Homework 3. Cable Theory

Hui Shi (added later bc of anonymous grading)
Based on Professor Winslow's lectures
EN.580.422 - System Bioengineering 2

February 22, 2018

Exercise 1. Consider the following bipolar neuron that has a soma and two dendrites. A synaptic input with conductance value G_e and reversal potential E is applied at the end of each dendrite. Assume that these synaptic conductances are sufficiently large that they set membrane potential at the site of application to the value E (another way of stating large G_e is $G_e >> G_\infty$. Assume the dendrites are identical and in the absence of synaptic input, have sealed end boundary conditions. The input conductance of each dendrite in the absence of synaptic input, looking towards the sealed-end, is G_{in} . The input conductance of a semi-infinite version of these cables is G_∞. For simplicity assume $G_{soma} = G_\infty$. Assume the neuron will fire if soma potential exceeds E/4.

Your task is to determine upper and lower bounds on the dendritic electrotonic length L such that this neuron implements an AND gate. That is, L must be large enough such that if only one synaptic input is applied, the neuron will not fire an action potential. At the same time, L must be small enough that the neuron will fire an action potential if two synaptic inputs are applied simultaneously. Perform a steady-state analysis. Note there is no closed-form analytical solution for the bounds on L, they must be determined numerically.

Answer : Consider two situations where in the first case there is only one synapse input and the second two synaptic inputs.

In the first case, assume G_e on the left branch is on and G_e on the right branch is off. The problem statement allows us assume $G_{soma} = G_\infty$. At the junction of the two dendrites and soma, since G_{out} dichotomizes into the soma(G_∞) and the right branch(G_{in}), we can derive from Kirchoff's law that $G_{out} = G_\infty + G_{in}$. Since the right branch doesn't have synaptic input, it is a sealed end condition. Therefore, the input conductance of the right branch is $G_{in} = G_\infty tanh(L)$. $\therefore G_{out} = G_{in} + G_\infty = G_\infty(tanh(L) + 1)$.

The voltage attenuation is:

$$\frac{V(L)}{V(0)} = \frac{1}{cosh(L) + (1 + tanh(L))sinh(L)}$$

Given that V(0)=E,

$$V(L) = \frac{E}{cosh(L) + (1 + tanh(L))sinh(L)}$$

Soma potential has to be lower than E/4 without invoking action potential. Therefore,

$$\frac{E}{cosh(L) + (1 + tanh(L))sinh(L)} < \frac{E}{4}$$
$$\frac{1}{cosh(L) + (1 + tanh(L))sinh(L)} < \frac{1}{4}$$

The plot and codes generated in this case is attached in the end of this document. The lower bound of the electrotonic length L is approximately 1.0847 for this relationship to be satisfied.

In the second scenario with the two synaptic inputs, since the system is linear, we can apply the superposition principle where we set one of the sources to zero and the other to E and calculate the soma voltage. In this case, we have:

$$G_{in} = G_\infty \frac{\frac{G_e}{\infty} + tanh(L)}{1 + \frac{G_e}{G_\infty} tanh(L)} = G_\infty \frac{\frac{G_e}{\infty} + tanh(L)}{1 + \frac{G_e}{G_\infty} tanh(L)} \frac{\frac{G_\infty}{G_e}}{\frac{G_\infty}{G_e}} = G_\infty \frac{1 + \frac{G_\infty}{G_e} tanh(L)}{\frac{G_\infty}{G_e} + tanh(L)}$$

Since $\frac{G_\infty}{G_e} = 0$,

$$G_{in} = G_\infty (\frac{1 + 0}{0 + tanh(L)}) = \frac{G_\infty}{tanh(L)}$$
$$\therefore G_{out} = G_{in} + G_\infty = G_\infty (\frac{1}{tanh(L)} + 1)$$

The voltage attenuation in this case is:

$$\frac{V(X)}{V(0)} = \frac{cosh(L - X) + \frac{G_{out}}{G_\infty} sinh(L - X)}{cosh(L) + \frac{G_{out}}{G_\infty} sinh(L)}$$
$$V(L) = \frac{E}{cosh(L) + (\frac{1}{tanh(L)} + 1)sinh(L)}$$

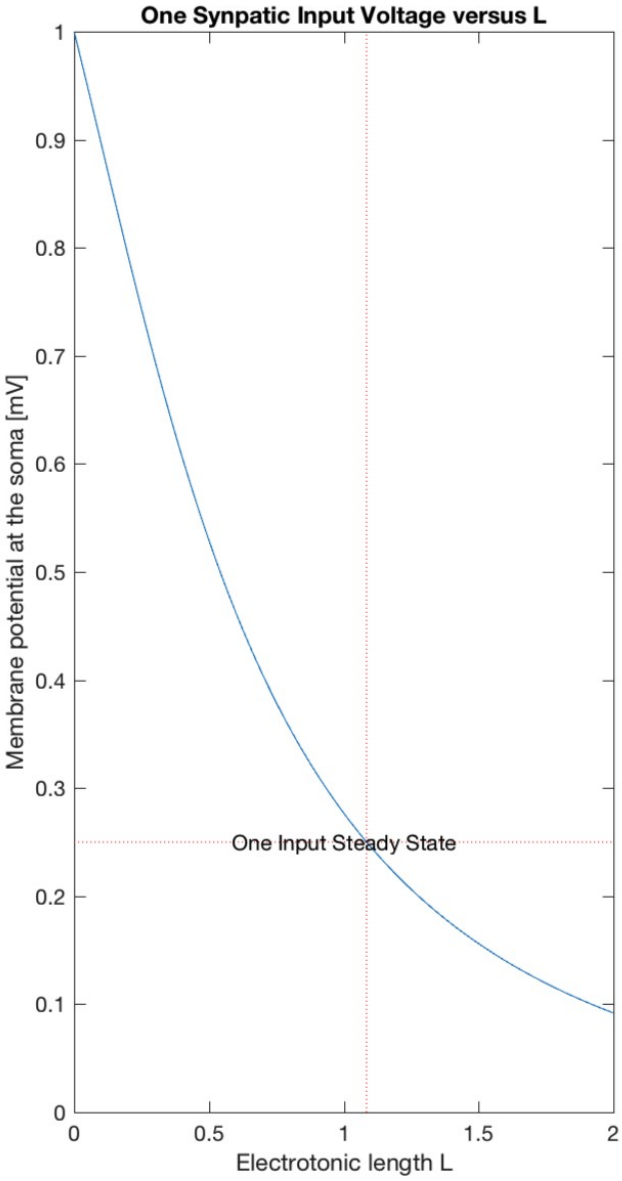
Then, applying the superposition principle on the linear system, we can add the two synaptic inputs together.

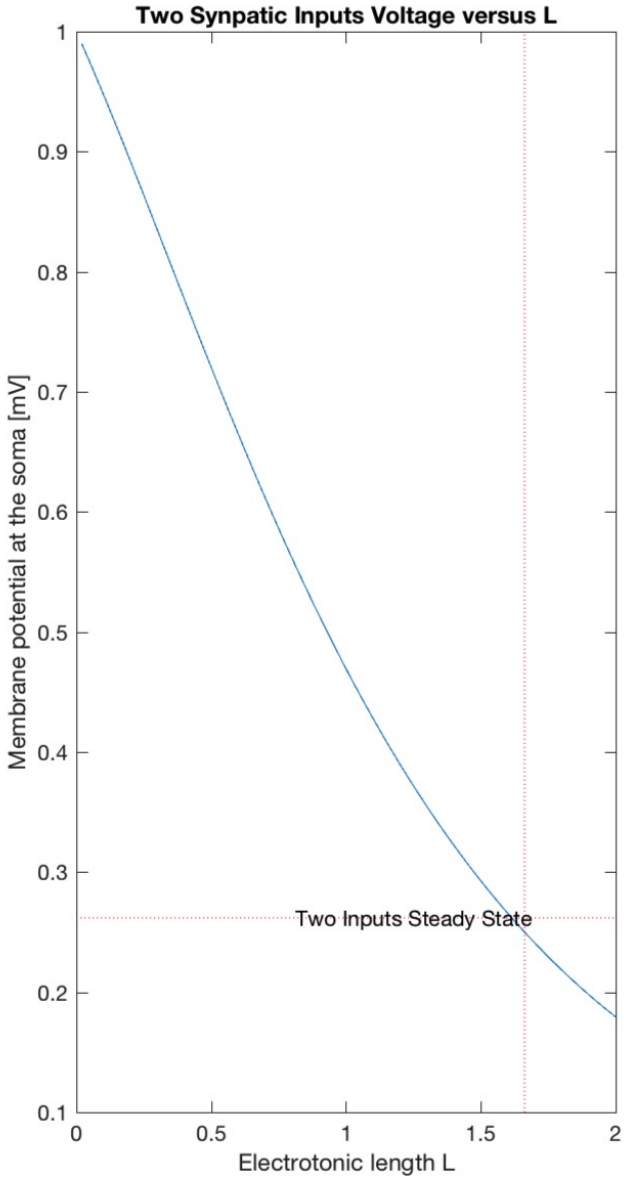
$$V_1(L) = \frac{2E}{cosh(L) + (\frac{1}{tanh(L)} + 1)sinh(L)} > \frac{E}{4}$$
$$\therefore \frac{1}{cosh(L) + (\frac{1}{tanh(L)} + 1)sinh(L)} > \frac{1}{8}$$

The plot and codes generated in the second case is also attached in the end of this document. The upper bound of the electrotonic length L is approximately 1.662 for this relationship to be satisfied.

In conclusion, the bounds on L is $1.0847 < L < 1.662$. This means that the synaptic input must be sufficiently far from the soma so that one input alone cannot generate an action potential. However, if the inputs are too far from the soma, action potential cannot be produced either.

Appendix





```
%% Problem 1
y1=@(L) cosh(L) + (1+tanh(L)) * sinh(L);
y2=@(L) cosh(L) + (1+1/tanh(L)) * sinh(L);
x0=0.001;
one_input=fzero(@(L) y1(L)-4,x0);
two_inputs=fzero(@(L) y2(L)-8,x0);
l = linspace(0,2,100); % Distance from soma

disp([num2str(one_input), ' < L < ',num2str(two_inputs)])

voltage1 = NaN(100,1);
voltage2 = voltage1;
for i = 1:100
voltage1(i) = (y1(l(i))).^-1;
voltage2(i) = 2* (y2(l(i))).^-1;
end

voltage11 = (y1(one_input)).^-1;
voltage22 = 2*(y1(two_inputs)).^-1;

figure(1);
subplot(121)
plot(l,voltage1,'DisplayName','One Input')
xlabel('Electrotonic length L'), ylabel('Membrane potential at the
soma [mV]')
vline(one_input);hline(voltage11);
text(one_input-0.5,voltage11,'One Input Steady State')
title('One Synpatic Input Voltage versus L')

subplot(122)
plot(l,voltage2,'DisplayName','Two Inputs')
xlabel('Electrotonic length L'), ylabel('Membrane potential at the
soma [mV]')
vline(two_inputs);hline(voltage22);
text(two_inputs-0.85,voltage22,'Two Inputs Steady State')
title('Two Synpatic Inputs Voltage versus L')
saveas(figure(1),'hw2.png')
```

====hline=====

```
function hhh=hline(y,in1,in2)
% function h=hline(y, linetype, label)
%
% Draws a horizontal line on the current axes at the location
specified by 'y'.  Optional arguments are
% 'linetype' (default is 'r:') and 'label', which applies a text label
to the graph near the line.  The
% label appears in the same color as the line.
%
% The line is held on the current axes, and after plotting the line,
the function returns the axes to
```

```
% its prior hold state.
%
% The HandleVisibility property of the line object is set to "off", so
not only does it not appear on
% legends, but it is not findable by using findobj.  Specifying an
output argument causes the function to
% return a handle to the line, so it can be manipulated or deleted.
Also, the HandleVisibility can be
% overridden by setting the root's ShowHiddenHandles property to on.
%
% h = hline(42,'g','The Answer')
%
% returns a handle to a green horizontal line on the current axes at
y=42, and creates a text object on
% the current axes, close to the line, which reads "The Answer".
%
% hline also supports vector inputs to draw multiple lines at once.
For example,
%
% hline([4 8 12],{'g','r','b'},{'l1','lab2','LABELC'})
%
% draws three lines with the appropriate labels and colors.
%
% By Brandon Kuczenski for Kensington Labs.
% brandon_kuczenski@kensingtonlabs.com
% 8 November 2001

if length(y)>1  % vector input
    for I=1:length(y)
        switch nargin
        case 1
            linetype='r:';
            label='';
        case 2
            if ~iscell(in1)
                in1={in1};
            end
            if I>length(in1)
                linetype=in1{end};
            else
                linetype=in1{I};
            end
            label='';
        case 3
            if ~iscell(in1)
                in1={in1};
            end
            if ~iscell(in2)
                in2={in2};
            end
            if I>length(in1)
```

```
                linetype=in1{end};
            else
                linetype=in1{I};
            end
            if I>length(in2)
                label=in2{end};
            else
                label=in2{I};
            end
        end
        h(I)=hline(y(I),linetype,label);
    end
else
    switch nargin
    case 1
        linetype='r:';
        label='';
    case 2
        linetype=in1;
        label='';
    case 3
        linetype=in1;
        label=in2;
    end



    g=ishold(gca);
    hold on

    x=get(gca,'xlim');
    h=plot(x,[y y],linetype);
    if ~isempty(label)
        yy=get(gca,'ylim');
        yrange=yy(2)-yy(1);
        yunit=(y-yy(1))/yrange;
        if yunit<0.2
            text(x(1)+0.02*(x(2)-
x(1)),y+0.02*yrange,label,'color',get(h,'color'))
        else
            text(x(1)+0.02*(x(2)-x(1)),y-
0.02*yrange,label,'color',get(h,'color'))
        end
    end

    if g==0
    hold off
    end
    set(h,'tag','hline','handlevisibility','off') % this last part is
so that it doesn't show up on legends
```

```
end % else

if nargout
    hhh=h;
end
```

====vline====

```
function hhh=vline(x,in1,in2)
% function h=vline(x, linetype, label)
%
% Draws a vertical line on the current axes at the location specified
by 'x'.  Optional arguments are
% 'linetype' (default is 'r:') and 'label', which applies a text label
to the graph near the line.  The
% label appears in the same color as the line.
%
% The line is held on the current axes, and after plotting the line,
the function returns the axes to
% its prior hold state.
%
% The HandleVisibility property of the line object is set to "off", so
not only does it not appear on
% legends, but it is not findable by using findobj.  Specifying an
output argument causes the function to
% return a handle to the line, so it can be manipulated or deleted.
Also, the HandleVisibility can be
% overridden by setting the root's ShowHiddenHandles property to on.
%
% h = vline(42,'g','The Answer')
%
% returns a handle to a green vertical line on the current axes at
x=42, and creates a text object on
% the current axes, close to the line, which reads "The Answer".
%
% vline also supports vector inputs to draw multiple lines at once.
For example,
%
% vline([4 8 12],{'g','r','b'},{'l1','lab2','LABELC'})
%
% draws three lines with the appropriate labels and colors.
%
% By Brandon Kuczenski for Kensington Labs.
% brandon_kuczenski@kensingtonlabs.com
% 8 November 2001

if length(x)>1  % vector input
    for I=1:length(x)
        switch nargin
        case 1
            linetype='r:';
            label='';
```

```
        case 2
            if ~iscell(in1)
                in1={in1};
            end
            if I>length(in1)
                linetype=in1{end};
            else
                linetype=in1{I};
            end
            label='';
        case 3
            if ~iscell(in1)
                in1={in1};
            end
            if ~iscell(in2)
                in2={in2};
            end
            if I>length(in1)
                linetype=in1{end};
            else
                linetype=in1{I};
            end
            if I>length(in2)
                label=in2{end};
            else
                label=in2{I};
            end
        end
        h(I)=vline(x(I),linetype,label);
    end
else
    switch nargin
    case 1
        linetype='r:';
        label='';
    case 2
        linetype=in1;
        label='';
    case 3
        linetype=in1;
        label=in2;
    end



    g=ishold(gca);
    hold on

    y=get(gca,'ylim');
    h=plot([x x],y,linetype);
```

```
    if length(label)
        xx=get(gca,'xlim');
        xrange=xx(2)-xx(1);
        xunit=(x-xx(1))/xrange;
        if xunit<0.8
            text(x+0.01*xrange,y(1)+0.1*(y(2)-
y(1)),label,'color',get(h,'color'))
        else
            text(x-.05*xrange,y(1)+0.1*(y(2)-
y(1)),label,'color',get(h,'color'))
        end
    end

    if g==0
    hold off
    end
    set(h,'tag','vline','handlevisibility','off')
end % else

if nargout
    hhh=h;
end
```