2D1435, Mathematical Modeling of Biological Systems, 2002

Lab 4: Biological Neural Networks

1 Objectives

In this exercise you will study how properties on the molecular and cellular levels can give rise to "behaviour" on the network level. The exercise consists of two parts. In the first, you will examine a model of the swimming pattern generator in the primitive vertebrate *Lamprey.*[1] In the second, you will study a simplified model of Hebbian cell assemblies. Hebb's cell assembly theory is fundamental to many modern theories of how the cerebral cortex represents and processes information. When you are finished you should:

- have some hands on experience with the simulation of biological neural networks.
- understand how a neuromodulator can have an effect at the network level.
- understand how properties of single cells can have an effect at the network level.
- be able to explain the basic principle of how the Lamprey central pattern generator (CPG) generates the swimming rhythm.
- understand the concept of *Hebbian cell assembly*
- understand how the concepts of *recurrent ANN* and *Hebbian cell assembly* relate to each other.
- understand the network level effects of *spike train adaptation* in the Lamprey CPG and in a *Hebbian cell assembly.*
- know why inhibition is important in a recurrent excitatory network.

2 Tasks

So far you have only experimented with single neurons. In this exercise you will see how neurons can interact in networks, and how new properties and functions can emerge through this interaction. It is assumed that you are familiar with the Swim simulator and have done the previous exercise in this series (lab 3: Neuron and Synapse Models).

In the first part of the exercise you will perform a series of simulations of a network of neurons from one segment of the Lamprey spinal cord. This is the neural circuit responsible for the production of swimming movements in this animal. You

[1] Resembles an eel (Swedish: "nejonöga").

will examine how the network behaves under exposure of different concentrations of drugs affecting the *AMPA/kainate*, *NMDA*, and, *serotonin* (5-HT) receptors.

In the second part you will first study the properties of two types of synapses. Then you will determine a suitable value for a critical parameter in a model of a Hebbian cell assembly: the *excitatory connection strength.* Finally, you will study different functional properties of this network.

3 Background for the Lamprey simulations

3.1 A central pattern generator in the Lamprey spinal cord

The network you are going to work with here is an abstraction of a central pattern generator in the Lamprey spinal cord which is responsible for the generation of swimming movements.

The Lamprey is a primitive vertebrate which diverged from our evolutionary line some 400 million years ago. It is believed to have been comparatively well preserved during evolution, so that it still shares many characteristics with our common ancestor. Therefore, studying the nervous system of this species reveals fundamental architectural features of our own nervous system.

A basic part of most neural circuits controlling locomotion is the central pattern generator. These CPGs are neural networks, located in the spinal cord, which generate patterns of action potentials to the motor neurons. The set of CPGs in the spinal cord serves as a repertoire of basic motor components upon which higher centra can act to achieve the final motor behaviour.

The CPG in this exercise generates rhythmic bursts of action potentials alternating between the two sides of the body. Such a burst leads to contraction of one side of the body. In the Lamprey, this circuit generates the swimming rhythm. It is believed that the same circuit has evolved to produce the motor patterns controlling walking in higher vertebrates.

When studying this circuit experimentally, physiologists dissect pieces of the living spinal cord out of the animal's body and place it in a chamber, bathing in a solution with nutrients. The network can now be excited by releasing drugs into the bath. The drugs go into the synaptic clefts and act directly on the post- and presynaptic receptors.

3.2 The CPG model

Although the real spinal cord contains about one hundred segments connected like a chain, we are only going to simulate one segment here. Another simplification is that we only simulate one neuron of each type. In reality there are around fifty.

Figure 1 shows a schematic drawing of the simplified network. The network may appear complicated, but is actually quite simple. It consists of 2 x 5 neurons symmetrically arranged on the two sides of the spinal cord. The motoneurons (MN) are not explicitly simulated since they don't participate in generating the locomotor pattern. At the start of a swim cycle the excitatory interneuron (EIN) and the interneuron with a crossed caudal axon (CCIN) on one side is active and the opposite side is kept quiet by the CCIN which has inhibitory connections to all interneurons on the contralateral[2] side. The activity will switch between the left and right sides due to different burst terminating mechanisms. Such a mechanism is provided by the LIN cell which is thought to contribute at higher swimming speeds. LIN inhibits CCIN and thereby reduces its inhibition of the contralateral neurons which become active. The same procedure is now repeated on the opposite side

[2] opposite side

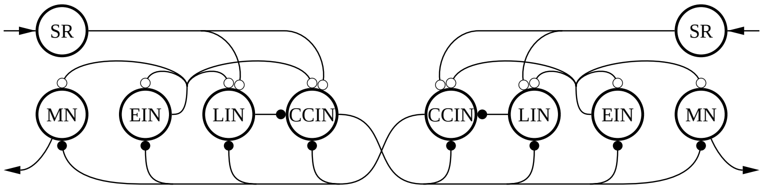

FIGURE 1: Local (segmental) network responsible for the generation of the basic rhythmicity of swimming. The network is symmetrically organized with motoneurons (MN) providing output to the muscles on the two sides of the body and stretch receptor neurons (SR) receiving information on the local curvature of the body. The basic rhythmicity is generated by the network of premotor interneurons. Crossed caudally projecting interneurons (CCIN) provide reciprocal inhibition. Small excitatory interneurons (EIN) maintain this ipsilateral activity while large laterally located interneurons (LIN) may contribute to the termination of the bursts to make the contralateral side take over.

after which the first side once again becomes active and the whole cycle is repeated. The normal locomotor pattern consists of different variations on this theme, some of which we will investigate below. The stretch receptor neurons (SR) are sensory neurons, more about them later.

4 The Lamprey CPG simulation environment

You should also create a directory (e.g. called “lamprey”) and copy the simulation specification files to that directory. Type the following commands at the shell prompt (>):

```
> mkdir lamprey
> cd lamprey
> cp -pr /info/biomod02/lab4/lamprey .
```

To be able to run the simulations in this exercise it is necessary that you have the right environment. If you haven’t already set up the environment by the command “course join biomod02”, or from lab3, please do so now. Either add to your path /info/biomod02/bin or copy the commands `swim`, `swim_new` and `xswimgraph` from that bin directory to your `lamprey`, or your own `bin`, directory.

5 Kainate swimming

`lamprey` is a shellscript that is used to start the Swim simulator and simulate the described network for 2 sec. It expects three parameters. They are used to specify the relative (simulated) concentration of three different drugs activating AMPA/kainate, NMDA and serotonin receptors. Normally the CPG is activated by synapses from reticulospinal neurons situated in the brainstem. However, as mentioned above (section 3.1), it can also be activated by administrating a drug to the extracellular bath solution. This is the type of activation simulated here. You can now start a simulation by typing the command

```
> lamprey 4 0 0
```

which will give some stimulation to the AMPA/kainate receptors, but no stimulation to either NMDA or serotonin receptors. This will bring up a window with graphs of the membrane potential in some compartment of the EIN and CCIN cells in the network.

- Verify that the function is as described above and note the burst frequency by which the activity alternates between the two sides.

The drugs cause a lasting activatation of different membrane channels (synaptic as well as extrasynaptic) and can induce a locomotor pattern similar to the natural one. By increasing the concentration of kainate it is possible to make the network "swim faster". Leave NMDA and serotonin at zero. Try

```
> lamprey 6 0 0
```

and observe what happens. However with too much excitation the locomotor pattern breaks down.

- What is the burst frequency just before this happens?

The burst frequency can be increased up to 10 Hz, as can be observed in the real lamprey preparation. This is accomplished by incorporating LINs into the network. Try

```
> lamprey_LIN 6 0 0
```

It is also possible to lower the kainate concentration, and thereby the burst frequency. Try

```
> lamprey 3 0 0
```

Still there is a regular burst pattern: *How is it possible that the network consisting of only EIN and CCIN cells can provide alternating activity between the sides*? The answer is that the AHP summation make the neurons fire slower towards the end of a burst, thereby giving the opposite side a chance to become active.

- You saw this phenomenon in exercise 1 as well. What is it called?

This mechanism is however not always stable and can give rise to irregular bursts as in

```
> lamprey 2.5 0 0
```

It should also be sensitive to drugs that affect the duration of the AHP, like serotonin. Let's try adding a bit of serotonin and see what happens:

```
> lamprey 3 0 x
```

where x is less than 0.1. (Try e.g 0.05 and 0.07).

- Compare this with the situation without serotonin. What is the difference?

- Why?

- Compare the relative sensitivity to serotonin for a low activation (AMPA/kainate field = 3) and a higher one (AMPA/kainate = 5).

6 NMDA swimming

At slow swimming there is one additional mechanism that significantly stabilizes the rhythm. Many synapses in the spinal cord contain NMDA receptors. The channels coupled to NMDA receptors display certain unique properties, including a strong voltage dependence and a permeability for calcium. Studies in the lamprey CNS has shown that a tonic activation of NMDA receptors can induce intrinsic oscillatory properties in spinal neurons. You can simulate NMDA mediated stimulation by executing the command

```
> lamprey 0 1 0
```

(The initial burst is much longer than the following ones, but that's an artifact due to an unfortunate choice of initial values.) You can try different values of the drug (a useful range is 0.5–1.5). You can also try mixing drugs affecting the NMDA receptors and drugs affecting kainate receptors in the bath. Don't expect any large variations in oscillation frequency with high (1.0) values for the NMDA-affecting drug.

- Describe the differences to the previous patterns with regard to oscillation frequency and stability.

7 Sensory feedback

Sensory feedback is an important part of almost any motor act. In this exercise we are going to study how the activity of sensory cells can modify the locomotor pattern of the network. For this purpose we introduce a new type of cells. They are a type of stretch receptors (SRs) situated at the lateral margin of the spinal cord. SRs have excitatory connections to the neurons on the ipsilateral[3] side. Stretching one side of the body (and thereby the spinal cord) activates the SR on that side, which in turn excites the interneurons and thereby synchronizes the CPG with the actual movements of the body. There are also inhibitory SRs (not shown) that work in the opposite way. In conclusion, SRs can modify the locomotor pattern of the CPG by entraining the frequency to that of the actual movement. To simulate the stretching of the spinal cord a time varying conductance with a sinusoidal time dependence is included in the SRs membrane.

To be able to do this exercise you must make a small modification to the file named CHANGE.ME. Remove all leading dashes from the include specifications. They are used to signify comments and when removed will enable Swim to include the specification files that define SRs and their synapses. After you have made the changes try execute the command below, and note the change in frequency as compared to the previous exercise.

```
> lamprey 0 1 0
```

You can change the frequency of the entraining movement by changing the "Freq" entry in the R MOD and L MOD definitions in the CHANGE.ME file.

- At what frequencies does the swimming rhythm get phase locked with sensory input?
- What happens if "Freq" is set to 6?
- Why do you think it is important that motor output is phase locked with sensory input?

[3] same side

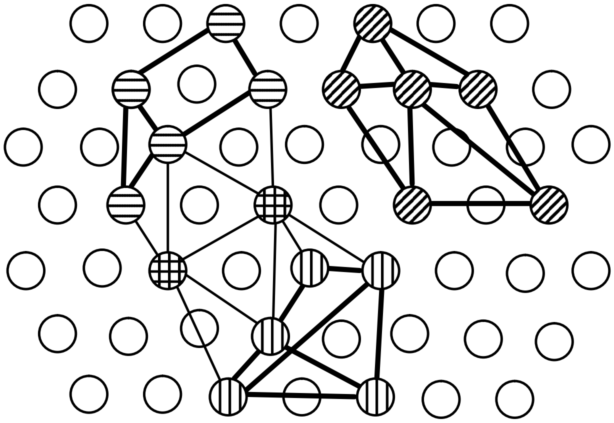

FIGURE 2: Three cell assemblies are seen, two of which are overlapping. Each circle represents a cell, each line the reciprocal connections between two cells. Only positive connections are shown. Negative connections exist between all cells which don't belong to the same assembly (pattern).

8 Background for the cell assembly simulations

8.1 Hebb's cell assembly theory

In his classical book, The Organization of Behavior, Donald Hebb introduced the concept of the *cell assembly* suggested to be the basic functional unit of information processing in the brain. It is a group of cells, strongly connected through excitatory synapses. The members of a cell assembly are not necessarily located close to each other but can be distributed over large areas of the brain. A cell can also, at least transiently, belong to more than one assembly. (See figure 2.) Such an assembly can emerge as a result of repeated co-activation of its constituent cells and the action of what we today refer to as Hebbian synapses, i.e. excitatory synapses enhanced by co-activity of the pre- and postsynaptic neuron. The resulting structure is analogous to a recurrent ANN.

Hebb proposed that the assembly so formed thereafter can serve as an internal representation of a corresponding object or category in the outside world. Activation of a few of its members can lead to activation of the rest, a phenomenon equivalent to *pattern completion* in recurrent ANNs. The cell assembly will then sustain its activity for some time by mutual excitation among its members, a phenomenon called *after-activity.*

According to the theory, the existence of cell assemblies in the brain can have an influence on system performance and eventually explain certain perceptual phenomena, e.g. perceptual completion. It has been suggested that competition between cell assemblies through lateral inhibition could explain phenomena like *figure-ground separation* and *perceptual rivalry.* Even chains of associations and, eventually, associative thought processes could be understood in terms of cell assemblies active in succession.

Hebb's cell assembly theory is still, in its general aspects, compatible with experimental findings relating to cortical architecture and physiology. Thus, despite the lack of direct experimental evidence, the cell assembly theory has remained one of the most viable hypotheses relating to cortical associative memory function.

Over the last few years, the computational advantages of neural networks of the kind suggested by Hebb have been thoroughly established in studies of abstract recurrent *artificial neural network* (ANN) models, e.g. Hopfield type attractor networks. Possibilities of comparing the output of these networks to biological experimental data have also been discussed. It has been demonstrated that small networks of cultured cells can indeed display activity of the sort described by these computational models.

In fact, a prototypical recurrent attractor ANN used as an auto-associative *content-addressable memory* (CAM) can be regarded as a mathematical realization of Hebb's basic idea. A "memory" in such a network, i.e. a group of units kept together by strong recurrent excitation produced by enhanced Hebbian synapses, corresponds closely to a cell assembly. The dynamic recall process converging to a low-energy stable state is analogous to the triggering of and after-activity in a cell assembly. It is interesting to establish the relations between neuro-psychological theories of cortical function and today's computational models, since results relating to storage capacity, prototype extraction capabilities etc. of the latter might then have a direct bearing on our understanding of brain function.

8.2 A simplified model of a cell assembly

The cell assembly model in this exercise is comprised of twenty-five excitatory neurons interacting through excitatory and inhibitory synapses. We will only study recall of patterns that have been previously stored, i.e. no learning will be involved, so the synaptic connectivity will be fixed.

The nerve cell models are of the same type as the ones in exercise 1 and in the previous part of this exercise, but their properties are modeled after the *regular spiking* and the *fast spiking* cells of the cerebral cortex. Synaptic interaction includes voltage gated NMDA receptors and conventional receptors.

Values for synaptic strengths are taken from a recurrent Bayesian ANN. The ANN has been trained with 5 patterns, each of which has 8 active units. Pairs of patterns share between 0 and 2 units. The Bayesian learning rule produces excitatory synapses within the patterns and inhibitory ones between them. The patterns are shown in figure 3. Altogether there are 268 excitatory synapses and 332 inhibitory synapses. The synapses are located on the middle dendritic compartment and on the soma respectively. The excitatory synapses are either of a conventional AMPA/kainate type or an NMDA type. Inhibitory synapses are modeled after the conventional postsynaptic inhibition, mediated via GABA$_A$.

9 The cell assembly simulation environment

Before continuing with this second part of the exercise, you should set up a new directory (e.g. called "hebbian") with the specification files that are needed. Type the following commands at the shell prompt:

```
> mkdir hebbian
> cd hebbian
> cp -pr /info/biomod02/lab4/hebbian .
```

Also here, if you haven't set your path (see Section 4), copy all commands (you will need most of them) from the `/info/biomod02/bin` directory to your `hebbian` (or your own `bin`) directory.

The directory `hebbian` now contains a file `lab.spec`. This is the main specification file. The simulation commands described in appendix A pass this file to the simulator (Swim). The file contains references to the other specification files which also will be read by the simulator. One file of particular importance is `change_me.spec`. It is documented in appendix B. During the course of the exercise you will edit this file to reflect the kind of stimulation given to the network. (Use the `Emacs` editor.)

```
                                                          cell
pattern:     1         2         3         4         5     numbering

           .....     *****     *....     .....     ..*..      1...5
           ..*..     ....*     *....     .*...     *****      6...10
           ..*..     ....*     *****     .*.*.     .....     11...15
           ..*..     ....*     *....     .****     *....     16...20
           *****     .....     .....     ...*.     ....*     21...25

pattern             cells active

   1       8,  13, 18, 21, 22, 23, 24, 25
   2       1,   2,  3,  4,  5, 10, 15, 20
   3       1,   6, 11, 12, 13, 14, 15, 16
   4       7,  12, 14, 17, 18, 19, 20, 24
   5       3,   6,  7,  8,  9, 10, 16, 25
```

FIGURE 3: In the upper half of the figure the whole matrix of cells is showed for each pattern. A star indicates a cell belonging to the pattern. The patterns are numbered 1–5. The figure to the right shows the cell numbering scheme. The lower half of the figure shows which cells are active in which pattern.

The activation of cell assemblies is simulated by means of a depolarizing current injection into cells belonging to the activation pattern. The activation patterns are illustrated in figure 3.

In the first part of this exercise, you were given detailed instructions about what to do at each step. In this part you will yourselves select among the available commands described in appendix A. Each command starts a simulation and presents one or two windows with simulation results on the screen. E.g., in order to run the cell assembly network with excitatory connection strength 5.0, type:

```
> esyng change_me.spec 5.0
```

You will now get two windows on your screen: In the Swimgraph plot the cells are plotted row wise, with the first cell (no. 1) in the the top trace and the last cell (no. 25) in the bottom trace. This graph tells you details of when cells fire and how long they are active. The square color plot codes membrane potential from purple (= large negative) to red (= large positive). A spike is seen as a red flash. This graph is mostly useful to see if the right cells are active in a pattern. Remember that the firing of action potentials is what matters for other cells, the membrane potential is only a matter for the cell itself.

- Don't forget to save your changes before executing.
- Use `^C` (Ctrl-C) to stop an execution.
- A SwimGraph window is deleted by selecting the window and typing `q`.
- You can print a copy of the graph by selecting the window and typing `p`.

10 Kainate and NMDA synapses

You will now compare the properties of two types of synapses. The properties of both cells and synapses has a large influence on the network behavior. You're now free to use the commands in appendix A as you want.

- Compare the rise and decay times of the EPSP for kainate and NMDA synapses. Which is slowest, i.e. has the largest rise and decay times?

- Compare the evolution over time for the postsynaptic membrane potentials behind a kainate and an NMDA synapse during temporal summation. Look at the curve shapes. Do you see any qualitative difference? Look at the derivatives!

- Try to suggest a mechanism which could be one cause for this difference.

(As an extra exercise, you might want to look at the firing properties of different cell types.)

11 Strength of recurrent excitation

In a network of biologically realistic cells, as the one we will study here, an activation of the cells in one of the stored assemblies/patterns can result in sustained spiking activity of all cells after the stimulation has ceased. The after-activity can be compared to the low-energy stable *attractor state* in an artificial recurrent neural network. The frequency during the after-activity is rather stable at some 50–100 Hz. For this activity to persist, some criteria must be met. First, the excitatory synaptic conductance must be of an appropriate strength. Furthermore, cell types should have repetitive firing characteristics that support this sustained activity.

One of the central features of a content-addressable memory is the ability to retrieve a complete pattern when a partial pattern is presented, so called pattern completion. In this context, good performance means that unstimulated cells belonging to a partially activated pattern should become active (produce action potentials) through the excitation originating from the other assembly members. Cells in other patterns should stay silent. However, we do not want a pattern to become completely active just because of *one* stimulated cell as such a stimuli could be caused by noise in the input. A noise tolerance of one cell does not seem much, but the pattern size is only 8 so it's not too bad.

11.1 After-activity

You will first study after-activity. The specification file `change_me.spec` contains information about which cell type, which excitatory synapse type, and, which stimulation pattern to use (see appendix B). Throughout this exercise, use the cell type `RS` and synapse type `kainate_dendr`. (This is the original values so if you haven't changed the file yourself, you won't need to edit it.)

Select one of the complete stimulation patterns in `change_me.spec`. If the file is unchanged, pattern 4 is selected, so if you are content with this pattern you won't have to edit the file. Some patterns are more difficult for the network than others. Now study the effect when the excitatory synaptic strength is varied. The ability to support after-activity and, as we also will see below, pattern completion depends heavily on the excitatory (and inhibitory) synaptic strengths. Start with the maximum value and go down, searching for the lowest value giving after-activity. The total simulation time is 100 ms, but a cell assembly can be said to have acceptable after-activity if all pattern members are active for more than 75 ms. The activity can gradually die out during the last 25 ms—this is OK.

- Which is the lower bound of the synaptic strength?

11.2 Noise tolerance

For studying noise tolerance, choose the pattern with one cell stimulated. Begin at the minimum value of synaptic strength and go up.

- Find the largest value that does not give pattern completion (one or two stray spikes are OK, but not more). This will be the upper bound on the synaptic strength.

11.3 Determining a suitable value for the excitatory synaptic strength

As a final check, pick a value in between the two end values you have found, and test this with a partial pattern with some 5 cells stimulated (the patterns with fewer stimulated cells are more difficult for the network than the ones with many stimulated, make your own choice) to see that pattern completion and after-activity works at the same time.

- The requirements of after-activity and pattern completion puts conflicting constraints on the connection strength, how?
- Which value did you pick? Does it work well? Use this value for the rest of your simulations!

If you like and have time, you can also do the test above for other cell types than `RS` and other synapse types than `kainate_dendr`. You will see that the cell type changes the network function a lot. Does after-activity and pattern completion still work?

12 Properties of a cell assembly

In this section you will study some more phenomena and make connections to artificial neural networks, biology and psychology. Use the same cell and synapse type as above to investigate some further assembly functions.

It is important to suppress noise in the input and retrieve a correct and noise free pattern. This is called noise tolerance. Further, if instances of two stored patterns are presented simultaneously a rivalry process will start as only one pattern can be active in an attractor network. This competition between the two patterns is due to inhibition between patterns, and makes one pattern suppress the other. Without the inhibition, the overlap between the stored patterns will lead to a spread of activation and pattern specificity will be lost. The time it takes from the beginning of the stimulation until a fairly stable activity pattern appears is called the *convergence time* in ANN terminology. If the axonal time delays in the network are too large, after-activity and pattern completion will not work properly.

In biological neural networks the electrical activity is only a small (but fast) component of the total activity. Most activity is biochemical reactions with different time constants from fractions of seconds to years. The complexity of these "computations" is probably huge and hence relatively little is known. One category of biomolecules is called modulators, as they change the properties of the neurons. One effect of such a modulator, serotonin, is a decreased AHP due to a decreased conductance of the Ca-dependent K channel. This will have effects on the network activity i.e. a chemical messenger influences network behaviour.

Investigate the noise tolerance and estimate the time it takes to get a stable firing pattern (convergence time). Use the stimulation types found in `change_me.spec`. Note that the stimulation starts 10 ms after the start.

- How many cells are needed in a pattern to suppress noise and complete the pattern?

- How long was the convergence time?

Look at the mixed pattern and estimate the convergence time.

- How long was the convergence time for the *mixed* pattern?

Look what happens if the inhibition is slightly lowered, and then what happens if inhibition is set to zero. (Since you won't use the `esyng` command here, you won't be able to use the value for synaptic strength which you determined in section 11.3. Instead a preselected value will be used.) Stimulate with a full pattern.

- What happens and why does is happen?

- This mechanism is involved in a neurological disorder, can you figure out which?

Investigate the tolerance to axonal delays. An increase of the size of the raster the cells are placed on gives an increased axonal delay. (Remember the fixed synaptic transmission delay of 1 ms.) Stimulate with some noisy pattern of a few cells (5 for example).

- How large maximal axonal delay can be accepted before assembly operations turn bad?

- Given an axonal transmission speed of 1 m/s, how far apart can the most separated cells be?

Look at the effect of simulating an increase or decrease in the normal level of a neuro modulator like serotonin. Stimulate with a full pattern.

- How is the after-activity affected?

- Describe how the modulator can have this effect.

- How does this affect the relative importance of internal (from other assembly members) compared to external excitation?

A Simulation commands

These are the commands you can use to study the network behavior. Not all commands will be used. Using these you can set one parameter at a time while leaving the others at the default value. The use of a command does not permanently change the value of that parameter, it only affects that particular simulation. The data file you give as an argument is `change_me.spec`.

`esyng` *filename strength(s)* Set the synaptic conductance scaling factor [0.0 – 10.0] of the excitatory synapse and simulate.

`isyng` *filename strength(s)* Set the synaptic conductance scaling factor [0.0 – 10.0] of the inhibitory synapse and simulate.

`ahp` *filename amount(s)* Set the level of modulator [-1 – 1] (that scales the conductance of the Ca-dependent K channel in the opposite direction) and simulate.

`delay` *filename delay(s)* Set the maximum axonal delay [0.0 – 50.0] (ms) in the simulated area and simulate.

`psth` *filename bin-width(s)* Set the bin width [1 - 9] (ms) in the histogram of number of spikes per time window, and simulate.

`tspike` *filename* Produce a spike plot of the cells and a total spike plot with all spikes summed. Here only the time of a spike is indicated with a short line.

`rundefault` *filename* Run a simulation with default values.

These are the commands you can use to study the cell and synapse characteristics.

`scell` *filename* Show all cell types while giving a short ($\approx 5\,ms$) stimulation. This displays the action potential shape of each cell type. The data file you give as an argument is `scell_instances.spec`.

`lcell` *filename* Show all cell types while giving continuous stimulation. This displays the continuous firing characteristics of each cell type. The data file you give as an argument is `lcell_instances.spec`.

`ssyn` *filename* Show all synapse types while giving a short ($\approx 5\,ms$) stimulation. This displays the PSP shape of each synapse type. The data file you give as an argument is `ssyn_instances.spec`.

`lsyn` *filename* Show all synapse types while giving continuous stimulation This displays the summing characteristics of each synapse type. The data file you give as an argument is `lsyn_instances.spec`.

arg(s) = multiple arguments can be given and will result in multiple simulations.

B Parameters in the file `change_me.spec`

Cell types:
`EIN` This is the excitatory inter neuron of the Lamprey spinal cord.
`RS` This is a regularly spiking cell type resembling a cortical pyramidal cell.
`FS` This is a fast spiking cell type resembling a small cortical inhibitory interneuron.
`IB` This is an intermittently bursting cell type resembling a special type of the cortical pyramidal cells.

Excitatory synapse types:
`kainate_soma` This is a kainate synapse located at the soma.
`kainate_dendr` This is a kainate synapse located in the middle of the dendritic compartment chain.
`NMDA_dendr` This is a NMDA synapse located in the middle of the dendritic compartment chain.

Stimulated pattern, choose one of these by setting its current to i.e. 0.4 nA.

Stimuli, complete patterns, stimulation time 20 ms:
`stim1` Stimulate with pattern #1.
`stim2` Stimulate with pattern #2.
`stim3` Stimulate with pattern #3.
`stim4` Stimulate with pattern #4.
`stim5` Stimulate with pattern #5.

Stimuli, partial patterns, stimulation time 20 ms:
`stimC1` Stimulate with pattern #4, stimulate 1 cell.
`stimC2` Stimulate with pattern #4, stimulate 2 cells.
`stimC3` Stimulate with pattern #4, stimulate 3 cells.
`stimC4` Stimulate with pattern #4, stimulate 4 cells.
`stimC5` Stimulate with pattern #4, stimulate 5 cells.
`stimC6` Stimulate with pattern #4, stimulate 6 cells.
`stimC7` Stimulate with pattern #4, stimulate 7 cells.

Stimuli, noisy patterns, stimulation time 20 ms:
`stimN1` Stimulate with pattern #4, stimulate 1 cell, and 7 cells in other patterns.
`stimN2` Stimulate with pattern #4, stimulate 2 cells, and 6 cells in other patterns.
`stimN3` Stimulate with pattern #4, stimulate 3 cells, and 5 cells in other patterns.
`stimN4` Stimulate with pattern #4, stimulate 4 cells, and 4 cells in other patterns.
`stimN5` Stimulate with pattern #4, stimulate 5 cells, and 3 cells in other patterns.
`stimN6` Stimulate with pattern #4, stimulate 6 cells, and 2 cells in other patterns.
`stimN7` Stimulate with pattern #4, stimulate 7 cells, and 1 cell in other patterns.

Stimuli, mixed pattern, stimulation time 20 ms:
`stim14` Stimulate with pattern #4, stimulate 4 cells and pattern #1, stimulate 4 cells.

Pattern stimulation length (seconds).
`stim_length` = 0.020