

---

**Rice/TMC/UH  
Computational  
Neuroscience REU**

---

Frequency  
Response of Model  
Cells

---

Sarah Fried, University of  
the South: Sewanee, TN

---

# Summary

This work was done in Dr. Peter Saggau's laboratory at Baylor College of Medicine with the support of the Rice/TMC/UH Computational Neuroscience REU and *NSF REU Grant DMS-0755294*.

## Introduction

One way to model neurons is as Resistor-Capacitor Circuits. From this, one can study the frequency response characteristics of the cell's membrane. It is known that a passive cell membrane acts as a low-pass filter. Active conductances, however, can change the frequency response of the cell, and allow it to filter high frequencies, low frequencies, or to resonate in specific frequency ranges. A cell membrane with a hyper-polarization activated current, so called h-current or  $I_h$ , is thought to behave as a low-pass filter, filtering out frequencies lower than the resonance frequency. In CA1 pyramidal cells, it has been shown that the magnitude of h-current varies with distance from the soma, so the site of current injection along the dendrite may also change the frequency response at the soma.

## Background

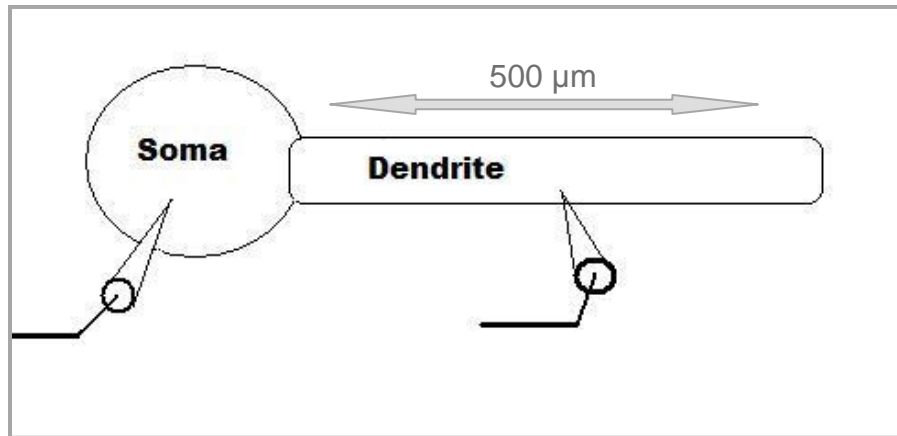
A low-pass filter allows low frequencies to pass through the filter, but reduces the amplitude of frequencies higher than the cutoff frequency. On the opposite hand, a high-pass filter allows high frequencies to pass through the filter, but reduces the amplitude of frequencies lower than the cutoff frequency. A combination of both a high-pass and low-pass filter is called a band-pass filter.

The Hodgkin-Huxley (H-H) channels used in one of the models is based on the experimentally determined sodium ( $\text{Na}^+$ ) and potassium ( $\text{K}^+$ ) channel mechanisms in the squid giant axon from the Hodgkin-Huxley paper published in 1952. In the more complex model cell, the ion channels are still comprised of both sodium and potassium channels, but the hyper-polarization ( $I_h$ ) channel has also been added. The  $\text{Na}^+$  and  $\text{K}^+$  channels in the complex model are based on experimentally determined values from rats' hippocampus CA1 pyramidal cells instead of the squid giant's axon cells.

*Since one neuron receives thousands of inputs from other cells, studying the frequency response can give a clue as to which inputs lead to an action potential and which are essentially 'ignored,' or filtered out.*

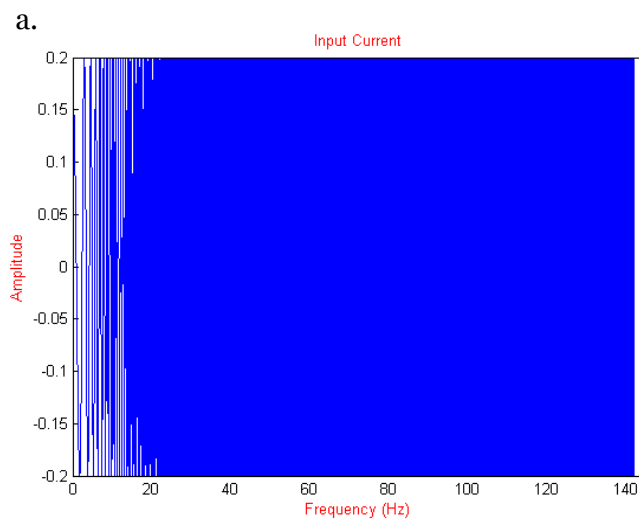
# Method

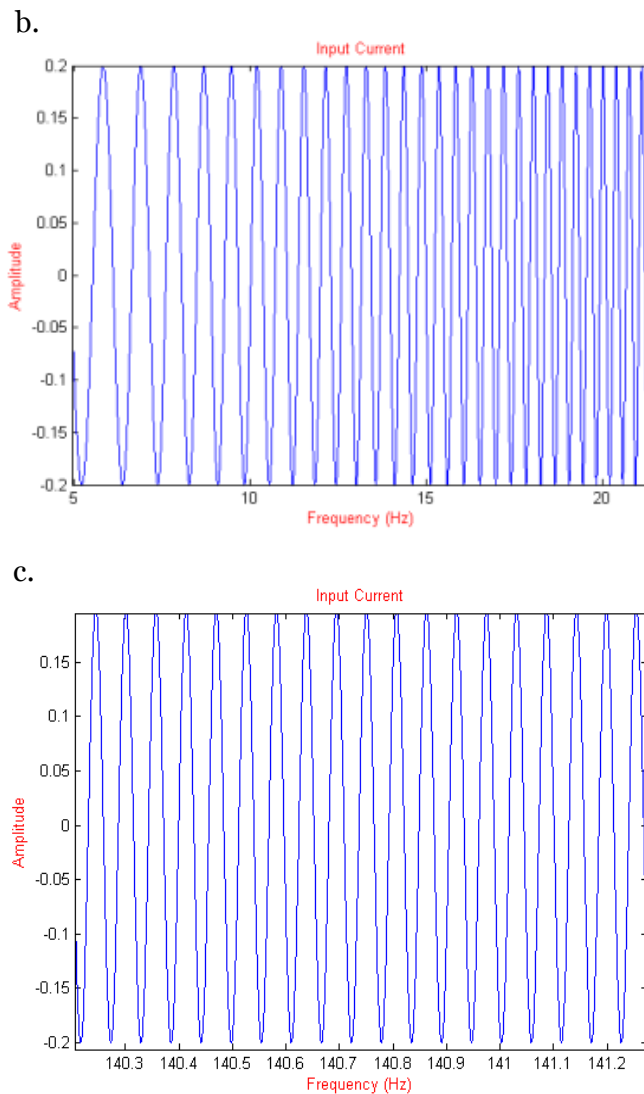
To test the somatic voltage response, a simple ball-and-stick cell was made in NEURON® (**figure 1**). The code could be rewritten to include or exclude Hodgkin-Huxley Sodium and Potassium ion channels in the cell membrane.



**Figure 1: Simple Ball-and-Stick Model**

The cell has one injection site along the dendrite for the current; and since it was thought that the frequency response at the soma is dependent on the placement of the injection site, the current clamp could be attached to different places on the dendrite. The voltage response for all the injections was recorded at the soma.

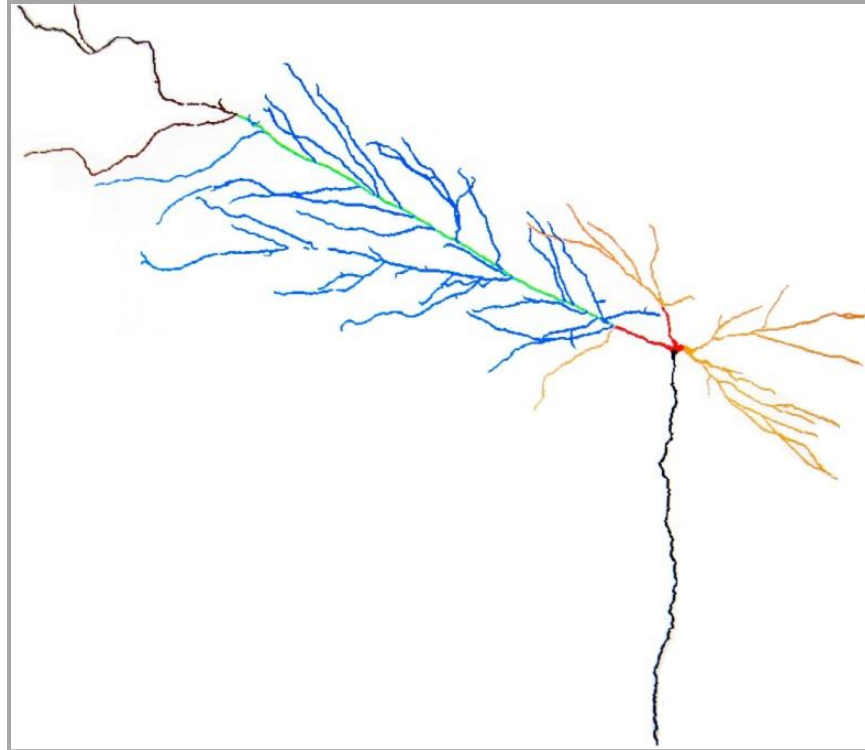




**Figure 2: a.) Entire Chirped Input Current, b.) Lower Frequency Chirped Input, c.) Higher Frequency Chirped Input**

To determine which range of frequencies was filtered out by the cell, a sinusoidal (chirped) current (**figure 2**) with a linearly increasing frequency was injected through the dendrite for ten seconds.

After the model of the simple cell was constructed and tested, a new program in NEURON® was used for a more complex model that had a variety of ion channels (**figure 3**).

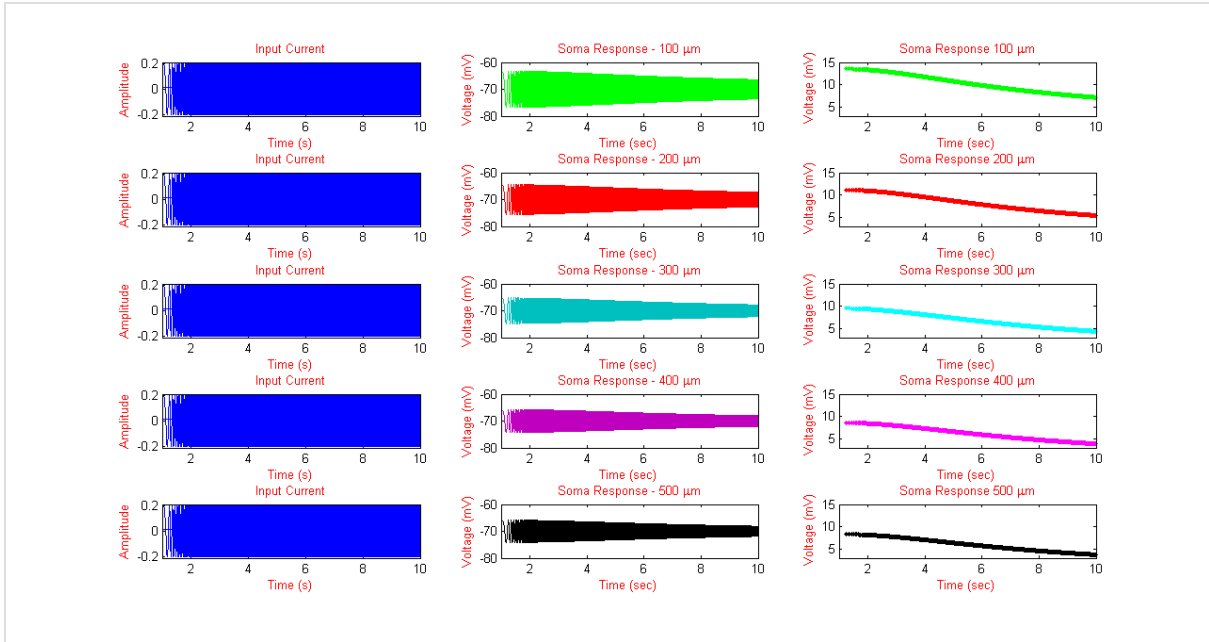


**Figure 3: Complex Cell Model**

This more complex model includes a variety of sodium ( $\text{Na}^+$ ), potassium ( $\text{K}^+$ ), and hyper-polarization ( $I_h$ ) activated channels; the proportion of each channel type, as well as the geometry of each dendritic section reflects what is known about real neurons.

## Analysis

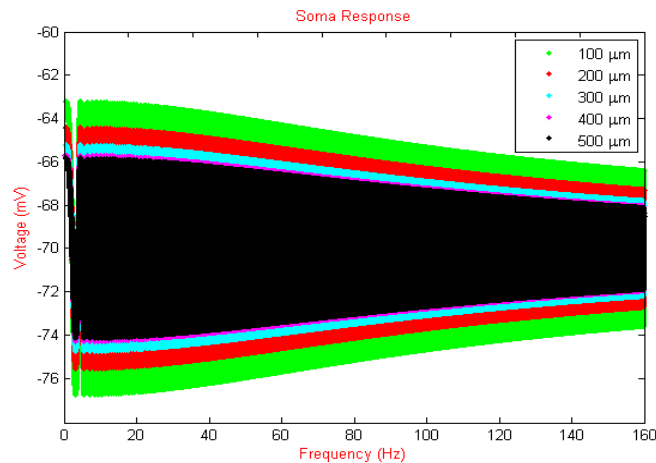
To analyze the data in MATLAB, a program was written that plots not only the input current vector and time vector (**figure 4.a**) but the voltage response at the soma with the time vector (**figure 4.b**). Since there were a lot of data points in the voltage response, the program includes code to plot the maxima and minima values, as well as the normalized difference between the maxima and minima values (**figure 4.c**), to make the visualization easier to interpret.



**Figure 4: Input Current, Voltage Response, and Normalized Difference of a Passive Cell**

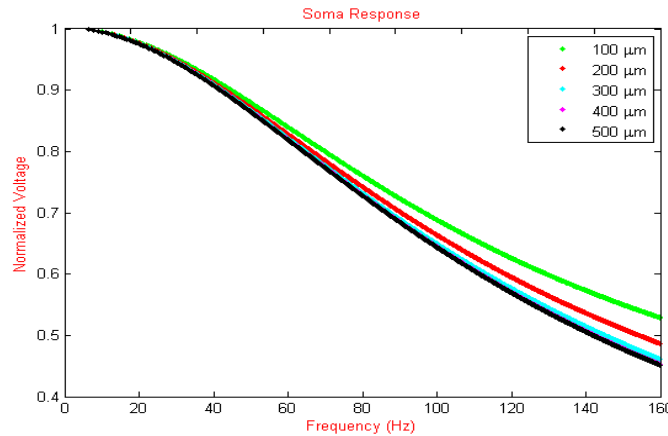
## Without Ion Channels

The somatic response of a passive cell shows a steady decrease in the amplitude with increasing frequencies, thus behaving more like a low pass filter (high frequency filter) (**figure 5**). The further the injection site from the soma, the more the injected current is filtered out (both low and high frequencies).



**Figure 5: Voltage Response for a Passive Cell with Various Injection Sites**

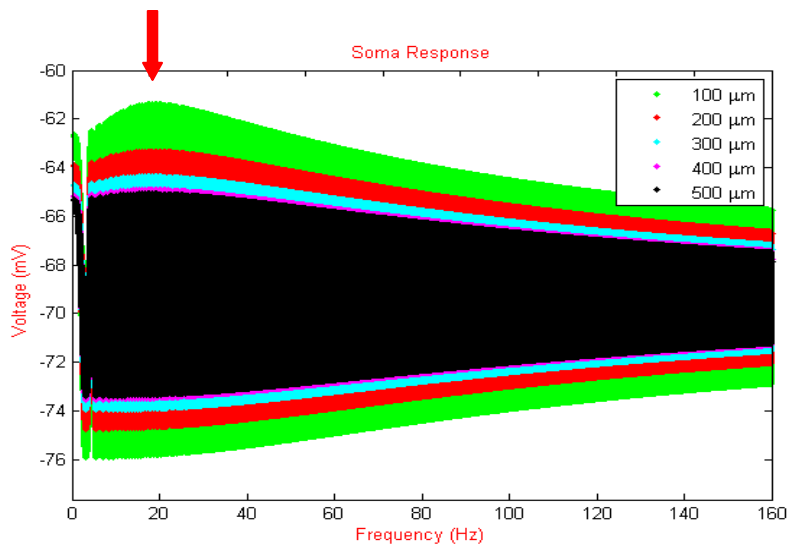
Overall, the more distal injection sites on the dendrite had more of an effect on higher frequencies than those attached near the soma (**figure 6**). At the distal injection sites, the higher frequencies were filtered out much more than the lower frequencies.



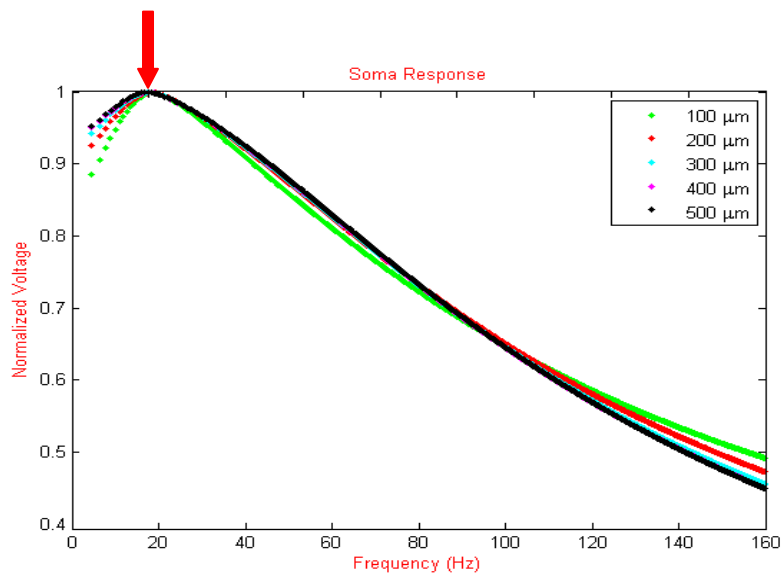
**Figure 6: Normalized Difference of Voltage Response of a Passive Cell with Various Injection Sites**

## Hodgkin-Huxley Channels

The soma's response of a cell with the standard Hodgkin-Huxley Sodium and Potassium ion channels also shows a decrease in the higher frequencies like a low pass filter (**figure 7**).



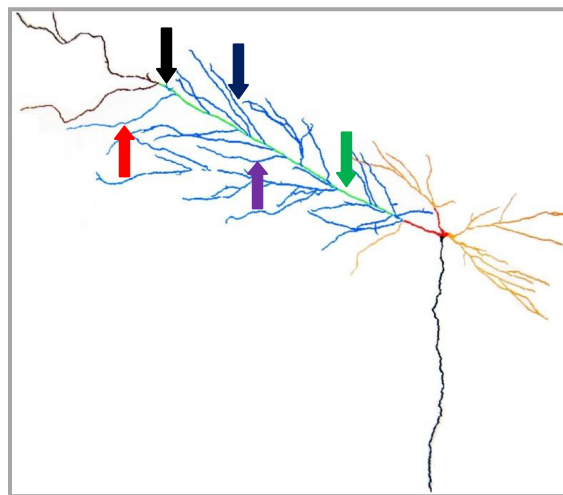
**Figure 7: Voltage Response for a H-H cell with Various Injection Sites**



**Figure 8: Normalized Difference of Voltage Response of H-H Cell with Various Injection Sites**

The maximum point in both figures seven and eight (*red arrow*), reflects a resonant frequency of the membrane imposed by the time dependence of the H-H channels. As in the passive cell, the most distal injection site on the dendrite still filters out more of the input's higher frequencies than the proximal end attached to the soma (**figure 8**).

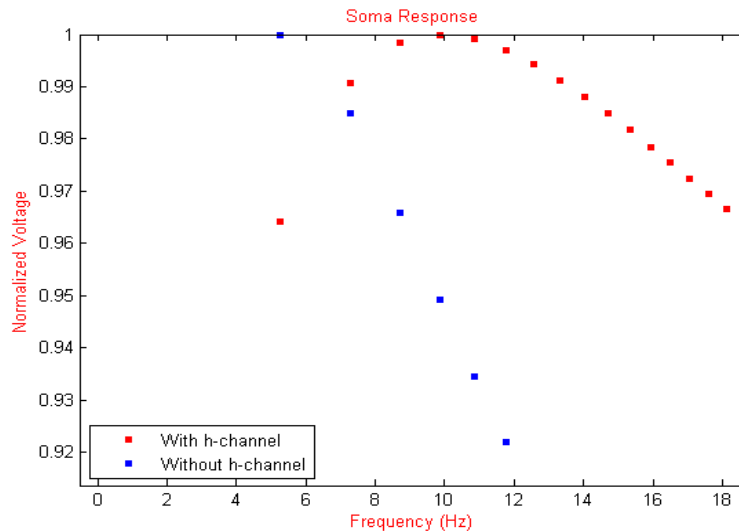
## Variety of Channels



**Figure 9: Complex Cell Stimulation Sites**



A chirped input was injected at a relatively distal oblique branch 42 (**figure 9: red arrow**) for six seconds. The voltage response with the h-channel at the soma (**figure 10**) shows that lower frequencies have been filtered out. The cell acts more like a high pass filter.

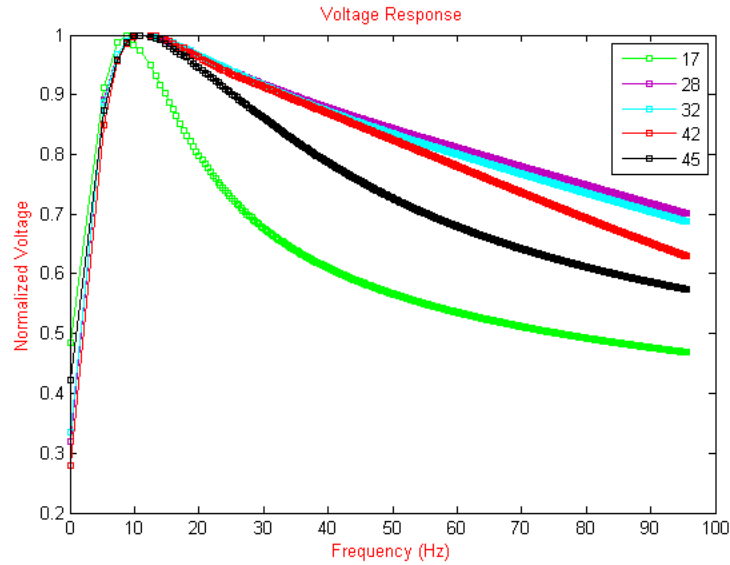


**Figure 10: Voltage Response with and without h-Channel at Branch 42**

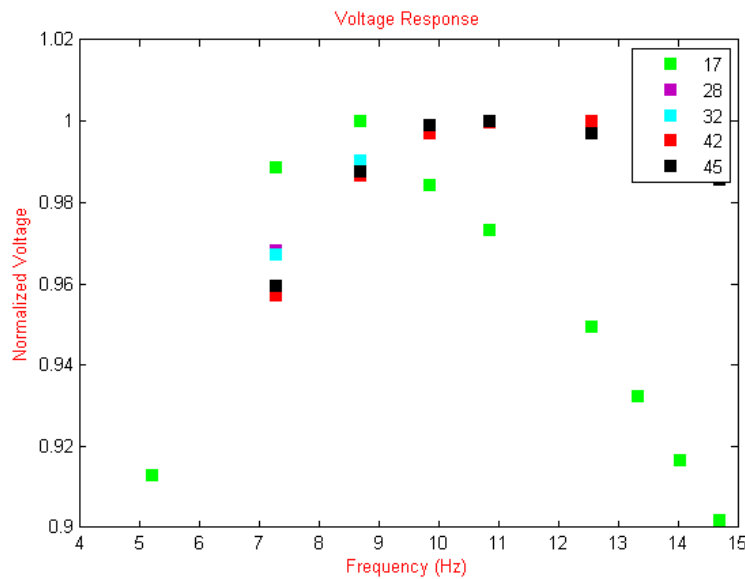
The response with the h-channel shows that the frequencies lower than 10 hertz have been filtered out; whereas, the response without the h-channel shows a response similar to the passive cell's response: a decreasing amplitude with increasing frequencies.

Injecting the chirped current into five different locations (**figure 9**) for six seconds shows similar behavior (**figure 11**) in all five branch locations. The branches located on the main dendrite have a similar response (**green** and **black**); whereas, the oblique dendrites show a similar pattern amongst themselves (**purple, blue, and red**).

Upon closer examination of the voltage responses, the branch closer to the soma (**figure 12: green**) along the main dendritic trunk has a resonance frequency around 9 hertz; whereas, the branch further away from the soma (**figure 12: black**) still along the main dendritic trunk has a resonant frequency between 10 and 11 hertz. The voltage responses of the oblique branches (**figure 12: purple, blue, and red**) also seem to have a resonant frequency between 10 and 11 hertz. It appears that the frequency response depends on the distance away from the soma of the injection site along the main dendritic branch; whereas, there does not seem to be a difference in resonance frequency for the oblique branches depending on distance.



**Figure 11: Voltage Response of 5 Branches of the Complex Model**



**Figure 12: Zoom of the Resonant Frequencies of the 5 Branches of the Complex Model**

## Acknowledgments

Special thanks dedicated to Dr. Peter Saggau and Keith Kelleher for support and mentorship, and Dr. Cox for the encouragement from the weekly meetings.

# Appendix of Code

## NEURON®:

### For the simple cell with the sinusoidal current injection:

```
//IClamp is in the dendrite; with
cosine wave; only one dendrite
objref istim, s, tvec, ivec, recv,
rect, g
create soma, dend
access soma
soma{
    diam = 10
    L = 10
    insert pas
    insert hh
    gnabar_hh = .5
// g_pas = .001
// e_pas = -58
}
dend{
    diam = 2
    L = 500
    nseg = 99
    insert pas
// g_pas = .001
// e_pas = -58
}
connect dend(0), soma(1)
v_init = -70
s = new Shape()
s.show(0)
tstop = 10000
access dend
//d is the delay at the start for
the steady state
//h is the amplitude of the cosine
wave
h = 0.35
d = 1000
rect = new Vector()
recv = new Vector()
//y is where the IClamp is placed
in the dendrite
y = 0.2
istim = new IClamp(y)
    istim.del = 0
    istim.dur = 10000
tvec = new Vector(tstop/.025)
ivec = new Vector(tstop/.025)
stepnum = 0
proc advance() {
    fadvance()
    if(t <= d) {
        istim.amp= h
    } else {
        istim.amp = h*cos(((t-
d)*(t-d)*0.0001))
    }
    tvec.x[stepnum] = t
    ivec.x[stepnum] = istim.amp
    stepnum = stepnum + 1
}
recv.record(&soma.v(0.5))
rect.record(&t)
run()
g = new Graph()
recv.line(g, rect)
//save data
objref f
strdef fname
sprintf(fname,
"C:/Users/Sarah/Desktop/TCN
'09/volt_%g_dend_%g.txt", h, y)
f = new File(fname)
f.wopen()
recv.printf(f)
f.close()
sprintf(fname,
"C:/Users/Sarah/Desktop/TCN
'09/iv_%g_dend_%g.txt", h, y)
f = new File(fname)
f.wopen()
ivec.printf(f)
f.close()
sprintf(fname,
"C:/Users/Sarah/Desktop/TCN
'09/tv_%g_dend_%g.txt", h, y)
f = new File(fname)
f.wopen()
tvec.printf(f)
f.close()
```

## For the complex cell with the sinusoidal current injection:

```
load_file("nrngui.hoc")
load_file("import3d_sec.hoc")
load_file("read_swc.hoc")
objref cell
cell = new Import3d_SWC_read()
cell.input("c.swc")
load_file("ParseNeuron5.hoc")
criticalRatio = .8
somaSize = 1
startParse()
initChannels()
// y is the branch number on the
neuron
y = 45
access K[y]
objref istim, tvec, ivec, recv,
rect, g
rect = new Vector()
recv = new Vector()
//d is the delay at the start for
the steady state
//h is the amplitude of the cosine
wave
h = -0.1
d = 150
istim = new IClamp((0.5))
    istim.del = 0
    istim.dur = 3000000000000
    istim.amp = -1
tstop = 6000
tvec = new Vector(tstop/.025)
ivec = new Vector(tstop/.025)
stepnum = 0
proc advance() {
    fadvance()
        if(t <= d) {
            istim.amp= h
        } else {
            istim.amp = h*cos(((t-
d)*(t-d)*0.0001))
        }
        tvec.x[stepnum] = t
        ivec.x[stepnum] = istim.amp
        stepnum = stepnum + 1
    }
    recv.record(&v(0.5))
    rect.record(&t)
    run()
    g = new Graph()
    recv.line(g, rect)
    //save data
    objref f
    strdef fname
    sprintf(fname,
"C:/Users/Sarah/Desktop/TCN
'09/n%gvolt.txt",y)
    f = new File(fname)
    f.wopen()
    recv.printf(f)
    f.close()
    sprintf(fname,
"C:/Users/Sarah/Desktop/TCN
'09/n%giv.txt",y)
    f = new File(fname)
    f.wopen()
    ivec.printf(f)
    f.close()
    sprintf(fname,
"C:/Users/Sarah/Desktop/TCN
'09/n%gtv.txt",y)
    f = new File(fname)
    f.wopen()
    tvec.printf(f)
    f.close()
    //objref ps
    //ps = new PlotShape()
    //tempSH.rotate(0,0,0,0,0,(PI/2))
    // for orientation
    //tempSH.show(0)
```

## MATLAB®:

### For the simple cell input:

```
function returnvariable=
plotinput(X,Y)
%% Need to import one file for
INPUT current; for INPUT only!
% assign Y & X.
% Y = input voltage data; X = input
time data
% For simple neuron model, plots
the power spec and current waveform
% This function automatically gets
rid of last 40000 points for x,y
x=x(1:end-40000); y=y(1:end-40000);

%% For the frequency
freq = ((0.0001*X)/(2*pi))*(1000);

%% the x and y data for the current
waveform

% Create figure
figure1 = figure;

% Create axes
axes('Parent',figure1);
box('on');
hold('all');
plot(freq,Y);
xlabel('Frequency (Hz)', 'Color', [1 0 0]);
ylabel('Amplitude', 'Color', [1 0 0]);
title('Input Current
Waveform', 'Color', [1 0 0]);

% Creates a max/min plot of the
current waveform
[Ymax, imax, Ymin, imin] =
extrema(Y);
Ymax = Ymax(2:end-1);
Ymin = Ymin(2:end-1);
imax = imax(2:end-1);
assignin('base', 'Ymax', Ymax);
assignin('base', 'imax', imax);
assignin('base', 'Ymin', Ymin);
assignin('base', 'imin', imin);

    if (length(Ymax) >
length(Ymin))

        Ymax =
Ymax(1:length(Ymin));
        end
        if (length(imax) >
length(imin))
            imax =
imax(1:length(imin));
            end

        if (length(Ymin) >
length(Ymax))
            Ymin =
Ymin(1:length(Ymax));
            end
            if (length(imin) >
length(imax))
                imin =
imin(1:length(imax));
                end

        [sortfreq, sortorder] =
sort(freq(imax));
        [sortFreq, sortorder2] =
sort(freq(imin));
        diff = Ymax(sortorder) -
Ymin(sortorder2);
        d = max(diff);
        plot(sortfreq, (diff./d),
'r. ');
        xlabel('Frequency
(Hz)', 'Color', [1 0 0]);
        ylabel('Normalized
Voltage', 'Color', [1 0 0]);
        title('Soma Response', 'Color',
[1 0 0]);

%% the x data and y data for the
power spec

Y1 = abs(fft(Y));

% Create figure
figure1 = figure;

% Create axes
axes('Parent',figure1);
box('on');
hold('all');
```

```

semilogy(freq,Y1);
xlabel('Frequency (Hz)', 'Color', [1
0 0]);
ylabel('Power', 'Color', [1 0 0]);
title('Power Spectrum of Input
Current Waveform', 'Color', [1 0
0]);% Creates a plot of the power
spec

% Creates a max/min plot of the
power response
[Ylmax, ilmax, Ylmin, ilmin] =
extrema(Y1);
Ylmax = Ylmax(2:end-1);
Ylmin = Ylmin(2:end-1);
ilmax = ilmax(2:end-1);
assignin('base', 'Ylmax', Ylmax);
assignin('base', 'ilmax', ilmax);
assignin('base', 'Ylmin', Ylmin);
assignin('base', 'ilmin', ilmin);

if (length(Ylmax) >
length(Ylmin))
    Ylmax =
Ylmax(1:length(Ylmin));
end
if (length(ilmax) >
length(ilmin))
    ilmax =
ilmax(1:length(ilmin));

```

```

end

if (length(Ylmin) >
length(Ylmax))
    Ylmin =
Ylmin(1:length(Ylmax));
end
if (length(ilmin) >
length(ilmax))
    ilmin =
ilmin(1:length(ilmax));
end

[sortpw, sortOrder] =
sort(freq(ilmax));
[sortPW, sortOrder2] =
sort(freq(ilmin));
diff2 = Ylmax(sortOrder) -
Ylmin(sortOrder2);
d2 = max(diff);
semilogy(sortfreq, (diff2./d2),
'r.');
```

```

xlabel('Frequency
(Hz)', 'Color', [1 0 0]);
ylabel('Normalized
Power', 'color', [1 0 0]);
title('Power Spec
Response', 'Color', [1 0 0])
end

```

### For the simple cell response:

```

function returnvariable =
plotresponse(X,Z)
%% Need to import one file for
RESPONSE current; for RESPONSE
only!
% assign X & Z.
% X = input time data; Z = response
voltage data
% For simple neuron model, plots
the power spec and current waveform
% This function automatically gets
rid of last 40000 points for X;1
point for Z
X=X(1:end-40000); Z=Z(1:end-1);
%% in order to get the frequency
freq = ((0.0001*X)/(2*pi))*(1000);
%% the x and y data for the current
waveform

% Create figure
figure1 = figure;

% Create axes
axes('Parent',figure1);
box('on');
hold('all');
plot(freq,Z);
xlabel('Frequency (Hz)', 'Color', [1
0 0]);
ylabel('Voltage (mV)', 'Color', [1 0
0]);
title('Soma Response', 'Color', [1 0
0]);

% Creates a max/min plot of the
current waveform response
[Zmax, imax, Zmin, imin] =
extrema(Z);
Zmax = Zmax(2:end-1);
Zmin = Zmin(2:end-1);
imax = imax(2:end-1);
assignin('base', 'Zmax', Zmax);

```

```

    assignin('base','imax',imax);
    assignin('base','Zmin',Zmin);
    assignin('base','imin',imin);

    if (length(Zmax) >
length(Zmin))
        Zmax =
Zmax(1:length(Zmin));
    end
    if (length(imax) >
length(imin))
        imax =
imax(1:length(imin));
    end

    if (length(Zmin) >
length(Zmax))
        Zmin =
Zmin(1:length(Zmax));
    end
    if (length(imin) >
length(imax))
        imin =
imin(1:length(imax));
    end

    [sortfreq, sortorder] =
sort(freq(imax));
    [sortFreq, sortorder2] =
sort(freq(imin));
    diff = Zmax(sortorder) -
Zmin(sortorder2);
    d = max(diff);
    plot(sortfreq, (diff./d),
'r.');
```

xlabel('Frequency (Hz)', 'Color', [1 0 0]);  
ylabel('Normalized Voltage', 'Color', [1 0 0]);  
title('Soma Response', 'Color', [1 0 0]);

[1 0 0];  
% the x data and y data for the power spec

```

Z1 = abs(fft(Z));

% Create figure
figure1 = figure;

% Create axes
axes('Parent',figure1);
box('on');
hold('all');
plot(freq,Z1);

xlabel('Frequency (Hz)', 'Color', [1 0 0]);
ylabel('Normalized Power', 'color', [1 0 0]);
title('Power Spec Response', 'Color', [1 0 0])
end

xlabel('Frequency (Hz)', 'Color', [1 0 0]);
ylabel('Power', 'Color', [1 0 0]);
title('Power Spectrum of Response Current Waveform', 'Color', [1 0 0]);

% Creates a max/min plot of the power response
[Z1max, ilmax, Z1min, ilmin] =
extrema(Z1);
Z1max = Z1max(2:end-1);
Z1min = Z1min(2:end-1);
ilmax = ilmax(2:end-1);
assignin('base','Z1max',Z1max);
assignin('base','ilmax',ilmax);
assignin('base','Z1min',Z1min);
assignin('base','ilmin',ilmin);

    if (length(Z1max) >
length(Z1min))
        Z1max =
Z1max(1:length(Z1min));
    end
    if (length(ilmax) >
length(ilmin))
        ilmax =
ilmax(1:length(ilmin));
    end

    if (length(Z1min) >
length(Z1max))
        Z1min =
Z1min(1:length(Z1max));
    end
    if (length(ilmin) >
length(ilmax))
        ilmin =
ilmin(1:length(ilmax));
    end

    [sortpw, sortOrder] =
sort(freq(ilmax));
    [sortPW, sortOrder2] =
sort(freq(ilmin));
    diff2 = Z1max(sortOrder) -
Z1min(sortOrder2);
    d2 = max(diff);
    semilogy(sortfreq, (diff2./d2),
'r.');
```

xlabel('Frequency (Hz)', 'Color', [1 0 0]);  
ylabel('Normalized Power', 'color', [1 0 0]);  
title('Power Spec Response', 'Color', [1 0 0])  
end

## For the complex cell:

```
function returnvariable =
plotneuron(X, Y, Z)
% X = timevec; Y = ivec; Z =
voltvec

%% For the complex neuron with h-,
sodium, potassium channels, etc.!
% This function plots the input
current, the voltage response at
the soma,
% the power spectrum, and the
normalized max/min plots.

%% in order to get the frequency
freq = ((0.0001*X)/(2*pi))*(1000);
%% Input Current and Voltage
Response

    % create the plot for the INPUT
current waveform
    plot(freq, Y);
    xlabel('Frequency
(Hz)', 'Color', [1 0 0]);
    ylabel('Amplitude', 'Color', [1
0 0]);
    title('Input Current', 'Color',
[1 0 0]);

    % create the plot for the
current waveform
    plot(freq, Z);
    xlabel('Frequency
(Hz)', 'Color', [1 0 0]);
    ylabel('Voltage (mV)', 'Color',
[1 0 0]);
    title('Soma Response', 'Color',
[1 0 0]);

    %creates the max/min plot of
the current waveform
    %use extrema.m file
    [Zmax, imax, Zmin, imin] =
extrema(Z);
    assignin('base', 'Zmax', Zmax);
    assignin('base', 'imax', imax);
    assignin('base', 'Zmin', Zmin);
    assignin('base', 'imin', imin);

    if (length(Zmax) >
length(Zmin))
        Zmax =
Zmax(1:length(Zmin));
    end

        if (length(imax) >
length(imin))
            imax =
imax(1:length(imin));
        end

        if (length(Zmin) >
length(Zmax))
            Zmin =
Zmin(1:length(Zmax));
        end
        if (length(imin) >
length(imax))
            imin =
imin(1:length(imax));
        end
        [sortfreq, sortorder] =
sort(freq(imax));
        [sortFreq, sortorder2] =
sort(freq(imin));
        diff = Zmax(sortorder) -
Zmin(sortorder2);
        d = max(diff);
        plot(sortfreq, (diff/d), 'r-');
        xlabel('Frequency
(Hz)', 'Color', [1 0 0]);
        ylabel('Normalized
Voltage', 'Color', [1 0 0]);
        title('Soma Response', 'Color',
[1 0 0]);
        %% Creates the power spec &
normalized max/min plot of the
power spec

        %z = abs(fft(Z));

        %create the plot for the power
spec
        %semilogy(freq, z);
        %xlabel('Frequency
(Hz)', 'Color', [1 0 0]);
        %ylabel('Power', 'Color', [1 0
0]);
        %title('Power Spec Current
Waveform', 'Color', [1 0 0]);

        %[zmax, izmax, zmin, izmin] =
extrema(z);
        % zmax = zmax(2:end-1);
        % zmin = zmin(2:end-1);
        % izmax = izmax(2:end-1);
        % assignin('base', 'zmax', zmax);
```



```

%
assignin('base','izmax',izmax);
%   assignin('base','zmin',zmin);
%
assignin('base','izmin',izmin);
%
%   if (length(zmax) >
length(zmin))
%       zmax =
zmax(1:length(zmin));
%   end
%   if (length(izmax) >
length(izmin))
%       izmax =
izmax(1:length(izmin));
%   end
%   if (length(zmin) >
length(zmax))
%       zmin =
zmin(1:length(zmax));
%   end

%   if (length(izmin) >
length(izmax))
%       izmin =
izmin(1:length(izmax));
%   end
%
%   [sortpw, sortOrder] =
sort(freq(izmax));
%   [sortPW, sortOrder2] =
sort(freq(izmin));
%   diff2 = zmax(sortOrder) -
zmin(sortOrder2);
%   d2 = max(diff);
%   %semilogy(sortfreq,
(diff2./d2), 'r. ');
%   xlabel('Frequency
(Hz)', 'Color', [1 0 0]);
%   ylabel('Normalized
Power', 'color', [1 0 0]);
%   title('Power Spec
Response', 'Color', [1 0 0])
end

```