Harvard-MIT Division of Health Sciences and Technology
HST.723: Neural Coding and Perception of Sound
Instructor: Bertrand Delgutte

HST.723 - Neural Coding and Perception of Sound
Spring 2004

Compartmental Model of Binaural Coincidence Detector Neurons

Bertrand Delgutte

Time and location: 4/9/2004, 9-11 am

Introduction

The purpose of this laboratory exercise is to give you hands-on experience with a compartmental model of a neuron. Compartmental models differ from point neuron models such as the Kalluri and Delgutte (2003) model studied in the Cochlear Nucleus theme in that they explicitly represent the geometry of the neuron. In a point neuron, all points inside the cell are assumed to have the same electrical potential. This assumption is appropriate for neurons that are "electrically small", i.e. small relative to their length constant. For neurons with long, thin dendrites, the point neuron assumption is inappropriate, and one must use a compartment model. In this type of model, the neuron's volume is divided into separate compartments, each with its own potential. Typically, there could be separate compartments for the cell body, the axon hillock, and several compartments for the axon and each of the dendrites. The potential at each point is determined by the distribution of both active and passive membrane conductances as well as the intrinsic resistance of the intracellular material. Obviously, compartment models can be computationally much more demanding than point neurons.

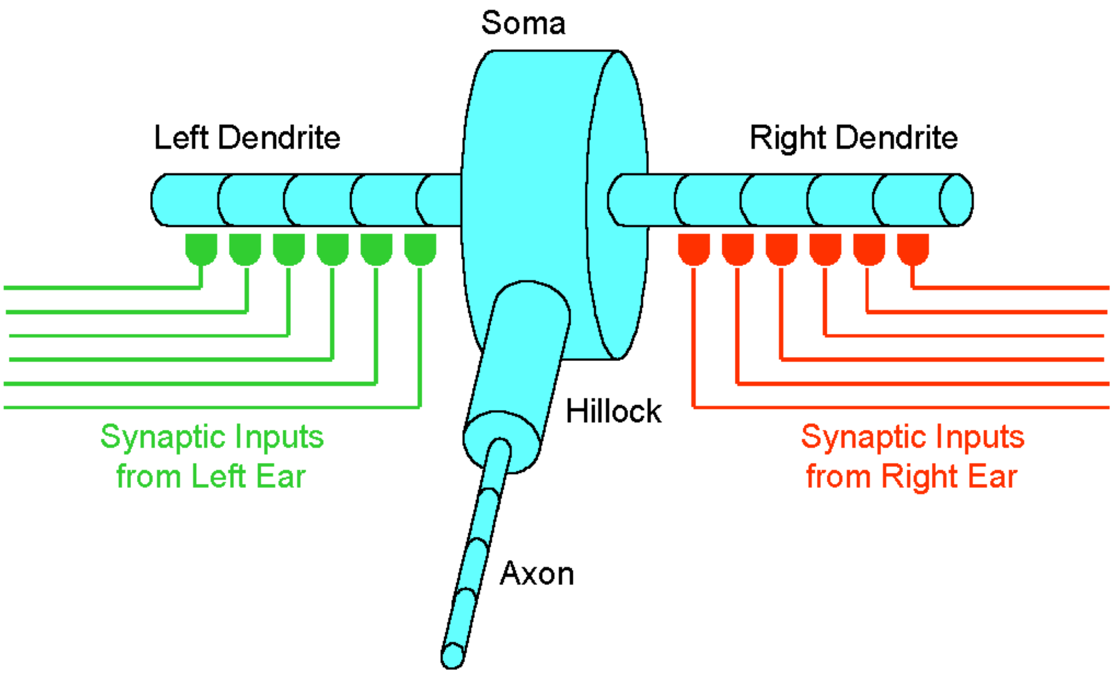
Figure removed
due to copyright
considerations.

Steps in building a compartmental model: A: Neuron geometry and physiological characterization (channel types). B: Cable representation of geometry. Right: Equivalent electrical circuit for small patch of cable. From Segev and Burke (1999).

The particular model that was selected to illustrate these concepts is a model for binaural coincidence detector neurons in the nucleus laminaris (NL) of the chick (Simon et al., 1999). NL is the avian homolog of the mammalian MSO, and, like MSO, contains neurons with bipolar dendrites, each one receiving inputs from the cochlear nucleus on one side. Like mammalian MSO neurons, NL neurons act as binaural coincidence detectors and are sensitive to interaural time differences (ITD). This neuron was selected because a great deal of anatomical and physiological data are available for constructing a compartmental model. A goal of this lab is for you to understand which neural properties are essential for a neuron to act as a binaural coincidence detector. It has been observed that the length of the dendrites of NL neurons varies inversely with the neuron's best frequency: low-CF neurons have longer dendrites than high CF neurons. You will use the model to investigate the consequences of this co-variation for ITD tuning using stimuli with different frequencies.

The model

A brief description of the model is available in the Simon et al. (1999) paper, which is rather hard to read. A description of a closely-related (but less detailed) model from the same laboratory is available in a paper by Agmon-Snir et al. (1998). This paper studies the role of dendrites in coincidence detection by comparing ITD tuning for a compartmental model having bipolar dendrites vs. ITD tuning for a point neuron model. Tuning of the compartmental model is found to be sharper, except at high frequencies. The Agmon-Snir model also provides a mechanistic rationale for the co-variation between best frequency and dendritic length by showing that optimal ITD tuning is obtained for a specific dendritic length at every frequency.



Schematic geometry of the NL neuron model. The drawing is not to scale, and not all compartments are shown explicitly. Any resemblance to a Martian weapon in a B sci-fi movie is purely accidental.

The NL neuron model includes a 15-mm soma (5 compartments), two 70-mm dendrites (10 compartments each), a 30-mm axon hillock (10 compartments), and an axon, itself comprising a 100-mm myelinated section (10 compartments) and unmyelinated node (1 compartment). The neural membrane model includes the standard Hodgkin-Huxley conductances (Na+, K+, and leak), as well as the low-threshold, outward-rectifier K+ channel ("Klva") studied by Manis and Marx (1991) in VCN bushy cells, which has also been found in NL neurons. There is also a high-threshold K+ channel ("Khva") found in MNTB neurons which is thought to help fast repolarization after a spike. These channels and other cellular specializations for fast timing are well described in a review paper by Trussell (1999).

Excitatory synaptic inputs to the model are provided by model spike trains from both cochlear nuclei. The synaptic conductance and the number of synapses can be varied, as well as their distribution along the dendrite. In the default configuration, there are 30 uniformly-distributed synapses on each dendrite, each one receiving a statistically independent input from a model CN neuron. The model also receives inhibitory inputs on the proximal end of each dendrite. These inputs are effectively sustained, lasting throughout the duration of the stimulus.

Spikes from the cochlear nuclei are simulated by a Poisson model with dead time (refractory period). The model only allows pure-tone stimuli presented either monaurally or binaurally with an adjustable interaural phase difference (IPD). Both the average discharge rate and the synchronization index (vector strength) of the phase-locked spikes can be controlled independently.

The model is implemented in NEURON, a general software environment for neural modeling developed by Michael Hines at Yale University. NEURON is freely available for download, and runs on Unix, MacOS, and Windows machines. The NEURON code implementing the NL-cell model can also be downloaded from Jonathan Simon's site at the University of Maryland.

References

1. Simon, J.Z. Carr, C.E., and Shamma, S.A (1999). A dendritic model of coincidence detection in the avian brainstem. *Neurocomputing* **26-27**, 263-269.
2. Agmon-Snir, H., Carr, C.E, and Rinzel, J. (1998). The role of dendrites in auditory coincidence detection. *Nature* **393**, 268-272.
3. Kalluri, S. and Delgutte, B. (2003). Mathematical model of cochlear nucleus onset neurons: I. Point neuron with many, weak synaptic inputs. *J. Comput. Neurosci.* **14**:71-90.
4. Trussell, L.O. (1999). Physiological mechanisms for coding timing in auditory neurons. *Ann. Rev. Physiol.* **61**:477-496.
5. Segev, I. and Burke, R.E. (1999). Compartmental models of complex neurons. In *Methods in Neuronal Modeling*, C. Koch and I. Segev (eds). MIT Press: Cambridge, MA, pp. 93-136.

NEURON orientation

You will be primarily using four NEURON panels in this lab exercise.

1. The **HST.723 Lab Exercise** panel is used for displaying the Simulation Control panels specific to each simulation, and for quitting Neuron.
2. The **File and Displays Control** panel is used for saving your own model configuration (*.hoc) files (from the *File* menu), for (re)displaying the other control panels, and for setting plotting options.
3. The **Run Control CD Lab** panel lets you start simulations, and stop them if you notice an error. You also use this panel to set the period of real time that is being simulated. The default (115 msec) is appropriate in most cases. The first 15 msec are discarded to let the model settle to a steady state, so you will be effectively using 100 msec of data.

Clicking on each button in the **HST.723 Lab Exercise** panel pops up the panel specific to the simulation you want to run. This panel allows you to specify which parameters of the model you want to vary, and over what range. Sometimes, you will also have a slave parameter co-varying with the primary parameter. **Using more than 8-10 parameter values creates very long simulations.**

Also useful is NEURON's **Print and File Window Manager**, with which you can select windows for printing (by clicking on them), then resize and print them.

Running a simulation

Once you have set all the parameters for a simulation, press ***Init & Run*** in the **Run Control CD Lab** panel. A number of plotting windows pop up, each one with several subplots. Each window (labeled 0 to n-1) is for one value of the primary parameter that is being varied. Each subplot in a window is for one interaural phase difference (IPD). By default, there are 5 values of IPD ranging from 0° to 180° in 45° steps. You can change this number by altering the **Cells per Array** parameter in the Simulation Control panel. Increasing this value allows you to study IPD tuning with finer resolution, at the price of a proportional increase in computational time. The default value is appropriate for most purposes.

Each subplot in a plotting window contains several traces, including the synaptic inputs, the dendritic, cell-body, and axonal potential, etc. You can specify which traces to display using the **File and Display Control** panel.

At the end of a simulation, the model creates 3 files: (1) a configuration (*.hoc) file, (2) a text file (*.txt) containing summary results, and (3) a Matlab (*.m) script for plotting the results. All three files have the same, unique name created by a rather obscure algorithm. To plot the summary results, enter the file name at the Matlab prompt. Both the average discharge rate and the synchronization index ("Vector Strength") of the model cell will be plotted as a function of IPD for each value of the primary parameter. To replot the same data on a normalized vertical (discharge rate) scale, enter PlotRateVSNorm at the Matlab prompt.

Default settings for most simulations

By default, two parameters in the model receive a special treatment which requires some explanation:

a. The "***Vector Strength of CN Inputs***" depends on frequency of the tone stimulus according to experimental data in chicks, to mimic the decrease in phase-locking with increasing frequency.

b. The "***Length of Dendrites***" is automatically derived from the stimulus frequency to mimic the decrease in dendritic length with increasing best frequency observed in NL neurons. This effectively assumes that a neuron is always stimulated at its best frequency.

In most stimulations, it is appropriate to leave these defaults as is, unless otherwise indicated in red.

Run two standard simulations

First, run the two simple simulations below to familiarize yourself with the operation of the software. The first one focuses on basic cellular properties *in vitro*, the second one on the effect of input phase locking on ITD tuning *in vivo*.

1. Threshold for intracellular current injections

This very simple simulation mimics intracellular current injections *in vitro*. Unlike the other, *in vivo* simulations, it requires no synaptic input, and is not concerned with ITD tuning. Examine the traces for membrane voltage in the soma and the axon, and determine the smallest current that produces an action potential propagating along the axon. What criteria do you use to determine that a spike occurred?

2. Effect of synchronization index of the CN inputs

With pure-tone stimuli, the synaptic inputs must be phase-locked for the coincidence detector in order to operate. Verify this by varying the vector strength (synchronization index) of the CN inputs. Compare the vector strength of the NL cell with that of its CN inputs for both IPD = 0° and IPD = 180°. How do you account for your observations? **For this simulation, check "*VS does not depend on frequency*".** Otherwise, the synchronization of the inputs is automatically derived from the frequency to mimic the decrease in phase locking with increasing frequency.

Pick one simulation for detailed study

Choose one additional simulation for further study. With your partner, formulate a specific hypothesis as to what you expect the parameter(s) of interest to do for ITD tuning. Present your hypothesis to the entire class, and modify it based on the feedback you get. Then, run the simulation, and present the results to the class. Discuss whether your hypothesis is supported by the results or whether it needs further modification

You can pick your simulation from the suggested list below, or come up with one of your own. In order to create a new simulation, use the **Simulation Control** and/or the **CD Lab Parameters** panels (You don't need these panels to do this lab exercise if you are happy with the suggested simulations below). When in doubt about the range over which to vary a parameter, find the default value, and vary around it (e.g. 1/3 default to 3 times default).

Suggested simulations

1. **Effect of number of synapses per dendrites**

Coincidence detection also requires synaptic inputs to be individually subthreshold. To verify this, vary the number N_s of synapses per dendrite while inversely varying the conductance of each synapse G_s so as to keep the net synaptic strength (the product $N_s G_s$) constant. For this purpose, make G_s a slave of the primary variable N_s. Do you get best ITD tuning with large or small N_s? Examine and print the individual voltage traces and their fluctuations to understand what is happening.

2. **Effect of synapse position along dendrite**

Yet another factor in coincidence detection is the distribution of synapses along the dendrites. By default, excitatory synapses uniformly cover the entire length of each dendrite, consistent with the anatomy. You can also create very narrow distributions (set the "***Excitatory Synapses Distribution Width"*** parameter to < 0.1), and vary the position of the center of this distribution from 0 (near the cell body) to 1 (the distal end of the dendrite). One difficulty is that the average firing rate of the model neuron becomes very low when the inputs are far on the dendrite (Why?). To meaningfully compare ITD tuning with proximal and distal inputs, you therefore need to boost the firing rate for distal inputs. The simplest way to do this is to co-vary the synaptic conductance G_s with synapse position by making G_s a slave variable as in Simulation 3.

3. **Effect of dendritic length**

The length of the dendrites has a profound effect on ITD tuning. Predict whether increasing dendritic length will improve or degrade tuning, then verify your prediction by varying dendritic length. Again, increasing dendritic length will greatly decrease the firing rate, so you need to compensate by co-varying the synaptic conductance G_s. In other simulations, dendritic length is automatically derived from the stimulus frequency. For this simulation, however, make sure to **uncheck** ***"Dendritic Length Decreases as Frequency Increases".***

4. Effect of tone frequency

The effect of tone frequency is a combination of the effects observed in Simulations S2 and 3. The vector strength of the CN inputs varies with frequency, since phase locking degrades at high frequencies. In addition we have seen that, in NL neurons, the length of dendrites is inversely correlated with best frequency. If you want to simulate the response of a tonotopic array of neurons to best-frequency tones, dendritic length must vary with frequency as well. To see how these effects interact, run the simulation two ways, with "***Dendritic Length Decreases as Frequency Increases***" first checked, then unchecked. Discuss the differences.

5. Effect of inhibitory synapses

Varying the conductance of inhibitory synaptic inputs alters the model cell's resistance, and therefore the membrane time constant and coincidence detection properties.

6. Effect of low-threshold K$^\pm$ conductance

The maximum conductance of the low-threshold K+ channel (*gKLVA* parameter) also affects the cell's resistance and time constant. This conductance is found both in the dendrites and the soma. To vary both together, make the dendritic *gKLVA* the primary parameter, and check "*Soma g's from Dendrite*".