Running the Java spiking neuron system from MATLAB

*Originally*:

>> setuprunningstring ;

>> system([commandtorun '-fileprefix ' fp\_macbook ' -d "/drive\_1.csv" -c "context\_1.csv" ' cf2]) ;

Where *setuprunningstring* sets up everything. Altering exactly what’s happening involves editing this script.

Here’s what it looks like:

% location of all files etc for this run

fileprefix = '"/Users/lss/Documents/workspace/PyramidalCells/Test\_nov2018/" ' ;

% all the rest of the flags

commandflags = '-d "driving\_6synapses.csv" -c "context\_6synapses.csv" -s 5000 -t 5.0 -wd drivingsynapses\_6.txt -wc contextsynapses\_6.txt -alpha\_driver 1000 -alpha\_context 300 -apical\_multiplier 8 -p\_refractory\_period 0.01 -i\_refractory\_period 0.01 -t\_basal 0.01 -t\_apical 0.1 -t\_inhib 0.05 -n "networkconfig.txt" -sout "t1outSpikes.csv"' ;

% actual command to run opyramidal neuron simulation

commandtorun = 'java -jar pyramidal.jar ' ;

fp = '/Users/lss/Documents/workspace/PyramidalCells/Test\_nov2018/' ;

cf1 = 'c "context\_6synapses.csv" -s 5000 -t 5.0 -wd drivingsynapses\_6.txt -wc contextsynapses\_6.txt -alpha\_driver 1000 -alpha\_context 300 -apical\_multiplier 8 -p\_refractory\_period 0.01 -i\_refractory\_period 0.01 -t\_basal 0.01 -t\_apical 0.1 -t\_inhib 0.05 -n "networkconfig.txt" -sout "t1outSpikes.csv"' ;

cf2 = '-s 5000 -t 5.0 -wd drivingsynapses\_6.txt -wc contextsynapses\_6.txt -alpha\_driver 1000 -alpha\_context 300 -apical\_multiplier 8 -p\_refractory\_period 0.01 -i\_refractory\_period 0.01 -t\_basal 0.01 -t\_apical 0.1 -t\_inhib 0.05 -n "networkconfig.txt" -sout "t1outSpikes.csv"' ;

Nasty! It would be a better idea to create a function in which all the parameters were initialized, but alterable using varargin.

***Now***:

I have written a function RunSpikeSimulator which replaces all the above. All the parameters (and the actual system function being run) are preset, but updatable using varargin. This means that one could repeatedly call the function to run the network multiple times with (e.g.) the same input and different internal parameters.

RunSpikeSimulator(‘c’, ‘filename.csv’) ; etc.

There is an example of this in runMultiple.m. Here, (currently), the parameters from createDataFiles (specifically N) are internal, so need altered to be the same by changing driveno and cointextno. The data and context filke roots also need to be the same as those in the directory.

This now returns a driveno by contextno array containing the number of spikes generated by neuron 1: this can then be plotted using surf().

One can use plotspikes to visualize the output: note that the output file is at

[fileprefix sout]!

Added, March 20, 2019

Script runmulmul: runs through a set of RunSpikeSimulator function calls, varying gradient and intercept for the tuft and basal compartments. Note that values are the same for both compartments.

Function [spikesOutArray] = runMultipleIntercept(fileprefix, varargin)

Runs through a set of intercept values using empty context and drive input files. Concept is that the intercept (-ve values) is like a current injection, and result is like a step version of the Kay & Phillips graph. Varargin allows low and high levels of each to be set, as well as allowing some other parameters (gradients, number of steps). Produces a labelled surface graph.