1 Specifying OMP parameters in ompsimulator.py

Execute ompsimulator.py code as a script, passing the values as "paramname=paramvalues" pairs, e.g.,

```
python3 ompsimulator.py 0 dpname=omp1 lamm=0.1 lama=0.01 \
lamh=0.00001 nax=10 nol=2 taug=30 fixedel=nrn_5 taus=200 \
jitter=1 spksig=sync nreps=2 nepochs=100 Tesec=10 name=myrun1
```

The translation table between the variable names in the manuscript and in the code is provided below, in Table 2. The fixed delays are specified via string "fdtype"_"value". In the paper we use mainl normalized random normal values specified with σ_D, which is passed as nrn_"σ_D", e.g., nrn_10. For the spiking signal, use spksig="signal spec", we use mainly 4 types shown in Figure 2, and an arbitrary mixture of those. For each pure signal we have a single letter abbreviation, as well as a longer name (sometimes two, to preserve back-compatibility). The specification is as follows:
{"sync", s}: correlated/time-locked poisson spiking
{"indep", i}: independent poisson spiking
{"regsync", r}: correlated/time-locked regular spiking
{"regindep", j}: independent regular spiking
{"msync_xyz"}: Mixed signals, specified with a character sequence xyz.

Mixed signals can be specified in other ways, but for the purpose of this manuscript the simple notation `msync_xxyz...` suffices, in which axons are evenly divided among an arbitrary number of concatenated characters specifying one of the four pure signals using their one-letter abbreviations (s,i,r,j), e.g., `msync_ssssi`, which was used in Figure 4C.

The synchronization profiles are saved in the numpy file results/"runname"-results.npy. When using saver=5 only the basic information will be saved. Use np.load("filename").tolist() to load the results file as a dictionary. The key 'tstdarr' containes the synchronization measure, σ_τ, saved as a numpy array with shape (nreps, nepochs, nol, ngroups), where ngroups indicates the number of different mixed signals. Once can already use saver argument to generate model high temporal resolution histories of the OMP variables. For example, saver=aa5 will save all replicates and all epochs, but is not recommended as it will yield enormous files. Use second to last character to specify how many replicates or epochs to save, for example saver=a15 will save all epochs of the first replicate run, while 3r515 will save epochs 3-5, inclusive, for the first replicate run. Histories will be saved in hrecs/directory with a name modelhistories-"runname".npy in a list of lists. This modelhist list of list contains the timecourses for OMP variables in modelhist[ireplicate][iepoch][1] array, while modelhist[ireplicate][iepoch][0] contains the time array. This nomenclature for choosing output is going to be simplified in the new ompmodel.py distribution, when it becomes available, which will use OMPmodel class.

2 Translating between manuscript and code variables

Description	Symbol	python variable
OMP model parameters		
number of axons	N_A	`nax`
number of OL	N_O	`nol`
OL time constant	τ_G	`taug`
mean interspike interval	τ_s	`taus`
M-factor production rate	λ_M	`lamm`
myelin addition rate	λ_A	`lama`
homeostatic rate	λ_H	`lamh`
maximal delay	$\tau_{\max}$	`taumax`
minimal delay	$\tau_{\min}$	`taumin`
nominal delay	τ_{nom}	`taunom`
Signal & Simulation Parameters		
Duration of epoch	T_e	`Tesec`
Number of replicates	n_r	`nreps`
Number of epochs	n_e	`nepochs`

Table 1: Table of python variable names. The signal types are specified with a spksig string, as described in the text, and the fixed delays are specified using nrn_σ_D nomenclature.