	57
27. Ionic currents	58
28. Ionic currents during the nerve impulse [22]	60
29. Temperature Effect Results.....	61
30. Action Potential: temperature effect [23]	62
31. Action potential during current injection.....	64
32. Current Injection [24]	65
33. Threshold.....	67
34. Threshold level [25]	68

Chapter 1 - Background

1.1 Introduction

The action potential is a source of communication between different neural structures located in different anatomical regions. For instance, communication between the brain and musculature is achieved by propagating an electrical signal from the brain to the specific muscle. Neurons are used for processing and transporting action potentials from one part of the body to another. Neurons are also responsible for propagating action potentials from one neuron to another. An action potential can be described as an electrical impulse which is a transient alteration of transmembrane voltage. During this event, the electrical membrane potential of a particular cell changes very rapidly from negative to positive and then reverses back to its baseline (or resting) voltage. This is done in a consistent pattern. There are different kinds of excitable cells that have this unique property, such as muscle cells, neurons and cardiomyocytes.

Cellular communication is a method by which living cells communicate with each other. Therefore an ability to send and receive messages is very important for cells to survive and function properly. Living cells communicate important information such as availability of nutrients and changes in the temperature. This information is communicated with the environment and other living cells. Cellular communications are achieved in different ways such as biochemical signaling and biophysical cellular communication [1]. Evidently, the action potential plays an important role in biophysical cellular communication processes [1]. It's also responsible for activating different intracellular processes such as muscle cells contraction. There are two main types of

action potentials. The first type of action potential is generated by voltage-gated sodium channels (NaV) and the second type is generated by voltage gated calcium channels (CaV). The main difference between these two types of action potentials is duration. Sodium based action potential lasts for a very short period of time, less than millisecond [2]. Calcium based action potential on the other hand may last over a period of 100 milliseconds or longer [2]. This paper is focused on the first type of action potentials, generated by voltage gated sodium channels.

Most human and animal cells are electrically polarized and act like small batteries. In other words, they have a specific voltage difference between the extracellular and intracellular environments. This voltage is also known as the membrane potential and it is measured in millivolts. The resting membrane potential is a membrane potential during a static state of a cell [3]. Consequently, during this state the cell membrane potential is at equilibrium. The resting membrane potential is negatively charged since the inside of excitable cells is approximately 70mV more negative than the outside of the cell. Therefore, the resting membrane potential is approximately -70mV. This value varies depending on neuron types and the species [3]. In this dissertation the resting membrane potential is assumed to be constant and equal to -70mV.

So what is the underlying mechanism of action potential initiation? Excitable cells have a means of transporting different types of ions from the intracellular to the extracellular environment and vice versa. This ion transport is achieved by specific ion channels with selective permeability. In other words, only specific types of ions can cross ion channels with selective permeability. For example sodium ions are transported through the cell membrane by the sodium ion channel, whereas potassium ions are transported through potassium ion channels. An important factor related to the potential

is the ionic concentration gradient. During normal conditions (resting state) there are a specific number of ions present inside and outside the cell. Since ions are charged particles, the difference in ion concentration between outside and inside of a cell membrane defines a resting voltage potential. There are two forces that move ions across the cell membrane. The first is the concentration gradient and the second is the charge gradient. An equilibrium potential is reached when the net flow of that particular ion is zero. Resting voltage potential is reached when the net flow of all types of ions is zero. In other words, the net flow of sodium, potassium and chloride ions is zero during the resting membrane potential [4].

An action potential is an event during which the electrical membrane potential changes its state from resting state. During this event the membrane potential rises rapidly and then falls. In order to initiate an action potential, the membrane potential has to increase and reach a certain level [4].

As we will see in later sections of this dissertation, action potential occurs in several phases: depolarization, repolarization, etc. All of these stages involve ion transfer from the extracellular environment to the intracellular environment and vice versa. For example, the depolarization phase can be characterized by rapid inflow of sodium ions through sodium ion channels. Repolarization phase on the other hand occurs when potassium ions flow out from the axoplasm through potassium selective ion channels.

In order to better understand all of the mechanisms involved in action potential initiation and propagation, several mathematical models have been developed such as Morris–Lecar model, the FitzHugh–Nagumo model, the Bonhoeffer–van der Pol model [5]. One such models is called the Hodgkin and Huxley model which is a mathematical model that describes how action potential is initiates and propagates along an axon. In

1963 Alan Hodgkin and Andrew Huxley received a Nobel Prize in physiology for their valuable contribution in neuroscience.

1.2 Neurons

Neurons are specialized nerve cells that have an ability to send and receive electrical signals from one neural structure to another. Neurons are also capable of receiving and sending signals to other neurons. Being able to transport and process electrical signals, neurons form a communication network between different organs in the body. Basic neuron structure includes a cell body, an axons and dendrites. The main function for axons is to transmit electrical impulses along its length. The axon is the longest part of a neuron responsible for propagation of the action potential from cell body (also known as the Soma) to axon terminal [6]. Dendrites, on the other hand, are the receiving part of a neuron. In other words, dendrites receive electrical impulses passed by other neurons and propagate them to the cell body.

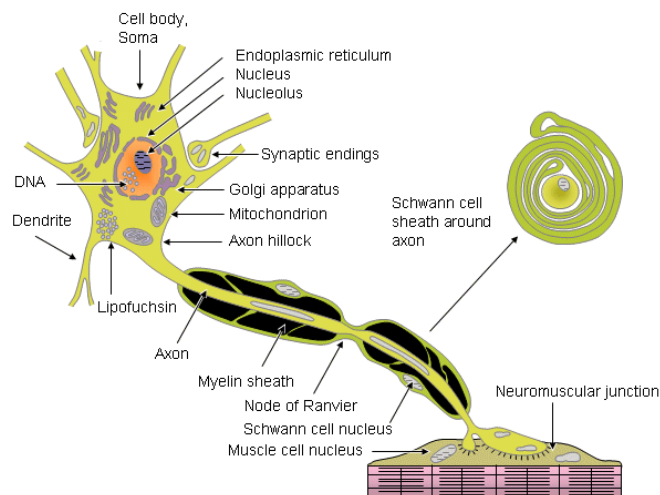


Figure 1: Neuron Components [4]

The soma, or cell body, is the processing center of a neuron. The unique feature of the cell body is that it is the only part of a neuron that contains a cell nucleus [7]. Most RNA in a neuron is produced by a cell nucleus. Therefore, most proteins are produced in the cell body and then transported to the entire neuron [7]. Another part of a neuron that plays an important role in action potential origination is the axon hillock. It is the part of a neuron from which axon originates. This region is characterized by a high amount of protein synthesis. Another important characteristic of the axon hillock is a presence of a specialized plasma membrane. This plasma membrane contains many voltage gated ion channels and it is responsible for action potential initiation.

Complex interactions between specialized protein structures, which are embedded in a cell membrane, allow initiation and propagation of action potential along neurons. These protein structures vary across a neuron thus giving various electrical properties and functions to different parts of the neuron. Therefore, some of the neuronal structures have excitable properties, while other parts of the neuron do not. In other words, given an external stimulus, excitable parts of a neuron can initiate an action potential. For example, myelinated parts of a neuron are not excitable and therefore are not involved in the action potential generation process. In these sections of a neuron, an action potential is propagated passively by means of electrotonic propagation. Unmyelinated sections of a neuron are called Nodes of Ranvier. These parts of the axon are capable of producing an action potential.

1.3 Cell Membrane

Electrical properties of human and animal cells are mostly determined by the structure of the cell membrane that separates the extracellular and intracellular

environments. A cell membrane is composed of a phospholipid bi-layer. Phospholipids are specialized molecules that are composed of two parts: phosphate and lipids. The phosphate head is a hydrophilic portion of the phospholipid which means that it attracts water molecules, whereas lipid tails are the hydrophobic portion of the phospholipid and they are repelled by the water molecules. Therefore, a phospholipid bilayer is formed due to the fact that lipid tails are repelled by aqueous solutions and are attracted facing each other (inward) whereas the phosphate head of the phospholipid molecule is facing outward [8]. Figure 2 demonstrates how the phospholipid bilayer is formed, identifying hydrophilic phosphate heads and hydrophobic lipid tails.

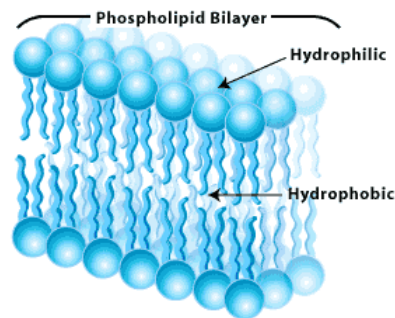


Figure 2: Phospholipid Bilayer [8]

So how do different ions move across the phospholipid bilayer? The lipid layer acts as an insulator since it is highly resistant to movement of ions. There are a number of different types of proteins embedded in a cell membrane. Some of these proteins form ion channels that provide means for different ion types to cross a cell membrane. Ion channels are large protein molecules embedded in a cell membrane (e.g. voltage gated channels, leak channels, etc.). Channel proteins form small openings for different molecules to pass from the extracellular to the intracellular medium and vice versa. Some of these protein channels are capable of actively moving specific types of ions

from one side of the membrane to another [9]. For example, the Na-K-Pump uses energy from adenosine triphosphate (ATP) to transfer sodium and potassium across a cell membrane. Receptor proteins act as molecular event triggers that initiate a specific cell response such as opening a channel such as the ligand gated channel [9]. In other words, the channel stays inactivated until a specific ion or molecule, called a ligand, is bound to the ligand binding site. This process will activate the channel allowing it to open for ions to pass through the cell membrane. Another type of a cell membrane protein is a carrier protein, which acts as a binding site that grabs a specific molecule and pulls it inside the cell into the intracellular medium [9]. Cell recognition proteins such as CD (cluster of differentiation) are useful in identification of cell types. This protein is an important part of the body's immune system. Another type of protein molecule is a voltage gated channel. This type of channels plays an important role in action potential activation and propagation.

1.4 Voltage Gated Channels

Voltage gated ion channels are a special type of ion channel which is embedded in a cell membrane. The key feature of voltage gated channels is that they are activated by a difference in membrane potential [9]. In other words, these types of channels open for some values of membrane potential while for other values of membrane potential these channels remain closed. Figure 3 shows different states of a voltage gated sodium channel. The first image shows the inactive state of a voltage gated ion channel. At this state, the channel remains closed and therefore no ion flow exists. This state is prevalent when the membrane is at rest [9]. The channel quickly begins to open as soon

as the membrane potential reaches a specific threshold value. Additionally, voltage gated ion channels are responsible for ion transport between the extracellular and intracellular environments. These channels are permeable to specific types of ions. For example, voltage gated sodium channels (NaV) are permeable to sodium (Na) ions only; whereas voltage gated Calcium channels (CaV) are permeable to calcium (Ca) ions only.

Voltage gated ion channels contribute to action potential generation and/or propagation by initiating a positive feedback loop. The opening of ion channels changes the membrane conductance to ionic current flow. In other words, a change in membrane potential will activate and open voltage gated ion channels and thus increase the membrane conductance to ionic current flow. As a result ions will flow into the axoplasm. The inflow or outflow of ions will lead to a change of an electrochemical gradient. Consequently, increase in electrochemical gradient will contribute to further change in a membrane potential. As a result more voltage gated ion channels will be activated, causing an increase in ion current.

An action potential is initiated when the membrane potential reaches a certain level referred to as the threshold potential. Reaching the threshold level will result in a rapid increase of membrane potential. The biophysical properties of voltage gated ion channels influence the shape of the action potential curve [8]. There are different types of voltage gated ion channels that can produce a positive feedback loop which could lead to initiation of an action potential. An example of such a channel is a voltage gated sodium channel. It contributes to action potential in nerve fibers. Voltage gated calcium channel, on the other hand, is an important component in muscle fibers since calcium initiates the contractile process [10].

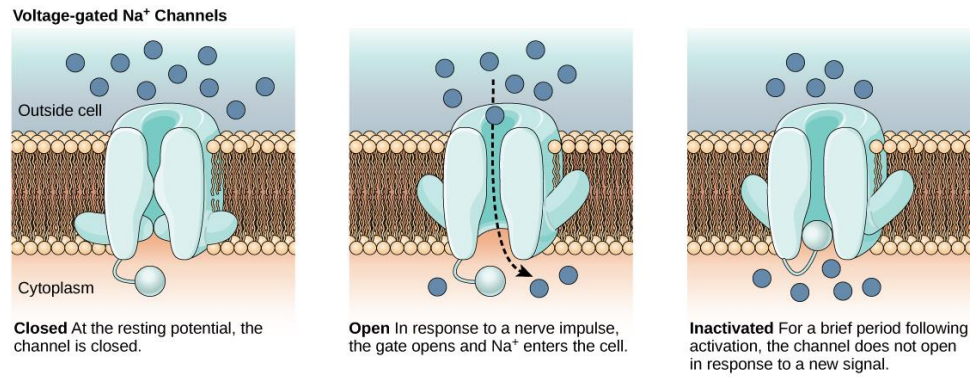


Figure 3: Voltage gated Sodium Channel [9]

Voltage-gated potassium channels (KV^+) open and/or close in response to changes in the transmembrane potential. These channels are potassium selective pores that allow only potassium ions to cross a cell membrane. Furthermore, potassium channels are found in most cell types and control a wide variety of cell functions. In excitable cells such as neurons, they shape the action potential curve and set the resting membrane potential [9]. By contributing to regulation of the action potential duration in cardiac muscle, malfunction of potassium channels may cause life-threatening arrhythmias [6]. Potassium channels may also be involved in maintaining vascular tone by having an impact on contractile properties of smooth muscle cells which are commonly found in arteries and arterioles [11].

1.5 Ion Pumps

Another type of transmembrane protein channel is an ion pump. The extracellular environment and the intracellular environment have different ionic composition. For example, Table 1 shows different intracellular and extracellular

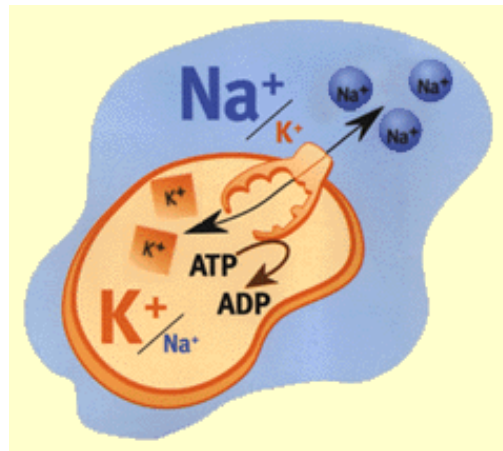
concentrations for different ions. Potassium concentration is much higher in the axoplasm, whereas sodium concentration is much higher outside the cell.

Table 1: Ion concentrations [4]

	Extracellular Concentration [mol/m ³]	Intracellular Concentration [mol/m ³]
Na ⁺	150	15
K ⁺	5.5	150
Cl ⁻	125	9

Ions move from areas of high concentrations to areas of low concentrations.

This process is called diffusion and it continues until the concentration is equalized on both sides [9]. Another process that moves ions in the opposite direction is electrostatic in nature and gives rise to the Nernst Potential. Therefore, there must be a mechanism to maintain a different ion concentration during the resting state. This process is achieved by ion pumps. The main difference between ion pumps and voltage gated channels is that ion pumps move ions across the plasma membrane against the ionic concentration gradient. This is the reason why ion pumps require energy to perform their action and are considered to be active channels. Ion pumps are enzymes that are capable of converting energy from different sources. For instance, Figure 4 illustrates how ATP is used in Na⁺K⁺ the pump to move sodium and potassium ions across the membrane against their concentration gradients.



Ion composition outside the cell differs from that inside. When e.g. a nerve is stimulated Na⁺ flows into the cell. The electrical potential arising across the membrane causes the nerve impulse to propagate itself along the nerve. The normal state is restored when Na⁺ has been pumped out and K⁺ pumped in.

Figure 4: Ion Pump [12]

Na⁺K⁺ pump can be found in all animal cells [12]. The main function of the Na⁺K⁺ pump is to maintain a stable resting potential in cells by maintaining a low sodium concentration and high potassium concentration in the intracellular environment. This gradient is maintained by moving three sodium ions out of a cell and moving two potassium ions into the cell. In other words, one positive charge is being removed during a single pump action.

1.6 Leak Channels

Another way for ions to cross a cell membrane is through leak channels. These channels are passive, or randomly gated [13]. In other words, these gates open and close at random. Unlike voltage gated channels, there is no specific event that can trigger a leak channel to open or close. This type of ion channel has its own intrinsic rate of change between open and closed states [13]. Figure 5 demonstrates how ions are being transported across the cell membrane via a leak channel. The right image

demonstrates how ions travel in the direction of the concentration gradient, from high ionic concentration to low ionic concentration.

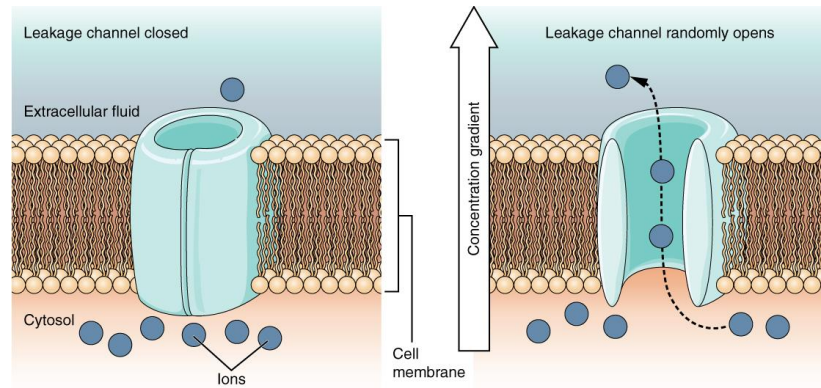


Figure 5: Leak channel [13]

Leak channels allow ions to pass from the extracellular medium to the intracellular medium and vice versa. Since ions flow in the direction of lower concentration, leak channels have a great impact on resting membrane potential.

1.7 Action Potential

The action potential can be described as an electrical impulse which is a transient alteration of transmembrane potential. The action potential is generated by series of consecutive events. The prerequisite for action potential initiation is the membrane potential reaching some threshold value. This value can be approximated by a constant value. The threshold level depends on many different factors such as the type of neuron, its location in the body as well as temperature and other factors. Therefore, in practical circumstances, the threshold level fluctuates [7]. For simplicity

this paper assumes that the threshold level is constant. Figure 6 depicts the movement of different ions during various stages of action potential generation (e.g. depolarization, repolarization, etc.). Threshold level is represented by a dashed line (approximately -60mV) and is located above the level of the resting potential (-70mV).

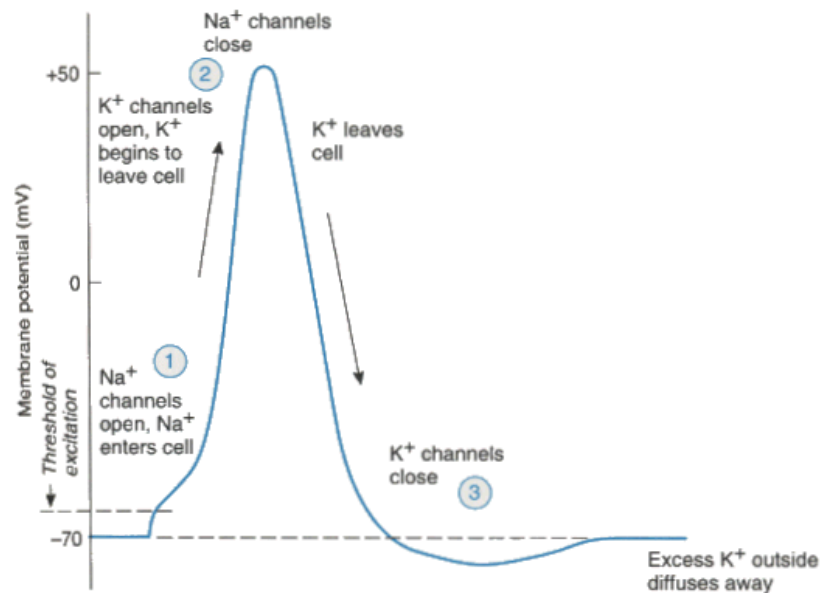


Figure 6: Ion Flow in Action Potential [7]

The first phase of the action potential initiation process is called depolarization. In order for the depolarization phase to occur, membrane potential needs to meet certain criteria. Therefore, when membrane potential reaches a certain threshold value, voltage gated sodium channels (NaV) will open allowing rapid influx of sodium ions from the extracellular environment to the intracellular environment. As more sodium ions enter the cell, the concentration of sodium ions inside the cell increases. Membrane potential increases since sodium ions are positively charged. The associated positive feedback loop will result in more voltage gated sodium channels opening and thus creating an upward shooting trend in the action potential curve [7]. As a result, the intracellular environment will become more positively charged relative to the extracellular

environment. This process will lead to a plasma membrane polarity switch. Voltage gated sodium channels will then deactivate. Since voltage gated sodium channels are opened for only a very short period of time, the bulk sodium concentration during cellular signaling changes only by a tiny amount that is a few orders of magnitude smaller than the bulk concentration [7]. When voltage gated sodium channels close, sodium ions can no longer enter the cell. The depolarization phase is the fastest phase in the action potential generation process.

The next phase of the action potential initiation process is called repolarization. During this phase voltage gated potassium channels open allowing the outflow of potassium ions from the intracellular environment to the extracellular environment. This process decreases the membrane potential of a cell. As a result, the intracellular environment will become more negatively charged relative to the extracellular environment. Just like sodium channels, potassium channels are opened for a very short period of time [7].

The third phase of the action potential initiation process is called hyperpolarization. During this phase, voltage gated potassium channels remain open and the intracellular environment is hyperpolarized. In other words, the membrane potential (-80mV) decreases below the level of the resting potential (-70mV). At this stage potassium ions will enter the cell mostly through leak channels and ion pumps [7]. This process will restore the membrane potential leveling it with the resting potential. This is the slowest phase in the action potential generation process.

Generally, there are enough sodium and potassium ions in both the extracellular and intracellular environments to support multiple action potentials. Action potential can

either fire or not fire. Action potential cannot fire partially neither it can fire too hard or too weak. In other words, the amplitude remains constant [14].

Resting potential is a level at which the membrane potential stays constant unless something disturbs or stimulates the excitable cell. Another important level of membrane potential is the threshold level. Reaching this level will result in the action potential initiation process. A typical value of resting potential at the axon hillock is around -70mV and the threshold value is around -55mV [7].

The shape of the action potential curve depends on many different factors such as properties of excitable cell membrane (e.g. number of voltage gated ion channels, leak channels, membrane capacitance) and temperature [7].

The most studied voltage gated ion channels are located in fast nerve conduction system [2]. As noted earlier in this paper Alan Hodgkin and Andrew Huxley won a Nobel Prize for their valuable contribution in the quantitative description of action potentials of a giant squid axon [2]. Their model is named after them. They developed a series of nonlinear differential equations that encompass the biophysics of the membrane which gives rise to the action potential. These equations approximate electrical properties of excitable cells such as the ones found in neurons and cardiac muscle cells. More specifically, Alan Hodgkin and Andrew Huxley modeled ionic currents involved in action potential initiation and propagation. These currents are:

- sodium current through voltage gated sodium channels
- potassium current through voltage gated potassium channels
- potassium, chloride and sodium currents through leak channels
- current through the cell membrane

1.8 Definition of the Hodgkin-Huxley Model

A semipermeable cell membrane separates the axoplasm from the extracellular medium and acts as a capacitor. If external current is applied to an excitable membrane, it can charge the membrane capacitance and/or diffuse through ion channels. Alan Hodgkin and Andrew Huxley modeled a total membrane current, taking into consideration different ions which utilize various ion channels to cross a cell membrane:

1. Sodium channel current (I_{Na})
2. Potassium channel current (I_K)
3. Leak channel current (I_{leak})
4. Capacitive current

According to the Hodgkin and Huxley model, each of these channels acts independently and carries only a specific type of ions. In other words, a sodium channel can only transport sodium ions. A leak channel on the other hand does not have selective permeability and therefore, different types of ions can move through it. And finally, capacitive current is a current through the phospholipid bilayer of an excitable cell.

Alan Hodgkin and Andrew Huxley represented the membrane model by using basic electrical components such as resistors and capacitors. This model is also known as a parallel conductance model [4]. Figure 7 shows schematics of the Hodgkin and Huxley model using capacitors and resistors. Current (I) is an electrical current flowing across the cell membrane during activation. Each channel (e.g. voltage gated sodium channel, voltage gated potassium channel and leak channel) is represented by a resistor and an

associated equilibrium potential is represented by a battery (e.g. E_K , E_{Na} and E_i). The equilibrium potential is the value of V_m where there is no net flow of ions between the extracellular environment and the intracellular environment.

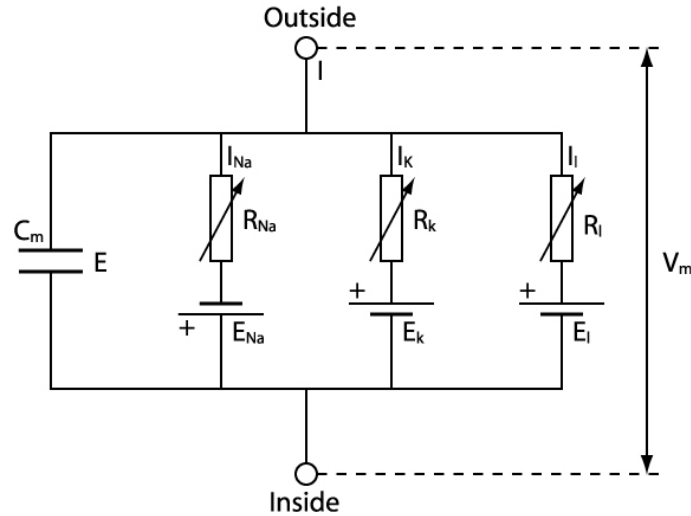


Figure 7: Schematic diagram for the Hodgkin-Huxley model. [15].

In this paper V_m is used to describe a membrane potential which is equal to the intracellular potential minus the extracellular potential. Ion equilibrium potentials for sodium, potassium and chloride ions are described by Nernst potentials and defined by the Nernst equation:

$$V_{Na} = -\frac{RT}{zF} \ln \frac{c_{i,Na}}{c_{o,Na}} \quad (1)$$

$$V_K = -\frac{RT}{zF} \ln \frac{c_{i,K}}{c_{o,K}} \quad (2)$$

$$V_{Cl} = -\frac{RT}{zF} \ln \frac{c_{i,Cl}}{c_{o,Cl}} \quad (3)$$

Note that subscripts “ c_i ” and “ c_o ” represent ion concentration in the intracellular and the extracellular environments respectively. The value of z for sodium and potassium ions is 1. The value of z for chloride ions is -1. Since the internal concentration of chloride is extremely low, it does not play an important role in evaluation of the action potential [4]. Therefore, (3) can be generalized to include not only chloride ions but all non-specific ions crossing the cell membrane. As was previously discussed, leak channels allow a small amount of different types of ions to pass from the intracellular environment to the extracellular environment and vice versa. Consequently, it is assumed that the conductance of a leak channel (G_L) is constant. The leak channel voltage (V_L) is also assumed to be constant since the sum of all ionic currents is equal to zero during the resting state. In their experiments, Alan Hodgkin and Andrew Huxley determined that the membrane conductance for a potassium channel (G_K) and the membrane conductance for a sodium channel (G_{Na}) are functions of transmembrane voltage and time [4]. There is no sodium current when $V_m = V_{Na}$. Therefore $I_{Na} = 0$. There is no potassium current ($I_K = 0$) when $V_m = V_K$.

According to Ohm’s law, the conductance per unit area for the potassium channels, sodium channels and leak channels can be calculated as shown below.

$$G_{Na} = \frac{I_{Na}}{V_m - V_{Na}} \quad (4)$$

$$G_K = \frac{I_K}{V_m - V_K} \quad (5)$$

$$G_L = \frac{I_L}{V_m - V_L} \quad (6)$$

where G_{Na} , G_K , G_L Conductance for sodium, potassium and leak channels [S/cm²]

I_{Na}, I_K, I_L	Current passed by sodium, potassium and leak channels [mA/cm ²]
V_{Na}, V_K, V_L	Voltage for sodium, potassium and leak channels [mV]
V_m	Resting voltage of cell membrane [mV]

The equation for the total transmembrane current can be written as a sum of currents from all channels (sodium, potassium and leak channels) and the membrane capacitance.

$$I_m = C_m \frac{dV_m}{dt} + (V_m - V_{Na})G_{Na} + (V_m - V_K)G_K + (V_m - V_L)G_L \quad (7)$$

where I_m	membrane current per unit area [mA/cm ²]
C_m	membrane capacitance per unit area [F/cm ²]
V_m	membrane voltage [mV]
V_{Na}, V_K, V_L	Nernst voltage for sodium, potassium and leakage ions [mV]
G_{Na}, G_K, G_L	sodium, potassium, and leakage conductance per unit area [S/cm ²]

1.9 Voltage Gated Sodium Channel

Sodium ions cross the cell membrane via voltage gated sodium channels. Sodium ion equilibrium is achieved when the condition $V_m = V_{Na}$ is satisfied. At the sodium equilibrium potential there is no net movement of sodium ions across the cell membrane because for each sodium ion that moves into the cell by diffusion, exactly one sodium ion moves out of the cell due to the electrostatic gradient. Therefore, the net current of sodium during this state is 0 ($I_{Na} = 0$). Any deviation from this state

$(V_m \neq V_{Na})$ will result in a non-zero current in the sodium channel. According to Ohm's Law the current in a sodium channel is equal to:

$$I_{Na} = (V_m - V_{Na})G_{Na} \quad (8)$$

where	V_m	membrane voltage [mV]
	I_{Na}	current passed by sodium channel [mA/cm ²]
	V_{Na}	Nernst voltage for sodium ions [mV]
	G_{Na}	sodium conductance per unit area [S/cm ²]

The Hodgkin and Huxley model is based on the assumption that sodium channels open and close according to the state of the m -particle [2]. The m -particle is an activation parameter for a sodium channel. Sodium channel m -particles have two states: permissive (the state when the sodium channel is open) and non-permissive (the state when the sodium channel is closed). The m -particle gives the proportion of sodium channels that are in the open state at a particular moment in time. Therefore, $(1-m)$ expresses the proportion of sodium channels that are currently in the closed state. Alan Hodgkin and Andrew Huxley assumed that it required three m -particles to activate a single sodium channel [2]. In other words, all three m -particles need to be in the permissive state in order to activate a single sodium channel. Sodium channels are deactivated by a deactivation particle h . This parameter represents the probability of an h -particle being in a non-activated state. Therefore, the probability of the h -particle being in an activated state is equal to $1-h$. Alan Hodgkin and Andrew Huxley assumed that it required one h -particle to deactivate a single sodium channel [2].

The relationship between the sodium conductance and the m -particles and h -particle is represented by (9).

$$G_{Na} = G_{Na \max} m^3 h \quad (9)$$

where $G_{Na \max}$ maximum value of sodium conductance [mS/cm²]

m obeys (10)

h obeys (11)

Sodium channel activation is represented by (10). Sodium channel deactivation is represented by (11).

$$\frac{dm}{dt} = \alpha_m(1 - m) - \beta_m m \quad (10)$$

$$\frac{dh}{dt} = \alpha_h(1 - h) - \beta_h h \quad (11)$$

Transfer rate coefficients for α_h ; α_m ; β_h ; β_m are voltage dependent. Therefore, transfer rate coefficients do not depend on time [4].

1.10 Voltage Gated Potassium Channel

Potassium ions cross the cell membrane via voltage gated potassium channels. The equilibrium for potassium ions is achieved when $V_m = V_K$. During this state, there is no potassium current going through the cell membrane ($I_K = 0$). Any deviation from this state ($V_m \neq V_K$) will result in a non-zero current in the potassium channel.

According to Ohm's Law, the current in a potassium channel is equal to:

$$I_K = (V_m - V_K)G_K \quad (12)$$

where	V_m	membrane voltage [mV]
	I_K	Current passed by potassium channel [mA/cm ²]
	V_K	Nernst voltage for potassium ions [mV]
	G_K	potassium conductance per unit area [S/cm ²]

The Hodgkin and Huxley model is based on the assumption that potassium channels open and close according to the state of the n -particle [2]. The n -particle is an activation parameter for a potassium channel. Potassium channel n -particles have two states: permissive (the state when the potassium channel is open) and non-permissive (the state when the potassium channel is closed). The n -particle defines the proportion of potassium channels that are in the open state at a particular moment in time. Therefore, $(1-n)$ defines the proportion of potassium channels that are currently in the closed state. Alan Hodgkin and Andrew Huxley assumed that it required four n -particles to activate a single potassium channel [2]. In other words, all four n -particles need to be in the permissive state in order to activate a single potassium channel. Potassium conductance is represented by (13).

$$G_K = G_{K \max} n^4 \quad (13)$$

where	$G_{K \max}$	maximum value of potassium conductance [mS/cm ²]
	n	obeys (14)

The behavior of the potassium channel activation is represented by (14).

$$\frac{dn}{dt} = \alpha_n(1 - n) - \beta_n n \quad (14)$$

1.11 Leak Channel

Different types of ions (e.g. potassium, calcium, sodium) cross the cell membrane via leak channels. Equilibrium in leak channels is achieved when the following condition is satisfied $V_m = V_L$. During this state there is no leak current going through the cell membrane and therefore $I_L = 0$. Any deviation from this state ($V_m \neq V_L$) will result in a non-zero current in the leak channel.

According to Ohm's Law, the current in a leak channel is equal to:

$$I_L = (V_m - V_L)G_L \quad (15)$$

where	V_m	membrane voltage [mV]
	I_L	Current passed by leak channel (mA/cm ²)
	V_L	Nernst voltage for ions in leak channel [mV]
	G_L	Leak channel conductance per unit area [S/cm ²]

1.12 Membrane Capacitance

Another component that contributes to the total membrane current is the membrane capacitance.

$$I_m = C_m \frac{dV_m}{dt} \quad (16)$$

where I_m membrane current per unit area [mA/cm²]

C_m membrane capacitance per unit area [F/cm²]

V_m membrane voltage [mV]

1.13 Action Potential Propagation

So far we have discussed the action potential origination that involved transmembrane currents only. This is an important building block which can be used to get a more complete neuron model. After initiation, the action potential propagates in outward directions and travels to the synaptic terminal region. It is important to define action potential propagation since the word “propagation” often refers to different actions. In many cases the word “propagation” defines the movement of an object from one point to another. For example, throwing a ball from one location to another defines propagation, or movement of the ball. This definition does not apply to action potential propagation. In case of action potential propagation one can think of a series of events

that have to happen in a specific order. Each of these successive events will eventually trigger the next event thus propagating it towards the end point.

Figure 8 demonstrates how action potential propagates gradually along the axons' membrane in outward directions from the point where the stimulus was applied. The action potential propagates in all or nothing fashion [16]. In other words, an action potential cannot propagate partially. A propagating action potential can be stopped as it propagates by a region of unmyelinated fiber such as in patients with multiple sclerosis [6].

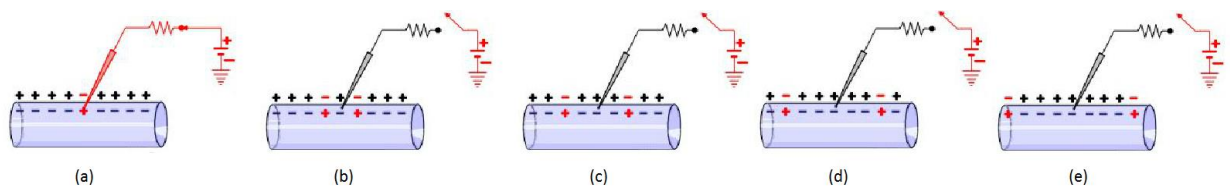


Figure 8: Action Potential propagation across a neuron [16]

Under resting conditions, the extracellular environment is positively charged whereas the intracellular environment is negatively charged. This arrangement is due to a high concentration of positively charged sodium ions in the extracellular environment. Let's consider an experiment in which an external stimulus is applied directly to the mid-section of an axon. If this stimulus is high enough to reach a certain level, called the threshold level, then an action potential will fire. As it was previously discussed, the first step will open voltage gated sodium channels which will result in a rapid inflow of sodium ions from the extracellular environment to the intracellular environment. This will result in an increase in membrane potential from its resting state (-70mV) to a positive 40mV [16]. The action potential, at its initiation site, will depolarize the membrane to either side of it and a subsequent action potential will be generated in the newly depolarized regions.

Action potential propagation can be compared to the waves in water. If one would throw a rock in calm water, the concentric circles or the ripples will originate and propagate in an outward direction away from the point where the rock landed. The main difference is that the ripples are usually rapidly distorted in a random manner and eventually destroyed completely. This is not true with an action potential which will propagate throughout the whole length of a neuron until it reaches the axon terminal.

1.14 Propagation Velocity

One of the most important properties in action potential propagation is the velocity. The human body has a wide variety of neuron types which can be characterized by the action potential velocity which is the speed at which the impulse is propagated across the entire length of an axon. It is an important factor that defines how fast the information is transferred from one part of the body to another. Some of the key factors influencing action potential propagation velocity are fiber diameter, myelination, age, sex and temperature [17]. Different types of neurons can be found in the human body and each of the neuron type has its unique purpose and properties. Action potential propagation velocity in neurons can vary from 100 meters per second (223.7 miles per hour) to a much lower speed, less than a tenth of a meter per second (0.6 miles per hour) [18]. Why do neurons have such a big difference in the propagation velocity? Different neurons in a human body have their own purpose. Some of the neurons must have a high propagation velocity in order to quickly deliver information from the body to the brain and vice versa. Other types of neurons propagate an action potential at much slower speeds. Axons are typically divided into three classes: type A fibers, type B fibers and type C fibers. Type C fibers have the slowest propagation

velocity, usually below 1.34 m/s [17]. These types of fibers are the autonomic motor neurons that carry an electrical impulse to cardiac muscles, smooth muscle, glands and other peripheral effectors. Some of the sensory neurons are also of type C fibers. They carry information about pain, general touch, pressure and temperature. This type of information is not required to be transferred at high speeds. Type C fibers are the smallest axons whereas type A fibers are usually the largest axons [17]. One of the main differences in the physical properties between type C fibers and type A is an absence of myelin in type C fibers. This directly correlates to the size of an axon fiber. Action potential propagation velocity in an axon increases as the diameter of the axon increases. Type A fibers have the fastest propagation velocity. For example type A sensory neurons carry information about balance, position, delicate touch and pressure. This type of information is required to be transferred in a very short period of time. Therefore, the propagation velocity for this type of information is high.

There are big differences between action potential propagation velocities across different fibers. In order to understand how these differences arise we need to examine properties of the axons. In particular, it is important to understand the concept of the time constant and the space constant.

1.15 Time Constant

Different types of axons have various action potential propagation velocities. Some of the axons propagate an electrical signal very fast (e.g. type A), while the other type of axons propagate an action potential at a much slower speed (e.g. type C). So why does the propagation velocity differ so much? The time constant is an intrinsic

property of the membrane and can be used to describe the difference in action potential propagation in neurons.

Before analyzing the time constant it may be beneficial to consider a heat transfer example. Let's consider a block of metal with the initial temperature of 10°C . If this block of metal is placed on a hot plate which has a temperature of 100°C , then the temperature of the metal block will gradually increase until it reaches 100°C . This temperature change from 10°C to 100°C will not occur instantly but rather gradually. Similar behavior is observed with excitable cells. Let's consider an example illustrated in the Figure 9.

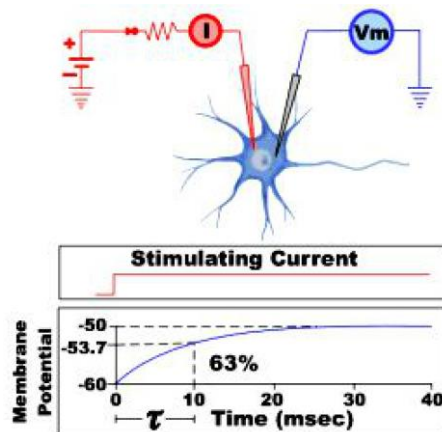


Figure 9: Time Constant [16]

The top image illustrates an experimental setup where an excitable cell is stimulated by the induced current and the voltmeter is used to measure a transmembrane voltage as a function of time. It is assumed that all regions within and around the excitable cell are equipotential. In other words, the same potential exists across all points of the excitable cell membrane. In the example above, the recording electrode measures the initial voltage difference of -60mV . This is the resting potential before applying the stimulus. At the time 0 an external current is applied to the excitable cell. This action is analogous to slamming the metal block on the hot plate. Even

though the stimulating current changed instantaneously from 0 to some positive value, the change in membrane potential will occur gradually and will not happen instantly. It will take some time for a voltage potential to change from -60mV to -50mV. Therefore, there is a 10mV depolarization which occurs gradually, as an exponential function of time.

There is a time constant index denoted by the symbol τ which represents how quickly the exponential function changes with time. More specifically, a time constant index defines how quickly an action potential reaches 63% of its maximum value. In the previous example, a membrane potential increases from -60mV to -50mV. Therefore, the time constant is equal to the time when -53.7mV is reached. According to the plot illustrated in Figure 9, the time constant is reached at 10msec. Therefore, the smaller the time constant, the faster the action potential reaches 63%. The neuron responds faster to the induced current. So if an excitable cell had a time constant of 5 milliseconds, it would reach 63% of the maximum potential in only 5 milliseconds. The time constant can be compared to the 0-60 car rating. Any car accelerates from 0mph to 60mph for a certain amount of time. The lower the 0-60 time of a given car, the faster it accelerates.

Time constant depends on two properties of a cell membrane: 1) membrane resistance (R_m) and 2) membrane capacitance (C_m). Membrane resistance opposes the flow of ions from the extracellular environment to the intracellular environment and vice versa. High membrane resistance means low permeability of a cell membrane. Which factors influence the membrane resistance? One of the factors that may contribute to the membrane resistance is the number of ion channels present in the cell membrane [16]. Low number of ion channels in the cell membrane results in low number of ions

crossing the cell membrane within a specific timeframe. Therefore, the membrane resistance is high and the membrane conductance is low.

The time constant also depends on the membrane capacitance [16]. The cell membrane acts as a capacitor since it has an ability to store energy. Membrane capacitance is charged when an external stimulus is applied. Discharging and charging of the membrane capacitance does not occur instantaneously but rather gradually.

The time constant can be represented by a product of membrane resistance and membrane capacitance:

$$\tau = R_m C_m \quad (17)$$

where τ time constant
 R_m membrane resistance
 C_m membrane capacitance

1.16 Space Constant

The space constant or length constant is another quantifier often used in neurobiology to describe properties of axons. The space constant is a mathematical value which quantifies the distance that an electrical impulse will travel along an axon via passive electrical conduction. The space constant, in a cell, dictates the distance across which a membrane depolarization will decay exponentially assuming a DC value of injected current. Therefore, the higher the space constant is, the longer this distance is. Let's consider an example where one end of the metal rod is placed on a hotplate. Let's assume that the initial temperature of the rod is 10C° and the temperature of the hotplate

is 100C° . The metal rod is left on the hotplate for a period of time, enough for the temperature to stabilize. If we were to measure the temperature across the metal rod we could conclude that the temperature decreases with the distance away from the hotplate. This is true due to the fact that the heat loss increases with the distance away from the heat source. Therefore, the temperature gradient can be described as a function of distance.

Similar behavior is observed within excitable cells. Figure 10 represents a nerve cell with electrodes v_0, v_1, v_2, v_4 in different locations along the axon, positioned at 10mm increments away from the cell body. The stimulating electrode is located inside the cell body and is used to deliver an electrical signal to the soma. The stimulating electrode provides enough power to change a membrane potential from -60mV to -50mV . Electrodes v_0 through v_4 are used for recording a voltage potential at different distances away from the cell body. Before stimulation is applied, the neuron has a resting potential of -60mV inside the soma as well as across the entire length of the axon. After an electrical impulse is applied to the soma, the membrane potential is changed from -60mV to -50mV . Is this true for the entire length of an axon? Figure 10 shows that the membrane potential is different at the various distances away from the soma. The cell body will have a voltage potential of -50mV since it is the area where the stimulation was applied. Moving along the axons' length we can notice that the voltage potential decreases as we move away from the soma. For example the membrane potential has a value of -56.3 at 1 mm away from the stimulation site. The membrane potential is very close to -60mV at 2mm away from the stimulation site. Membrane potential is even closer to -60mV at the distances 4mm and further.

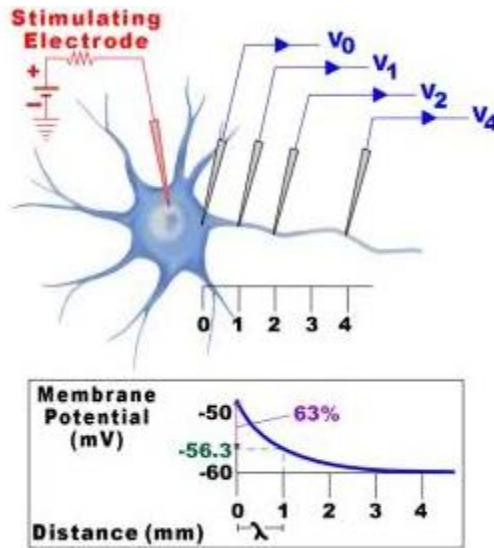


Figure 10: Length (Space) Constant

The length constant or space constant is an index that represents how far a membrane depolarization will spread along the axons' length in response to a stimulus. The space constant is denoted by the symbol λ . In the previous example space constant was 1 mm. At this distance the membrane potential will decay by 63% from its maximal value. The membrane potential spreads further for larger space constants. In other words, if the space constant was 2mm then the membrane potential would spread further along the axon and vice versa. Similar to the time constant, the space constant is also a passive property of the cell membrane. Therefore, these properties influence the rate of action potential propagation. The space constant can be compared to the race track surface. If the surface is smooth then the race car will decelerate very slowly due to lack of friction. If the surface is very sandy and muddy, then the car will decelerate much faster due to all the resistance from the sand and mud.

The space constant can be calculated by the following formula:

$$\lambda = \sqrt{\frac{dR_m}{4R_i}} \quad (18)$$

where	λ	space constant [cm]
	R_m	membrane resistance [m Ω]
	R_i	axoplasm resistance [m Ω]
	d	diameter of an axon [cm]

Intracellular resistance is the opposition of ion propagation in the intracellular environment along the length of the axon. An axon stimulation results in a change in charge distribution which propagates along the axon. During this process some of the charges will diffuse back to the extracellular environment whereas the other charges will continue to move along the length of the axons. Therefore, if the membrane resistance is high, then fewer ions will diffuse to the extracellular environment and more ions will stay inside the axoplasm. Since the diffusion rate is low, ions move further along the axon.

1.17 Propagation Velocity

So how does the action potential propagation velocity depend on the time constant and the length constant? As it was discussed in previous sections, a small time constant means that the adjacent regions of an axon will depolarize more quickly. More rapid depolarization of the adjacent regions will bring the adjacent regions to the threshold value sooner. Based on this information we can conclude that the small time constant results in a high action potential propagation velocity.

The space constant also has a direct impact on the action potential propagation velocity. A large space constant means that the change in a voltage potential spreads to a greater distance away from the stimulation site. This brings the distant regions closer to the threshold level. Based on this information we can conclude that a high space constant results in a high action potential propagation velocity.

Therefore, action potential propagation velocity is inversely proportional to the time constant and directly proportional to the space constant. These relationships can be combined in the following equation for the action potential propagation velocity:

$$\text{Velocity} \propto \frac{1}{C_m} \sqrt{\frac{d}{4R_m R_i}} \quad (19)$$

where	d	diameter of a neuron [cm]
	C_m	membrane capacitance per unit area [F/cm ²]
	R_m	membrane resistance [mΩ]
	R_i	Intracellular resistance [mΩ]

According to (19) we can identify which properties have an impact on an action potential propagation velocity in a neuron. For instance if we were to model a neuron with a high action potential propagation velocity, we could increase the diameter of an axon. There is a limitation to modifying only the diameter of the axon. In order to double the action potential propagation velocity, we must quadruple the diameter. The same principle can be applied to membrane resistance and intracellular resistance. In order to double the velocity, we must decrease the intracellular resistance by 4 times. Therefore, the most efficient way to increase propagation velocity is to decrease the capacitance of the membrane. In order to double action potential propagation velocity, we must half the

capacitance of the membrane. One way to decrease membrane capacitance is to introduce some kind of insulation layer to the membrane. This layer is known as myelin sheath.

In order to generate an action potential, any axon is required to have voltage gated sodium channels and voltage gated potassium channels. If these ion channels are blocked by an insulation layer, the action potential will have no location to be initiated.

1.18 Action Potential Propagation in Myelinated Fibers

The central nervous system has different types of neurons each having its distinct characteristics and physical properties. For example, type A and type B neurons are myelinated fibers whereas type C neurons are unmyelinated. As it was discussed in previous sections, type C fibers have the slowest action potential propagation velocity. This property can be explained by the absence of the insulation layer called the myelin. Myelin serves as an electrical insulator which facilitates rapid action potential propagation velocity. Myelin is present in type A and type B neurons, but not in type C neurons [17]. Myelin is composed of a number of layers of the specialized plasma membrane of oligodendrocytes. Oligodendrocytes are specialized cells with most of the cytoplasm excluded [6]. The main function of these cells is to provide the physical support and insulation properties to the axons. The oligodendrocyte membrane contains specialized proteins and lipids that greatly contribute to an orderly packed sheath. Axons are encapsulated in a small blocks of myelin, called internodes. These insulation

blocks are roughly a few hundred microns in length [6]. These internodes are separated by small gaps of unmyelinated sections of axon which are called the nodes of Ranvier. Even though myelinated axons are present in all parts of the human brain, they are most commonly found in the white matter. This is the most common area for the myelin diseases. For example multiple sclerosis is a disease that involves the immune system attacking the central nervous system. More specifically, during multiple sclerosis the myelin is damaged. As a result scar tissues impair action potential propagation [6].

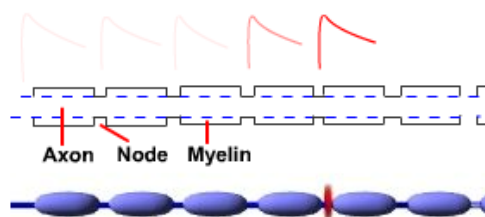


Figure 11: Action potential propagation in myelinated fiber

Figure 11 represents action potential propagation in a myelinated fiber. The action potential propagates from the leftmost internode to the right. The action potential propagates actively at the nodes of Ranvier. The mechanism of action potential propagation at the nodes of Ranvier was discussed in previous sections. So how does the action potential propagate within the internodes? These regions of axons have a myelin layer which acts as an insulator and prevent any ion movement across the cell membrane. In fact, internodes do not have any ion channels for ions to move across the cell membrane [6]. Therefore, the action potential is produced at the nodes of Ranvier. Action potential propagation across internodes is achieved by passively spreading the action potential until the next node of Ranvier is reached. Just as the temperature would spread across the metal rod, sodium and potassium ions will be pushed from one node of Ranvier to the next one. As the action potential spreads away from its origin, it will decrease in amplitude. In other words, the action potential will be smaller by the time it

reaches the next node of Ranvier. If action potential is not high enough to reach a certain threshold at the next node of Ranvier, a new action potential will not fire and action potential propagation will be terminated.

This process will repeat itself until one of two conditions is met: 1) the action potential reaches its final destination, the axon terminal, or 2) the action potential stops at a certain node of Ranvier due to the inability to reach the specific threshold required to originate a new action potential.

1.19 Action Potential Propagation: Neuron Properties

The analysis of action potential propagation should consider axial currents as well as transmembrane currents. Figure 12 illustrates the schematics of action potential propagation along an axon. The total length of an axon can be divided into a number of units. Each unit of length can be represented by the Hodgkin and Huxley model.

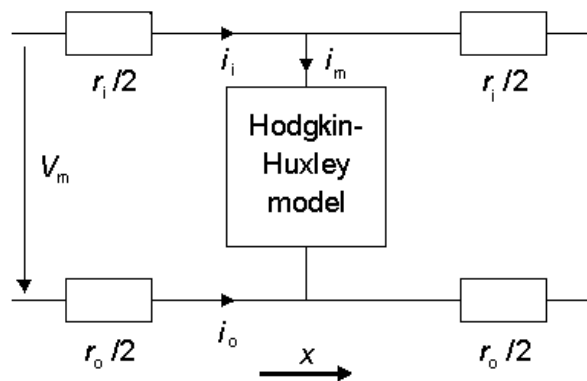


Figure 12: Schematics of Action potential propagation [4]

Figure 12 shows the schematics of the unit length of an axon. In the schematic r_i represents the intracellular resistance per unit length, whereas r_o represents the

extracellular resistance per unit length. The Hodgkin and Huxley model describes the movement of electrical current between extracellular and intracellular environments.

The total membrane current per unit length can be described by the following equation [4]:

$$i_m = \frac{1}{r_i + r_o} \frac{\partial^2 V_m}{\partial x^2} = 2\pi a I_m \quad (20)$$

where	i_m	total membrane current per unit length [$\mu\text{A}/\text{cm}$]
	I_m	membrane current per unit area [$\mu\text{A}/\text{cm}^2$]
	r_i	intracellular membrane resistance [$\text{m}\Omega$]
	r_o	extracellular membrane resistance [$\text{m}\Omega$]
	a	axon radius [cm]
	V_m	membrane voltage [mV]

The intracellular resistance per unit length is represented in (21)

$$r_i = \frac{\rho_i}{\pi a^2} \quad (21)$$

where	ρ_i	axoplasm resistivity [$\text{k}\Omega\text{cm}$]
	a	axon radius [cm]
	r_i	axoplasm resistance per unit length [$\text{k}\Omega/\text{cm}$]

In reality the extracellular medium is very wide. Therefore, extracellular resistance per unit area can be omitted ($r_o = 0$) [4]. Solving (20) for I_m we obtain the equation for propagating nerve impulse:

$$I_m = \frac{a}{2\rho_i} \frac{\partial^2 V_m}{\partial x^2} \quad (22)$$

The Hodgkin and Huxley model is based on the assumption that a neuron has a circular cross-section. Most neuron fibers do not have a perfectly circular shape [2]. Neuron fibers have convoluted elliptical shapes. Under steady conditions, action potential propagation velocity is constant [4]. Therefore, the membrane potential obeys the wave equation:

$$\frac{\partial^2 V_m}{\partial x^2} = \frac{1}{\theta^2} \frac{\partial^2 V_m}{\partial t^2} \quad (23)$$

(23) is substituted into equation (20) and solved for I_m . The result is substituted into (7) and forms the equation for action potential propagation.

$$\frac{a}{2\rho_i\theta^2} \frac{\partial^2 V_m}{\partial t^2} = C_m \frac{\partial V_m}{\partial t} + (V_m - V_{Na})G_{Na} + (V_m - V_K)G_K + (V_m - V_L)G_L \quad (24)$$

where a axon radius [cm]
 ρ_i axoplasm resistivity [kΩcm]
 θ velocity of conduction [m/s]

The action potential propagation model can be represented as a sequence of Hodgkin and Huxley models connected in series. The output from each unit is used as an input to the next unit, where each unit is represented by a single instance of the Hodgkin and Huxley model. In more complex geometries, where multiple neurons are connected to multiple neurons, the same principle can be applied. Each neuron can be represented as a series of Hodgkin and Huxley models.

The human body has a wide variety of different neurons with different physical properties. Depending on the location in the body, axons have different characteristics. For instance, the axon length differs greatly depending on where the cell is located in the body [6]. Another variable property of axons is their diameter. These properties have a great impact on action potential propagation [19]. Therefore, it is important to analyze how the physical properties of a neuron influence action potential propagation along the length of the axon.

The diameter of an axon varies from $0.1\mu\text{m}$ to $20\mu\text{m}$ for large myelinated fibers in vertebrates [6]. Axons of invertebrates can have even larger diameters. In the case of a squid giant axon which Alan Hodgkin and Andrew Huxley used in their experiments, the size of the axon diameter is measured in the millimeter range [20]. The reason invertebrates have such a large diameter is that invertebrate axons are unmyelinated [20]. As previously stated, myelin contributes greatly to action potential propagation velocity. Action potential propagation velocity is an important factor for different organisms. It influences reflexes and allows organisms to escape from predators. This aspect is important especially for those creatures, whose bodies are unable to produce myelin. Therefore, it may be beneficial to investigate the effect of axon diameter.

So how does the axon diameter influence action potential propagation? One can argue that the larger the axon diameter the lower the membrane resistance inside the cell. In other words, axons with larger diameter have more ion channels that can be activated at the same time. This property leads to a higher number of ions crossing cell membrane at the same time. Therefore the result is a lower membrane resistance.

Another property that influences action potential propagation is intracellular resistance. During the depolarization phase, ion currents move from the depolarized membrane to adjacent polarized areas [19]. This phenomenon results in the depolarization of neighboring areas of an axon. Axoplasmic current also moves away from depolarized to polarized regions. Axoplasmic currents then cross the cell membrane and move into the extracellular environment. This process is known as electrotonic conduction during which ions move passively from depolarized to polarized regions. Electrotonic conduction has three distinct components: 1) ion movement through the extracellular environment; 2) ion movement through the intracellular environment; 3) ion movement across the cell membrane [19].

So which factors influence electrotonic conduction? Extracellular conduction is impacted by the extracellular resistance. Increase in extracellular resistance will result in the decrease in extracellular conductance and therefore weaker electrotonic propagation. The Hodgkin and Huxley model omits this factor since extracellular resistance is very small and constant along the entire length of an axon [4].

On the other hand, intracellular conductance is impacted by axoplasmic resistance. Intracellular resistance or axoplasmic resistance is the electrical resistance of the intracellular environment. The significance of this property is important because it affects the rate at which the action potential propagates along the axon [19]. The

intracellular environment of excitable cell contains a wide variety of molecules. Some of these molecules are electrically conductive (e.g. sodium ions, potassium ions), whereas some of these molecules are not-electrically conductive (e.g. lipids, proteins).

Therefore, if the intracellular environment contains a lot of not-electrically conductive molecules, then more ions (both electrically conductive and not-electrically conductive) will travel through the membrane channels. This characteristic will result in high intracellular resistivity and therefore it will decrease the action potential propagation velocity [19].

Chapter 2 - Modeling

2.1 Modeling the Hodgkin and Huxley Model

Equation 7 can be modeled using Simulink by utilizing subsystem blocks which would represent the voltage gated sodium channel (NaV), the voltage gated potassium channel (KV), the leak channel (Leak) and the membrane capacitance (Membrane). Each of these blocks, except for the Membrane capacitance block, takes voltage (V) as an input and returns current (I) as an output. The membrane block takes the sum of current from all ion channels as an input and returns voltage as an output. The scope utility can be used for output verification purposes. Figure 13 illustrates one of the many possible ways to implement the Hodgkin and Huxley model using Simulink.

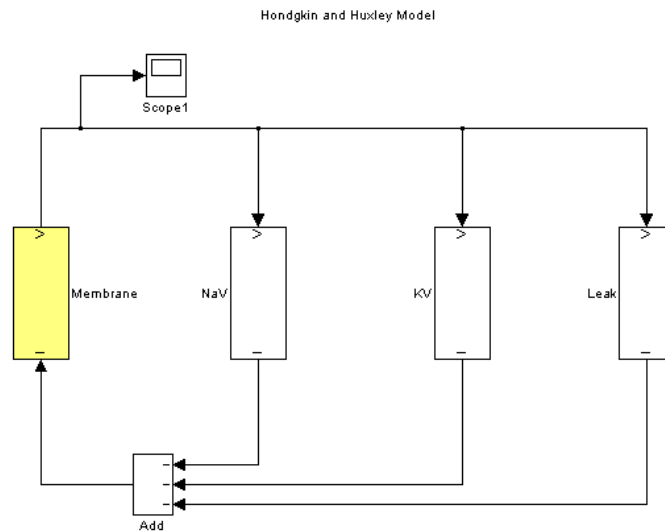


Figure 13: High Level Schematics of the Hodgkin and Huxley model

Alan Hodgkin and Andrew Huxley developed the following equations for transfer rate coefficients α_n ; α_h ; α_m ; β_n ; β_h ; β_m :

$$\alpha_n = \frac{0.1 - 0.01V'}{e^{(1-0.1V')} - 1} \quad (25)$$

$$\beta_n = \frac{0.125}{e^{0.0125V'}} \quad (26)$$

$$\alpha_m = \frac{2.5 - 0.1V'}{e^{(2.5-0.1V')} - 1} \quad (27)$$

$$\beta_m = \frac{4}{e^{\left(\frac{V'}{18}\right)}} \quad (28)$$

$$\alpha_h = \frac{0.07}{e^{0.05V'}} \quad (29)$$

$$\beta_h = \frac{1}{e^{(3-0.1V')} + 1} \quad (30)$$

where	V'	$V' = V_m - V_r$ [mV]
	V_r	the resting voltage [mV]
	α_n	transfer rate coefficient for n -particle from closed to opened state
	β_n	transfer rate coefficient for n -particle from opened to closed state
	α_m	transfer rate coefficient for m -particle from closed to opened state
	β_m	transfer rate coefficient for m -particle from opened to closed state
	α_h	transfer rate coefficient for h -particle from inactivated to activated state
	β_h	transfer rate coefficient for h -particle from activated to inactivated state

2.2 Modeling Voltage Gated Sodium Channel

Figure 14 shows one of the many ways to implement the voltage gated sodium channel using Simulink. The upper portion of the model describes the behavior of

sodium channel activation and the behavior of sodium channel inactivation. The behavior is modeled by implementing equations (10) and (11). The lower part of the model implements the sodium conductance, represented by (9). And finally, the sodium channel is modeled according to equation (8).

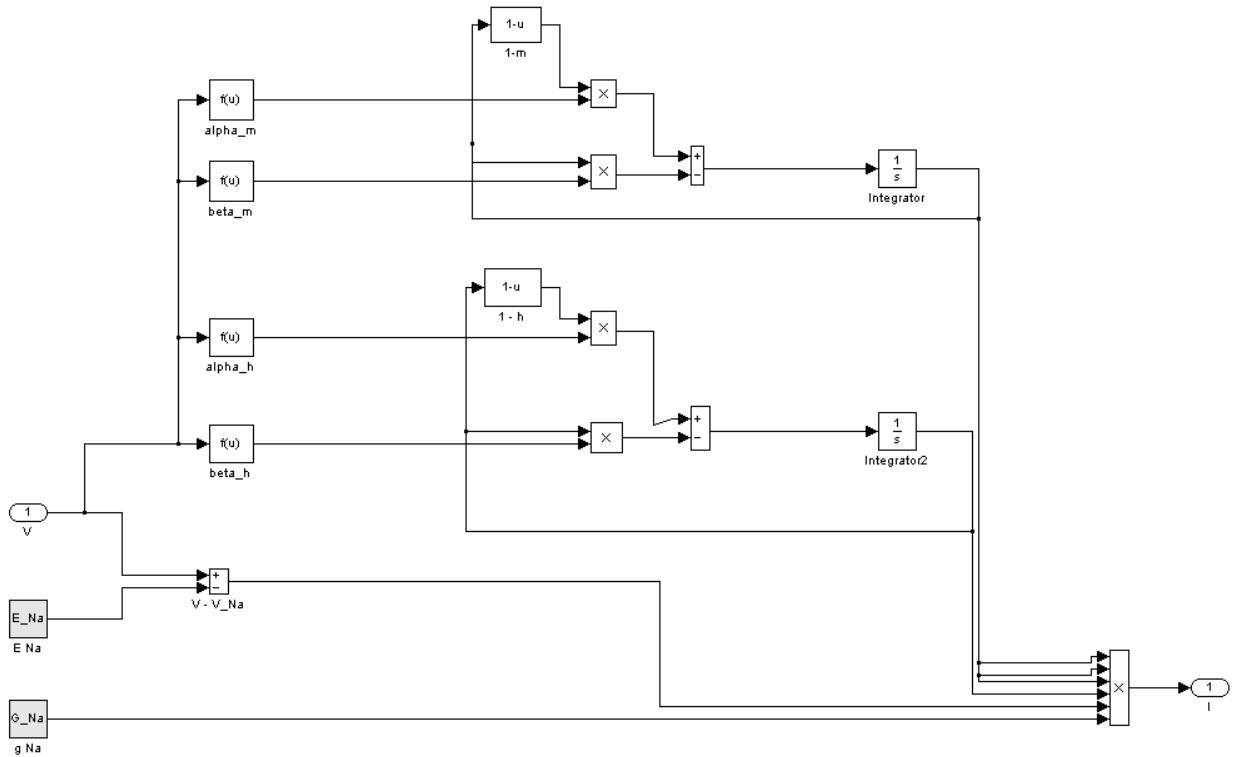


Figure 14: Voltage Gated Sodium channel Simulink model

2.3 Modeling Voltage Gated Potassium Channel

Figure 15 shows one of the many ways to implement a voltage gated potassium channel using Simulink. The upper portion of the model describes the behavior of potassium channel activation. This behavior is simulated by implementing equation (14). The lower part of the model implements the potassium conductance, represented by equations (13). And finally, the potassium channel is modeled according to (12).

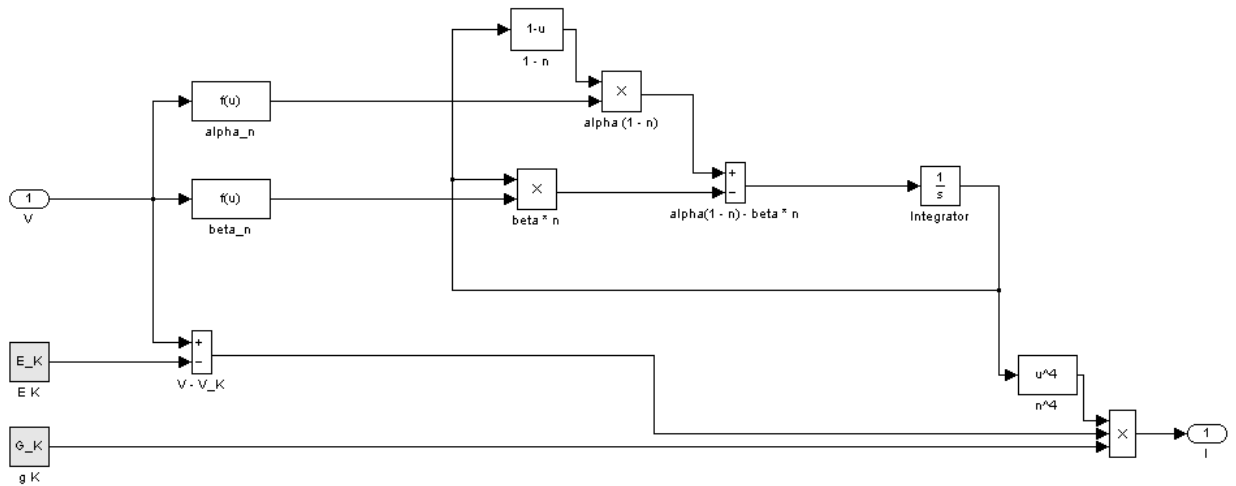


Figure 15: Voltage Gated Potassium Channel Simulink model

2.4 Modeling Leak Channel

Figure 16 shows one of the possible ways that to model the leak channel using Simulink. This scheme implements equation 15.

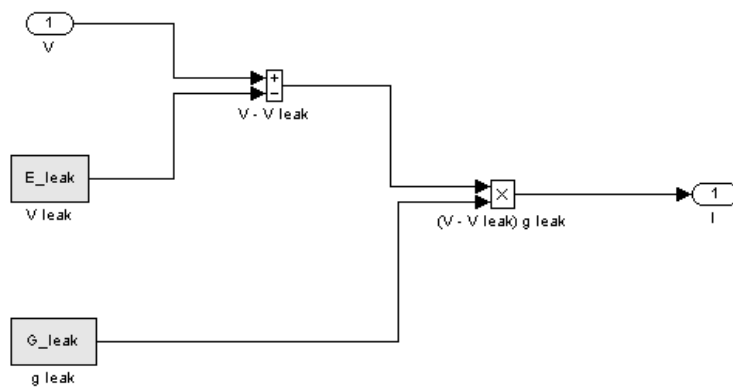


Figure 16: Leak Channel Schematics

2.5 Modeling Membrane Capacitance

Figure 17 shows one of the possible ways to model the membrane capacitance using Simulink. The membrane capacitance subsystem block takes the current (I) as an input and returns voltage (V) as its final result. This subsystem block implements (16) for the voltage (V_m). In this model, the current (I_m) is divided by the capacitance (C_m) and then integrated. The current (I_m) consist of the capacitive current.

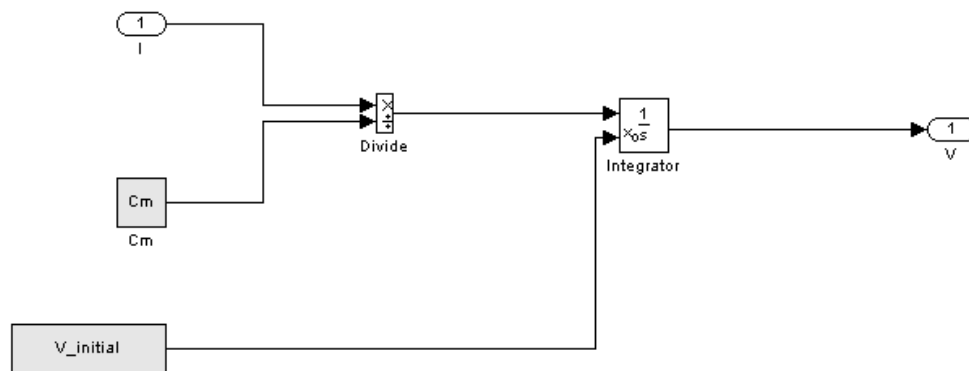


Figure 17: Schematics of membrane capacitance

The output from the Hodgkin and Huxley model implemented using Simulink is shown in Figure 18. The membrane potential starts rising from its resting state (at -70mV) and reaches its threshold value at approximately -60mV. Reaching the threshold point will initiate an action potential. Voltage gated sodium channels open rapidly and depolarize the excitable cell. This state represents the start of the depolarization phase. Membrane potential reaches its highest point at positive 18mV. At this point voltage gated potassium channels open causing an outflow of potassium ions. This state represents the start of the repolarization phase. The excitable cell repolarizes until the membrane potential reaches -80mV. At this point the excitable cell

is hyperpolarized since its membrane potential (-80mV) is lower than its resting membrane potential (-70mV). During the next phase, the action potential depolarizes to its resting state (-70mV). This is the slowest phase which takes more than 12ms to complete. According to the result from the Simulink model we can conclude that it takes approximately 20ms for a single action potential complex to occur. As we will see in later sections, the action potential propagation velocity depends on different experimental conditions (i.e. temperature) and physical properties of a neuron.

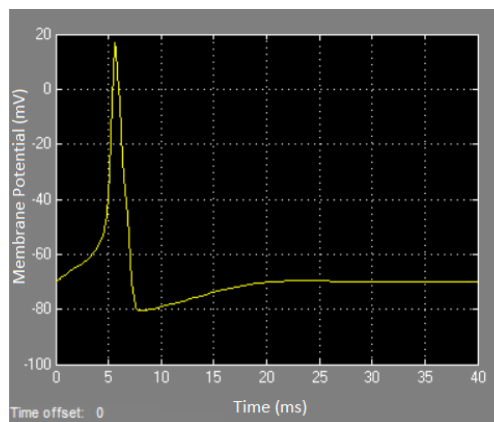


Figure 18: Action Potential

An action potential as represented in Figure 18 does not propagate since it is based on the basic Simulink model which does not implement action potential propagation. As we will see later in this paper, the Simulink model needs to be modified in order to simulate action potential propagation.

2.6 Modeling Periodic Stimulation

Figure 19 shows the modified Hodgkin and Huxley model that simulates action potentials by periodic stimulation of the excitable cell. The signal builder block defines the stimulation signal that is added to the total current I_T .

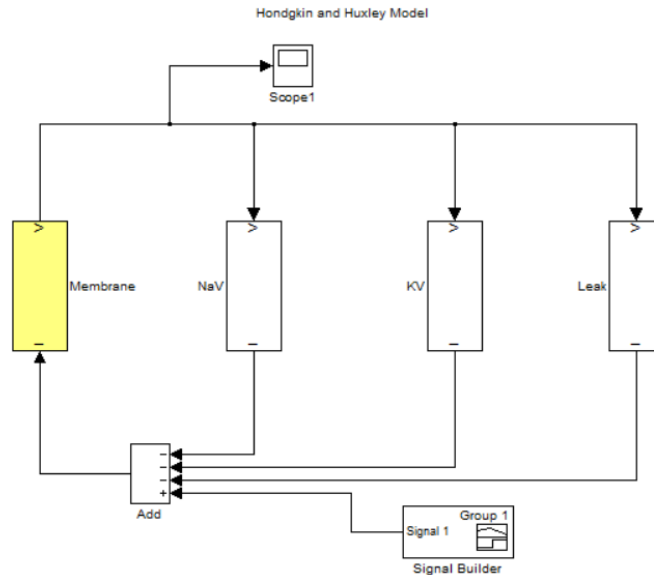


Figure 19: The Hodgkin and Huxley model (External stimulation)

As a result of this change, an action potential is generated every time the Signal Builder outputs a signal. Figure 20 shows the output from the Signal Builder block and the output from the Hodgkin and Huxley model aligned along the x-axis. The x-axis represents time in milliseconds. The simulation runs for 150ms. The initial action potential complex is generated by the Hodgkin and Huxley model without any output from the Signal Builder block. This is possible due to initial conditions of n -particle, m -particle and h -particle. All consecutive action potential complexes are initiated by the periodic stimulation originated by the Signal Builder block. For simplicity, stimulation intervals are selected so that the membrane potential has enough time to return to its resting state of -70mV. In the example below the stimulation interval is set to 20ms.

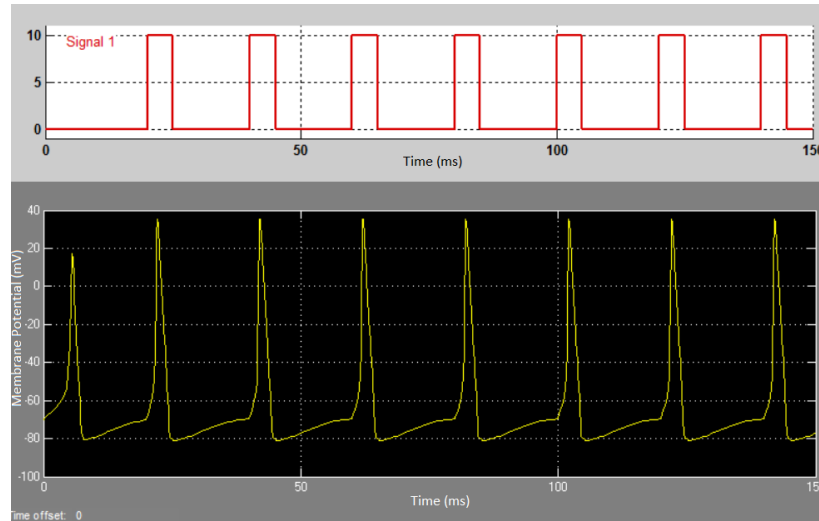


Figure 20: Action potential – periodic stimulation

2.7 Modeling Temperature Effect

Temperature is an important factor which is almost always taken into consideration during experiments. Body temperature influences many processes in the body. In most experimental studies the temperature is held uniform and many human and animal models are evaluated at their normal body temperature, i.e. approximately 37°C . In reality, body temperature may vary in human and/or animal systems. These variations could be caused by different inflammatory responses to a disease, foreign material in the body such as a prosthesis, or other special circumstances, such as surgery. Temperature in cold-blooded living systems, such as the squid, varies with that of the environment. The experimental temperature value used by the Hodgkin and Huxley model is 6.3°C [2].

Temperature differences between the nominal room temperature and the normal human body temperature are sufficient to produce a noticeable difference in numerical results. Temperature variations can be expressed in many different ways. For example,

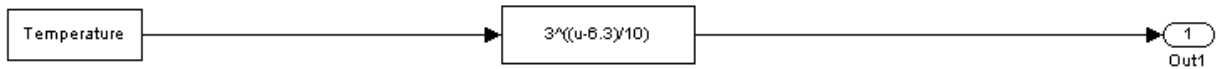
a sharper action potential peak may represent a higher experimental temperature. Temperature variances may produce significant differences in the experimental results and therefore in the overall performance of living tissue. For that reason, it is important to consider temperature in the model.

The effect of temperature can be represented by (31) [4].

$$3^{(T-6.3)/10} \tag{31}$$

where T temperature [C°]

In order to take into account the effect of temperature, the Simulink model has to be modified. Figure 21 shows the implementation of the Temperature subsystem.



The effect of the temperature is taken care in the model so that the right-hand sides of the Equations for n, m and h are multiplied by the factor $3^{(T-6.3)/10}$

Figure 21: Temperature Effect

The model of the Potassium channel has to be modified in order to take into consideration the effect of temperature. The right hand side of the (32) needs to be multiplied by the temperature factor, represented by equation 31 [4].

$$\frac{dn}{dt} = \alpha_n(1 - n) - \beta_n n \tag{32}$$

Therefore, the potassium channel subsystem needs to be modified as it is represented in Figure 22. The temperature factor which is represented by the “Temperature” subsystem is multiplied by $\alpha_n(1 - n) - \beta_n n$.

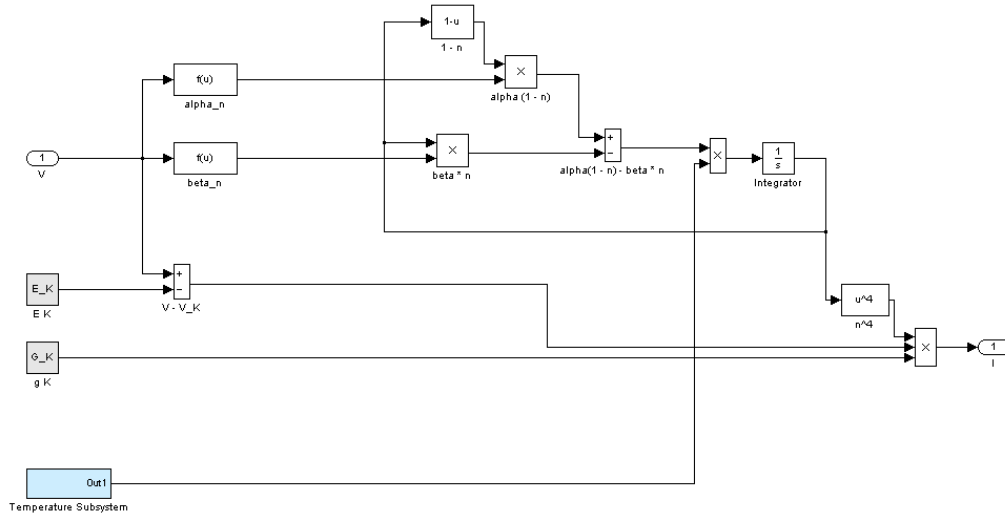


Figure 22: Potassium Channel (with temperature effect)

The model of Sodium channel has to be modified in order to account for the temperature effect. The right hand side of (33) and (34) have to be multiplied by the temperature factor, represented by (31) [4].

$$\frac{dm}{dt} = \alpha_m(1 - m) - \beta_m m \quad (33)$$

$$\frac{dh}{dt} = \alpha_h(1 - h) - \beta_h h \quad (34)$$

Therefore, the sodium channel subsystem needs to be modified as it is represented in Figure 23. The temperature factor which is represented by the “Temperature” subsystem is multiplied by $\alpha_m(1 - m) - \beta_m m$ and by $\alpha_h(1 - h) - \beta_h h$.

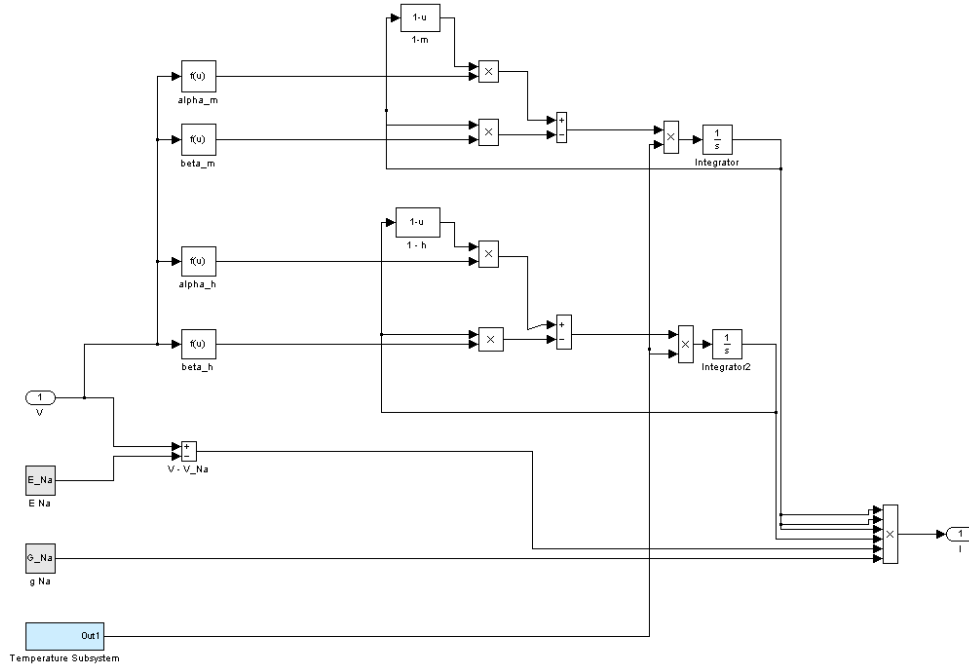


Figure 23: Sodium Channel (with temperature effect)

These modifications to the Hodgkin and Huxley model are required to enhance the model to take into consideration the effect of temperature. Two variables were conveniently introduced to the Simulink model: “Temperature_Bit” and “Temperature”. The “Temperature_Bit” variable is used to enable and to disable the analysis of temperature effect. For instance, by setting “Temperature_Bit” to 1, the user can enable the analysis of temperature effects. Therefore, setting “Temperature_Bit” to 0 disables the effect of temperature.

The “Temperature” variable is used for setting the experimental temperature. Most of the results described in this work are based on the giant squid axon model. Therefore, 6.3C° is a default temperature value which is used for simulation of action potential initiation and action potential propagation.

Chapter 3 – Simulation Results

3.1 Membrane Conductance

The membrane impedance decreases significantly during action potential activation. This behavior is a result of an increase in the membrane conductance. In other words, the membrane capacitance does not vary during action potential activation. Therefore, membrane capacitance is represented by a constant value in the Hodgkin and Huxley model. Figure 24 shows the increase in sodium conductance (G_{Na}), potassium conductance (G_K) and their sum ($G_K + G_{Na}$).

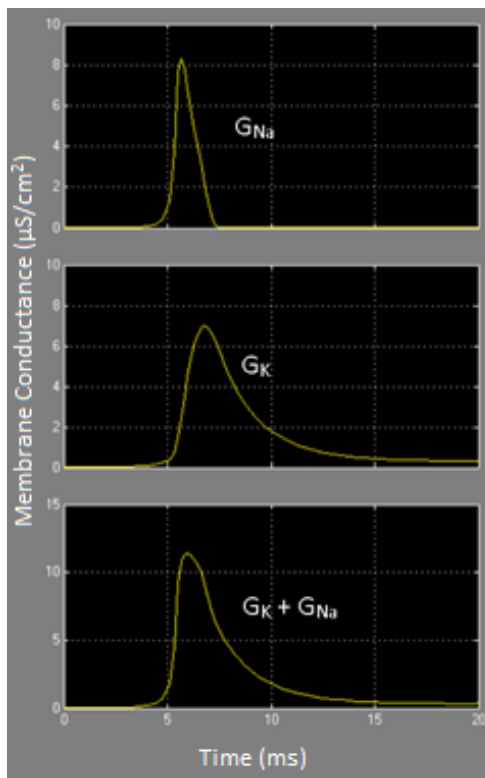


Figure 24: Sodium and Potassium conductance and their sum

These increases in conductance are observed during action potential activation. Therefore, sodium conductance (G_{Na}) starts its upward trend when voltage gated sodium channels are activated. This leads to a rapid inflow of sodium ions. Potassium conductance (G_K) does not start its upward trend at the same time. Instead, potassium conductance starts its rapid increase when the sodium conductance is almost at its maximum level. At this point voltage gated sodium channels are being deactivated. At the same time voltage gated potassium channels are activated and the repolarization phase takes place. Potassium conductance takes more time than the sodium conductance. After the cell membrane is hyperpolarized its resting potential is slowly restored. This is the longest phase in the action potential initiation. Mostly potassium ions are involved in this phase. Potassium ions enter the intracellular environment through leak channels and active pumps.

3.2 Gating Variables

Gating variables n , m and h are activation and inactivation particles that play an important role in opening and closing voltage gated sodium channels and voltage gated potassium channels.

Potassium channels open and close in accordance with the state of particle n [2]. The n -particles have two states: the permissive, the state when the potassium channel is open and non-permissive, the state when potassium channel is closed. In order to activate a single potassium channel, four n particles need to be in the permissive state.

Sodium channels open and close in accordance with the state of particle m and particle h [2]. Particle m is an activation parameter for the sodium channel. m -particles have two states: permissive (the state when the sodium channel is open) and non-

permissive (the state when the sodium channel is closed). In order to activate a single sodium channel, three m particles need to be in the permissive state. The sodium channel is deactivated by the deactivation particle h . It requires one h particle to deactivate a sodium channel [2].

Figure 25 illustrates how values of these particles change during action potential activation.

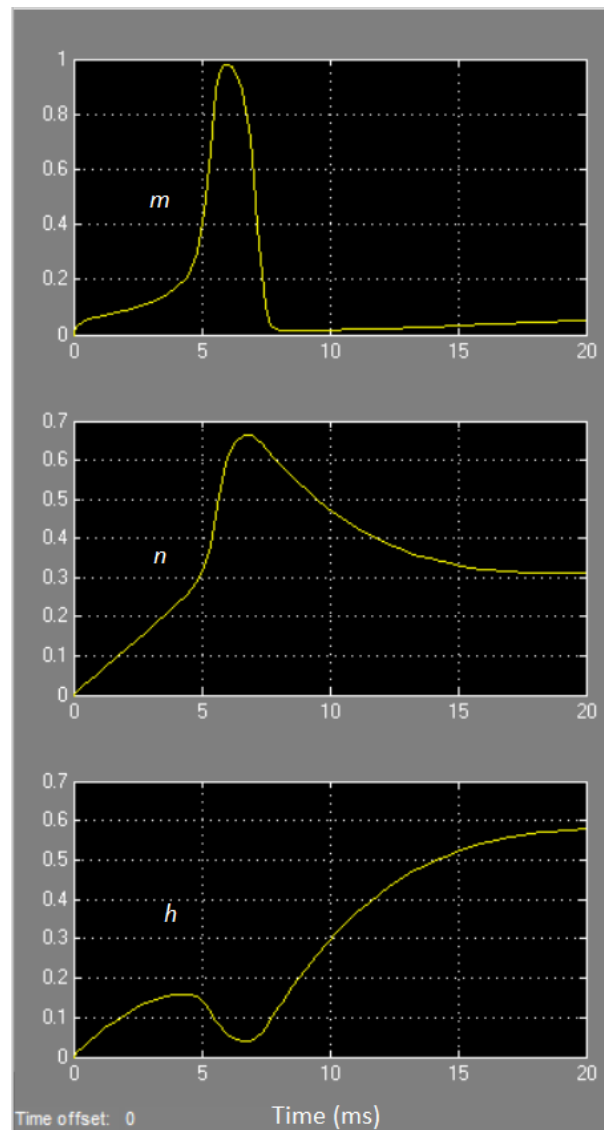


Figure 25: Gating variables (n , m and h) during action potential activation

In this model the initial values for the n , m and h particles is set to 0. Membranes of excitable cells continuously exist throughout a lifetime of those cells [2]. In other words, there is no specific mechanism that starts or stops a cell. Therefore, there are no such conditions where all of the three particles (n , m and h) would have a value of 0. The purpose of this simulation is to determine the initial values for the n , m and h particles. The initial values for gating variables m , n , h should be equal to the same values obtained during the resting membrane potential (-70mV). By using the graph represented in Figure 25 we can determine that the values of n , m and h particles should be initialized to 0.3176, 0.0529 and 0.5919 respectively. The simulation presented in this work does not initialize m , n , h variables and therefore the initial values of these parameters is 0.

These simulation results are consistent with the results described in the simulation of the membrane action potential published in “Principles of computational modeling in neuroscience” [21]. Figure 26 displays the time courses of membrane potential, conductance and gating variables during an action potential.

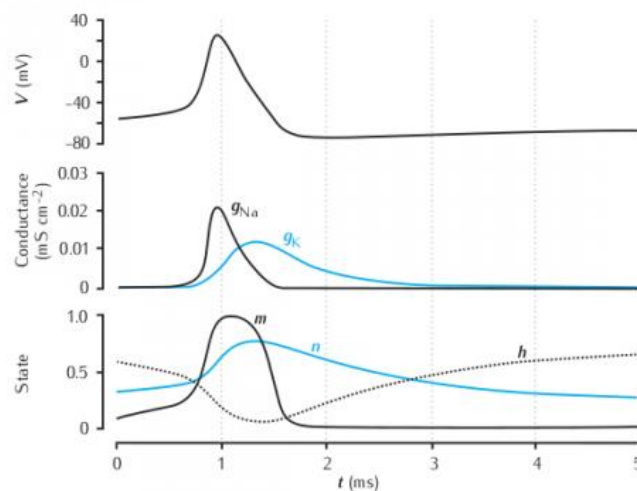


Figure 26: Membrane Action Potential [21]

3.3 Ionic Currents

An action potential involves currents from different ion channels. Ion flows can be broken down into categories based on ion channels (e.g. I_{Na} , I_K , I_{Leak} and I_{Total}). The Hodgkin and Huxley model can be used for studying the behavior of different ion channels. This analysis provides a better understanding of the behavior of different ion channels during action potential initiation.

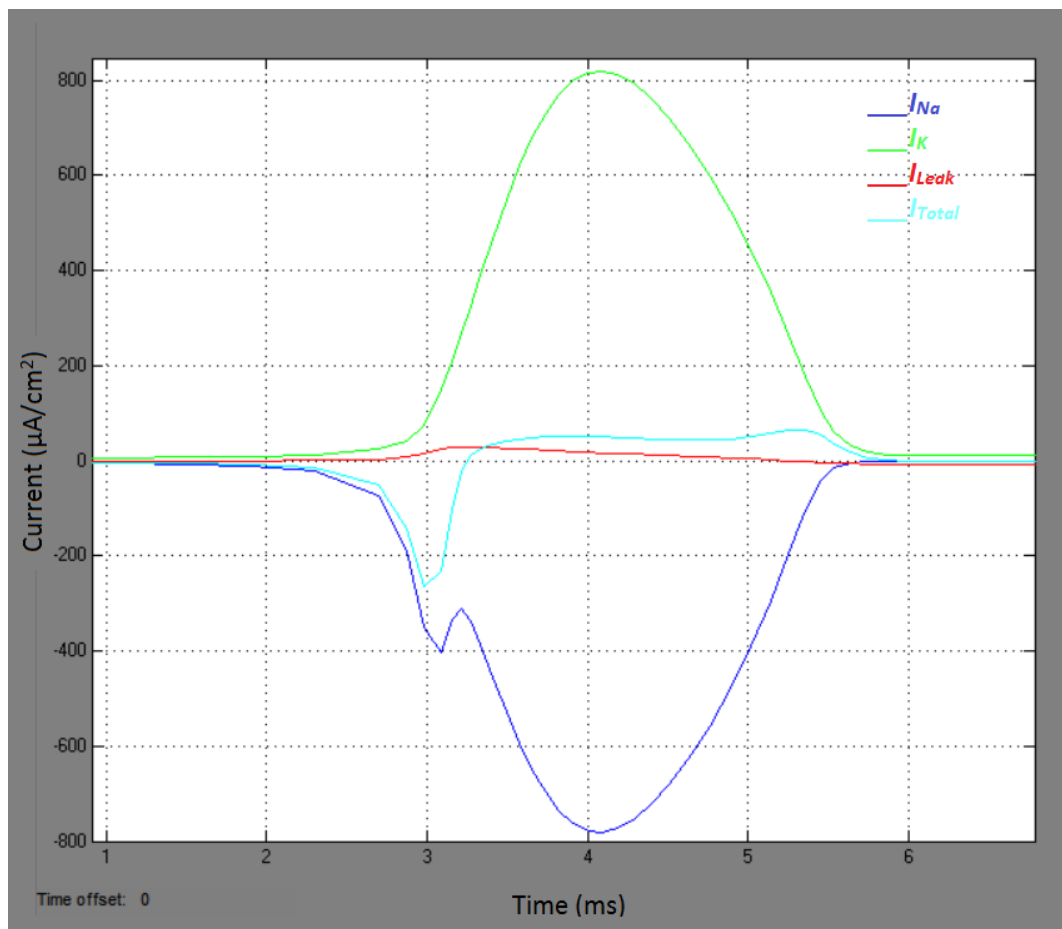


Figure 27: Ionic currents

Figure 27 illustrates changes in ionic currents during an action potential. The negative current represents an inflow of ions from the extracellular environment to the intracellular environment. The positive current represents an outflow of ions from the

axoplasm to the extracellular environment. A zero-value represents the equilibrium state in which there are no ionic movements across the cell membrane.

When the cell membrane is at rest, the total ion current is equal to zero. During the resting membrane potential the ionic currents are balanced such that there is no net current flow across the membrane. According to the graph the total current (I_{total}) is equal to 0 when the cell membrane is at rest. This condition changes when the net ion current changes to the non-zero state. In the simulation described in Figure 27, the depolarization phase starts approximately at 1.1ms. During this stage the sodium current initiates and sodium ions enter the excitable cell. This is represented in the downward slope of the I_{Na} curve. The repolarization phase starts at approximately 1.5ms. During this stage voltage gated potassium channels open allowing potassium ions to escape into the extracellular environment. At the same time voltage gated sodium channels start to close. The total current (I_{total}) represents the sum of currents from all ion channels and is shown in red color on the Figure 27. During the initial phase of depolarization (from 1.1ms to 1.7ms), the total current consists primarily of the sodium current. During this period the I_{total} curve is in negative value range. The spike on the sodium current curve represents the time when the sodium channels start to deactivate. At 2.8ms the sodium current reaches its highest value of $-830\mu A/cm^2$ and then starts to decay. At the same time the potassium current reaches its highest point at $850\mu A/cm^2$ and starts to decline due to inactivation of the voltage gated potassium channels. Voltage gated sodium channels are closed at 4.1ms. Therefore, sodium current through voltage-gated sodium channels is equal to zero. At this point sodium ions can cross the cell membrane only through the leak channels and active pumps. During the action potential some ions leak from axoplasm into the extracellular environment and vice versa. This movement is done primarily through the leak channels. The positive leak

current represents an outflow of ions from the axoplasm to the extracellular environment. In the example shown in Figure 27 this stage starts during repolarization, at approximately 1.5ms. Potassium ions escape from the axoplasm to the extracellular environment. The negative leak current represents an inflow of ions from the extracellular environment to the intracellular environment. In the example shown in Figure 27 this stage starts during hyperpolarization, at approximately 3.7ms. Sodium ions enter the axoplasm and membrane potential is restored to its resting state.

These results are consistent with the results Francisco Bezanilla described in his work “Ionic currents during the nerve impulse” [22]. Simulation results described in these work are similar to the results displayed in Figure 28.

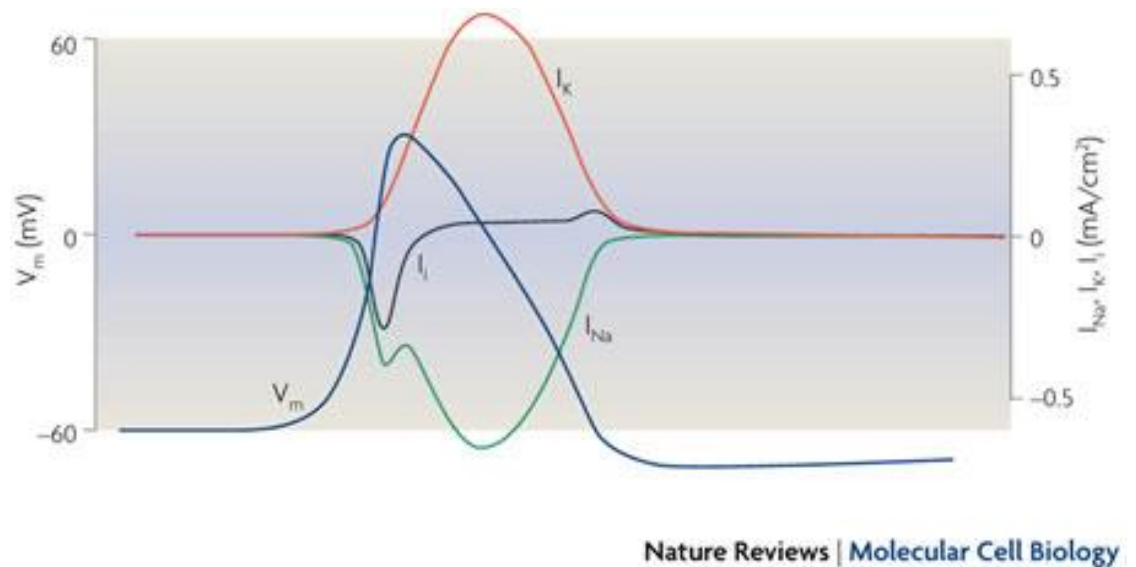


Figure 28: Ionic currents during the nerve impulse [22]

3.4 Temperature Effect

During their experiments, Alan Hodgkin and Andrew Huxley were using 6.3°C as their baseline temperature. This temperature is consistent with the model presented in this work. In order to investigate the temperature effect, the “Temperature” variable was set to several different values. Initially the temperature variable was set to 6.3°C . Action potential generated under this condition was used as a reference value.

Figure 29 illustrates the comparison of action potentials generated at 6.3°C and 12.3°C . The difference in scale in the two graphs is noteworthy. We can see that the action potential amplitude is slightly greater in the 6.3°C simulation and it reaches 16.4mV at its highest point. In comparison the action potential reaches only 4mV at its highest point in the 12.3°C simulation. According to the results of this simulation, action potential amplitude decreases with a temperature rise.

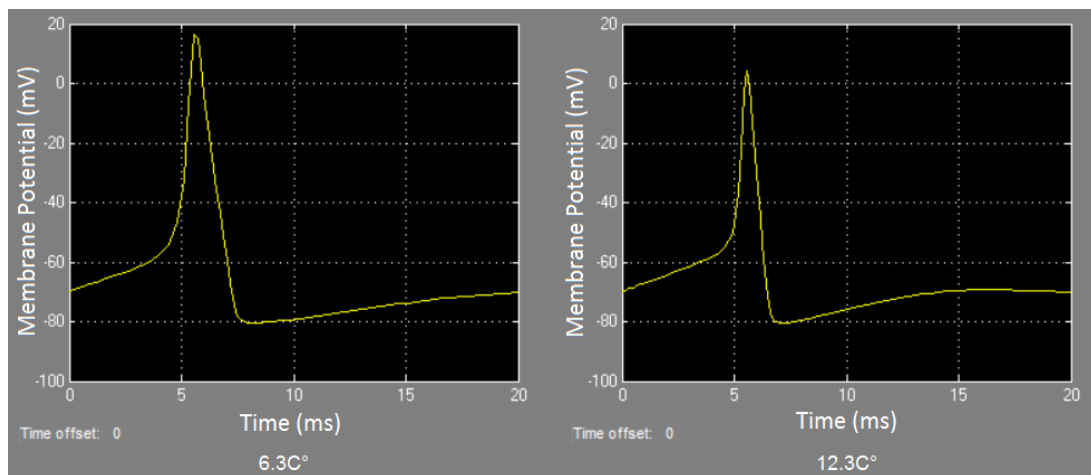


Figure 29: Temperature Effect Results

One can also notice that the action potential complex is slightly wider in the 6.3°C experiment. It reaches its lowest point of -80.5mV at 8.2ms . The action potential then

slowly returns to -70 at 20ms. In the 12.3 C° experiment the action potential has a narrow complex which reaches its lowest point of -80.45mV at 7.15ms and reaches -70mV (resting potential) at 14.2ms. At this stage all potassium channels are closed. The only active channels at this time are the leak channels and active pumps. These channels are responsible for restoring the resting state of an action potential.

These simulation results are consistent with the results described in “Warm Body Temperature Facilitates Energy Efficient Cortical Action Potentials” [23].

Figure 30 illustrates the action potential of classical HH model neuron generated at 6, 18, and 27 C°, respectively. Bottom panel shows the corresponding Na⁺ and K⁺ currents during action potential generation for temperature at 6, 18, and 27 C°, respectively.

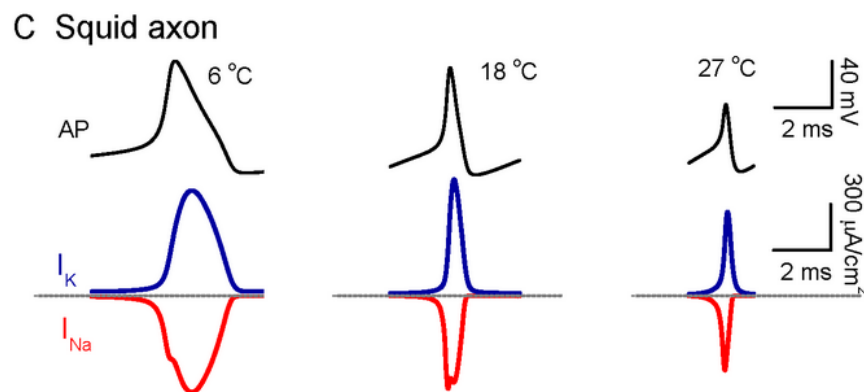


Figure 30: Action Potential: temperature effect [23]

3.5 Current Injection

It is beneficial to investigate how current injection impacts the change in voltage in the excitable cell. In this experiment electrodes are used to introduce an electrical current into the excitable cell. A voltmeter is used for measuring the voltage potential of

the cell. Another way to study current injection is to use the Hodgkin and Huxley model. The model can be modified to introduce an external current. Figure 19 shows the modified Hodgkin and Huxley model which can be used for the current injection experiment.

Figure 31 illustrates the results of injecting different electrical currents into the Hodgkin and Huxley model. The y-axis represents voltage $v(t)$ in millivolts. The x-axis represents time in milliseconds. The simulation runs for 100ms. The first voltage strip demonstrates how the membrane potential changes when $12\mu\text{A}$ current is injected into the model. This level of injected current results in consistent generation of action potentials. In the second experiment the current was reduced from $12\mu\text{A}$ to $8\mu\text{A}$. The second strip shows the results for the second experiment. In this experiment an action potential is fired consistently but at the lower rate. In other words, there are 8 action potential complexes initiated during the $12\mu\text{A}$ current injection, while there are only 7 action potential complexes originated with the 8nA current injection.

The third voltage strip of Figure 31 demonstrates the voltage change with the $4\mu\text{A}$ current injection. At this level of current injection, a single action potential complex is generated. The fourth voltage strip of Figure 31 is a reference experiment. There is no current ($0\mu\text{A}$) injected during the reference experiment. There is no action potential generated during the reference experiment. Therefore, the fourth voltage strip of Figure 31 contains a flat line. The fifth voltage strip of Figure 31 represents a $-5\mu\text{A}$ current injection and just like in the reference experiment, there is no action potential generated.

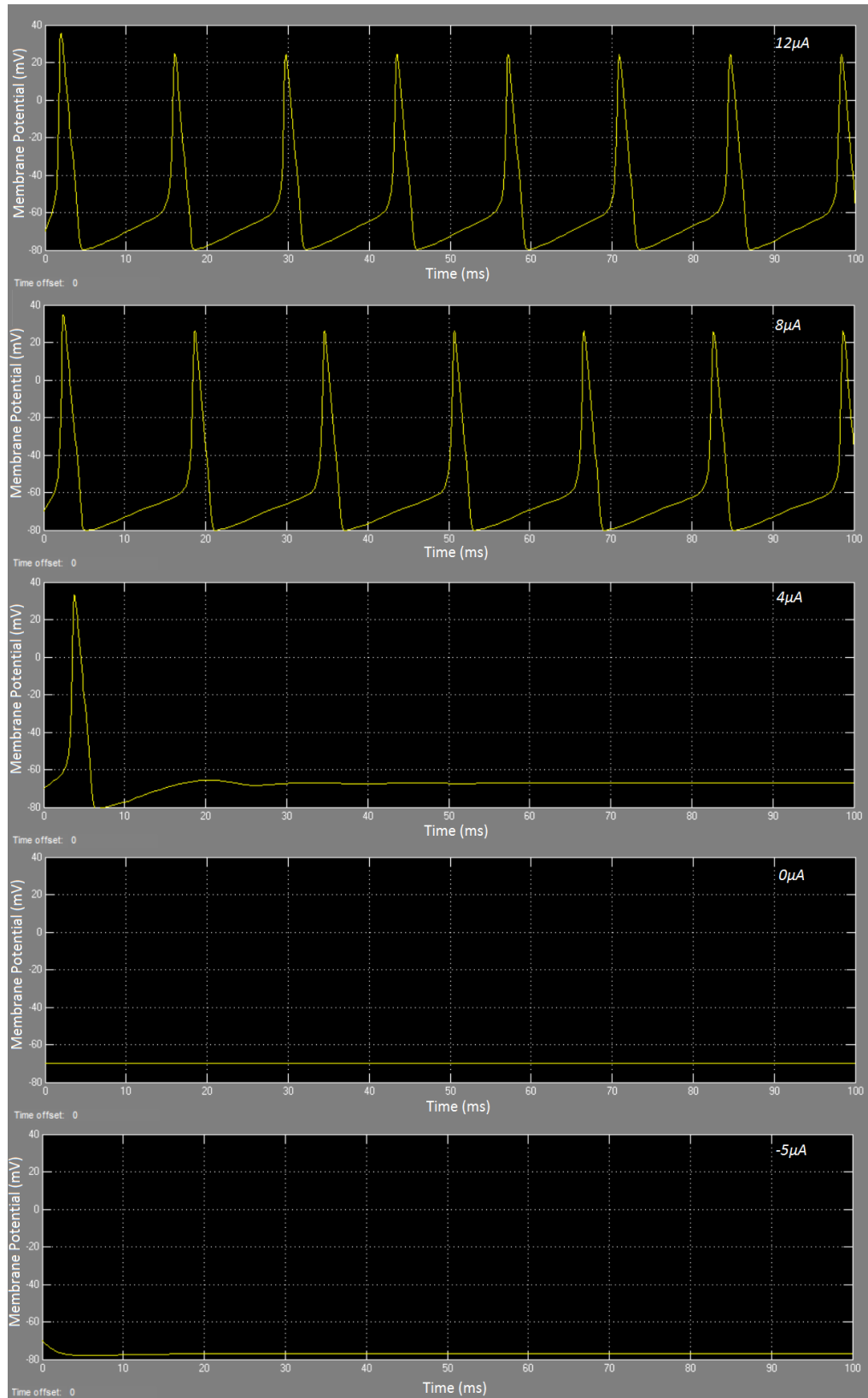


Figure 31: Action potential during current injection

The current injection can be split into three distinct phases: equilibrium stage, single spike stage and limit cycle stage. The single spike stage is a stage where only a single action potential complex is generated. This stage continues until we get to a point where the action potential fires consistently. At this level of current we have reached a minimum firing rate. So what does the minimum firing rate mean? The action potential can either fire or not. The action potential cannot fire partially or fire with low amplitude. Due to the “all or nothing” principle, there is no visible increase in the action potential amplitude. Instead there is a sudden voltage jump which is consistent with an action potential complex. In the example below this level is reached at $8\mu\text{A}$. In other words, action potentials fires continuously at $8\mu\text{A}$ or higher.

An increase in injected current (I) will lead to an increase in firing rate in a neuron. Figure 31 shows that the number of action potential complexes increase from 0 to 8 when the injected current increases from $0\mu\text{A}$ to $12\mu\text{A}$.

These simulation results are consistent with the results described in “Current Clamp and Modeling Studies of Low-Threshold Calcium Spikes in Cells of the Cat’s Lateral Geniculate Nucleus” [24]. Figure 32 illustrates the relationship between current injection (400-ms pulses) and the evoked action potential responses.

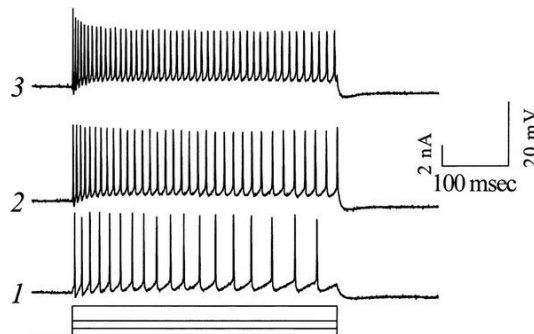


Figure 32: Current Injection [24]

3.6 Threshold

Threshold level defines the minimum level that a membrane potential has to reach in order for an action potential to fire. If this level is not reached, the potential will return to the resting voltage and an action potential will not fire.

Threshold level depends greatly on resting membrane potential. In all of the simulations discussed in this work, the resting membrane potential was set to -70mV . In other words, -70mV is the initial value used for the membrane potential in all simulations discussed in this dissertation. Threshold level also depends on experimental temperature. During most of the experiments discussed in this work, the temperature level was set to 6.3C° .

In order to identify the threshold level, a set of experiments should be performed where different levels of current (I) are injected into the excitable cell. In this work the current is injected in the Simulink model. In order to get the most accurate estimates for the threshold level, the current (I) is increased gradually until the action potential initiates.

Let us consider an experiment below. In order to identify the threshold level for the model described in this work, different levels of current have been injected into the model. The results of this experiment are depicted in Figure 33. The y-axis scale of all experiments runs from -100mV to 40mV . Each experiment runs for 15ms . For the baseline experiment there was no current injection. Therefore, the action potential was not initiated. This can be seen on the first image of Figure 33. In order to determine the threshold level, an external current of $1.5\mu\text{A}$ was injected into the model. As a result of this stimulation, the potential increased and reached -67.5mV . This simulation is represented in the second image of Figure 33. After an external stimulus is applied,

potential starts to increase until -67.5mV is reached. This increase was due to the external current applied to the excitable cell. After reaching -67.5mV , the potential starts to decline and eventually returns to the resting voltage. The applied current in this experiment was not high enough to produce an action potential. In other words, the potential did not reach the threshold level. Therefore, the external stimulus should be increased even further.

In the next simulation (shown in the third image in Figure 33) the current (I) is increased to $2.2567\mu\text{A}$. As in the previous example, this level of current ($I = 2.2567\mu\text{A}$) is not sufficient to initiate an action potential. In other words, during this simulation the membrane potential does not reach the threshold level and returns to its resting state (-70mV). An external stimulus is applied and the potential starts to increase from -70mV to -59.178mV . After reaching its highest level of -59.178mV , the potential starts to decay and eventually returns to its resting level. Comparing results with the previous simulation we notice that the voltage potential reaches a higher level (-59.178mV) when a higher current ($I = 2.2567\mu\text{A}$) is applied. In other words, a higher current depolarizes cell more and therefore the membrane potential can reach a higher level. Reaching potential of -59.178mV does not result in an action potential. Therefore, the external stimulus has to be increased.

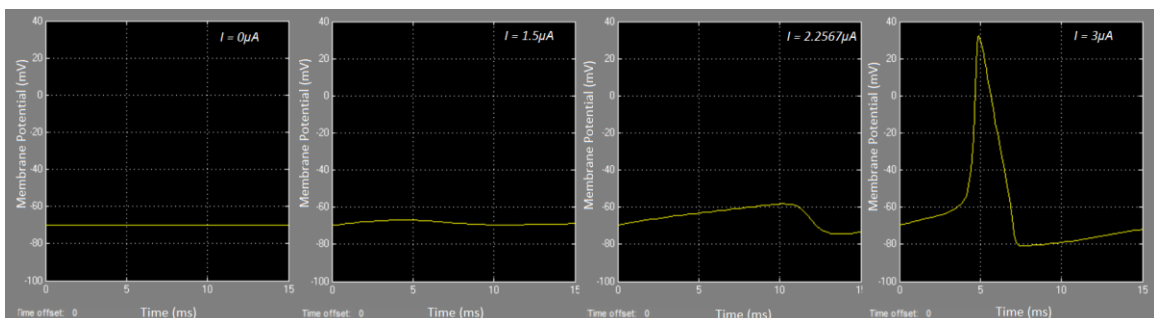


Figure 33: Threshold

In the last simulation the current (I) was increased to $3\mu\text{A}$. The results of this experimental run are shown in the last image of Figure 33. An external stimulus, applied to the model, initiates an action potential complex. Therefore, we can conclude that the threshold level was reached by applying an external stimulus of $3\mu\text{A}$ during a 6.3C° experiment. By examining the test results depicted in the last graph in Figure 33, we can see that the action potential starts its initiation process at approximately -60mV . Voltage gated sodium channels start to open when the membrane potential reaches -59.178mV . This results in a rapid inflow of sodium ions into the cell. On the graph this process is shown as a rapid raise in the action potential curve. Therefore, -59.178mV is the threshold level for a giant squid axon model which is simulated at 6.3C° .

Threshold simulation results are consistent with the theoretical data described in “Nerve Cells and Electric Signaling” [25]. Figure 34 illustrates the changes in the membrane potential in response to a different stimulus strength.

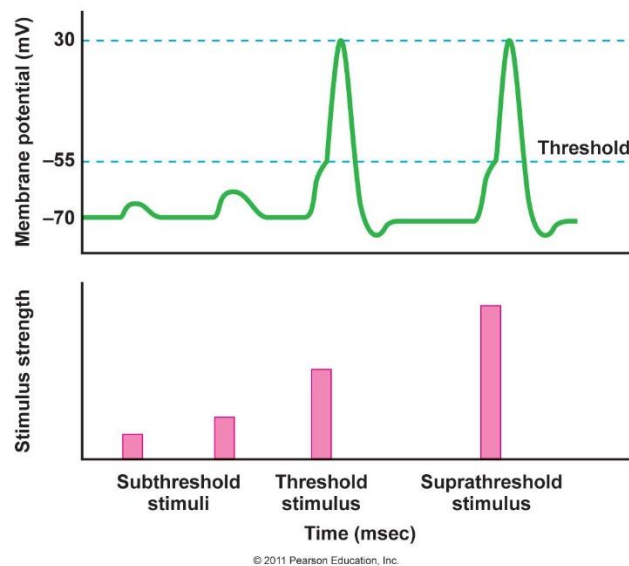


Figure 34: Threshold level [25]

Chapter 4

4.1 Future Work

Without any doubts the Hodgkin and Huxley model is a quantitatively useful model for neuromodulation studies. This model can be extended and improved even further.

Future enhancements to the model may include the following work:

- Modeling and analyzing the function of active pumps, such as the sodium and potassium pumps, that are actively involved in maintaining the resting membrane potential.
- Utilizing the Hodgkin and Huxley model in order to study different types of stimulation techniques and their effects on action potential initiation and propagation.
- Extending the Hodgkin and Huxley model to create a closed loop system which would dynamically optimize the stimulation signal depending on an output from the Hodgkin and Huxley model.

The current model lacks the implementation of active pumps, such as Na^+K^+ pumps. Excitable cell membranes contain multiple ion channel types. One such ion channel is the active pump which is responsible for maintaining a resting membrane potential. It may be beneficial to investigate the significance of active pumps and determine their role in an action potential initiation and propagation. This investigation will involve performing voltage clamp simulations on a giant squid axon model and finding the mathematical relationships between voltage, current and conductance.

A giant squid axon can be used for this simulation. In order to block voltage gated sodium channels, tetrodotoxin (TTX) can be introduced into a giant squid axon.

Tetraethylammonium (TEA) can be used to block voltage gated potassium channels.

Tetrodotoxin is an alkaloid neurotoxin that is commonly found in puffer fish and several species of frogs. This molecule has the ability to selectively block voltage gated sodium channels which will prevent sodium ions from crossing the cell membrane from the extracellular environment to the intracellular environment and vice versa [26]. TTX is often used in experiments where the sodium ion concentration is kept constant.

Tetraethylammonium is a quaternary ammonium compound. It has the ability to selectively block voltage gated potassium channels which will prevent potassium ions from crossing from the intracellular environment to the extracellular environment and vice versa [27]. TEA is often used in experiments where the potassium concentration is kept constant.

By deactivating both voltage gated potassium and voltage gated sodium channels it may be possible to investigate and model the impact of active pumps. This study will provide a better understanding of the role of active pumps in the action potential initiation process.

The Simulink model, described in the work, can be improved by introducing a custom signal builder, which can be used to study different types of stimulation techniques and their effects on an action potential initiation and propagation. In today's world of implantable devices it is very important to optimize and maximize a battery life. Prolongation of the battery life would require less battery exchanges and therefore less surgical procedures.

In order to achieve this goal, the battery consumption needs to be minimized. By introducing a custom signal builder we can minimize the level of energy used for cell

stimulation. This can be done by using biphasic stimulation, monophasic stimulation, burst stimulation and combinations of these stimulations.

Most neuromodulation devices are blindly delivering the therapy to patients. These therapy devices do not receive any kind of feedback notifying the device if the therapy was delivered successfully or not. These types of therapies are based on previously programmed values that can only be changed by a human interaction. Ideally, neuromodulation devices would operate with a minimum of human intervention. Apart from inconvenience, there are many risks involved in giving patients the ability to program their own devices. It is not uncommon for people to make mistakes while programming their own devices. Another risk factor is a security. No matter how well the security system is designed, there will always be people trying to find ways to penetrate that security system. This may result in potential risk for a patient. By modifying the programmed therapy, an intruder may unwillingly or willingly create a life threatening situation for a patient. Therefore, it might be worth building a closed loop system that will be able to analyze the result of a delivered therapy and adjust the output accordingly. This study can be achieved by extending the Simulink model and implementing the input side of the circuit. By implementing a close loop system we can eliminate the need for a patient to interfere with a neuromodulation device.

4.2 Conclusion

Neural communication involves a transfer of information from the brain to different parts of the body (e.g. arms, legs, vasculature, etc.) and vice versa. This communication is achieved by means of neurons that process and transmit information through electrical and chemical signals. Neural communication has a number of important functions such as sensory communication, motor communication, etc.

Neural communication is necessary for a being to functioning properly. All neural communications are achieved by means of action potentials. The action potential is an event during which an electrical impulse travels from one location in the body to another. By stimulating an excitable cell, the cell membrane goes through a sequence of events such as depolarization from its resting state followed by repolarization back to the resting state. As a result, the polarity of excitable cell is reversed for a brief period of time (from negative to positive) and then returns back to its initial state by reestablishing the rest potential.

One of the most important contributions to neuroscience has been made by Alan Hodgkin and Andrew Huxley. They used a giant squid axon to derive a mathematical model which analyzed and explained the ionic mechanisms underlying the initiation and propagation of action potentials. In 1963, Hodgkin and Huxley received a Nobel Prize in Physiology. The Hodgkin and Huxley model is a mathematical model that describes how the action potential is initiated and propagated in a neuron. It is a set of nonlinear ordinary differential equations that approximates the electrical characteristics of excitable cells.

This model simulates an action potential origination and propagation in an axon. It also provides the ability to investigate different properties of a neuron such as diameter and operating temperature and their impact on the action potential.

There have been many improvements introduced to the Hodgkin and Huxley model. A lot of new and more complex models have been developed over the years. The Hodgkin and Huxley model was the foundation for most of these models.

With the aid of advanced modeling tools such as MATLAB Simulink a lot of useful and complex simulations can be performed. These tools provide a practical way to model different properties of a neuron and to analyze their impact on the electrical signal.

References

- [1] M. R. James Neitzel, "Cell Communication," [Online]. Available: <http://www.nature.com/scitable/topic/cell-communication-14122659>. [Accessed 27 10 2013].
- [2] R. Plonsey and R. C. Barr, "Bioelectricity - A Quantative Approach," p. 50, 2007.
- [3] Boundless, "Nerve Impulse Transmission within a Neuron," Boundless, [Online]. Available: <https://www.boundless.com/biology/the-nervous-system/how-neurons-communicate/nerve-impulse-transmission-within-a-neuron-resting-potential/>. [Accessed 07 04 2014].
- [4] J. Malmivuo and R. Plonsey, Bioelectromagnetism, New York; Oxford: OXFORD UNIVERSITY PRESS, 1995.
- [5] M. D, Y. A and R. S, "Exploring Biological Neuron Models," [Online]. Available: <http://www.iitk.ac.in/directions/feb2006/PRINT~KALRA.pdf>. [Accessed 21 11 2013].
- [6] K. Abbas, Robbin's Basic Pathology, Elsevier Health Sciences, 2007.
- [7] N. R. Carlson, Foundations of Physiological Psychology, Allyn & Bacon, 2004, pp. 53-55.
- [8] S. Muskopf, "Chapter 5 - Membrane Structure and Function," [Online]. Available: http://www.biologycorner.com/APbiology/cellular/notes_cell_membrane.html. [Accessed 9 March 2013].
- [9] C. Biology, "How Neurons Communicate," 10 May 2013. [Online]. Available: <http://cnx.org/content/m44748/latest/>.
- [10] W. A. Catterall, "Voltage-Gated Calcium Channels," 11 07 2011. [Online]. Available: <http://intl-cshperspectives.cshlp.org/content/3/8/a003947.full>. [Accessed 07 04 2014].
- [11] C. G. Sobey, "Potassium Channel Function in Vascular Disease," *Arteriosclerosis, Thrombosis, and Vascular Biology.*, no. 10, pp. 28-39, 2000.
- [12] N. Media, "The Nobel Prize in Chemistry 1997," 2013. [Online]. Available: http://www.nobelprize.org/nobel_prizes/chemistry/laureates/1997/illpres/skou.html.
- [13] C. A. a. Physiology, "The Action Potential," 28 June 2013. [Online]. Available:

<http://cnx.org/content/m46526/latest/>. [Accessed 29 Sep 2013].

- [14] UIC.edu, "The Body's Communication Systems: The Endocrine and Nervous Systems," [Online]. Available: <http://www.uic.edu/classes/bios/bios100/lectures/nervous.htm>. [Accessed 11 March 2013].
- [15] S. Alford, "The Hodgkin-Huxley equations." *Cell and Molecular Neurobiology*.
- [16] J. H. Byrne, "Chapter 3: Propagation of the Action Potential," 1997. [Online]. Available: <http://neuroscience.uth.tmc.edu/s1/chapter03.html>.
- [17] "Peripheral Axons: Characteristics," 10 12 2013. [Online]. Available: <http://neuromuscular.wustl.edu/nother/axon.htm>.
- [18] CUNeurons, "Section 2 - Myelination & Diameter," CUNeurons, [Online]. Available: <https://sites.google.com/site/cuneurons/section-2>. [Accessed 15 1 2014].
- [19] S. Sircar, *Principles of Medical Physiology*, New York, NY: Thieme, 2008, p. 91.
- [20] L. Squire, *Fundamental Neuroscience*, Oxford: Elsevier Inc, 2013, p. 62.
- [21] B. G. A. G. a. D. W. David Sterratt, "Principles of Computational Modeling in Neuroscience," 2011. [Online]. Available: <http://www.compneuroprinciples.org/code-examples/all/all>. [Accessed 11 06 2014].
- [22] F. Bezanilla, "Ionic currents during the nerve impulse," *Nature Reviews: Molecular Cell Biology*, 01 04 2008. [Online]. Available: http://www.nature.com/nrm/journal/v9/n4/box/nrm2376_BX2.html. [Accessed 28 03 2014].
- [23] H. A. M. D. Yu Y, "Warm Body Temperature Facilitates Energy Efficient Cortical Action Potentials," *PLoS Comput Biol*, vol. 8, no. 4, 2012.
- [24] C. L. C. ., J. R. ., S. M. S. X. J. Zhan, "Current Clamp and Modeling Studies of Low-Threshold Calcium Spikes in Cells of the Cat's Lateral Geniculate Nucleus," *Journal of Neurophysiology*, vol. 81, pp. 2360-2373, 1999.
- [25] M. J. College, "Chapter 7 - Nerve Cells and Electric Signaling," Modesto Junior College: MJC, [Online]. Available: http://droualb.faculty.mjc.edu/Course%20Materials/Physiology%20101/Chapter%20Notes/Fall%202011/chapter_7%20Fall%202011.htm. [Accessed 12 06 2014].
- [26] NCBI, "Tetrodotoxin," [Online]. Available:

<http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=20382>. [Accessed 29 1 2014].

- [27] NCBI, "Tetraethylammonium," [Online]. Available: <http://pubchem.ncbi.nlm.nih.gov/summary/summary.cgi?cid=5413>. [Accessed 29 1 2014].

Appendix

Appendix A: Simulink Constants and Variables

```
clear;

% *****
% CONSTANTS
% *****

V_initial=0;           % Initial voltage (mV)
                        % used for Anode break

V_resting=-70;        % Resting membrane potential (mV)

E_Na=115;             % Vr - VNa (mV)
E_K=-12;              % Vr - VK (mV)
E_leak=10.613;       % Vr - VL (mV)

Cm=1;                 % membrane capacitance per unit area ( $\mu\text{F}/\text{cm}^2$ )

G_leak=0.3;           % leakage conductance per unit area ( $\text{mS}/\text{cm}^2$ )
G_Na=120;             % sodium conductance per unit area ( $\text{mS}/\text{cm}^2$ )
G_K=36;               % potassium conductance per unit area ( $\text{mS}/\text{cm}^2$ )

% *****
% VARIABLES
% *****

n_initial=0.3176;     % Initial n gate variable
```

```
m_initial=0.0529;      % Initial m gate variable
h_initial=0.5919;     % Initial h gate variable

% TEMPERATURE
Temperature_Bit=1;    % To enable the analysis of the temperature
effect set bit to 1

                        % and to disable the effect of the
temperature set

                        % the bit to any value not equal to 1
Temperature=6.3;      % Temperature in Celsius
```

Appendix B: Simulink Model Source Code

```
Model {
  Name "MyModel"
  Version 7.7
  MdlSubVersion 0
  GraphicalInterface {
    NumRootInports 0
    NumRootOutports 0
    ParameterArgumentNames ""
    ComputedModelVersion "1.258"
    NumModelReferences 0
    NumTestPointedSignals 0
  }
  SavedCharacterEncoding "windows-1252"
  slprops.hdlmdlprops {
    $PropName "HDLParams"
    $ObjectID 1
    Array {
      Type "Cell"
      Dimension 2
      Cell "HDLSubsystem"
      Cell "MyModel"
      PropName "mdlProps"
    }
  }
  SaveDefaultBlockParams on
  ScopeRefreshTime 0.035000
  OverrideScopeRefreshTime on
  DisableAllScopes off
  DataTypeOverride "UseLocalSettings"
  DataTypeOverrideAppliesTo "AllNumericTypes"
  MinMaxOverflowLogging "UseLocalSettings"
  MinMaxOverflowArchiveMode "Overwrite"
  MaxMDLFileLineLength 120
  UserBdParams
  "PhysicalModelingChecksum;PhysicalModelingParameterChecksum;PhysicalMod
  elingProducts"
  PhysicalModelingChecksum "2725184017"
  PhysicalModelingParameterChecksum "1666047934"
  PhysicalModelingProducts "Simscape"
  Created "Mon Jan 14 21:36:21 2013"
  Creator "Vladimir"
  UpdateHistory "UpdateHistoryNever"
  ModifiedByFormat "%<Auto>"
  LastModifiedBy "ruzovv01"
  ModifiedDateFormat "%<Auto>"
  LastModifiedDate "Fri May 09 15:13:27 2014"
  RTWModifiedDateStamp 321543468
  ModelVersionFormat "1. %<AutoIncrement:258>"
  ConfigurationManager "None"
  SampleTimeColors off
  SampleTimeAnnotations off
  LibraryLinkDisplay "none"
  WideLines off
}
```



```

ShowLineDimensions      off
ShowPortDataTypes      off
ShowLoopsOnError       on
IgnoreBidirectionalLines off
ShowStorageClass       off
ShowTestPointIcons     on
ShowSignalResolutionIcons on
ShowViewerIcons       on
SortedOrder            off
ExecutionContextIcon   off
ShowLinearizationAnnotations on
BlockNameDataTip      off
BlockParametersDataTip off
BlockDescriptionStringDataTip off
ToolBar               on
StatusBar             on
BrowserShowLibraryLinks on
BrowserLookUnderMasks on
SimulationMode        "normal"
LinearizationMsg      "none"
Profile               off
ParamWorkspaceSource  "MATLABWorkspace"
AccelSystemTargetFile "accel.tlc"
AccelTemplateMakefile "accel_default_tmf"
AccelMakeCommand      "make_rtw"
TryForcingSFcnDF     off
Object {
    $PropName          "DataLoggingOverride"
    $ObjectID          2
    $ClassName          "Simulink.SimulationData.ModelLoggingInfo"
    model_             "simulink1"
    overrideMode_     [0.0]
    Array {
        Type           "Cell"
        Dimension      1
        Cell            "simulink1"
        PropName        "logAsSpecifiedByModels_"
    }
    Array {
        Type           "Cell"
        Dimension      1
        Cell            []
        PropName        "logAsSpecifiedByModelsSSIDs_"
    }
}
RecordCoverage        off
CovPath               "/"
CovSaveName           "covdata"
CovMetricSettings     "dw"
CovNameIncrementing   off
CovHtmlReporting      on
CovForceBlockReductionOff on
covSaveCumulativeToWorkspaceVar on
CovSaveSingleToWorkspaceVar on
CovCumulativeVarName  "covCumulativeData"
CovCumulativeReport   off
CovReportOnPause      on

```

```

CovModelRefEnable      "Off"
CovExternaleMLEnable  off
ExtModeBatchMode       off
ExtModeEnableFloating on
ExtModeTrigType        "manual"
ExtModeTrigMode        "normal"
ExtModeTrigPort        "1"
ExtModeTrigElement     "any"
ExtModeTrigDuration    1000
ExtModeTrigDurationFloating "auto"
ExtModeTrigHoldOff    0
ExtModeTrigDelay       0
ExtModeTrigDirection  "rising"
ExtModeTrigLevel       0
ExtModeArchiveMode     "off"
ExtModeAutoIncOneShot  off
ExtModeIncDirWhenArm   off
ExtModeAddSuffixToVar  off
ExtModeWriteAllDataToWs off
ExtModeArmWhenConnect  on
ExtModeSkipDownloadWhenConnect off
ExtModeLogAll          on
ExtModeAutoUpdateStatusClock on
BufferReuse            off
ShowModelReferenceBlockVersion off
ShowModelReferenceBlockIO off
Array {
  Type          "Handle"
  Dimension      1
  Simulink.ConfigSet {
    $ObjectID    3
    Version      "1.11.0"
    Array {
      Type          "Handle"
      Dimension     10
      Simulink.SolverCC {
        $ObjectID    4
        Version      "1.11.0"
        StartTime    "0.0"
        StopTime     "100"
        AbsTol       "1e-6"
        FixedStep    "auto"
        InitialStep  "auto"
        MaxNumMinSteps "-1"
        MaxOrder     5
        ZcThreshold  "auto"
        ConsecutiveZCsStepRelTol "10*128*eps"
        MaxConsecutiveZCs "1000"
        ExtrapolationOrder 4
        NumberNewtonIterations 1
        MaxStep        "auto"
        MinStep        "auto"
        MaxConsecutiveMinStep "1"
        RelTol         "1e-3"
        SolverMode     "Auto"
        ConcurrentTasks off
        Solver         "ode45"
      }
    }
  }
}

```

```

SolverName                "ode45"
SolverJacobianMethodControl "auto"
ShapePreserveControl      "DisableAll"
ZeroCrossControl          "UseLocalSettings"
ZeroCrossAlgorithm        "Nonadaptive"
AlgebraicLoopSolver       "TrustRegion"
SolverResetMethod         "Fast"
PositivePriorityOrder      off
AutoInsertRateTranBlk    off
SampleTimeConstraint      "Unconstrained"
InsertRTBMode             "Whenever possible"
}
Simulink.DataIOCC {
  $ObjectID                5
  Version                  "1.11.0"
  Decimation               "1"
  ExternalInput            "[t, u]"
  FinalStateName          "xFinal"
  InitialState             "xInitial"
  LimitDataPoints         on
  MaxDataPoints            "1000"
  LoadExternalInput       off
  LoadInitialState        off
  SaveFinalState          off
  SaveCompleteFinalSimState off
  SaveFormat               "Array"
  SignalLoggingSaveFormat "ModelDataLogs"
  SaveOutput              on
  SaveState                off
  SignalLogging           on
  DSMLogging              on
  InspectSignalLogs      off
  SaveTime                on
  ReturnWorkspaceOutputs off
  StateSaveName           "xout"
  TimeSaveName            "tout"
  OutputSaveName          "yout"
  SignalLoggingName       "logstdout"
  DSMLoggingName          "dsmout"
  OutputOption            "RefineOutputTimes"
  OutputTimes             "[]"
  ReturnWorkspaceOutputsName "out"
  Refine                  "1"
}
Simulink.OptimizationCC {
  $ObjectID                6
  Version                  "1.11.0"
  Array {
    Type                   "Cell"
    Dimension              8
    Cell                   "BooleansAsBitfields"
    Cell                   "PassReuseOutputArgsAs"
    Cell                   "PassReuseOutputArgsThreshold"
    Cell                   "ZeroExternalMemoryAtStartup"
    Cell                   "ZeroInternalMemoryAtStartup"
    Cell                   "OptimizeModelRefInitCode"
    Cell                   "NoFixptDivByZeroProtection"
  }
}

```

```

    Cell                "UseSpecifiedMinMax"
    PropName            "DisabledProps"
}
BlockReduction        off
BooleanDataType        on
ConditionallyExecuteInputs on
InlineParams          on
UseIntDivNetSlope      off
UseSpecifiedMinMax     off
InlineInvariantSignals off
OptimizeBlockIOStorage on
BufferReuse           off
EnhancedBackFolding    off
StrengthReduction      off
ExpressionFolding      on
BooleansAsBitfields    off
BitfieldContainerType  "uint_T"
EnableMemcpy          on
MemcpyThreshold        64
PassReuseOutputArgsAs "Structure reference"
ExpressionDepthLimit   2147483647
FoldNonRolledExpr      on
LocalBlockOutputs      on
RollThreshold          5
SystemCodeInlineAuto   off
StateBitsets           off
DataBitsets            off
UseTempVars           off
ZeroExternalMemoryAtStartup on
ZeroInternalMemoryAtStartup on
InitFltsAndDblsToZero off
NoFixptDivByZeroProtection off
EfficientFloat2IntCast off
EfficientMapNaN2IntZero on
OptimizeModelRefInitCode off
LifeSpan              "inf"
MaxStackSize          "Inherit from target"
BufferReusableBoundary on
SimCompilerOptimization "Off"
AccelVerboseBuild     off
AccelParallelForEachSubsystem on
}
Simulink.DebuggingCC {
    $ObjectID          7
    Version            "1.11.0"
    RTPrefix           "error"
    ConsistencyChecking "none"
    ArrayBoundsChecking "none"
    SignalInfNanChecking "none"
    SignalRangeChecking "none"
    ReadBeforeWriteMsg "UseLocalSettings"
    WriteAfterWriteMsg "UseLocalSettings"
    WriteAfterReadMsg "UseLocalSettings"
    AlgebraicLoopMsg "warning"
    ArtificialAlgebraicLoopMsg "warning"
    SaveWithDisabledLinksMsg "warning"
    SaveWithParameterizedLinksMsg "warning"
}

```

```

CheckSSInitialOutputMsg on
UnderspecifiedInitializationDetection "Classic"
MergeDetectMultiDrivingBlocksExec "none"
CheckExecutionContextPreStartOutputMsg off
CheckExecutionContextRuntimeOutputMsg off
SignalResolutionControl "UseLocalSettings"
BlockPriorityViolationMsg "warning"
MinStepSizeMsg "warning"
TimeAdjustmentMsg "none"
MaxConsecutiveZCsMsg "error"
MaskedZcDiagnostic "warning"
IgnoredZcDiagnostic "warning"
SolverPrmCheckMsg "none"
InheritedTsInSrcMsg "warning"
DiscreteInheritContinuousMsg "warning"
MultiTaskDSMMsg "error"
MultiTaskCondExecSysMsg "error"
MultiTaskRateTransMsg "error"
SingleTaskRateTransMsg "none"
TasksWithSamePriorityMsg "warning"
SigSpecEnsureSampleTimeMsg "warning"
CheckMatrixSingularityMsg "none"
IntegerOverflowMsg "warning"
Int32ToFloatConvMsg "warning"
ParameterDowncastMsg "error"
ParameterOverflowMsg "error"
ParameterUnderflowMsg "none"
ParameterPrecisionLossMsg "warning"
ParameterTunabilityLossMsg "warning"
FixptConstUnderflowMsg "none"
FixptConstOverflowMsg "none"
FixptConstPrecisionLossMsg "none"
UnderSpecifiedDataTypeMsg "none"
UnnecessaryDatatypeConvMsg "none"
VectorMatrixConversionMsg "none"
InvalidFcnCallConnMsg "error"
FcnCallInpInsideContextMsg "Use local settings"
SignalLabelMismatchMsg "none"
UnconnectedInputMsg "warning"
UnconnectedOutputMsg "warning"
UnconnectedLineMsg "warning"
SFcnCompatibilityMsg "none"
UniqueDataStoreMsg "none"
BusObjectLabelMismatch "warning"
RootOutputRequireBusObject "warning"
AssertControl "UseLocalSettings"
EnableOverflowDetection off
ModelReferenceIOMsg "none"
ModelReferenceMultiInstanceNormalModeStructChecksumCheck
"error"
ModelReferenceVersionMismatchMessage "none"
ModelReferenceIOMismatchMessage "none"
ModelReferenceCSMismatchMessage "none"
UnknownTsInhSupMsg "warning"
ModelReferenceDataLoggingMessage "warning"
ModelReferenceSymbolNameMessage "warning"
ModelReferenceExtraNoncontSigs "error"

```

```

StateNameClashWarn          "warning"
SimStateInterfaceChecksumMismatchMsg "warning"
SimStateOlderReleaseMsg     "error"
InitInArrayFormatMsg        "warning"
StrictBusMsg                 "ErrorLevel1"
BusNameAdapt                 "WarnAndRepair"
NonBusSignalsTreatedAsBus   "none"
LoggingUnavailableSignals   "error"
BlockIODiagnostic            "none"
SFUnusedDataAndEventsDiag   "warning"
SFUnexpectedBacktrackingDiag "warning"
SFInvalidInputDataAccessInChartInitDiag "warning"
SFNoUnconditionalDefaultTransitionDiag "warning"
SFTransitionOutsideNaturalParentDiag "warning"
}
Simulink.HardwareCC {
  $ObjectID          8
  Version             "1.11.0"
  ProdBitPerChar     8
  ProdBitPerShort    16
  ProdBitPerInt      32
  ProdBitPerLong     32
  ProdBitPerFloat    32
  ProdBitPerDouble   64
  ProdBitPerPointer  32
  ProdLargestAtomicInteger "Char"
  ProdLargestAtomicFloat  "None"
  ProdIntDivRoundTo     "Undefined"
  ProdEndianess         "Unspecified"
  ProdWordSize          32
  ProdShiftRightIntArith on
  ProdHWDeviceType      "32-bit Generic"
  TargetBitPerChar     8
  TargetBitPerShort    16
  TargetBitPerInt      32
  TargetBitPerLong     32
  TargetBitPerFloat    32
  TargetBitPerDouble   64
  TargetBitPerPointer  32
  TargetLargestAtomicInteger "Char"
  TargetLargestAtomicFloat  "None"
  TargetShiftRightIntArith on
  TargetIntDivRoundTo     "Undefined"
  TargetEndianess         "Unspecified"
  TargetWordSize          32
  TargetTypeEmulationWarnSuppressLevel 0
  TargetPreprocMaxBitsSint 32
  TargetPreprocMaxBitsUint 32
  TargetHWDeviceType      "Specified"
  TargetUnknown           off
  ProdEqTarget            on
}
Simulink.ModelReferenceCC {
  $ObjectID          9
  Version             "1.11.0"
  UpdateModelReferenceTargets "IfOutOfDateOrStructuralChange"
  CheckModelReferenceTargetMessage "error"
}

```

```

EnableParallelModelReferenceBuilds off
ParallelModelReferenceErrorOnInvalidPool on
ParallelModelReferenceMATLABWorkerInit "None"
ModelReferenceNumInstancesAllowed "Multi"
PropagateVarSize "Infer from blocks in model"
ModelReferencePassRootInputsByReference on
ModelReferenceMinAlgLoopOccurrences off
PropagateSignalLabelsOutOfModel off
SupportModelReferenceSimTargetCustomCode off
}
Simulink.SFSimCC {
  $ObjectID 10
  Version "1.11.0"
  SFSimEnableDebug on
  SFSimOverflowDetection on
  SFSimEcho on
  SimBlas on
  SimCtrlC on
  SimExtrinsic on
  SimIntegrity on
  SimUseLocalCustomCode off
  SimParseCustomCode on
  SimBuildMode "sf_incremental_build"
}
Simulink.RTWCC {
  $BackupClass "Simulink.RTWCC"
  $ObjectID 11
  Version "1.11.0"
  Array {
    Type "Cell"
    Dimension 8
    Cell "IncludeHyperlinkInReport"
    Cell "GenerateTraceInfo"
    Cell "GenerateTraceReport"
    Cell "GenerateTraceReportSl"
    Cell "GenerateTraceReportSf"
    Cell "GenerateTraceReportEml"
    Cell "GenerateSLWebview"
    Cell "GenerateCodeMetricsReport"
    PropName "DisabledProps"
  }
  SystemTargetFile "grt.tlc"
  GenCodeOnly off
  MakeCommand "make_rtw"
  GenerateMakefile on
  TemplateMakefile "grt_default_tmf"
  GenerateReport off
  SaveLog off
  RTWVerbose on
  RetainRTWFile off
  ProfileTLC off
  TLCDebug off
  TLCCoverage off
  TLCAssert off
  ProcessScriptMode "Default"
  ConfigurationMode "Optimized"
  ConfigAtBuild off
}

```

```

RTWUseLocalCustomCode    off
RTWUseSimCustomCode     off
IncludeHyperlinkInReport off
LaunchReport             off
TargetLang               "C"
IncludeBusHierarchyInRTWFileBlockHierarchyMap off
IncludeERTFirstTime      off
GenerateTraceInfo        off
GenerateTraceReport      off
GenerateTraceReportSl    off
GenerateTraceReportSf    off
GenerateTraceReportEml   off
GenerateCodeInfo         off
GenerateSLWebview        off
GenerateCodeMetricsReport off
RTWCompilerOptimization "Off"
CheckMdlBeforeBuild      "Off"
CustomRebuildMode        "OnUpdate"
Array {
  Type           "Handle"
  Dimension      2
  Simulink.CodeAppCC {
    $ObjectID    12
    Version      "1.11.0"
    Array {
      Type       "Cell"
      Dimension  21
      Cell       "IgnoreCustomStorageClasses"
      Cell       "IgnoreTestpoints"
      Cell       "InsertBlockDesc"
      Cell       "InsertPolySpaceComments"
      Cell       "SFDataObjDesc"
      Cell       "MATLABFcnDesc"
      Cell       "SimulinkDataObjDesc"
      Cell       "DefineNamingRule"
      Cell       "SignalNamingRule"
      Cell       "ParamNamingRule"
      Cell       "InlinedPrmAccess"
      Cell       "CustomSymbolStr"
      Cell       "CustomSymbolStrGlobalVar"
      Cell       "CustomSymbolStrType"
      Cell       "CustomSymbolStrField"
      Cell       "CustomSymbolStrFcn"
      Cell       "CustomSymbolStrFcnArg"
      Cell       "CustomSymbolStrBlkIO"
      Cell       "CustomSymbolStrTmpVar"
      Cell       "CustomSymbolStrMacro"
      Cell       "ReqsInCode"
      PropName   "DisabledProps"
    }
    ForceParamTrailComments off
    GenerateComments         on
    IgnoreCustomStorageClasses on
    IgnoreTestpoints         off
    IncHierarchyInIds        off
    MaxIdLength              31
    PreserveName              off
  }
}

```



```

    PreserveNameWithParent    off
    ShowEliminatedStatement  off
    IncAutoGenComments       off
    SimulinkDataObjDesc      off
    SFDataObjDesc            off
    MATLABFcnDesc            off
    IncDataTypeInIds         off
    MangleLength              1
    CustomSymbolStrGlobalVar "$R$N$M"
    CustomSymbolStrType      "$N$R$M"
    CustomSymbolStrField     "$N$M"
    CustomSymbolStrFcn       "$R$N$M$F"
    CustomSymbolStrFcnArg    "rt$I$N$M"
    CustomSymbolStrBlkIO     "rtb_ $N$M"
    CustomSymbolStrTmpVar    "$N$M"
    CustomSymbolStrMacro     "$R$N$M"
    DefineNamingRule         "None"
    ParamNamingRule          "None"
    SignalNamingRule         "None"
    InsertBlockDesc          off
    InsertPolySpaceComments  off
    SimulinkBlockComments   on
    MATLABSourceComments     off
    EnableCustomComments     off
    InlinedPrmAccess         "Literals"
    ReqsInCode               off
    UseSimReservedNames     off
}
Simulink.GRTTargetCC {
    $BackupClass             "Simulink.TargetCC"
    $ObjectID                 13
    Version                   "1.11.0"
    Array {
        Type                   "Cell"
        Dimension               16
        Cell                    "GeneratePreprocessorConditionals"
        Cell                    "IncludeMdlTerminateFcn"
        Cell                    "CombineOutputUpdateFcns"
        Cell                    "SuppressErrorStatus"
        Cell                    "ERTCustomFileBanners"
        Cell                    "GenerateSampleERTMain"
        Cell                    "GenerateTestInterfaces"
        Cell
    }
    "ModelStepFunctionPrototypeControlCompliant"
    Cell                       "CPPClassGenCompliant"
    Cell                       "MultiInstanceERTCode"
    Cell                       "PurelyIntegerCode"
    Cell                       "SupportComplex"
    Cell                       "SupportAbsoluteTime"
    Cell                       "SupportContinuousTime"
    Cell                       "SupportNonInlinedSFcns"
    Cell                       "PortableWordSizes"
    PropName                   "DisabledProps"
    }
    TargetFcnLib                "ansi_tfl_table_tmw.mat"
    TargetLibSuffix             ""
    TargetPreCompLibLocation   ""
}

```

```

TargetFunctionLibrary    "ANSI_C"
UtilityFuncGeneration    "Auto"
ERTMultiwordTypeDef     "System defined"
FunctionExecutionProfile off
CodeExecutionProfiling  off
ERTCodeCoverageTool     "None"
ERTMultiwordLength      256
MultiwordLength         2048
GenerateFullHeader      on
GenerateSampleERTMain   off
GenerateTestInterfaces  off
IsPILTarget             off
ModelReferenceCompliant on
ParMdlRefBuildCompliant on
CompOptLevelCompliant  on
IncludeMdlTerminateFcn on
GeneratePreprocessorConditionals "Disable all"
CombineOutputUpdateFcns off
CombineSignalStateStructs    off
SuppressErrorStatus        off
ERTFirstTimeCompliant     off
IncludeFileDelimiter       "Auto"
ERTCustomFileBanners      off
SupportAbsoluteTime       on
LogVarNameModifier        "rt_"
MatFileLogging            on
MultiInstanceERTCode      off
SupportNonFinite          on
SupportComplex            on
PurelyIntegerCode        off
SupportContinuousTime     on
SupportNonInlinedSFcns   on
SupportVariableSizeSignals off
EnableShiftOperators      on
ParenthesesLevel         "Nominal"
PortableWordSizes        off
ModelStepFunctionPrototypeControlCompliant off
CPPClassGenCompliant     off
AutosarCompliant        off
UseMalloc                off
ExtMode                  off
ExtModeStaticAlloc      off
ExtModeTesting          off
ExtModeStaticAllocSize  1000000
ExtModeTransport        0
ExtModeMexFile          "ext_comm"
ExtModeIntrfLevel       "Level1"
RTWCAPISignals          off
RTWCAPIParams           off
RTWCAPIStates           off
RTWCAPIRootIO           off
GenerateASAP2           off
}
  PropName                "Components"
}
}
hdlcoderui.hdlcc {

```

```

$ObjectID          14
Version            "1.11.0"
Description        "HDL Coder custom configuration
component"
Name               "HDL Coder"
Array {
  Type             "Cell"
  Dimension        1
  Cell             ""
  PropName        "HDLConfigFile"
}
HDLCActiveTab     "0"
}
SSC.SimscapeCC {
$ObjectID          15
Version            "1.0"
Array {
  Type             "Cell"
  Dimension        1
  Cell             "Name"
  PropName        "DisabledProps"
}
Array {
  Type             "Handle"
  Dimension        1
  MECH.SimMechanicsCC {
    $ObjectID      16
    Version        "1.11.0"
    Name           "SimMechanics"
    WarnOnRedundantConstraints on
    WarnOnSingularInitialAssembly off
    ShowCutJoints off
    VisOnUpdateDiagram off
    VisDuringSimulation off
    EnableVisSimulationTime on
    VisSampleTime "0"
    DisableBodyVisControl off
    ShowCG        on
    ShowCS        on
    ShowOnlyPortCS off
    HighlightModel on
    FramesToBeSkipped "0"
    AnimationDelay "3"
    RecordAVI     off
    CompressAVI   on
    AutoFitVis    "off"
    EnableSelection on
    LastVizWinPosition "[-1 -1 -1 -1]"
    CamPosition     "[0 0 0]"
    CamTarget       "[0 0 -1]"
    CamUpVector     "[0 1 0]"
    CamHeight       "-1"
    CamViewAngle    "0"
    VisBackgroundColor "[0.9 0.9 0.95]"
    DefaultBodyColor "[1 0 0]"
    MDLBodyVisualizationType "Convex hull from body CS
locations"

```

```

        OVRRIDBodyVisualizationType "NONE"
    }
    PropName          "Components"
}
Name                "Simscape"
EditingMode         "Full"
ExplicitSolverDiagnosticOptions "warning"
InputDerivativeDiagnosticOptions "warning"
GlobalZcOffDiagnosticOptions "warning"
SimscapeLogType    "none"
SimscapeLogName    "simlog"
SimscapeLogDecimation 1
SimscapeLogLimitData on
SimscapeLogDataHistory 5000
}
PropName          "Components"
}
Name                "Configuration"
CurrentDlgPage     "Diagnostics"
ConfigPrmDlgPosition [ 400, 132, 1281, 864 ]
}
PropName          "ConfigurationSets"
}
Simulink.ConfigSet {
    $PropName          "ActiveConfigurationSet"
    $ObjectID          3
}
}
BlockDefaults {
    ForegroundColor    "black"
    BackgroundColor    "white"
    DropShadow         off
    NamePlacement      "normal"
    FontName           "Helvetica"
    FontSize           10
    FontWeight         "normal"
    FontAngle          "normal"
    ShowName           on
    BlockRotation      0
    BlockMirror        off
}
}
AnnotationDefaults {
    HorizontalAlignment "center"
    VerticalAlignment  "middle"
    ForegroundColor    "black"
    BackgroundColor    "white"
    DropShadow         off
    FontName           "Helvetica"
    FontSize           10
    FontWeight         "normal"
    FontAngle          "normal"
    UseDisplayTextAsClickCallback off
}
}
LineDefaults {
    FontName           "Helvetica"
    FontSize           9
    FontWeight         "normal"
    FontAngle          "normal"
}
}

```

```

}
BlockParameterDefaults {
  Block {
    BlockType          ActionPort
    InitializeStates   "held"
    PropagateVarSize   "Only when execution is resumed"
    ActionType         "unset"
  }
  Block {
    BlockType          BusCreator
    Inputs             "4"
    DisplayOption      "none"
    OutDataTypesStr    "Inherit: auto"
    NonVirtualBus      off
  }
  Block {
    BlockType          Constant
    Value              "1"
    VectorParams1D     on
    SamplingMode       "Sample based"
    OutMin             "[]"
    OutMax             "[]"
    OutDataTypesStr    "Inherit: Inherit from 'Constant value'"
    LockScale          off
    SampleTime         "inf"
    FramePeriod        "inf"
    PreserveConstantTs off
  }
  Block {
    BlockType          Demux
    Outputs            "4"
    DisplayOption      "none"
    BusSelectionMode   off
  }
  Block {
    BlockType          Fcn
    Expr               "sin(u[1])"
    SampleTime         "-1"
  }
  Block {
    BlockType          FromWorkspace
    VariableName       "simulink_input"
    SampleTime         "-1"
    Interpolate        on
    ZeroCross          off
    OutputAfterFinalValue "Extrapolation"
  }
  Block {
    BlockType          If
    NumInputs          "1"
    IfExpression       "u1 > 0"
    ShowElse           on
    ZeroCross          on
    SampleTime         "-1"
  }
  Block {
    BlockType          Inport

```

```

    Port                "1"
    OutputFunctionCall  off
    OutMin              "[]"
    OutMax              "[]"
    OutDataTypeStr     "Inherit: auto"
    LockScale          off
    BusOutputAsStruct  off
    PortDimensions     "-1"
    VarSizeSig         "Inherit"
    SampleTime         "-1"
    SignalType         "auto"
    SamplingMode       "auto"
    LatchByDelayingOutsideSignal off
    LatchInputForFeedbackSignals off
    Interpolate        on
}
Block {
    BlockType          Integrator
    ExternalReset      "none"
    InitialConditionSource "internal"
    InitialCondition   "0"
    LimitOutput        off
    UpperSaturationLimit "inf"
    LowerSaturationLimit "-inf"
    ShowSaturationPort off
    ShowStatePort      off
    AbsoluteTolerance  "auto"
    IgnoreLimit        off
    ZeroCross          on
    ContinuousStateAttributes ""
}
Block {
    BlockType          Merge
    Inputs             "2"
    InitialOutput      "[]"
    AllowUnequalInputPortWidths off
    InputPortOffsets  "[]"
}
Block {
    BlockType          Outport
    Port                "1"
    OutMin              "[]"
    OutMax              "[]"
    OutDataTypeStr     "Inherit: auto"
    LockScale          off
    BusOutputAsStruct  off
    PortDimensions     "-1"
    VarSizeSig         "Inherit"
    SampleTime         "-1"
    SignalType         "auto"
    SamplingMode       "auto"
    SourceOfInitialOutputValue "Dialog"
    OutputWhenDisabled "held"
    InitialOutput      "[]"
}
Block {
    BlockType          Product

```

```

Inputs                "2"
Multiplication         "Element-wise (.*)"
CollapseMode          "All dimensions"
CollapseDim           "1"
InputSameDT           on
OutMin                "[]"
OutMax                "[]"
OutDataTypeStr        "Inherit: Same as first input"
LockScale             off
RndMeth               "Zero"
SaturateOnIntegerOverflow on
SampleTime            "-1"
}
Block {
  BlockType            Scope
  ModelBased           off
  TickLabels           "OneTimeTick"
  ZoomMode             "on"
  Grid                 "on"
  TimeRange            "auto"
  YMin                 "-5"
  YMax                 "5"
  SaveToWorkspace      off
  SaveName              "ScopeData"
  LimitDataPoints      on
  MaxDataPoints        "5000"
  Decimation           "1"
  SampleInput          off
  SampleTime           "-1"
}
Block {
  BlockType            SubSystem
  ShowPortLabels       "FromPortIcon"
  Permissions           "ReadWrite"
  PermitHierarchicalResolution "All"
  TreatAsAtomicUnit    off
  CheckFcnCallInpInsideContextMsg off
  SystemSampleTime     "-1"
  RTWFCnNameOpts       "Auto"
  RTWFileNameOpts      "Auto"
  RTWMemSecFuncInitTerm "Inherit from model"
  RTWMemSecFuncExecute "Inherit from model"
  RTWMemSecDataConstants "Inherit from model"
  RTWMemSecDataInternal "Inherit from model"
  RTWMemSecDataParameters "Inherit from model"
  SimViewingDevice     off
  DataTypeOverride     "UseLocalSettings"
  DataTypeOverrideAppliesTo "AllNumericTypes"
  MinMaxOverflowLogging "UseLocalSettings"
  Variant               off
  GeneratePreprocessorConditionals off
}
Block {
  BlockType            Sum
  IconShape            "rectangular"
  Inputs               "++"
  CollapseMode         "All dimensions"

```

```

CollapseDim      "1"
InputSameDT      on
AccumDataTypeStr "Inherit: Inherit via internal rule"
OutMin           "[]"
OutMax           "[]"
OutDataTypeStr   "Inherit: Same as first input"
LockScale        off
RndMeth          "Floor"
SaturateOnIntegerOverflow on
SampleTime       "-1"
}
Block {
  BlockType      UnaryMinus
  SampleTime     "-1"
  SaturateOnIntegerOverflow off
}
}
System {
  Name           "MyModel"
  Location       [2124, 319, 3456, 1049]
  Open           on
  ModelBrowserVisibility on
  ModelBrowserWidth 200
  ScreenColor    "white"
  PaperOrientation "landscape"
  PaperPositionMode "auto"
  PaperType      "usletter"
  PaperUnits     "inches"
  TiledPaperMargins [0.500000, 0.500000, 0.500000, 0.500000]
  TiledPageScale 1
  ShowPageBoundaries off
  ZoomFactor      "100"
  ReportName      "simulink-default.rpt"
  SIDHighWatermark "1093"
  Block {
    BlockType      Scope
    Name           "Action Potential"
    SID            "70"
    Ports          [1]
    Position       [340, 44, 370, 76]
    Floating       off
    Location       [2180, 588, 3112, 921]
    Open           off
    NumInputPorts  "1"
    List {
      ListType      AxesTitles
      axes1         "%<SignalLabel>"
    }
    TimeRange      "150"
    YMin           "-5e+007"
    YMax           "7e+008"
    SaveName       "ScopeData1"
    DataFormat     "StructureWithTime"
    SampleTime     "0"
  }
}
Block {
  BlockType      Sum

```



```

Name          "Add"
SID           "75"
Ports        [4, 1]
Position      [230, 352, 260, 398]
BlockMirror   on
ShowName      off
Inputs        "----+"
InputSameDT   off
OutDataTypeStr "Inherit: Inherit via internal rule"
SaturateOnIntegerOverflow off
}
Block {
  BlockType      Sum
  Name           "Add1"
  SID            "1065"
  Ports         [2, 1]
  Position       [260, 43, 290, 72]
  NamePlacement  "alternate"
  ShowName       off
  InputSameDT    off
  OutDataTypeStr "Inherit: Inherit via internal rule"
  SaturateOnIntegerOverflow off
}
Block {
  BlockType      BusCreator
  Name           "Bus\nCreator"
  SID            "1078"
  Ports         [4, 1]
  Position       [390, 546, 395, 584]
  ShowName       off
  DisplayOption  "bar"
}
Block {
  BlockType      Constant
  Name           "Constant"
  SID            "1066"
  Position       [110, 39, 205, 61]
  BackgroundColor "[0.901961, 0.901961, 0.901961]"
  ShowName       off
  Value          "V_resting"
}
Block {
  BlockType      Constant
  Name           "Constant Stimulation"
  SID            "1091"
  Position       [865, 375, 945, 405]
  BlockMirror    on
  Value          "3"
}
Block {
  BlockType      Scope
  Name           "Currents"
  SID            "1077"
  Ports         [1]
  Position       [530, 548, 560, 582]
  Floating       off
  Location       [610, 157, 1274, 736]
}

```

```

Open                off
NumInputPorts      "1"
List {
ListType           AxesTitles
axes1              "%<SignalLabel>"
}
TimeRange          "10"
YMin               "-1000"
YMax               "1000"
SaveName           "ScopeData3"
DataFormat         "StructureWithTime"
SampleTime         "0"
}
Block {
BlockType          Scope
Name               "Gating Variables"
SID                "1071"
Ports              [3]
Position           [520, 313, 550, 347]
Floating           off
Location           [1314, 467, 1780, 1064]
Open               off
NumInputPorts      "3"
TickLabels         "on"
ZoomMode           "yonly"
List {
ListType           AxesTitles
axes1              "%<SignalLabel>"
axes2              "%<SignalLabel>"
axes3              "%<SignalLabel>"
}
YMin               "-5~-5~-5"
YMax               "5~5~5"
DataFormat         "StructureWithTime"
SampleTime         "0"
}
Block {
BlockType          SubSystem
Name               "KV"
SID                "48"
Ports              [1, 2]
Position           [459, 180, 501, 280]
BlockRotation      270
BlockMirror        on
NamePlacement      "alternate"
MinAlgLoopOccurrences off
PropExecContextOutsideSubsystem off
RTWSystemCode      "Auto"
FunctionWithSeparateData off
Opaque             off
RequestExecContextInheritance off
MaskHideContents   off
Port {
PortNumber         1
ShowPropagatedSignals "on"
RTWStorageClass    "Auto"
DataLoggingNameMode "SignalName"
}
}

```

```

}
System {
Name          "KV"
Location      [278, 98, 1342, 850]
Open          off
ModelBrowserVisibility on
ModelBrowserWidth 200
ScreenColor   "white"
PaperOrientation "landscape"
PaperPositionMode "auto"
PaperType     "usletter"
PaperUnits    "inches"
TiledPaperMargins [0.500000, 0.500000, 0.500000, 0.500000]
TiledPageScale 1
ShowPageBoundaries off
ZoomFactor    "100"
Block {
  BlockType    Inport
  Name         "V"
  SID          "49"
  Position     [65, 188, 95, 202]
  IconDisplay  "Port number"
}
Block {
  BlockType    Fcn
  Name         "1 - n"
  SID          "182"
  Position     [450, 49, 490, 81]
  Expr        "1-u"
}
Block {
  BlockType    Constant
  Name         "E K"
  SID          "68"
  Position     [65, 305, 95, 335]
  BackgroundColor "[0.901961, 0.901961, 0.901961]"
  Value       "E_K"
}
Block {
  BlockType    Integrator
  Name         "Integrator"
  SID          "167"
  Ports       [2, 1]
  Position     [765, 187, 795, 218]
  InitialConditionSource "external"
}
Block {
  BlockType    Product
  Name         "Product"
  SID          "977"
  Ports       [2, 1]
  Position     [715, 176, 730, 209]
  ShowName    off
  InputSameDT off
  OutDataTypeStr "Inherit: Inherit via internal rule"
  SaturateOnIntegerOverflow off
}
}

```

```

Block {
  BlockType      Product
  Name           "Product1"
  SID            "76"
  Ports          [3, 1]
  Position       [910, 357, 930, 393]
  ShowName      off
  Inputs         "3"
  InputSameDT   off
  OutDataTypeStr "Inherit: Inherit via internal rule"
  SaturateOnIntegerOverflow off
}
Block {
  BlockType      SubSystem
  Name           "Temperature Subsystem"
  SID            "963"
  Ports          [0, 1]
  Position       [65, 491, 145, 519]
  BackgroundColor "[0.803922, 0.929412, 0.996078]"
  MinAlgLoopOccurrences off
  PropExecContextOutsideSubsystem off
  RTWSystemCode  "Auto"
  FunctionWithSeparateData off
  Opaque         off
  RequestExecContextInheritance off
  MaskHideContents off
  System {
    Name          "Temperature Subsystem"
    Location      [2375, 212, 3422, 1027]
    Open          off
    ModelBrowserVisibility on
    ModelBrowserWidth 200
    ScreenColor    "white"
    PaperOrientation "landscape"
    PaperPositionMode "auto"
    PaperType      "usletter"
    PaperUnits     "inches"
    TiledPaperMargins [0.500000, 0.500000, 0.500000,
0.500000]
    TiledPageScale 1
    ShowPageBoundaries off
    ZoomFactor      "100"
    Block {
      BlockType      Constant
      Name           "Constant"
      SID            "964"
      Position       [80, 200, 175, 230]
      ShowName      off
      Value          "Temperature_Bit"
    }
    Block {
      BlockType      If
      Name           "If"
      SID            "965"
      Ports          [1, 2]
      Position       [265, 196, 365, 234]
      IfExpression   "u1 == 1"
    }
  }
}

```

```

}
Block {
  BlockType      SubSystem
  Name           "If Action\nSubsystem"
  SID           "966"
  Ports         [0, 1, 0, 0, 0, 0, 0, 1]
  Position       [465, 242, 535, 278]
  ShowName      off
  TreatAsAtomicUnit on
  MinAlgLoopOccurrences off
  PropExecContextOutsideSubsystem off
  RTWSystemCode "Auto"
  FunctionWithSeparateData off
  Opaque        off
  RequestExecContextInheritance off
  MaskHideContents off
  System {
    Name           "If Action\nSubsystem"
    Location       [648, 110, 1575, 870]
    Open          off
    ModelBrowserVisibility on
    ModelBrowserWidth 200
    ScreenColor   "white"
    PaperOrientation "landscape"
    PaperPositionMode "auto"
    PaperType     "usletter"
    PaperUnits    "inches"
    TiledPaperMargins [0.500000, 0.500000, 0.500000, 0.500000]
    TiledPageScale 1
    ShowPageBoundaries off
    ZoomFactor     "100"
  }
  Block {
    BlockType      ActionPort
    Name           "Action Port"
    SID           "967"
    Position       [225, 15, 284, 43]
    ActionType     "else"
  }
  Block {
    BlockType      Constant
    Name           "Constant"
    SID           "968"
    Position       [170, 70, 200, 100]
    ShowName      off
  }
  Block {
    BlockType      Outputport
    Name           "Out1"
    SID           "969"
    Position       [325, 78, 355, 92]
    IconDisplay    "Port number"
  }
  Line {
    SrcBlock      "Constant"
    SrcPort       1
    DstBlock      "Out1"
    DstPort       1
  }

```

```

    }
  }
}
Block {
  BlockType      SubSystem
  Name           "If Action\nSubsystem1"
  SID           "970"
  Ports         [0, 1, 0, 0, 0, 0, 0, 1]
  Position      [465, 167, 535, 203]
  ShowName      off
  TreatAsAtomicUnit  on
  MinAlgLoopOccurrences  off
  PropExecContextOutsideSubsystem  off
  RTWSystemCode "Auto"
  FunctionWithSeparateData  off
  Opaque        off
  RequestExecContextInheritance  off
  MaskHideContents  off
  System {
    Name           "If Action\nSubsystem1"
    Location      [695, 634, 1622, 1394]
    Open         off
    ModelBrowserVisibility  on
    ModelBrowserWidth  200
    ScreenColor  "white"
    PaperOrientation  "landscape"
    PaperPositionMode  "auto"
    PaperType     "usletter"
    PaperUnits    "inches"
    TiledPaperMargins  [0.500000, 0.500000, 0.500000, 0.500000]
    TiledPageScale  1
    ShowPageBoundaries  off
    ZoomFactor    "100"
  }
  Block {
    BlockType      ActionPort
    Name           "Action Port"
    SID           "971"
    Position      [345, 15, 404, 43]
    ActionType    "then"
  }
}
Block {
  BlockType      Constant
  Name           "Constant"
  SID           "972"
  Position      [25, 70, 105, 100]
  ShowName      off
  Value         "Temperature"
}
Block {
  BlockType      Fcn
  Name           "Fcn"
  SID           "973"
  Position      [320, 68, 435, 102]
  ShowName      off
  Expr          "3^((u-6.3)/10)"
}
Block {

```

```

        BlockType      Outport
        Name           "Out1"
        SID            "974"
        Position       [715, 78, 745, 92]
        IconDisplay    "Port number"
    }
    Line {
        SrcBlock       "Constant"
        SrcPort        1
        DstBlock       "Fcn"
        DstPort        1
    }
    Line {
        SrcBlock       "Fcn"
        SrcPort        1
        DstBlock       "Out1"
        DstPort        1
    }
    Annotation {
        Name           "The effect of the temperature is
taken care in the model so that the right-hand sides of the Equations
f"
        Position      [376, 179]
    }
}
Block {
    BlockType      Merge
    Name           "Merge"
    SID            "975"
    Ports          [2, 1]
    Position       [585, 195, 625, 235]
}
Block {
    BlockType      Scope
    Name           "Scope"
    SID            "1055"
    Ports          [1]
    Position       [795, 294, 825, 326]
    Floating       off
    Location       [188, 390, 512, 629]
    Open           off
    NumInputPorts "1"
    List {
        ListType    AxesTitles
        axes1       "%<SignalLabel>"
    }
    SaveName       "ScopeData6"
    DataFormat     "StructureWithTime"
    SampleTime     "0"
}
Block {
    BlockType      Outport
    Name           "Out1"
    SID            "976"

```

```

        Position          [770, 208, 800, 222]
        IconDisplay      "Port number"
    }
    Line {
        SrcBlock          "Merge"
        SrcPort           1
        Points            [75, 0]
        Branch {
            DstBlock      "Out1"
            DstPort       1
        }
        Branch {
            Points        [0, 95]
            DstBlock      "Scope"
            DstPort       1
        }
    }
    Line {
        SrcBlock          "If Action\nSubsystem"
        SrcPort           1
        Points            [15, 0; 0, -35]
        DstBlock          "Merge"
        DstPort           2
    }
    Line {
        SrcBlock          "If Action\nSubsystem1"
        SrcPort           1
        Points            [15, 0; 0, 20]
        DstBlock          "Merge"
        DstPort           1
    }
    Line {
        SrcBlock          "If"
        SrcPort           1
        Points            [85, 0; 0, -53]
        DstBlock          "If Action\nSubsystem1"
        DstPort           ifaction
    }
    Line {
        SrcBlock          "If"
        SrcPort           2
        Points            [130, 0]
        DstBlock          "If Action\nSubsystem"
        DstPort           ifaction
    }
    Line {
        SrcBlock          "Constant"
        SrcPort           1
        DstBlock          "If"
        DstPort           1
    }
    Annotation {
        Name              "Analysis of the Effect of temperature"
        Position          [418, 38]
    }
}
}

```



```

Block {
  BlockType      Sum
  Name           "V - V_K"
  SID            "66"
  Ports          [2, 1]
  Position       [205, 296, 220, 329]
  Inputs         "+-"
  InputSameDT    off
  OutDataTypeStr "Inherit: Inherit via internal rule"
  SaturateOnIntegerOverflow off
}
Block {
  BlockType      Product
  Name           "alpha (1 - n)"
  SID            "164"
  Ports          [2, 1]
  Position       [520, 97, 550, 128]
  InputSameDT    off
  OutDataTypeStr "Inherit: Inherit via internal rule"
  SaturateOnIntegerOverflow off
}
Block {
  BlockType      Sum
  Name           "alpha(1 - n) - beta * n"
  SID            "166"
  Ports          [2, 1]
  Position       [615, 167, 630, 198]
  Inputs         "+-"
  InputSameDT    off
  OutDataTypeStr "Inherit: Inherit via internal rule"
  SaturateOnIntegerOverflow off
}
Block {
  BlockType      Fcn
  Name           "alpha_n"
  SID            "82"
  Position       [205, 105, 265, 135]
  Expr           "(0.1-(0.01*u))/(exp(1-(0.1*u))-1)"
}
Block {
  BlockType      Product
  Name           "beta * n"
  SID            "165"
  Ports          [2, 1]
  Position       [455, 172, 485, 203]
  InputSameDT    off
  OutDataTypeStr "Inherit: Inherit via internal rule"
  SaturateOnIntegerOverflow off
}
Block {
  BlockType      Fcn
  Name           "beta_n"
  SID            "83"
  Position       [205, 180, 265, 210]
  Expr           "0.125/(exp(0.0125*u))"
}
Block {

```

```

BlockType      Constant
Name           "g K"
SID            "69"
Position       [65, 370, 95, 400]
BackgroundColor "[0.901961, 0.901961, 0.901961]"
Value         "G_K"
}
Block {
BlockType      Constant
Name           "n initial"
SID            "1079"
Position       [615, 304, 685, 326]
BackgroundColor "[0.901961, 0.901961, 0.901961]"
ShowName      off
Value         "n_initial"
}
Block {
BlockType      Fcn
Name           "n^4"
SID            "183"
Position       [840, 315, 885, 345]
Expr          "u^4"
}
Block {
BlockType      Outport
Name           "I_Potassium"
SID            "50"
Position       [970, 368, 1000, 382]
IconDisplay   "Port number"
}
Block {
BlockType      Outport
Name           "n_out"
SID            "1070"
Position       [970, 273, 1000, 287]
Port          "2"
IconDisplay   "Port number"
}
Line {
Labels         [0, 0]
SrcBlock       "v"
SrcPort        1
Points         [75, 0]
Branch {
Points         [0, 110]
DstBlock       "V - V_K"
DstPort        1
}
Branch {
DstBlock       "beta_n"
DstPort        1
}
Branch {
Points         [0, -75]
DstBlock       "alpha_n"
DstPort        1
}
}

```

```

}
Line {
  SrcBlock      "E K"
  SrcPort      1
  DstBlock      "V - V_K"
  DstPort      2
}
Line {
  SrcBlock      "Product1"
  SrcPort      1
  DstBlock      "I_Potassium"
  DstPort      1
}
Line {
  SrcBlock      "alpha_n"
  SrcPort      1
  DstBlock      "alpha (1 - n)"
  DstPort      2
}
Line {
  SrcBlock      "beta_n"
  SrcPort      1
  DstBlock      "beta * n"
  DstPort      2
}
Line {
  SrcBlock      "alpha (1 - n)"
  SrcPort      1
  Points      [35, 0; 0, 60]
  DstBlock      "alpha(1 - n) - beta * n"
  DstPort      1
}
Line {
  SrcBlock      "beta * n"
  SrcPort      1
  DstBlock      "alpha(1 - n) - beta * n"
  DstPort      2
}
Line {
  SrcBlock      "1 - n"
  SrcPort      1
  Points      [5, 0; 0, 40]
  DstBlock      "alpha (1 - n)"
  DstPort      1
}
Line {
  SrcBlock      "n^4"
  SrcPort      1
  Points      [0, 35]
  DstBlock      "Product1"
  DstPort      1
}
Line {
  SrcBlock      "V - V_K"
  SrcPort      1
  Points      [335, 0; 0, 60]
  DstBlock      "Product1"
}

```

```

    DstPort          2
  }
  Line {
    SrcBlock         "g K"
    SrcPort          1
    DstBlock         "Product1"
    DstPort          3
  }
  Line {
    SrcBlock         "Temperature Subsystem"
    SrcPort          1
    Points           [550, 0]
    DstBlock         "Product"
    DstPort          2
  }
  Line {
    SrcBlock         "Integrator"
    SrcPort          1
    Points           [25, 0; 0, 60]
    Branch {
      Points         [-420, 0; 0, -85]
      Branch {
        DstBlock     "beta * n"
        DstPort      1
      }
      Branch {
        Points       [0, -115]
        DstBlock     "1 - n"
        DstPort      1
      }
    }
    Branch {
      Points         [0, 15]
      Branch {
        DstBlock     "n^4"
        DstPort      1
      }
      Branch {
        DstBlock     "n_out"
        DstPort      1
      }
    }
  }
}
Line {
  SrcBlock         "Product"
  SrcPort          1
  DstBlock         "Integrator"
  DstPort          1
}
Line {
  SrcBlock         "alpha(1 - n) - beta * n"
  SrcPort          1
  DstBlock         "Product"
  DstPort          1
}
Line {
  SrcBlock         "n initial"

```

```

        SrcPort          1
        Points           [60, 0]
        DstBlock         "Integrator"
        DstPort          2
    }
}
}
Block {
    BlockType           SubSystem
    Name                "Leak"
    SID                 "51"
    Ports               [1, 1]
    Position             [599, 180, 641, 280]
    BlockRotation       270
    BlockMirror         on
    NamePlacement       "alternate"
    MinAlgLoopOccurrences off
    PropExecContextOutsideSubsystem off
    RTWSystemCode       "Auto"
    FunctionWithSeparateData off
    Opaque              off
    RequestExecContextInheritance off
    MaskHideContents   off
    Port {
        PortNumber      1
        ShowPropagatedSignals "on"
        RTWStorageClass "Auto"
        DataLoggingNameMode "SignalName"
    }
    System {
        Name            "Leak"
        Location        [546, 277, 1593, 1092]
        Open            off
        ModelBrowserVisibility on
        ModelBrowserWidth 200
        ScreenColor     "white"
        PaperOrientation "landscape"
        PaperPositionMode "auto"
        PaperType       "usletter"
        PaperUnits      "inches"
        TiledPaperMargins [0.500000, 0.500000, 0.500000, 0.500000]
        TiledPageScale  1
        ShowPageBoundaries off
        ZoomFactor      "100"
        Block {
            BlockType     Inport
            Name           "V"
            SID           "52"
            Position      [75, 123, 105, 137]
            IconDisplay    "Port number"
        }
    }
    Block {
        BlockType       Product
        Name            "(V - V leak) g leak"
        SID             "63"
        Ports           [2, 1]
        Position        [375, 231, 390, 249]
    }
}

```

```

    InputSameDT          off
    OutDataTypeStr       "Inherit: Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType           Sum
    Name                "V - V leak"
    SID                 "61"
    Ports               [2, 1]
    Position             [230, 160, 240, 180]
    Inputs              "+-"
    InputSameDT         off
    OutDataTypeStr       "Inherit: Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType           Constant
    Name                "V leak"
    SID                 "60"
    Position             [65, 215, 115, 245]
    BackgroundColor     "[0.901961, 0.901961, 0.901961]"
    Value               "E_leak"
}
Block {
    BlockType           Constant
    Name                "g leak"
    SID                 "62"
    Position             [65, 320, 120, 350]
    BackgroundColor     "[0.901961, 0.901961, 0.901961]"
    Value               "G_leak"
}
Block {
    BlockType           Outport
    Name                "I_Leak "
    SID                 "53"
    Position             [505, 233, 535, 247]
    IconDisplay         "Port number"
}
Line {
    SrcBlock            "V leak"
    SrcPort             1
    Points              [50, 0; 0, -55]
    DstBlock            "V - V leak"
    DstPort             2
}
Line {
    SrcBlock            "V"
    SrcPort             1
    Points              [60, 0; 0, 35]
    DstBlock            "V - V leak"
    DstPort             1
}
Line {
    SrcBlock            "V - V leak"
    SrcPort             1
    Points              [65, 0; 0, 65]
    DstBlock            "(V - V leak) g leak"
}

```

```

    DstPort          1
  }
  Line {
    SrcBlock         "g leak"
    SrcPort          1
    Points           [190, 0; 0, -90]
    DstBlock         "(V - V leak) g leak"
    DstPort          2
  }
  Line {
    SrcBlock         "(V - V leak) g leak"
    SrcPort          1
    DstBlock         "I_Leak "
    DstPort          1
  }
}
}
Block {
  BlockType         SubSystem
  Name              "Membrane"
  SID              "54"
  Ports            [1, 1]
  Position          [139, 180, 181, 280]
  BlockRotation    270
  BackgroundColor  "[1.000000, 1.000000, 0.501961]"
  MinAlgLoopOccurrences off
  PropExecContextOutsideSubsystem off
  RTWSystemCode    "Auto"
  FunctionWithSeparateData off
  Opaque           off
  RequestExecContextInheritance off
  MaskHideContents off
  System {
    Name            "Membrane"
    Location        [-30, 69, 1221, 874]
    Open            off
    ModelBrowserVisibility on
    ModelBrowserWidth 200
    ScreenColor    "white"
    PaperOrientation "landscape"
    PaperPositionMode "auto"
    PaperType      "usletter"
    PaperUnits     "inches"
    TiledPaperMargins [0.500000, 0.500000, 0.500000, 0.500000]
    TiledPageScale  1
    ShowPageBoundaries off
    ZoomFactor      "100"
  }
  Block {
    BlockType      Inport
    Name           "I"
    SID           "55"
    Position       [150, 103, 180, 117]
    IconDisplay    "Port number"
  }
}
Block {
  BlockType      Constant
  Name           "Cm"
}

```

```

    SID                "71"
    Position            [150, 205, 180, 235]
    BackgroundColor    "[0.901961, 0.901961, 0.901961]"
    Value              "Cm"
}
Block {
    BlockType          Product
    Name               "Divide"
    SID                "73"
    Ports              [2, 1]
    Position            [310, 153, 320, 177]
    Inputs             "*/"
    InputSameDT        off
    OutDataTypeStr     "Inherit: Inherit via internal rule"
    RndMeth            "Floor"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType          Integrator
    Name               "Integrator"
    SID                "74"
    Ports              [2, 1]
    Position            [460, 155, 490, 190]
    InitialConditionSource "external"
}
Block {
    BlockType          Constant
    Name               "V1"
    SID                "125"
    Position            [85, 295, 180, 325]
    BackgroundColor    "[0.901961, 0.901961, 0.901961]"
    ShowName           off
    Value              "V_initial"
}
Block {
    BlockType          Outport
    Name               "V"
    SID                "56"
    Position            [650, 168, 680, 182]
    IconDisplay        "Port number"
}
Line {
    SrcBlock           "I"
    SrcPort            1
    Points              [55, 0; 0, 50]
    DstBlock           "Divide"
    DstPort            1
}
Line {
    SrcBlock           "Cm"
    SrcPort            1
    Points              [55, 0; 0, -50]
    DstBlock           "Divide"
    DstPort            2
}
Line {
    SrcBlock           "V1"

```



```

        SrcPort          1
        Points           [260, 0]
        DstBlock         "Integrator"
        DstPort          2
    }
    Line {
        SrcBlock         "Integrator"
        SrcPort          1
        DstBlock         "V"
        DstPort          1
    }
    Line {
        SrcBlock         "Divide"
        SrcPort          1
        DstBlock         "Integrator"
        DstPort          1
    }
    Annotation {
        Name              "Membrane"
        Position          [302, 20]
    }
    Annotation {
        Name              "Anode break analysis"
        Position          [130, 342]
    }
}
Block {
    BlockType            SubSystem
    Name                 "NaV"
    SID                 "45"
    Ports                [1, 3]
    Position              [304, 180, 346, 280]
    BlockRotation        270
    BlockMirror          on
    NamePlacement        "alternate"
    MinAlgLoopOccurrences off
    PropExecContextOutsideSubsystem off
    RTWSystemCode        "Auto"
    FunctionWithSeparateData off
    Opaque               off
    RequestExecContextInheritance off
    MaskHideContents    off
    Port {
        PortNumber        1
        ShowPropagatedSignals "on"
        RTWStorageClass   "Auto"
        DataLoggingNameMode "SignalName"
    }
    System {
        Name              "NaV"
        Location          [278, 98, 1325, 833]
        Open              off
        ModelBrowserVisibility on
        ModelBrowserWidth 200
        ScreenColor       "white"
        PaperOrientation  "landscape"
    }
}

```

```

PaperPositionMode "auto"
PaperType         "usletter"
PaperUnits        "inches"
TiledPaperMargins [0.500000, 0.500000, 0.500000, 0.500000]
TiledPageScale    1
ShowPageBoundaries off
ZoomFactor        "100"
Block {
  BlockType      Inport
  Name           "V"
  SID            "46"
  Position       [100, 458, 130, 472]
  IconDisplay    "Port number"
}
Block {
  BlockType      Fcn
  Name           "1 - h"
  SID            "181"
  Position       [480, 270, 525, 300]
  Expr           "1-u"
}
Block {
  BlockType      Fcn
  Name           "1-m"
  SID            "180"
  Position       [485, 61, 525, 89]
  Expr           "1-u"
}
Block {
  BlockType      Constant
  Name           "E Na"
  SID            "59"
  Position       [100, 510, 130, 540]
  BackgroundColor "[0.901961, 0.901961, 0.901961]"
  Value         "E_Na"
}
Block {
  BlockType      Integrator
  Name           "Integrator"
  SID            "168"
  Ports          [2, 1]
  Position       [795, 192, 825, 223]
  InitialConditionSource "external"
}
Block {
  BlockType      Integrator
  Name           "Integrator2"
  SID            "169"
  Ports          [2, 1]
  Position       [805, 402, 835, 433]
  InitialConditionSource "external"
}
Block {
  BlockType      Product
  Name           "Product"
  SID            "961"
  Ports          [2, 1]
}

```

```

Position          [755, 181, 770, 214]
ShowName          off
InputSameDT       off
OutDataTypeStr    "Inherit: Inherit via internal rule"
SaturateOnIntegerOverflow off
}
Block {
BlockType         Product
Name              "Product1"
SID              "962"
Ports             [2, 1]
Position          [770, 391, 785, 429]
ShowName          off
InputSameDT       off
OutDataTypeStr    "Inherit: Inherit via internal rule"
SaturateOnIntegerOverflow off
}
Block {
BlockType         Product
Name              "Product3"
SID              "124"
Ports             [6, 1]
Position          [980, 575, 995, 635]
ShowName          off
Inputs           "6"
InputSameDT       off
OutDataTypeStr    "Inherit: Inherit via internal rule"
SaturateOnIntegerOverflow off
}
Block {
BlockType         Product
Name              "Product4"
SID              "176"
Ports             [2, 1]
Position          [565, 119, 585, 141]
ShowName          off
InputSameDT       off
OutDataTypeStr    "Inherit: Inherit via internal rule"
SaturateOnIntegerOverflow off
}
Block {
BlockType         Product
Name              "Product5"
SID              "177"
Ports             [2, 1]
Position          [565, 179, 585, 206]
ShowName          off
InputSameDT       off
OutDataTypeStr    "Inherit: Inherit via internal rule"
SaturateOnIntegerOverflow off
}
Block {
BlockType         Product
Name              "Product6"
SID              "178"
Ports             [2, 1]
Position          [565, 393, 590, 417]

```

```

    ShowName          off
    InputSameDT       off
    OutDataTypeStr    "Inherit: Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType         Product
    Name              "Product7"
    SID               "179"
    Ports             [2, 1]
    Position          [565, 304, 585, 331]
    ShowName          off
    InputSameDT       off
    OutDataTypeStr    "Inherit: Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType         Sum
    Name              "Subtract4"
    SID               "174"
    Ports             [2, 1]
    Position          [645, 172, 655, 203]
    ShowName          off
    Inputs            "+-"
    InputSameDT       off
    OutDataTypeStr    "Inherit: Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType         Sum
    Name              "Subtract5"
    SID               "175"
    Ports             [2, 1]
    Position          [645, 380, 660, 415]
    ShowName          off
    Inputs            "+-"
    InputSameDT       off
    OutDataTypeStr    "Inherit: Inherit via internal rule"
    SaturateOnIntegerOverflow off
}
Block {
    BlockType         SubSystem
    Name              "Temperature Subsystem"
    SID               "946"
    Ports             [0, 1]
    Position          [100, 696, 180, 724]
    BackgroundColor   "[0.803922, 0.929412, 0.996078]"
    MinAlgLoopOccurrences off
    PropExecContextOutsideSubsystem off
    RTWSystemCode     "Auto"
    FunctionWithSeparateData off
    Opaque            off
    RequestExecContextInheritance off
    MaskHideContents off
    System {
        Name          "Temperature Subsystem"
        Location       [783, 172, 1674, 995]
    }
}

```

```

Open                off
ModelBrowserVisibility on
ModelBrowserWidth  200
ScreenColor         "white"
PaperOrientation    "landscape"
PaperPositionMode  "auto"
PaperType           "usletter"
PaperUnits          "inches"
TiledPaperMargins  [0.500000, 0.500000, 0.500000,
0.500000]
TiledPageScale     1
ShowPageBoundaries off
ZoomFactor          "100"
Block {
  BlockType         Constant
  Name              "Constant"
  SID               "949"
  Position          [80, 200, 175, 230]
  ShowName          off
  Value             "Temperature_Bit"
}
Block {
  BlockType         If
  Name              "If"
  SID               "950"
  Ports            [1, 2]
  Position          [265, 196, 365, 234]
  IfExpression      "u1 == 1"
}
Block {
  BlockType         SubSystem
  Name              "If Action\nSubsystem"
  SID               "951"
  Ports            [0, 1, 0, 0, 0, 0, 0, 1]
  Position          [465, 242, 535, 278]
  ShowName          off
  TreatAsAtomicUnit on
  MinAlgLoopOccurrences off
  PropExecContextOutsideSubsystem off
  RTWSystemCode     "Auto"
  FunctionWithSeparateData off
  Opaque            off
  RequestExecContextInheritance off
  MaskHideContents off
  System {
    Name            "If Action\nSubsystem"
    Location        [648, 110, 1575, 870]
    Open            off
    ModelBrowserVisibility on
    ModelBrowserWidth 200
    ScreenColor     "white"
    PaperOrientation "landscape"
    PaperPositionMode "auto"
    PaperType       "usletter"
    PaperUnits      "inches"
    TiledPaperMargins [0.500000, 0.500000, 0.500000, 0.500000]
    TiledPageScale  1
  }
}

```

```

ShowPageBoundaries      off
ZoomFactor               "100"
Block {
  BlockType              ActionPort
  Name                   "Action Port"
  SID                    "952"
  Position               [225, 15, 284, 43]
  ActionType             "else"
}
Block {
  BlockType              Constant
  Name                   "Constant"
  SID                    "953"
  Position               [170, 70, 200, 100]
  ShowName               off
}
Block {
  BlockType              Outport
  Name                   "Out1"
  SID                    "954"
  Position               [325, 78, 355, 92]
  IconDisplay            "Port number"
}
Line {
  SrcBlock               "Constant"
  SrcPort                1
  DstBlock               "Out1"
  DstPort                1
}
}
Block {
  BlockType              SubSystem
  Name                   "If Action\nSubsystem1"
  SID                    "955"
  Ports                  [0, 1, 0, 0, 0, 0, 0, 1]
  Position               [465, 167, 535, 203]
  ShowName               off
  TreatAsAtomicUnit     on
  MinAlgLoopOccurrences off
  PropExecContextOutsideSubsystem off
  RTWSystemCode          "Auto"
  FunctionWithSeparateData off
  Opaque                 off
  RequestExecContextInheritance off
  MaskHideContents      off
  System {
    Name                  "If Action\nSubsystem1"
    Location              [648, 110, 1575, 870]
    Open                  off
    ModelBrowserVisibility on
    ModelBrowserWidth    200
    ScreenColor           "white"
    PaperOrientation      "landscape"
    PaperPositionMode     "auto"
    PaperType             "usletter"
    PaperUnits            "inches"
  }
}

```

```

TiledPaperMargins [0.500000, 0.500000, 0.500000, 0.500000]
TiledPageScale    1
ShowPageBoundaries off
ZoomFactor        "100"
Block {
  BlockType      ActionPort
  Name           "Action Port"
  SID           "956"
  Position       [345, 15, 404, 43]
  ActionType     "then"
}
Block {
  BlockType      Constant
  Name           "Constant"
  SID           "957"
  Position       [25, 70, 105, 100]
  ShowName      off
  Value         "Temperature"
}
Block {
  BlockType      Fcn
  Name           "Fcn"
  SID           "958"
  Position       [320, 68, 435, 102]
  ShowName      off
  Expr          "3^((u-6.3)/10)"
}
Block {
  BlockType      Output
  Name           "Out1"
  SID           "959"
  Position       [715, 78, 745, 92]
  IconDisplay   "Port number"
}
Line {
  SrcBlock      "Fcn"
  SrcPort       1
  DstBlock      "Out1"
  DstPort       1
}
Line {
  SrcBlock      "Constant"
  SrcPort       1
  DstBlock      "Fcn"
  DstPort       1
}
Annotation {
  Name          "The effect of the temperature is
taken care in the model so that the right-hand sides of the Equations
f"
  Position      [376, 179]
}
}
Block {

```

```

        BlockType      Merge
        Name            "Merge"
        SID             "960"
        Ports           [2, 1]
        Position        [585, 195, 625, 235]
    }
    Block {
        BlockType      Outport
        Name            "Out1"
        SID             "948"
        Position        [770, 208, 800, 222]
        IconDisplay     "Port number"
    }
    Line {
        SrcBlock        "Constant"
        SrcPort         1
        DstBlock        "If"
        DstPort         1
    }
    Line {
        SrcBlock        "If"
        SrcPort         2
        Points           [130, 0]
        DstBlock        "If Action\nSubsystem"
        DstPort         ifaction
    }
    Line {
        SrcBlock        "If"
        SrcPort         1
        Points           [85, 0; 0, -53]
        DstBlock        "If Action\nSubsystem1"
        DstPort         ifaction
    }
    Line {
        SrcBlock        "If Action\nSubsystem1"
        SrcPort         1
        Points           [15, 0; 0, 20]
        DstBlock        "Merge"
        DstPort         1
    }
    Line {
        SrcBlock        "If Action\nSubsystem"
        SrcPort         1
        Points           [15, 0; 0, -35]
        DstBlock        "Merge"
        DstPort         2
    }
    Line {
        SrcBlock        "Merge"
        SrcPort         1
        DstBlock        "Out1"
        DstPort         1
    }
    Annotation {
        Name            "Analysis of the Effect of temperature"
        Position        [418, 38]
    }
}

```



```

    }
  }
  Block {
    BlockType      Sum
    Name           "V - V_Na"
    SID            "65"
    Ports          [2, 1]
    Position       [260, 510, 275, 530]
    Inputs        "+-"
    InputSameDT    off
    OutDataStr     "Inherit: Inherit via internal rule"
    SaturateOnIntegerOverflow off
  }
  Block {
    BlockType      Fcn
    Name           "alpha_h"
    SID            "94"
    Position       [260, 310, 295, 340]
    Expr           "0.07/(exp(0.05*u))"
  }
  Block {
    BlockType      Fcn
    Name           "alpha_m"
    SID            "93"
    Position       [260, 119, 295, 151]
    Expr           "(2.5-(0.1*u))/(exp(2.5-(0.1*u))-1)"
  }
  Block {
    BlockType      Fcn
    Name           "beta_h"
    SID            "95"
    Position       [260, 395, 295, 425]
    Expr           "1/(exp(3-(0.1*u))+1)"
  }
  Block {
    BlockType      Fcn
    Name           "beta_m"
    SID            "96"
    Position       [260, 184, 295, 216]
    Expr           "4/(exp(u/18))"
  }
  Block {
    BlockType      Constant
    Name           "g Na"
    SID            "58"
    Position       [100, 615, 130, 645]
    BackgroundColor "[0.901961, 0.901961, 0.901961]"
    Value         "G_Na"
  }
  Block {
    BlockType      Constant
    Name           "h initial"
    SID            "1081"
    Position       [635, 504, 700, 526]
    BackgroundColor "[0.901961, 0.901961, 0.901961]"
    ShowName      off
    Value         "h_initial"
  }

```

```

}
Block {
  BlockType      Constant
  Name           "m initial"
  SID           "1080"
  Position       [625, 268, 700, 292]
  BackgroundColor "[0.901961, 0.901961, 0.901961]"
  ShowName      off
  Value         "m_initial"
}
Block {
  BlockType      Outport
  Name           "I_Sodium"
  SID           "47"
  Position       [1045, 598, 1075, 612]
  IconDisplay    "Port number"
}
Block {
  BlockType      Outport
  Name           "h_out"
  SID           "1067"
  Position       [1045, 458, 1075, 472]
  Port          "2"
  IconDisplay    "Port number"
}
Block {
  BlockType      Outport
  Name           "m_out"
  SID           "1068"
  Position       [1045, 398, 1075, 412]
  Port          "3"
  IconDisplay    "Port number"
}
Line {
  SrcBlock      "v"
  SrcPort       1
  Points        [25, 0]
  Branch {
    Points      [30, 0; 0, 50]
    DstBlock    "V - V_Na"
    DstPort     1
  }
  Branch {
    Points      [0, -55; 75, 0]
    Branch {
      DstBlock  "beta_h"
      DstPort   1
    }
  }
  Branch {
    Points      [0, -85]
    Branch {
      DstBlock  "alpha_h"
      DstPort   1
    }
  }
  Branch {
    Points      [0, -125]
  }
}

```

```

        DstBlock      "beta_m"
        DstPort      1
    }
    Branch {
        Points      [0, -65]
        DstBlock    "alpha_m"
        DstPort    1
    }
}
}
}
}
Line {
    SrcBlock      "E Na"
    SrcPort      1
    DstBlock      "V - V_Na"
    DstPort      2
}
Line {
    SrcBlock      "Product3"
    SrcPort      1
    DstBlock      "I_Sodium"
    DstPort      1
}
Line {
    SrcBlock      "alpha_m"
    SrcPort      1
    DstBlock      "Product4"
    DstPort      2
}
Line {
    SrcBlock      "alpha_h"
    SrcPort      1
    DstBlock      "Product7"
    DstPort      2
}
Line {
    SrcBlock      "beta_m"
    SrcPort      1
    DstBlock      "Product5"
    DstPort      2
}
Line {
    SrcBlock      "beta_h"
    SrcPort      1
    DstBlock      "Product6"
    DstPort      2
}
Line {
    SrcBlock      "Product6"
    SrcPort      1
    DstBlock      "Subtract5"
    DstPort      2
}
Line {
    SrcBlock      "Product7"
    SrcPort      1
}

```

```

    Points          [25, 0; 0, 70]
    DstBlock        "Subtract5"
    DstPort         1
  }
  Line {
    SrcBlock        "Product5"
    SrcPort         1
    DstBlock        "Subtract4"
    DstPort         2
  }
  Line {
    SrcBlock        "Product4"
    SrcPort         1
    Points          [25, 0; 0, 50]
    DstBlock        "Subtract4"
    DstPort         1
  }
  Line {
    SrcBlock        "1-m"
    SrcPort         1
    Points          [10, 0; 0, 50]
    DstBlock        "Product4"
    DstPort         1
  }
  Line {
    SrcBlock        "1 - h"
    SrcPort         1
    Points          [10, 0; 0, 25]
    DstBlock        "Product7"
    DstPort         1
  }
  Line {
    SrcBlock        "V - V_Na"
    SrcPort         1
    Points          [340, 0; 0, 100]
    DstBlock        "Product3"
    DstPort         5
  }
  Line {
    SrcBlock        "g Na"
    SrcPort         1
    DstBlock        "Product3"
    DstPort         6
  }
  Line {
    SrcBlock        "Temperature Subsystem"
    SrcPort         1
    Points          [555, 0; 0, -295]
    Branch {
      DstBlock        "Product"
      DstPort         2
    }
    Branch {
      Points          [0, 5]
      DstBlock        "Product1"
      DstPort         2
    }
  }

```

```

}
Line {
  SrcBlock      "Integrator2"
  SrcPort       1
  Points        [30, 0; 0, 40]
  Branch {
    Points      [0, 20]
    Branch {
      Points    [0, 130]
      DstBlock  "Product3"
      DstPort   4
    }
    Branch {
      Points    [-405, 0; 0, -80]
      Branch {
        DstBlock "Product6"
        DstPort  1
      }
      Branch {
        DstBlock "1 - h"
        DstPort  1
      }
    }
  }
  Branch {
    Points      [0, 5]
    DstBlock    "h_out"
    DstPort     1
  }
}
Line {
  SrcBlock      "Product1"
  SrcPort       1
  DstBlock      "Integrator2"
  DstPort       1
}
Line {
  SrcBlock      "Subtract5"
  SrcPort       1
  DstBlock      "Product1"
  DstPort       1
}
Line {
  SrcBlock      "Product"
  SrcPort       1
  DstBlock      "Integrator"
  DstPort       1
}
Line {
  SrcBlock      "Subtract4"
  SrcPort       1
  DstBlock      "Product"
  DstPort       1
}
Line {
  SrcBlock      "Integrator"
  SrcPort       1
}

```

```

Points          [45, 0; 0, 20]
Branch {
  Points        [-405, 0; 0, -60]
  Branch {
    Points      [0, 15]
    DstBlock    "Product5"
    DstPort     1
  }
  Branch {
    DstBlock    "1-m"
    DstPort     1
  }
}
Branch {
  Points        [0, 165]
  Branch {
    Points      [0, 185]
    Branch {
      Points    [0, 10]
      Branch {
        Points  [0, 10]
        DstBlock "Product3"
        DstPort  3
      }
    }
    Branch {
      DstBlock  "Product3"
      DstPort   2
    }
  }
  Branch {
    DstBlock    "Product3"
    DstPort     1
  }
}
Branch {
  Points        [0, 10]
  DstBlock      "m_out"
  DstPort       1
}
}
}
Line {
  SrcBlock      "m initial"
  SrcPort       1
  Points        [75, 0]
  DstBlock      "Integrator"
  DstPort       2
}
Line {
  SrcBlock      "h initial"
  SrcPort       1
  Points        [85, 0]
  DstBlock      "Integrator2"
  DstPort       2
}
Annotation {
  Position      [798, 414]
}

```

```

    }
  }
}
Block {
  BlockType          SubSystem
  Name               "Signal Builder"
  SID               "1092"
  Tag               "STV Subsys"
  Ports             [0, 1]
  Position          [860, 460, 945, 500]
  BlockMirror       on
  CopyFcn           "sigbuilder_block('copy');"
  DeleteFcn         "sigbuilder_block('delete');"
  LoadFcn           "sigbuilder_block('load');"
  ModelCloseFcn     "sigbuilder_block('modelClose');"
  PreSaveFcn        "sigbuilder_block('preSave');"
  StartFcn          "sigbuilder_block('start');"
  StopFcn           "sigbuilder_block('stop');"
  NameChangeFcn     "sigbuilder_block('namechange');"
  ClipboardFcn      "sigbuilder_block('clipboard');"
  OpenFcn           "sigbuilder_block('open',[70.5 105 547.5
236.25 ]);"
  CloseFcn          "sigbuilder_block('close');"
  MinAlgLoopOccurrences off
  PropExecContextOutsideSubsystem off
  RTWSystemCode     "Auto"
  FunctionWithSeparateData off
  Opaque            off
  RequestExecContextInheritance off
  MaskHideContents on
  MaskType          "Sigbuilder block"
  MaskDescription   "Create and generate interchangeable
groups of signals whose waveforms are piecewise linea"
  "r."
  MaskInitialization "if
~strcmp(get_param(bdroot(gcbh),'SimulationStatus'),'stopped') tuvar =
sigbuilder_blo"
  "ck('maskInit'); end"
  MaskSelfModifiable on
  MaskDisplay       "plot(0, 0, 100, 100,[2, 2, 32, 32, 2], [68, 8,
8, 68, 68],[32, 2], [38, 38], [32, 19, 2],[53,"
  " 60, 44], [32, 17, 17, 2],[16, 16, 31, 31]);txt =
getActiveGroup(gcbh);text(2, 100, txt,'verticalAlignment', 'to"
  "p');"
  MaskIconFrame     on
  MaskIconOpaque    off
  MaskIconRotate    "port"
  MaskPortRotate    "default"
  MaskIconUnits     "autoscale"
  MaskCapabilities  "slmaskedcaps(gcbh)"
  System {
    Name           "Signal Builder"
    Location       [480, 85, 980, 386]
    Open           off
    ModelBrowserVisibility on
    ModelBrowserWidth 200
    ScreenColor    "white"

```

```

PaperOrientation "landscape"
PaperPositionMode "auto"
PaperType "usletter"
PaperUnits "inches"
TiledPaperMargins [0.500000, 0.500000, 0.500000, 0.500000]
TiledPageScale 1
ShowPageBoundaries off
ZoomFactor "100"
SIDHighWatermark "17"
Block {
  BlockType Demux
  Name "Demux"
  SID "1092:14"
  Tag "STV Demux"
  Ports [1, 1]
  Position [150, 42, 160, 618]
  Outputs "1"
}
Block {
  BlockType FromWorkspace
  Name "FromWs"
  SID "1092:2"
  Tag "STV FromWs"
  Position [30, 300, 115, 350]
  VariableName "tuvar"
  SampleTime "0"
  ZeroCross on
  SigBuilderData "DataTag0"
  VnvData "DataTag1"
}
Block {
  BlockType Outport
  Name "Signal 1"
  SID "1092:15"
  Tag "STV Outport"
  Position [280, 30, 300, 40]
  IconDisplay "Port number"
}
Line {
  SrcBlock "FromWs"
  SrcPort 1
  DstBlock "Demux"
  DstPort 1
}
Line {
  SrcBlock "Demux"
  SrcPort 1
  DstBlock "Signal 1"
  DstPort 1
}
}
Block {
  BlockType UnaryMinus
  Name "Unary Minus"
  SID "1083"
  Position [210, 565, 240, 595]
}

```



```

    ShowName                off
}
Line {
  SrcBlock                  "Membrane"
  SrcPort                   1
  Points                    [0, -70; 85, 0]
  Branch {
    Points                  [80, 0]
    Branch {
      Points                [155, 0]
      Branch {
        DstBlock            "KV"
        DstPort              1
      }
      Branch {
        Points              [140, 0]
        DstBlock            "Leak"
        DstPort              1
      }
    }
    Branch {
      DstBlock              "NaV"
      DstPort                1
    }
  }
  Branch {
    DstBlock                "Add1"
    DstPort                  2
  }
}
Line {
  SrcBlock                  "Add"
  SrcPort                   1
  Points                    [-40, 0]
  Branch {
    Points                  [-25, 0]
    DstBlock                "Membrane"
    DstPort                  1
  }
  Branch {
    Points                  [0, 205]
    DstBlock                "Unary Minus"
    DstPort                  1
  }
}
Line {
  SrcBlock                  "NaV"
  SrcPort                   1
  Points                    [0, 75; -20, 0]
  Branch {
    DstBlock                "Add"
    DstPort                  1
  }
  Branch {
    Points                  [0, 190]
    DstBlock                "Bus\nCreator"
    DstPort                  1
  }
}

```

```

    }
  }
  Line {
    SrcBlock          "KV"
    SrcPort           1
    Points            [0, 85; -190, 0]
    Branch {
      DstBlock        "Add"
      DstPort         2
    }
    Branch {
      Points          [0, 190]
      DstBlock        "Bus\nCreator"
      DstPort         2
    }
  }
  Line {
    SrcBlock          "Leak"
    SrcPort           1
    Points            [0, 95; -310, 0]
    Branch {
      DstBlock        "Add"
      DstPort         3
    }
    Branch {
      Points          [0, 190]
      DstBlock        "Bus\nCreator"
      DstPort         3
    }
  }
  Line {
    SrcBlock          "Add1"
    SrcPort           1
    DstBlock          "Action Potential"
    DstPort           1
  }
  Line {
    SrcBlock          "Constant"
    SrcPort           1
    DstBlock          "Add1"
    DstPort           1
  }
  Line {
    SrcBlock          "NaV"
    SrcPort           3
    Points            [0, 35]
    DstBlock          "Gating Variables"
    DstPort           1
  }
  Line {
    SrcBlock          "KV"
    SrcPort           2
    Points            [0, 45]
    DstBlock          "Gating Variables"
    DstPort           2
  }
  Line {

```

```

        SrcBlock          "NaV"
        SrcPort           2
        Points            [0, 55]
        DstBlock          "Gating Variables"
        DstPort           3
    }
    Line {
        SrcBlock          "Bus\nCreator"
        SrcPort           1
        DstBlock          "Currents"
        DstPort           1
    }
    Line {
        SrcBlock          "Unary Minus"
        SrcPort           1
        DstBlock          "Bus\nCreator"
        DstPort           4
    }
    Line {
        SrcBlock          "Constant Stimulation"
        SrcPort           1
        DstBlock          "Add"
        DstPort           4
    }
    Annotation {
        Name              "Hondgkin and Huxley Model"
        Position          [394, 19]
    }
    Annotation {
        Position          [262, 63]
    }
    Annotation {
        Position          [248, 382]
    }
    Annotation {
        Position          [357, 77]
    }
}
}

# Finite State Machines
#
#   Stateflow Version 7.5 (R2010a) dated Mar  5 2011, 00:01:50
#
#

Stateflow {
    machine {
        id                1
        name              "MyModel"
        created           "31-Jan-2013 22:29:51"
        isLibrary         0
        firstTarget       2
        sfVersion         75014001
    }
    target {

```

```
id          2
name        "sfun"
description "Default Simulink S-Function Target."
machine     1
linkNode    [1 0 0]
}
}
```