

Code Instructions

Wilten Nicola, 06/2019

The code in these folders contains the scripts necessary to reproduce some of the figures in the manuscript: *A Diversity of Interneurons and Hebbian Plasticity Facilitates Rapid Compressible Learning in Hippocampus*, (2019), *Nature Neuroscience*, In Press, Wilten Nicola and Claudia Clopath. The DOI code for this manuscript is **10.1038/s41593-019-0415-2**.

Note that the background current parameters for the RO-CA1 network in Figure 5 were mislabeled as -37.5 pA (RO-CA1 I) and -40 pA (RO-CA1E). The values used in Figure 5 were actually -40 pA (RO-CA1I) and -42.5 (RO-CA1E). We apologize for this confusion. This code was prototyped on MATLAB V. 2016a.

Note that there are a sequence of steps required in order for the figure scripts to run without error. These steps are:

Steps for Running Code

Step 1: Run FORCE_training.m

Due to file size issues, the user has to run the training process themselves before activating the other scripts, in order to generate the trained weight matrices in FORCE_trained.mat and the phase preferences/ordering for the excitatory neurons in the file sortingid.mat. **Thus, the user should first run the script FORCE_training.m to generate these files for subsequent use**, other wise MATLAB error messages for missing files will appear when the figure scripts are used. The training script should take on the order of an hour with reasonable computing resources while the other scripts vary in duration, but are generally faster to run. Finally, the data file generated for the FORCE_training script is 1-2 gigabytes. The FORCE_training.m automatically calls the sorting_script.m to phase sort the weights/neurons.

Note that the random number seed was not fixed when we conducted these simulations in the manuscript, so some small numerical differences in the outputs may occur due to differences in the final trained weight matrix. However, we have fixed the random number seed in this code to (1) and recommend leaving it at that value in order to implement the reversion neuron population (Figure 2). The reason for this is that the reversion neuron implementation is dependent on a phase relationship between the reversion interneuron population and the SHOT-interneuron population. We manually searched for a component in the SHOT-CA3 FORCE components that has the correct phase relationship and the implementation is based on this (See Materials and Methods in manuscript). If the random number seed is altered, the user has to manually search the integer parameter “qin” in the code for Figure_3.m in order to implement the reversion. This typically only takes 1-2 minutes as the phase relationship need not be exact (see Materials and Methods of Manuscript)

Step 2: Run Training_Reversion_Neurons.m

This script trains a population of reversion neurons with a series of weights (determined offline, not through FORCE) in order for the reversion interneuron population to decode a bank of oscillators. The

decoded FORCE trained oscillators (decoded from the SHOT-CA3 population) serve as the supervisors for the reversion interneuron population. One of these reversion-interneuron-decoded oscillators is then used to implement the reversion current in Figure_3.m (the “qin” parameter selects the particular oscillator used).

After Steps 1. And 2. are completed, one can run the following script files:

Figure_1.m: Runs the population of SHOT-CA3 neurons post training with INP-MS on, and then off at time $t = t_{crit}$.

Figure_2.m: Runs the population of SHOT-CA3 neurons with recurrent excitatory connections. The default parameter values in this script consider the case where both E-to-E and E-to-I connections are present.

Figure_3.m: Runs the mirror reversion protocol as in Figure 3 of the main manuscript with a population of reversion interneurons, and the SHOT-CA3 network.

Figure_5.m: Runs the SHOT-CA3 and RO-CA1 training task with the local Fourier rule operating as in Figure 5 of the main manuscript. Note that for the time interval considered where the Medial Septum is off, different random seeds of this simulation may yield different results (or phase distortion), or different levels of activity in RO-CA1.