Utrecht University

Bachelor thesis

Computational modeling of the shunting effect in a single neuron

Author:
Lotte Spanjers

Supervisors:
Dr. Corette Wierenga
Dr. Gerhard Blab

January 18, 2017

Utrecht University

Abstract

Using the NEURON simulation environment, I examine the effect of the location of synapses on the shunting effect of the signal. I do this by first off recreating the results of C. Koch's, T. Poggio's and V. Torre's "Nonlinear interactions in a dendritic tree: localization, timing, and role in information processing" [3] and look at the conditions for which these results hold. I find that on path shunting is only more effective than shunting at the location of excitation, when the peak conductance of the excitatory synapse and membrane resistivity are sufficiently high. Then I try to recreate the results of A. Gidon's and I. Segev's "Principles governing the operation of synaptic inhibition in dendrites" [4]. I do not succeed in measuring the input resistance while the synapses are firing, and therefore cannot recreate their results.

Contents

1 Introduction **3**

2 Background Theory **3**
2.1 Neuroscience 4
2.2 NEURON 6

3 Previous results **6**
3.1 By Koch et al. 6
3.2 By Gidon and Segev 7

4 Experimental Methods and Results **9**
4.1 Model Structure 9
4.2 Results 9

5 Discussion **22**

6 Conclusions **24**

7 Further Research **24**

A
Appendix A **26**

B
Appendix B **31**

1 Introduction

In neuroscience a lot of research is done by experimenting. One can research what the effect of certain proteins or neurotransmitters is on the growth of synapses by experiment. Or what happens with boutons when nearby excitatory synapses are stimulated. There can even be measured what happens to the soma potential when a current is injected in the dendrite. But some things are hard to research by just experimenting, because you cannot control and change things as precisely as you would want to. At such moments it is easier to make a computational model. There are already a lot of ways to model a neuron, of which I am going to use a couple in a computational program called NEURON. One of the things that are easier to research by modeling, is the effect of a single synapse on the signal as received by the soma, which is what I am going to do.

The locations where an axon and a dendrite are connected through inhibitory synapses are sometimes quite locally positioned, and sometimes spread over a large part of the dendrite tree. This brings to question what the effect of location is on the shunting effect of these synapses. Is it more effective to place them close to each other, or spread them out widely? This may be dependent on the shape of the dendrite. To research this, it is important to know how the position a single inhibitory synapse affects the signal of an excitatory synapse. When we know this, we can see what more synapses will do, and maybe even make an analytic model. I am going to look at the shunting effect of an inhibitory synapse: Can I reproduce the results out of certain papers in the program NEURON? And how stable are these results in my simulation? I will do this by first off learning how to make models of simple neurons in the program NEURON. At first, I will look at the effect of the place of the excitatory synapse(s) on the signal as recorded in the soma. Then I will try to reproduce the results of earlier papers by Koch et al.[3] and Gidon and Segev[4] in my simulation and find out on which conditions these results are valid. Both these papers concentrate on the shunting effect of synapses and how the placing of these synapses influences this effect.

2 Background Theory

Before a physicist can work in biophysics, one should study some biology. This thesis falls in the domain of neuroscience, which we will explore in the following section. We should also look more into the program in which this model is made: NEURON. In this section I will introduce all one should know to read and understand what I am doing in this thesis.

2.1 Neuroscience

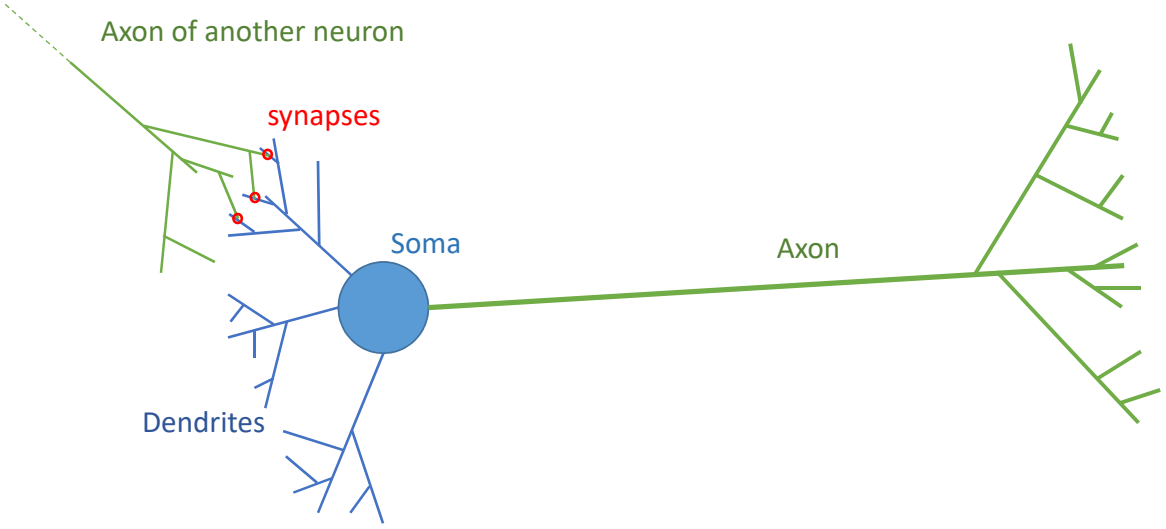


Figure 1: A schematic representation of a neuron. The direction of the signal is to the right.

The way a brain works is through electrical and chemical signals in neurons. A neuron consists of a soma, an axon and dendrites (see figure 1). The electrical signals are carried by the following ions: K^+, Na^+, Ca^+ and Cl^-. Between neurons, signals are passed through synapses. In this thesis we will focus on the passage of the electrical signal from the synapse in a dendrite to the soma. In rest, the whole neuron is in its resting membrane potential. There are many different kinds of neurons, each with their own typical resting potential. Most neurons have a resting potential between $-100mV$ and $0mV$. This resting potential can be depolarized and hyperpolarized by synaptic inputs. When the resting potential of the neuron is sufficiently depolarized, there is a spike induced, which we call an action potential. When a action potential is induced, synapses in the axon of this neuron will be activated and pass the signal through synapses to another dendrite. The movement of ions in the neuron can be described by laws and relations of diffusion, drift, mobility and space-charge neutrality. Because the signal passed is an electrical one, we can also learn more about this by making an equivalent circuit model.

There are roughly two kinds of synapses, inhibitory and excitatory synapses. A reversal potential of a synapse is the potential at which a neurotransmitter of that synapse will not cause a net current flow of ions through that ion channel. Inhibitory synapses have a reversal potential equal to or less than the resting potential of the dendrite, and can therefore hyperpolarize the neuron. Excitatory synapses have a reversal potential higher than the resting potential of the dendrite, and this creates a depolarization. The excitatory synapses have a reversal potential around the $0mV$. The inhibitory synapse can never induce an potential. When the inhibitory synapse fires, it does change the conductance and can therefor shunt a signal from another synapse.

There are different ways to simulate a neuron, but one of the most used is the cable model. Here we assume that the dendrite is passive and is shaped like a cable, with one sealed end and the other end connected to the soma. Being passive means that the cable has a certain membrane resistance, r_m, a membrane conductivity, c_m and an axial resistance, r_i, which are linear and uniform throughout the dendrite. We also assume the current flow is in just one dimension; along the cable. To calculate the way this action-potential travels through the dendrite we have Rall's cable equation [2]:

$$\frac{1}{r_i}\frac{\partial^2 V_m}{\partial x^2} = c_m\frac{\partial V_m}{\partial t} + \frac{V_m}{r_m} \tag{1}$$

With x the distance (cm), t the time ($msec$), V_m the membrane potential (mV), r_i the axial resistance (Ω/cm), r_m the membrane resistance (Ωcm) and c_m the membrane capacitance (F/cm). These relate to cable specific parameters as follows:

$$R_i = \pi a^2 r_i \tag{2}$$

$$R_m = 2\pi a r_m \tag{3}$$

$$C_m = \frac{c_m}{2\pi a} \tag{4}$$

With a the radius of the dendrite, R_i the specific intracellular resistivity (Ωcm), R_m the specific membrane resisitivity (Ωcm^2) and C_m the specific membrane capacitance (F/cm^2).

With this equation the simplest models of neurons are made. The easiest way to think about this model, is to picture it as a water hose with the water pressure as voltage. An inhibitory synapse can shunt the signal of an excitatory synapse by making a hole in the hose, it does this by lowering the input resistance locally.

We can make the model more physically accurate by adding simulations of ionchannels, which allows the ions to move trough the membrane, down their concentration gradient. This makes the model active, and adds sometimes non-linear compounds. These channels are influenced by the transmembrane voltage. The channels are in their open state with a probability of a and closed with a probability of $(1 - a)$. There are two types of these channels. If the membrane of one of these is depolarized, the energy available for switching from open to closed decreases, and therefore the probability of the gate being open increases. Hyperpolarization of the membrane will do the opposite to this channel. The other type of channel works in the exact opposite way, and is activated by membrane hyperpolarization. When the potential of a dendrite comes above a certain threshold, the channels respond by opening or closing, and thereby letting extra negatively loaded ions in, and thereby depolarizing the cell further. This happens rapidly and induces a spike in the membrane potential. After this spike, there is a short period of hyperpolarizing which makes is impossible for the signal spike to travel backwards.

In the soma we use the Hudgkin-Huxley model, or the HH-model. With this model Hudgkin and Huxley described the behavior of conduction, nerve excitation

and Na^+ and K^+ channels. They came up with eleven equations called the Hodgkin and Huxley equations which they solved numerically. They pictured the ion channels as parallel conductance channels in a electric circuit, and because the channels were controlled by gating particles they could come up with equations using gating variables and maximum conductances. Then they experimentally determined the precise equations and found those fit the experimental data really well. These equations can be found in Foundations of Cellular Neurophysiology [2]. You can use this HH-model in NEURON really easily, as it is already programmed in and you can call it with a simple commando.

2.2 NEURON

The NEURON simulation environment is a environment where one can construct and apply mechanisms to simulate the working of neurons and neural networks. I use the C-based language .hoc to write my code, this language is developed especially for the NEURON program. I used The Neuron Book [1] to learn to use this program, and subscribed to the NEURON forum for more background information. In NEURON I was able to create an active soma with a passive dendrite and measure for instance the peak voltage at the soma. The program can run simulations and numerically calculate voltage, input resistance and such, but some a lot easier than others.

3 Previous results

3.1 By Koch et al.

In their groundbreaking paper of 1983 [3], Koch et al. showed that inhibition on path can be more effective than inhibition at the place of excitation by solving the system of Volterra integral equations. They computed the F-factor, which is the ratio of the maximum of the somatic depolarization without inhibition to the somatic depolarization with inhibition.

$$\text{F factor} = \frac{\Delta V_{\text{soma, without inhibition}}}{\Delta V_{\text{soma, with inhibition}}} \tag{5}$$

They found that sometimes inhibition on path is more effective than inhibition at the place of excitation. They said that when g_e and g_i are small, the optimal location of inhibition is at the location of the excitatory synapse and, when g_e and/or R_m increase, the optimal location moves along the direct path toward the soma[3]. This was an important discovery, because prior to this publication it was thought that inhibition at the place of excitation was always the most effective. The conditions, under which they showed this was the case, were as follows: The membrane resistance $R_m = 14k\Omega/cm^2$, the conductance peak of the excitatory synapse $g_e = 100nS$, the membrane capacity $C_m = 1\mu F/cm^2$, and the input resistance $R_i = 70\Omega cm$.

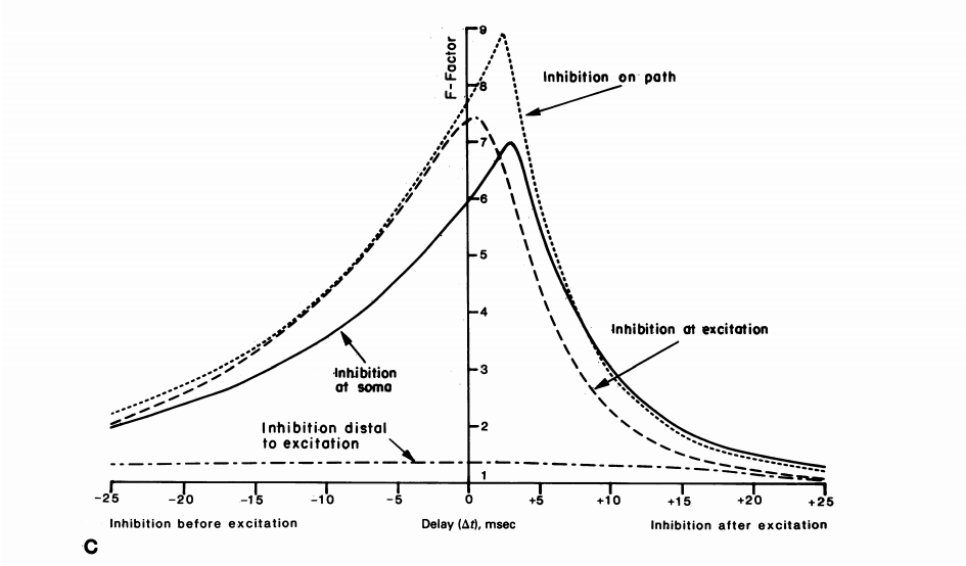


Figure 2: The F-factor with $R_m = 14k\Omega/cm^2$, $g_e = 100nS$, as calculated and shown by Koch et al. [3] They show on the x-axis the delay of the inhibitory synapse relative to the excitatory synapse.

3.2 By Gidon and Segev

In Principles Governing the Operation of Synaptic Inhibition in Dendrites by Gidon and Segev [4], it is shown that in certain cases, off path inhibition is more effective than on path inhibition. They show this by making a numerical simulation of an isopotential soma with a sealed-end dendrite and putting a hotspot at $X = 0.6L$. A hotspot in their case is a spot where twenty NMDA synapses, which are a type of excitatory synapses, are firing at a rate of $20Hz$. In the supplementary information they show that the same conclusions can be reached with a single, very strong, synapse, and this is what I used. They use in their figures and further analytic calculations the shunt-level, SL, which measures how much the inhibitory synapse shunts the excitatory synapse. They define the shunt-level at d as follows:

$$SL_d = \frac{\Delta R_d}{R_d} \tag{6}$$

with R_d the input-resistance (Ω) at d and ΔR_d the difference in input-resistance (Ω) at d due to a synaptic conductance perturbation.

With this definition they plot the SL as a function of the distance of the inhibitory synapse to the excitatory synapse. They show that the shunt level can be higher off path than on path at the same distance from the hotspot, which means off path inhibition can be more effective than on path inhibition. In their supplementary information they show how different electrotonic lengths influence the SL, although the conclusion stays the same.

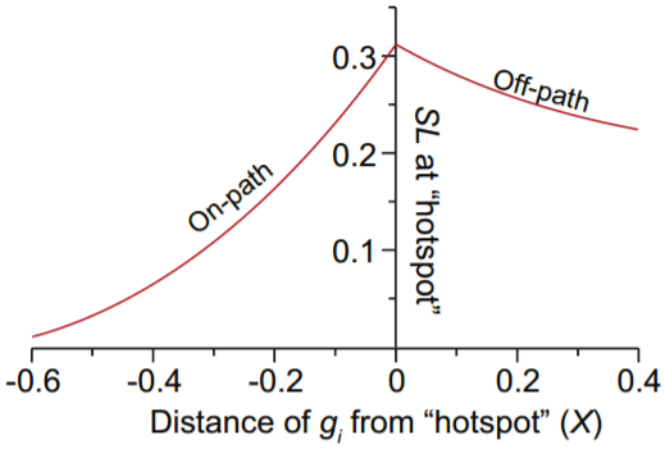


Figure 3: The value of SL at the hotspot in a ball and stick model as a function of the distance of the inhibitory synapse from the hotspot as depicted by Gidon and Segev [4]. Off-path inhibition attenuates less steeply compared to the respective on-path inhibition.

They also show that when inhibitory synapses are placed at the same spot on different branches, the shunt level is higher at the junction of these branches than at the place of inhibition when enough, in their case eight, branches are used. This number may vary when varying the parameters, for which they used: the conductance perturbation $g_i = 1nS$, and the distance from each of the perturbations to the junction $X = 0.4$. For two synapses they could calculate this condition. The shunt level at the junction is higher than at one of the synapse loci, i or j, when SL^j_{junction} is larger than SL^i_i. For more than two synapses they didn't calculate this condition and they concluded that for two synapses, the best shunting place is always at the location of one of the synapses. They also note that where the shunt level may be higher at the junction, but in passive models the voltage change is always highest at one of the synaptic loci.

They also calculate the shunt level at d, (SL_d), for two conductance perturbations, i.e. two synapses:

$$SL_d = \frac{SL^i_d + SL^j_d - 2SL^j_j SL^i_i A_{i,j} A_{j,d} A_{d,i}}{1 - SL^j_j SL^i_j} \tag{7}$$

with SL^a_b the shunt level at b due to a conductance perturbation at a (dimensionless), and $A_{c,d}$ the steady voltage attenuation, $(\frac{V_j}{V_i})$, from c to d (dimensionless). The shunt level at d due to a single conductance perturbation at i can be calculated as follows:

$$SL^i_d = [\frac{g_i R_i}{1 + g_i R_i}] A_{i,d} A_{d,i} \tag{8}$$

with i the location of the conductance perturbation, g_i the conductance perturbation at i, R_i the input resistance at i, and $A_{i,d}$ as defined before.

4 Experimental Methods and Results

4.1 Model Structure

As mentioned before, I am using NEURON's simulation environment to simulate neurons. I do this by first loading the neuron GUI library so I can use the pre-programmed functions. Then I set up the general shape of the neuron by creating the soma and dendrites and connecting them. These I give some biophysical properties like length, diameter, membrane resistance, number of segments, etc.. Thereafter, I declare functions, proc's, and data storages and I set the simulation controls. Finally, I run the simulation while I let NEURON store the data to .dat files, which I can open and process in Mathematica. In the appendix you can find two of the models I made. In appendix A you can find the code for the nine branches experiment (figure 8), and in appendix B you can find the code I used to calculate the shunt level (figure 19).

4.2 Results

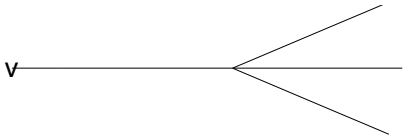


Figure 4: The shape of the neuron used in the first simulations: the isopotential soma on the left and one branched dendrite. The soma is spherical and has a diameter of $30\mu m$. All the branches have a length of $100\mu m$, a diameter of $1\mu m$

To get a feeling for NEURON, we begin with a simple simulation where I placed one excitatory synapse on different locations on a neuron with an isopotential soma and a passive branched dendrite with sealed ends. I ran a simulation for each location, measured the potential at the soma, and plotted the maximums (figure 5). You notice that, when the excitatory synapse is close enough to the soma, it produces a spike in the action-potential. Furthermore, there seems to be a slight discontinuity at the place where the child dendrites are connected to the parent dendrite. I thought this may be this way because all dendrite parts have the same diameter, and therefore do not follow the power rule for dendrites as told by the Rall model:

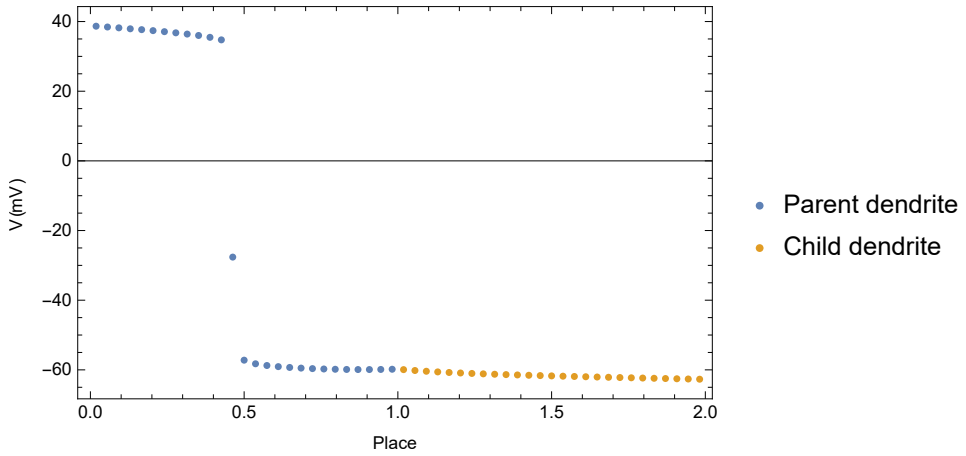
$$(d_{\text{parent}})^{\frac{3}{2}} = \sum (d_{\text{child}})^{\frac{3}{2}} \tag{9}$$

with d the diameter. With this power rule a semi-infinite branched dendrite can be simplified to a mathematically and electrically equivalent semi-infinite cable. For finite cables the extra assumption is made that they all end at the same electrotonic length. The electrotonic length at which dendrites end can be calculated in Rall's model by the following equation [2]:

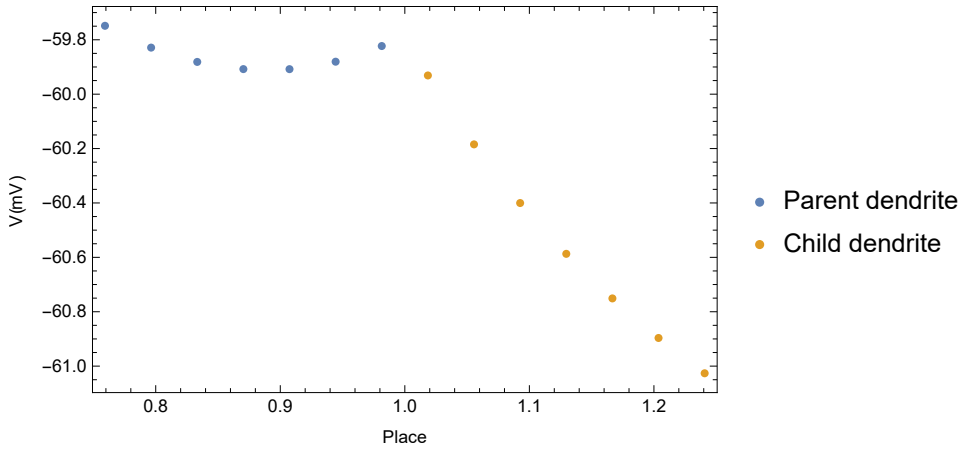
$$L = \frac{l_{parent}}{\lambda_{parent}} + \frac{l_{child}}{\lambda_{child}} \tag{10}$$

$$\lambda = \sqrt{\frac{dR_m}{4R_i}} \tag{11}$$

with L the electrotonic length (dimensionless), l the physical length (μm), d the diameter of the dendrite (μm), and λ the length or space constant (μm). With these equations I calculated that the equivalent cable should, in this case, be $100 + 100 \cdot 3^{1/3} \mu m$ long.



(a) Overview



(b) A zoomed in picture of the transition from parent branch to child branch

Figure 5: The potential as measured in the soma for different places of an excitatory synapse. The place is the relative place in the dendrite, with the places of the child dendrite added to 1 to ensure easy reading. The excitatory synapse has a reversal potential of $0mV$, and peak conductance of $0.03\mu S$

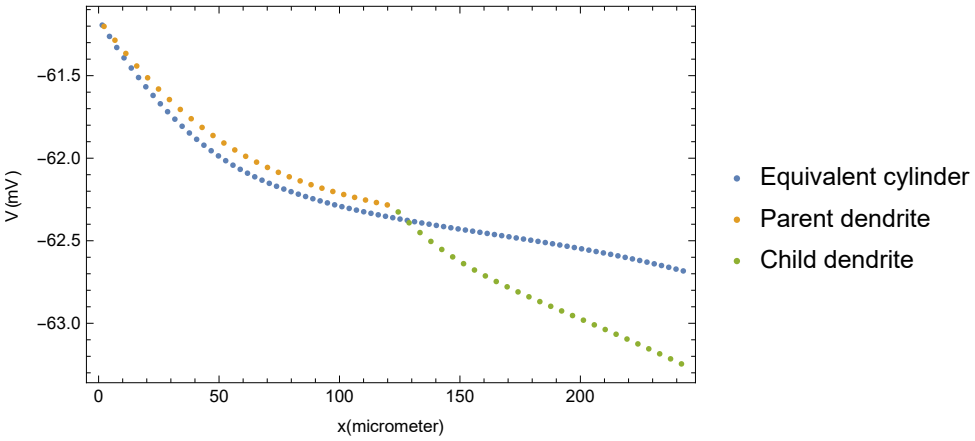


Figure 6: The potential as measured in the soma for different places of an excitatory synapse. The location x is measured from the soma. The excitatory synapse has a reversal potential of $0mV$, and peak conductance of $0.03\mu S$. In this simulation the diameter of the parent dendrite is changed from $d = 1\mu m$ to $d = 3^{2/3}\mu m$. The equivalent cylinder has the same properties as the parent and child dendrites, but a length of $100 + 100 \cdot 3^{1/3}\mu m$ and a diameter of $3^{2/3}\mu m$.

The bump in the potential becomes much smaller, but does not totally disappear when I change the diameter of the parent dendrite to $3^(2/3)\mu m$ (see figure 6), so the diameters of the dendrite parts cannot be the sole reason for this bump. The equivalent cable I simulate shows almost the same results, until I place the excitatory synapse on one of the child dendrites. When the excitatory synapse is placed on the child dendrite, the potential of the anatomically correct model suddenly drops, whereas the potential of the equivalent cylinder does not show this sudden drop.

To look at the effect of multiple synapses, we place two excitatory synapses on different locations on the child dendrites of the same model and let one excitatory synapse 'walk' over the dendrites.

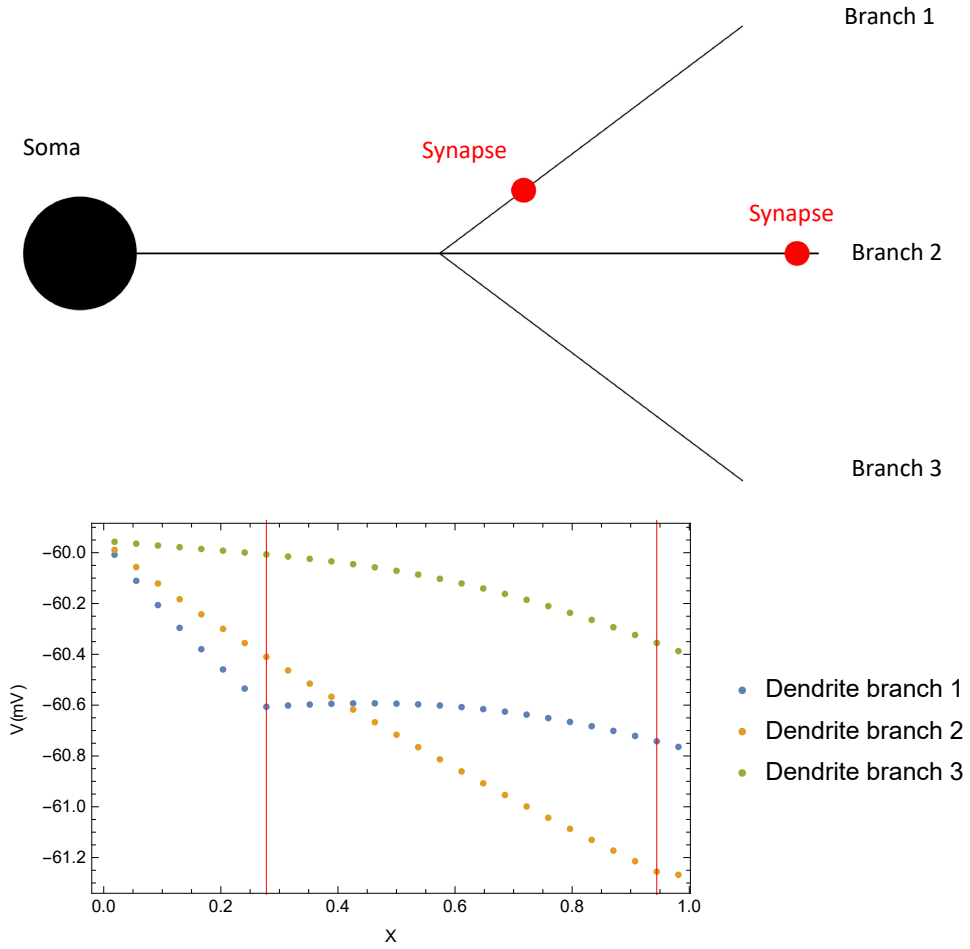


Figure 7: The maximums of the voltage as measured at the soma. The place X is where an excitatory synapse is placed on the branch of the dendrite, $0 \leq X \leq 1$ with 0 the zero end of the branch that is connected to the parent branch, and 1 the sealed end. There are also two excitatory synapses placed as seen above (the red dots), at 5/18 on the first branch and 17/18 at the second branch (red lines in the plot). The resting potential is set at $-65mV$. All the synapses are the same: the onset is at $0.5ms$, they excite the membrane to $0mV$, their peak conductance change is $0.01\mu S$, and time constant τ is $0.1ms$.

You can easily see in figure 7 that excitatory synapses on different branches of a dendrite can be more effective than when the synapses are placed on the same branch. This is partly because when they are on the same branch, they shunt each others signal, and therefore the whole signal gets less effective.

To look more closely at the shunting, we can do the same simulation again, but with an inhibitory synapse to 'walk' over the dendrites. The inhibitory synapse itself does not create a difference in the resting potential of the soma or dendrites, so the shunting effect is the only effect created. The shunting effect is the highest when the potential as measured at the soma is the lowest.

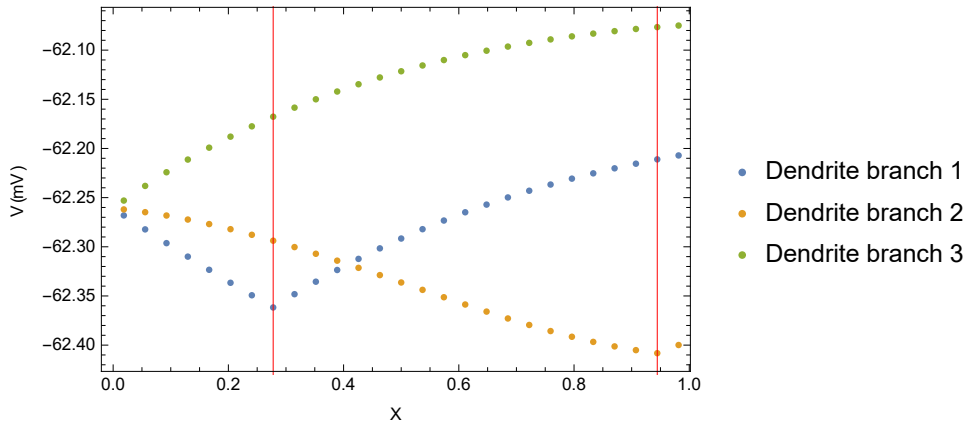


Figure 8: The maximums of the voltage as measured at the soma. The place X is where an inhibitory synapse is placed on the branch of the dendrite. The two excitatory synapses are placed as before, at 5/18 on the first branch and 17/18 at the second branch (red lines in the plot). The resting potential is set at $-65mV$, which is also the reversal potential of the inhibitory synapse. Without the inhibitory synapse, the maximal potential as measured at the soma is $V = -61.9637mV$.

In figure 8 you can see that the shunting effect of the inhibitory synapse is the most effective when placed at the same spot as the excitatory synapse. But you can also see, when looking very closely, that at the end of branch 2, the off path shunting is less effective than the on path shunting at the same distance from the excitatory synapse. This is the opposite of what Gidon and Segev have shown in their paper. It is also interesting to think about the fact that apparently, it is more effective to shunt a more distally located excitatory synapse than a more closely located one. Because that is when the best shunting effect is reached, by placing an inhibitory synapse at the same spot as the most distally placed excitatory synapse, as seen in dendrite branch 2 in figure 8. Placing the inhibitory synapse on the end of the branch without any synapses, still has a significant effect. The effect of this is $\Delta V = -61.964mV + 62.075mV = 0.111mV$.

To further explore the effects of multiple synapses, I wondered what would happen when I gave the dendrite more branches and put an excitatory synapse on a different place on each of the branches. I expected to see one dip in the maximum potential on each branch, at the spot of the excitatory synapse. This proved to be wrong, as there were more effects at work.

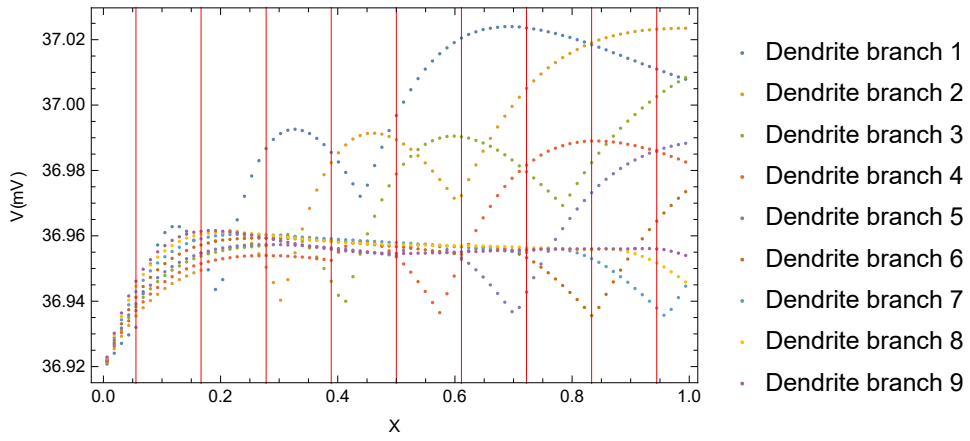


Figure 9: The maximums of the voltage as measured at the soma. The place X is where a inhibitory synapse is placed on the branch of the dendrite. There is an excitatory synapse on each branch at $\frac{\text{number branch}+1}{2}/18$ (the red lines in the plot, the first is on branch 1, the second on branch 2, etc.), with the same strength as before. The resting potential is set at $-65mV$, which is also the reversal potential of the inhibitory synapse. All the synapses have a peak conductance of $g_{\max} = 0.01\mu S$

The potential in figure 9 is a lot higher than in the first two figures. This is the case, because with so many excitatory synapses an action-potential is generated. More interesting is that there are multiple dips in this potential per branch, not only at the place of excitation. The maximum shunt is at the 0-end of all branches, which seems like an effect that Gidon and Segev [4] have also shown, where the shunting level of multiple inhibitory synapses is higher on the junction of different dendrites than on the place of the synapses themselves. Furthermore, the dips in voltage after this first dip are not at the exact place of excitation, but after. This may have an similar reason. It also seems that the place where the inhibition is most effective on the 1-side of the branches is at the end around $X = 0.95$ of the 7th branch. It looks like this is the same result as before, that shunting more distally placed excitatory synapses is more effective than shunting the closely located ones. The location where the inhibitory synapse is the least effective is almost an intuitive one: we would expect that this is at the end of the first dendrite branch. This is almost true, except that it is not the end of the branch, but around $X = 0.7$. It seems like the pattern of multiple dips is the reason for this place of the minimum in shunting.

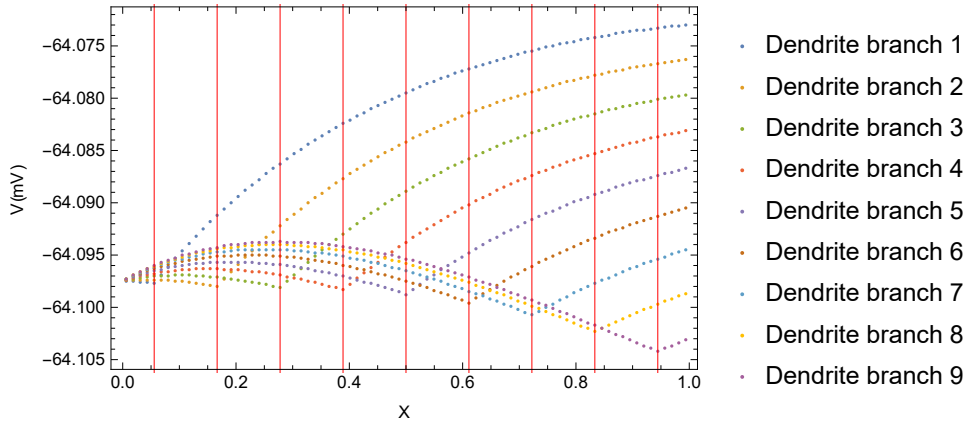


Figure 10: The maximums of the voltage as measured at the soma. The place X is where a inhibitory synapse is placed on the branch of the dendrite. There is an excitatory synapse on each branch at $\frac{\text{number branch}+1}{2}/18$ (the red lines in the plot, the first is on branch 1, the second on branch 2, etc.), with the same strength as before. The resting potential is set at $-65mV$, which is also the reversal potential of the inhibitory synapse. All the excitatory synapses have a peak conductance of $g_{\max} = 0.001\mu S$, and the inhibitory synapse has a peak conductance of $g_{\max} = 0.01\mu S$

These multiple potential dips per synapse disappear when we let the peak conductance of all excitatory synapses be ten times lower $g_{\max} = 0.001\mu S$, so no action potential is generated. The best place on each branch for the inhibitory synapse to shunt the signal is at the place of excitation. The best place for inhibition overall, is at the place of excitation on branch 9. This is again the most distally placed excitatory synapse, just like in figure 8.

Now we are somewhat familiar with the program NEURON and we can begin to simulate a neuron with the same physiological properties as Koch et al. and see if we can recreate their results: the fact that inhibition on path can be more effective than at the place of excitation. The first simulation I ran did not show this effect, inhibition at the location of the excitation was more effective than inhibition on path, and inhibition at the soma was the least effective. I used the same model as seen in figure 4.

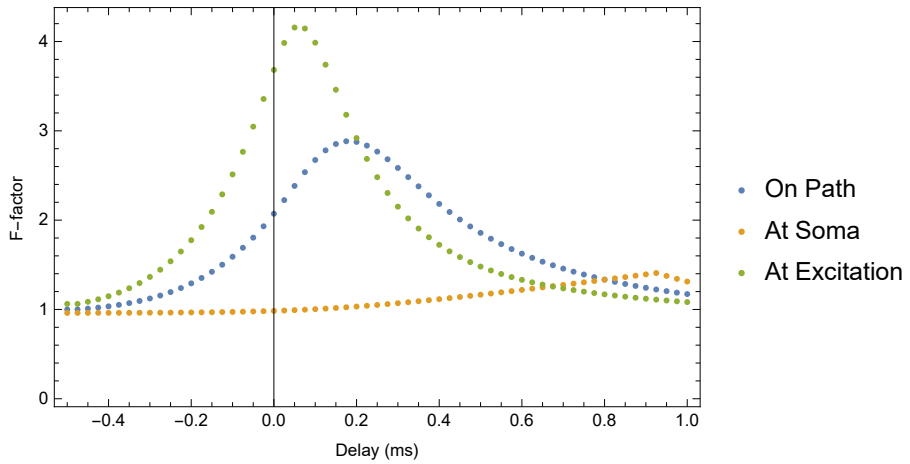


Figure 11: The F factor for different places of the inhibitory synapse. The excitatory synapse is located at $\frac{17}{18}$ of one of the child dendrites, the on path synapse is located at $\frac{5}{18}$. $R_m = 5k\Omega * cm$, $C_m = 1\mu F/cm^2$, $R_a = 100\Omega cm$.

A explanation for the different results is that I may have used slightly different parameters. Thus I altered several different parameters, and I got similar results when I increased the membrane resistance, $R_m = 14k\Omega/cm^2$, and the conductance peak of the excitatory synapse, $g_e = 100nS$. This makes sense, because when the resistance is higher, a conductance peak has more influence on the signal. Think of a water hose: with a higher membrane resistivity it is less leaky, then when you poke a hole it in, it will have a bigger effect than when the hose was leaky in the first place. When alternating other parameters, like the length of the dendrite, the cytoplasmic resistivity and the place of the excitatory synapse, the peaks of the F-factor did change, but the inhibition at the excitation kept being the most effective.

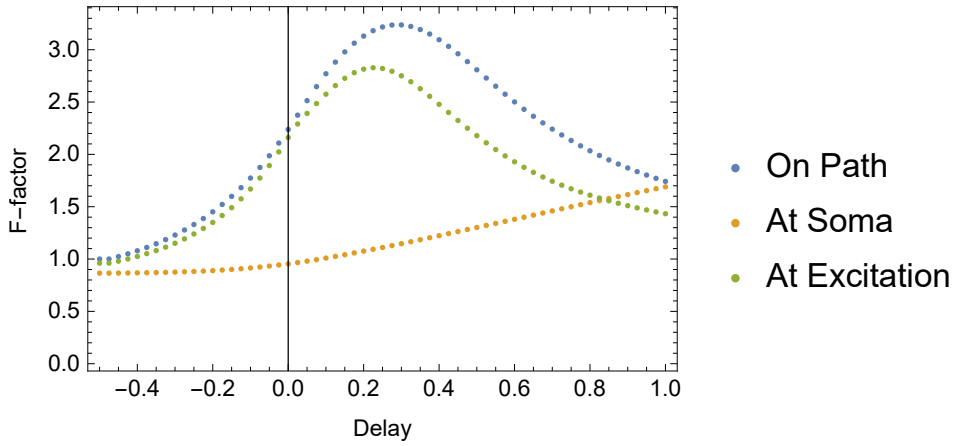


Figure 12: The F-factor with $R_m = 14k\Omega/cm^2$, $g_e = 100nS$

When I made a model to calculate the shunt factor like Gidon and Segev, I got different results than the results they showed in their paper. One of the biggest differences was that they found that the peak in the shunt factor was always precisely at the point of excitation, the peak I found was often off-path and

seemed to be asymptotic. I also found that my SL was much lower than theirs: while their maximum shunt level in one of the figures was around 0.3, mine kept being under 0.01. Also the 'peak' in the shunt level was a minimum, when we would expect a maximum (see figure 13). I tried to differ several parameters, but I never got the same or comparable results. I even found more asymptotic behavior and seemingly random peaks. It is hard to get exactly the same results, partly because I sometimes could not find which parameters Gidon and Segev used so I had to guess, and partly because NEURON did not facilitate all the models that they used. For example: One problem is that they use a semi-infinite cylindrical dendrite, and this is not possible to simulate in this numerical program.

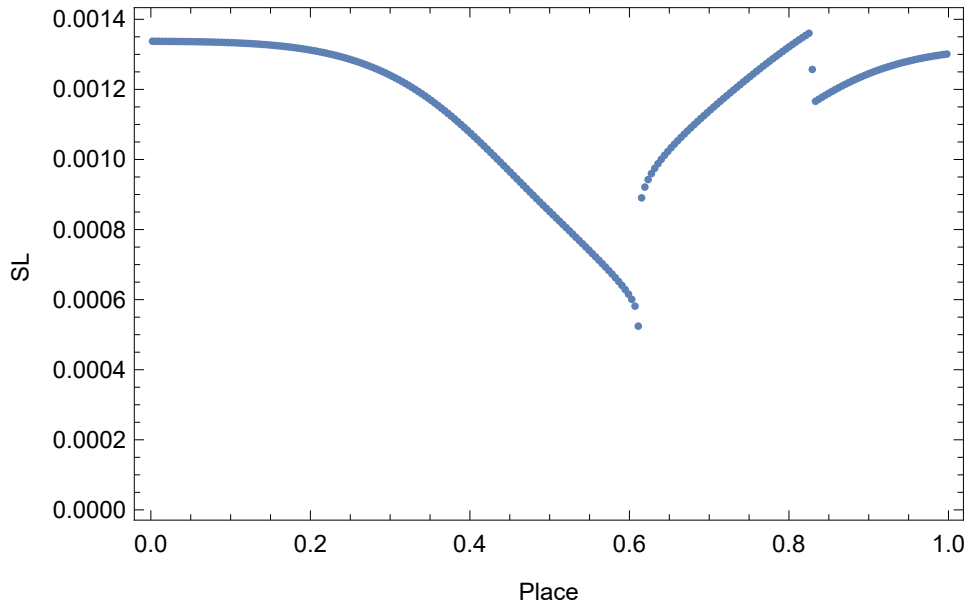
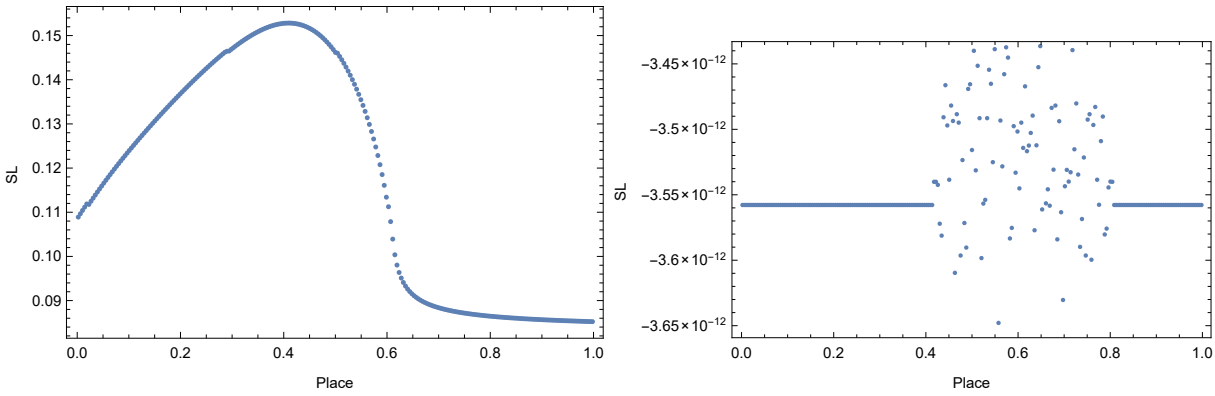


Figure 13: Shunt level at $\frac{11}{18}$ as a function of the place of one of the conductance perturbations. The other conductance perturbation is placed at $\frac{11}{18}$.

As we look at figure 13, we see that there are two abnormal points: around 0.61 and 0.82. The first case of asymptotic behavior coincides with the point where the excitatory synapse is placed, but the second one seems to be completely random, as there is no point process or anything located at that point, so it should not be any different from the others. Even if it coincided with something, we should not expect asymptotic behavior in a realistic model.
For these reasons I do not think I can trust my results, but I really do not know how to improve them. I have tried altering a lot of different simulation parameters, but it never got any closer. When, for example, I tried a much longer or shorter dendrite (my shorter dendrite was $100\mu m$ and my longer one $10000\mu m$), I got these two plots:



(a) for $L = 100\mu m$ (b) for $L = 10000\mu m$

Figure 14: The shunt level at the place of the excitatory synapse (X = 0.6L) as a function of the place of the inhibitory synapse.

In figure 14a you can see that on path shunting is much more effective than off path shunting in a short dendrite. This is the opposite of what Gidon and Segev showed. Other than that, this is at least a smooth function. From all the starting parameters I tried changing, the length of the dendrite gave the most interesting figures. In figure 14b you can see what happens when I elongate the dendrite enough, the simulation seems to be completely unstable around the place of the excitatory synapse, which may be caused by rounding errors. Furthermore, the shunt level is next to nothing and thus completely insignificant for the whole length of the dendrite, this happened even when I cranked up the maximum peak conductance of both synapses to $100nS$.

These discrepancies did not show in the input resistance I measured prior to putting the synapses on the dendrite. This input resistance was always a smooth function with the lowest value at the 0-end of the dendrite and the highest value at the 1-end. So I think something goes wrong when measuring the input resistance after putting the synapses in place, or possibly the results of Gidon and Segev do not hold up in this kind of simulation.

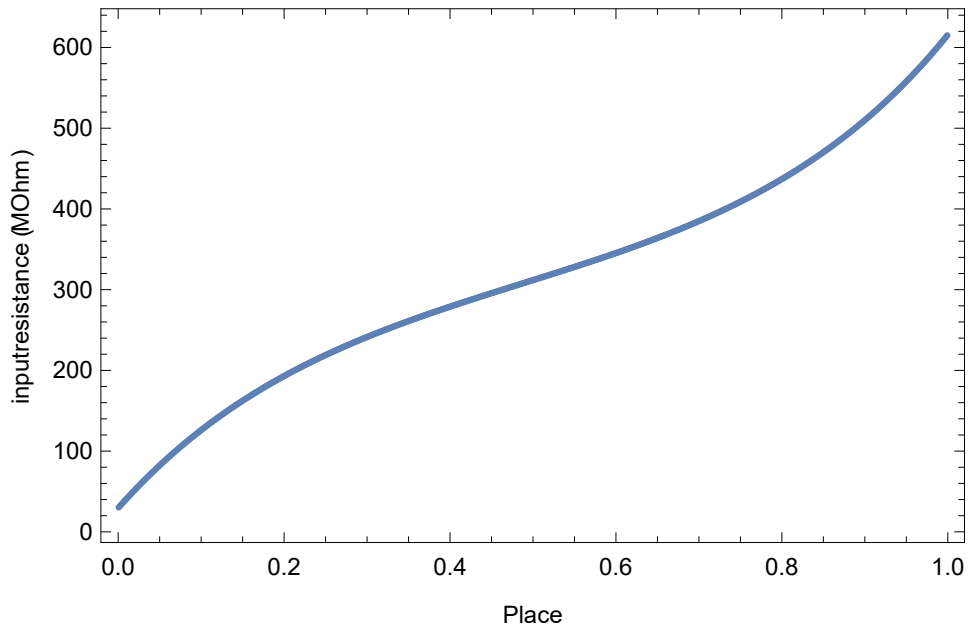


Figure 15: Input resistance as a function of place on the dendrite without any synapses.

The input resistance as a function of the location of the inhibitory synapse after putting the synapses on the dendrite is where weird things start happening. I think this may be some kind of rounding error, for the input resistance after putting the synapses on the dendrite is almost the same as the input resistance at $\frac{11}{18}$before putting the synapses on the dendrite, leading me to believe that while running a simulation, and thereby firing the synapses, NEURON cannot measure the input resistance.

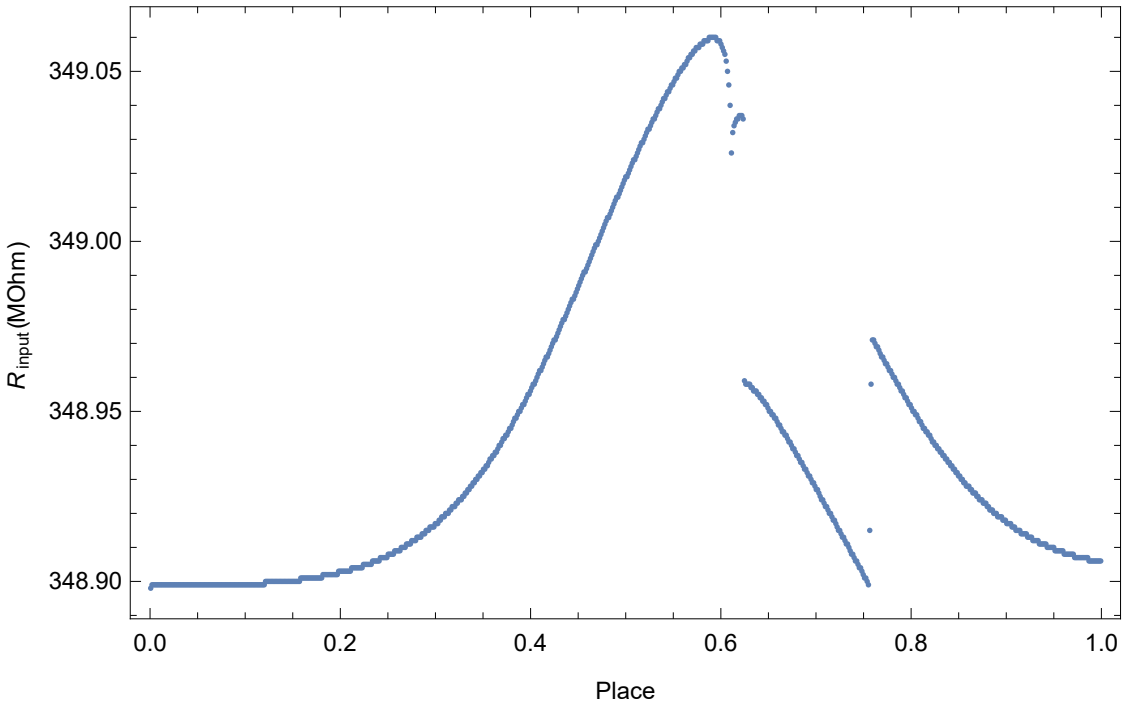
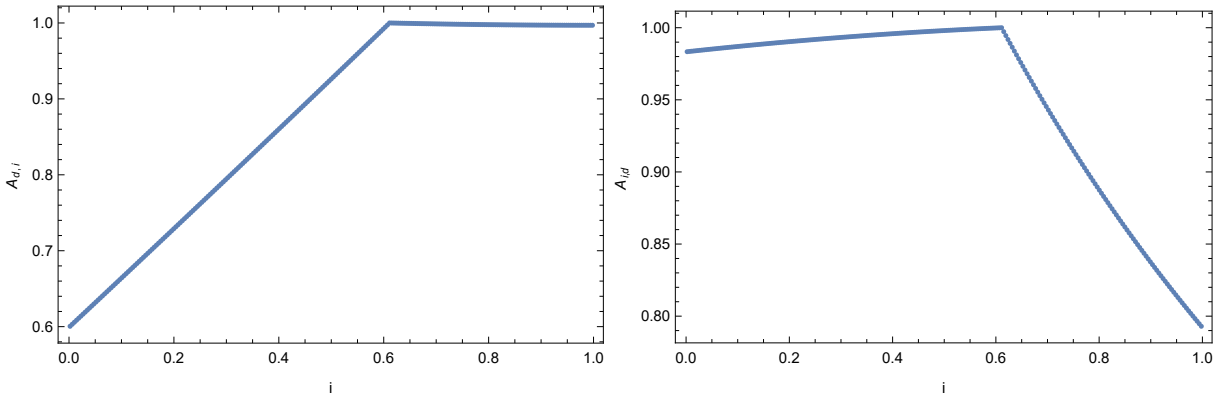


Figure 16: The input resistance at the place of excitation ($X = \frac{11}{18}$) as a function of the location of inhibitory synapse.

I also tried to use the equation that Gidon and Segev used to calculate the shunting level due to two perturbation levels (see equation 7). For this, I let NEURON measure the voltage attenuation $A_{d,i}$ and $A_{i,d}$ for a fixed $d = \frac{11}{18}$, and a varying i. The results of these are shown in figure 17. These values for the attenuation seem logical, for the voltage from the dendrite end to the soma attenuates for both more steeply than the other way around. Also the voltage attenuation for $d = i$ is one, which also makes sense as this is measured at the spot of injection.



(a) The voltage attenuation from d to i.

(b) The voltage attenuation from i to d.

Figure 17: The voltage attenuation for a fixed $d = \frac{11}{18}$ and a varying i.

Furthermore, I measured the input resistance for all i. This also seemed fine;

the input resistance closer to the soma is less then further along the dendrite and between the 100 and 300 $M\Omega$.

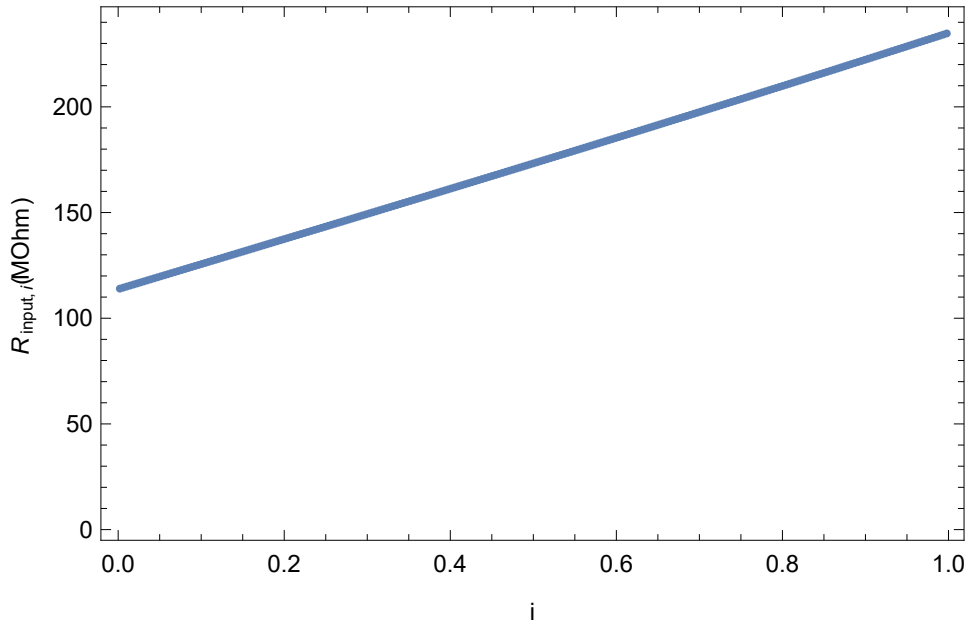


Figure 18: The input resistance at i, in the dendrite of the ball and stick model. The dendrite is $100\mu m$ long.

Figure 19 shows the calculated shunt level with the values for voltage attenuation and input resistance as shown in figures 17 and 18, and a peak conductance for both synapses of $g_{\max} = 1\mu S$. This is an unbelievably high shunt level, we expect it to be more in the range of 0 to 0.3, besides that we expect it to vary. This result didn't change significantly when, for example, the conductance perturbations ware set to be $g_{\max} = 0.1\mu S$ or the length of the dendrite was altered.

The shunt level looks more realistic when the second synapse is not taken into account. Still, where Gidon and Segev show that the shunt level stays under 0.3, ours climbs to almost 1, when the synapse is at the place of measuring. This may be a nudge in the right direction of where it went wrong, but I cannot see where exactly the calculation went wrong.

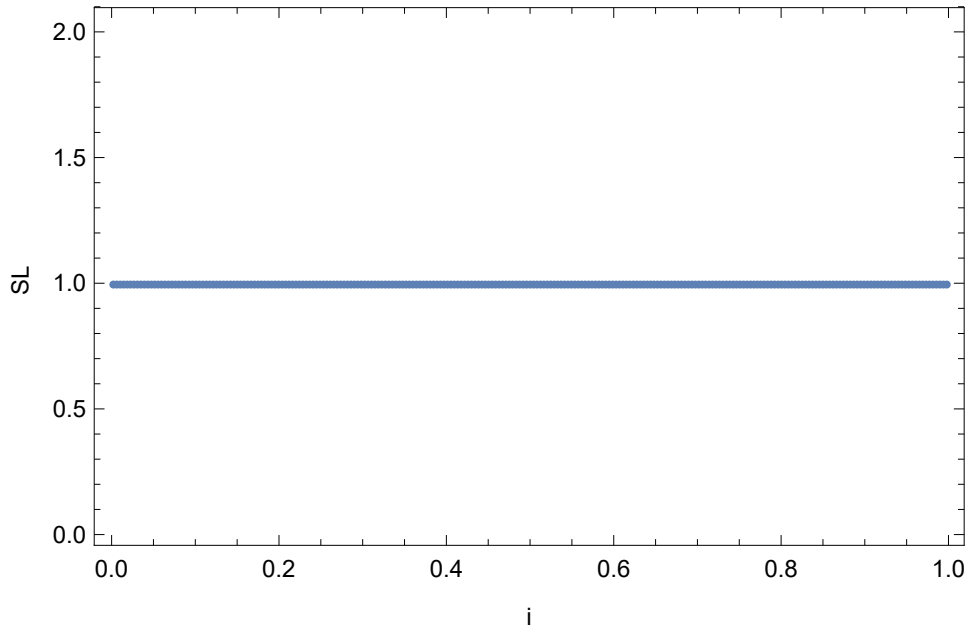


Figure 19: The shunt level as calculated by equation 7. The shunt level is constant at a value of $SL = 0.994727$.

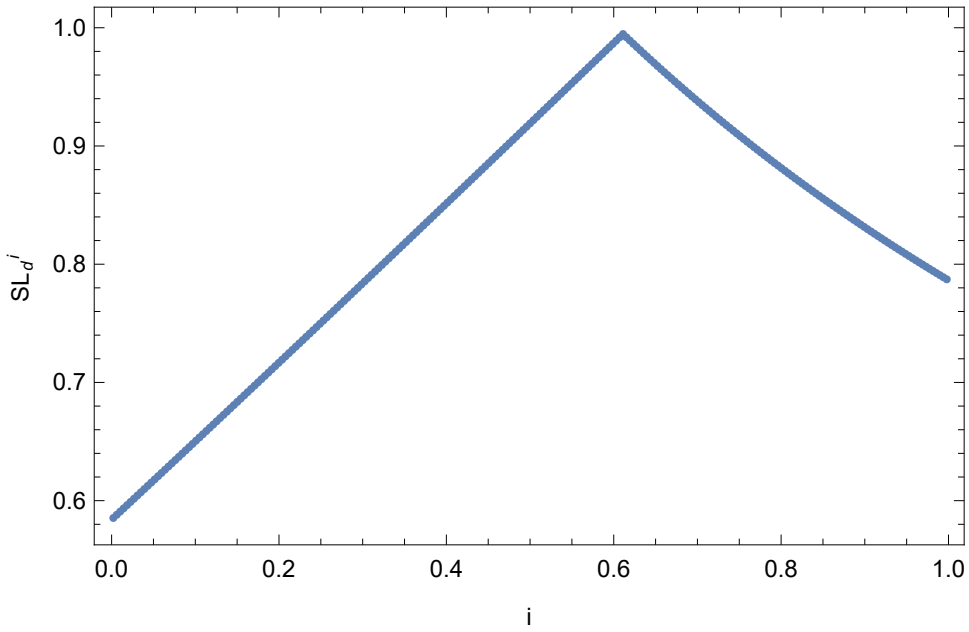


Figure 20: The shunt level at $X = \frac{11}{18}$ as a function of i, the place of the inhibitory synapse.

5 Discussion

The irregularity in figure 5b can be made smaller by satisfying the 3/2 power rule, see figure 6. It is not a rule of nature that dendrite branches follow this power rule, but, according to Johnson and Mio-Sin Wu [2], a number of studies

have suggested that in spinal motor neurons, cortical neurons, and hippocampal neurons the dendrite branching do seem to approximately follow this rule. But it was not true that by this rule we could find a equivalent cylinder which would give the same results as the branched dendrite. Maybe for this to work we should also alter the peak conductance of reversal potential of the synapse. It would make sense that, to make the whole neuron into an equivalent single cable model, we should also scale some parameters of the synapses. Also, we learned from looking at other models, that what happens on different branches influences each other, in sometimes anti-intuitive ways. This means that we can never catch the whole functionality of a dendrite tree by an equivalent cylinder model.

The result of a more branched dendrite as seen in figure 9 leads me to think that the maximum shunting effect, as seen in figure 8 and figure 10, may also not be exactly at the place of excitation, but this deviation is too small to be visible in the simulation I made. This means that synapses on other branches have a big effect on where the point of the best shunting effect is located. So when thinking about whether inhibition on a certain branch is the most effective on path, off path, or at the excitation, you cannot only consider the synapse on that branch. Every extra synapse we consider, even on another branch, has an effect on the most effective shunting place.

Evidently my results concerning the shunt levels calculated by measuring the input resistance with and without synapses do not seem that believable. I tried to fix this by altering the length of the dendrite, the peak conductances of both the excitatory and the inhibitory synapse, the membrane resistivity, the diameter of the dendrite, the number of segments in the dendrite, and the general ordering of my code. Nothing seemed to change my results in the direction of what Gidon and Segev found. Maybe there is something in the library of NEURON that works really different from what I expected, and therefore my use of it is wrong. Maybe the program NEURON just is not usable for the kind of simulation I wanted to make, but this seems rather an extreme conclusion to draw. Maybe I made an assumption in the model which I should not have made, but I could not find everything I needed in the paper (with supplementary information) I based this model on. The impedance class is a tool in NEURON with which you can measure the input resistance, voltage attenuation and such. I now suspect that this impedance class, which I use to measure the input resistance, cannot be used during a simulation, and therefore only measures the resistance when the synapses are inactive. This is obviously not what I wanted, and would also explain the really low shunt levels I found. It would, however, not explain the weird asymptotic behavior. This would mean that the way you measure the input resistance is different from, for example, the way you measure the membrane potential. For I can measure the membrane potential by running a simulation, and after the simulation ask NEURON for the membrane potential at a certain timepoint or the maximum potential etc.. I tried to use the '.record' command, which saves the values of a certain variable during a simulation, but then the input resistance seemed constant and unrelated to the place of the inhibitory synapse. The reason that NEURON can't measure the input resistance while at the same time running a simulation, may be that it's a voltage based model, in stead of a conductance based model.

The fact that even when I use the exact equation as Gidon and Segev and still don't get similar results, I cannot explain. I believe I use realistic parameters, and when possible the same as Gidon and Segev. I got the same results with a equation I simplified myself as with the un-simplified equation.

I would also not recommend any other researchers to try learning .hoc when trying to model neurons. I think it is easier to use Python if you want to use NEURON, because the documentation of this language is much larger and therefore it is easier to learn and use. I often could not find how to use the .hoc code or what certain commands really did. There are just two or three sources where you can find out more about the language .hoc and a lot more to find documentation of python, because it is more widely used. When you use a more documented and less specific language, the time I used digging deep in the NEURON forum to find out what exactly the right syntax is, can be used to construct some functionality yourself and thereby get a better understanding of how neurons work.

6 Conclusions

The calculated results of Koch et al. hold up in the simulation I set up in NEURON. These only hold when the excitatory peak conductance and the membrane resistivity are relatively high, which is also what Koch et al. found.

I have not succeeded to recreate the results of Gidon and Segev with a simulation I set up in NEURON. This is probably because I cannot find or do not understand some of the functionality of NEURON. Or because there is something in the equations of Gidon and Segev that needs alternate interpretation.

The location of the inhibitory synapse has an effect on the shunting of a signal from an excitatory synapse. The two ways we looked at shunting in this thesis, may not give the same results, i.e. looking at SL may give different conclusions that looking at the change in potential. But to say where the most effective place for shunting is, is probably dependent on the exact situation. For sometimes it will be most effective to shunt at the place of excitation, but adding a synapse in a whole other part of the dendrite tree might make it more effective to shunt slightly off path. The most notable finding of this thesis is that when synapses of the same strength are located on different branches of the passive dendrite tree, it is the most effective to place the inhibitory synapse at the place of the most distally located excitatory synapse (when no spike is generated).

7 Further Research

Interesting would be to see what effect adding ion-channels to the dendrites would have on the shunting effects and when they are most effective. One can do this in the program NEURON and add it to the simulation I already made (see for example appendix A).

It would clearly also be interesting to get the simulation to measure the shunt levels working, if necessary in another simulation environment. Here we can also

add active channels to measure these shunt levels and get a more complete and realistic picture of how neurons work.

References

[1] N.T. Carnevale, M.L. Hines: The Neuron Book, Cambridge University Press, 2005

[2] D. Johnston, S. Miao-Sin Wu.: Foundations of Cellular Neurophysiology, Massachusetts Institute of Technology, 1995

[3] C. Koch, T. Poggio, V. Torre: Nonlinear interactions in a dendritic tree: localization, timing, and role in information processing. Proc Natl Acad Sci USA. Vol. 80, pp. 2799-2802, May 1983

[4] A. Gidon, I. Segev: Principles governing the operation of synaptic inhibition in dendrites. Neuron. Volume 75, Issue 2, p330–341, 26 July 2012

A
Appendix A

```
load_file("nrngui.hoc")
load_file("shapebiophysics.hoc")
load_file("putsyn2inhibitory.hoc")
load_file("simulationcontrol.hoc")

//simulation control
dt = 0.025
tstop = 5
v_init = -65

objref dends
dends = new SectionList()
forsec "dend" dends.append()
forsec dends print secname()

objref location, amplitude
location = new Vector()     // stores locations along the dendrite
amplitude = new Vector()    // stores peak amplitude at each location

objref vm                   //makes a vector to store the data
vm = new Vector()
vm.record(&soma.v(0.5))     //sets location to record potential during simulation

objref data
data = new File()

//synapses

access dend[1]
objectvar syn1
syn1 = new AlphaSynapse(5/18)
syn1.onset =    0.5
syn1.tau =  0.1
syn1.gmax = 0.01
syn1.e =    0
syn1.i  =   0


access dend[2]
objectvar syn2
syn2 = new AlphaSynapse(17/18)
syn2.onset =    0.5
syn2.tau =  0.1
syn2.gmax = 0.01
```

```
syn2.e =    0
syn2.i =    0



forsec dends {
  print "in ", secname()
  for (x,0) {   // iterates over internal nodes of
                // currently accessed section
                // statements that use distance() to calculate
                // distance from x in currently accessed section
                // to a reference point, e.g. soma's 0 end
    putsyn(syn, x)      // move syn to x on currently accessed section
    run()               // execute a simulation
        vmmax = vm.max()            /* find maximum element in vm */
        amplitude.append(vmmax)     /* append this to amplitude vector */
        location.append(x)          /* append x to location vector */
    data.wopen("file4.1in.dat")         //open data file to write
    data.printf("location amplitude \n")
    data.printf("%d\n", location.size())
    for i=0, location.size()-1 {        //write all locations and amplitudes to datafile
        data.printf("%g %g\n", location.x(i), amplitude.x(i))
    }
    data.close()
    print " at ", x

  }
}
```

```
//simulation control
dt = 0.025
tstop = 5
v_init = -65

proc initialize(){
 finitialize(v_init)
 fcurrent()
}

proc integrate(){
 g.begin()
 while (t<tstop){
  fadvance()
  g.plot(t)
 }
 g.flush
}

proc go(){
 initialize()
 integrate()
}

```

```
load_file("nrngui.hoc")

//shape

ndend = 4  //number of dendrites

create soma, dend[ndend]

//connect dendrites and soma with the correct ends
connect dend[0](0), soma(1)
connect dend[1](0), dend[0](1)
connect dend[2](0), dend[0](1)
connect dend[3](0), dend[0](1)

topology() //show me how this looks

//biophysics

soma{
 L=30        //micrometers, Lenght
 diam=30     //micrometers, diameter
 nseg=1      //number of segments
 insert hh   //insert HH-model
}

dend[0]{
 L=100
 diam=1
 nseg=27
 insert pas      //insert passive model
 g_pas=0.0002    //conductance
 e_pas=-65       //mV, the leakage equilibrium potential
}

dend[1]{
 L=100
 diam=1
 nseg=27
 insert pas
 g_pas=0.0002
 e_pas=-65
}

dend[2]{
```

```
 L=100
 diam=1
 nseg=27
 insert pas
 g_pas=0.0002
 e_pas=-65
}

dend[3]{
 L=100
 diam=1
 nseg=27
 insert pas
 g_pas=0.0002
 e_pas=-65
}

forall{
 Ra=100      //Ohm cm, cytoplasmic resistivity
 cm=1        //uf/cm^2, specific membrane capacitance
}
```

```
/* putsyn.hoc for somatic epsp as a function of synaptic location
   last modified 7/15/99 NTC
*/

    // accessible to the Gather Values tool


synloc=0

objref syn
soma syn = new AlphaSynapse(0.5)

syn.onset = 0.5
syn.tau =   0.1
syn.gmax =  0.01
syn.e =     -65
syn.i   =   0

func putsyn() { local synloc
  if ($2 < 0 || $2 > 1) {
    print "ERROR--location must be in the range [0, 1]"
    synloc = -1
  } else {
    // say what we want
    synloc = $o1.loc($2)
    // find out what we got
    synloc = $o1.get_loc()
   /* Note: get_loc() pushes the section of the target point
        process onto the section stack, so that it becomes the
        currently accessed section.  We must restore the currently
        accessed section to what it was before get_loc(). */
    pop_section()
  }
   return synloc
}
```

B
Appendix B

```
load_file("nrngui.hoc")

//shape, this creates the shape of the neuron

ndend = 1   //number of dendrites
Rm = 10000 //Ohm cm, the membrane resistance

create soma, dend[ndend]

connect dend[0](0), soma(1) //here I connect the dendrites 0 end to the 1 end of the soma



topology() //with this command, the shape of the neuron will be shown when running your code

//biophysics

soma{
 L=30 //um      //Length
 diam=30 //um   //diameter
 nseg=1         //number of segments
 insert hh      //inserts the HH-model
}

dend[0]{
 L=100 //um
 diam=1 //um
 nseg=243
 insert pas     //inserts the passive model
 g_pas = 1/Rm   //conductance
 e_pas = -65    //mV, the leakage equilibrium potential
}

forall{
 Ra=100 //Ohm cm, cytoplasmic resistivity
 cm=1 //uf/cm^2, specific membrane capacitance
}


//data storage
objref location, SL, Aid, Adi, RNi, Aii, Aij, Aji
location = new Vector()     // stores locations along the dendrite
SL = new Vector()           // stores the calculated shunt levels
```

```
Aid = new Vector()          // stores the measured voltage attenuation levels A_(i,d)
Adi = new Vector()          // stores the measured voltage attenuation levels A_(d,i)
RNi = new Vector()          // stores the measured input resistance R_(N,i)
Aii = new Vector()          // stores the measured voltage attenuation levels A_(i,i)
Aij = new Vector()          // stores the measured voltage attenuation levels A_(i,j)
Aji = new Vector()          // stores the measured voltage attenuation levels A_(j,i)


objref data, data1, data2, data3
data = new File()
data1 = new File()
data2 = new File()
data3 = new File()


//impedance stukje. With zz I can use the impedance tools,
//which allow me to measure the input resistance (.input) and the voltage attenuation (.ratio)
objref zz
zz = new Impedance()

FREQ = 0 // Hz

proc calcZ() {
  dend[0] distance(0, $1)
  dend[0] zz.loc($1)  // sets origin for impedance calculations
  zz.compute(FREQ, 1) // takes the impedance contributions of
                      // gating state differential equations into account
                      // but requires mechanisms to be compatible with CVODE
}


access dend[0]
calcZ(11/18)

/*measure the voltage attenuation A_(i,d) and input resistance in dend[0] for all segments*/
for(x,0){ //for every segment in dend[0]
    attenuationid = zz.ratio(x)
    inputresistancei = zz.input(x)
    print attenuationid, inputresistancei
    Aid.append(attenuationid) /*append the attenuationid at this segment to the Aid vector*/
    RNi.append(inputresistancei) /*append the input resistance at this segment to the RNi vector*/
    location.append(x)  /* append x to location vector */
}
```

```
87
88  /*measure the voltage attenuation A_(d,i) in dend[0] for all segments*/
89  for(x,0){ //for every segment in dend[0]
90      calcZ(x)
91      attenuationdi = zz.ratio(11/18)
92      print attenuationdi
93      Adi.append(attenuationdi) /*append the attenuationdi at this segment to the Adi vector*/
94  }
95
96  /*measure the voltage attenuation A_(i,i) in dend[0] for all segments*/
97  for(x,0){ //for every segment in dend[0]
98      calcZ(x)
99      attenuationii = zz.ratio(x)
100     print attenuationii
101     Aii.append(attenuationii) /*append the attenuationii at this segment to the Aii vector*/
102 }
103
104 //Because we place our excitatory synapse at 11/18, this is the place we want to measure Rj
105 Rj = RNi.x(243*(11/18)+1/2)
106 print "Rj =", Rj
107 Ajj = Aii.x(243*(11/18)+1/2)
108 print "Ajj =", Ajj
109
110 calcZ(11/18)
111 Ajd = zz.ratio(11/18)
112 Adj = zz.ratio(11/18)
113
114 Aij = Aid
115 Aji = Adi
116
117 gi = 1
118 gj = 1
119
120 for(x,0){
121     SLid = ((gi * RNi.x(i))/(1 + gi * RNi.x(i))) * Aid.x(i) * Adi.x(i)
122     /*calculate SL^i_d*/
123     SLjd = (gj * Rj)/(1 + gj * Rj) * Ajd * Adj
124     /*calculate SL^j_d*/
125     SLjj = (gj * Rj)/(1 + gj * Rj) * Ajj * Ajj
126     /*calculate SL^j_j*/
127     SLii = ((gi * RNi.x(i))/(1 + gi * RNi.x(i))) * Aii.x(i) * Aii.x(i)
128     /*calculate SL^i_i*/
129     SLij = ((gi * RNi.x(i))/(1 + gi * RNi.x(i))) * Aij.x(i) * Aji.x(i)
```

```
    /*calculate SL^i_j*/
    shuntlevel = (SLid + SLjd - 2 * SLjj * SLii * Aij.x(i) * Ajd * Adi.x(i))/(1 - SLjj * SLij)
    /*calculate the shunting level at d for a synapse at i and d*/
    SL.append(shuntlevel)
    print shuntlevel
}


 /*write data files for generating graphics later*/
    //the Voltage Attenuation from i to d at all the internal nodes of dend[0]

data1.wopen("Segev1.14Aid.1.dat")
data1.printf("location A_(i,d) \n")
data1.printf("%d\n", location.size())
for i=0, location.size()-1 {
    data1.printf("%g %g\n", location.x(i), Aid.x(i))
}
data1.close()

    //the input resistance at all the internal nodes of dend[0]
data2.wopen("Segev1.14RN.1.dat")
data2.printf("location input resistance \n")
data2.printf("%d\n", location.size())
for i=0, location.size()-1 {
    data2.printf("%g %g\n", location.x(i), RNi.x(i))
}
data2.close()

    //the Voltage Attenuation from d to i at all the internal nodes of dend[0]

data3.wopen("Segev1.14Adi.1.dat")
data3.printf("location A_(d,i) \n")
data3.printf("%d\n", location.size())
for i=0, location.size()-1 {
    data3.printf("%g %g\n", location.x(i), Adi.x(i))
}
data3.close()

    //the Shunting Level at all the internal nodes of dend[0]
data.wopen("Segev1.14SL.1.dat")
data.printf("location Shunting Level \n")
data.printf("%d\n", location.size())
for i=0, location.size()-1 {
    data.printf("%g %g\n", location.x(i), SL.x(i))
}
data.close()
```