Multiarea Hill-Tononi thalamocortical network model 3.1

Ricardo Murphy
University of Oslo, Domus Medica, Division of Physiology,
Pb. 1103, Blindern, 0317 Oslo, Norway

Model description

We have developed a 'toy brain' network model comprising left (L) and right (R) hemispheres, each with three cortical areas (C1-C3) and associated thalamic and reticular nuclei. Each cortical area consists of three interconnected neuronal layers: a supragranular layer (L2/3), an infragranular layer (L5/6) and layer 4 (L4). Each thalamic nucleus consists of a single neuronal layer. Excitatory and inhibitory neurons within each layer are modeled as hybrid Hodgkin-Huxley conductance-based/integrate-and-fire (IAF) neurons based closely on the neuron model of Hill and Tononi (2005). Ipsilateral connectivities are based on Esser, Hill et al. (2009), while homotopic and heterotopic interhemispheric connectivities are based on data from various sources (Wise and Jones 1976, Douglas and Martin 2004, Petreanu, Huber et al. 2007, Harris and Shepherd 2015). The model was implemented in versions 2.12.0 and 2.16.0 of the neural simulator NEST (Fardet 2020) without MPI. A paper describing some results obtained with the model is currently in preparation.

Using the model

To access the files described below, download and extract the file multiarea_ht_model_3.1.zip.Then copy the files in folder ht_neuron to the NEST folder "models", and the files in ht_kernel to the NEST folder "nestkernel" before compiling NEST. This will replace the model ht_neuron supplied with NEST. The new ht_neuron has a couple of extra settable parameters:

(1) b_theta. This affects the equilibrium threshold (θ_{eq}) such that below –55 mV $\theta_{eq} = \theta_{eq}(-55\text{mV}) + \text{b\_theta}*(V+55)$, where $\theta_{eq}(-55\text{mV})$ is given by the ht_neuron settable parameter theta_eq. Default value: b_theta = 0.0.

(2) tau_D_KNa_75. This affects the time constant τ_D of the depolarization-activated potassium current I_KNa. For $V \leq -75$ mV, τ_D = tau_D_KNa_75. For $V \geq -55$ mV, τ_D = tau_D_KNa. For $-75 < V < -55$ mV, τ_D varies linearly between tau_D_KNa_75 and tau_D_KNa. Default value: tau_D_KNa_75 = tau_D_KNa = 1250.0 ms.

Note also that the number of activation particles for I_NaP and I_T is unity, in contrast to the ht_neuron supplied with NEST. Full details of ht_neuron, including all other settable parameters, can be found in the NEST documentation.

The network model is run with the python script Run_multiarea_HT_network_3.1.py. This calls the scripts Build_multiarea_HT_network_3.1.py and Simulate_multiarea_HT_network_3.1.py. The first of these builds the network and adds recorders and any synaptic bombardment. The second controls the simulation and saves the recorded data to an output folder. The files

Functions_multiarea_HT_network_3.1.py and User_functions_multiarea_HT_network_3.1.py must also be present (although the latter can have a different name; see below).

Model output

Simulation results are saved to an output folder (default name “output”) which in turn contains subfolders. Subfolder names begin with “L” or “R” followed by:

C#L4 – cortical layer 4
C#L23 – cortical layers 2/3
C#L56 – cortical layers 5/6
R# – reticular thalamic nucleus
TC# – thalamic core cells
TM# – thalamic matrix cells
TI# – thalamic inhibitory cells

with # = 1-3, indicating the brain area. Each subfolder contains excitatory and inhibitory cell spike times in files ...Ex_sp.txt and ...In_sp.txt (if you elected to save these data), as well as membrane potential (V_m) data in files ..._V_m.txt. The latter contains mean V_m data for the layer as well as V_m for some randomly chosen cells. Spike times for these cells are in the ...Ex_Vm_sp.txt and ... In_Vm_sp.txt files. Mean cellular firing rates per sampling interval are in files …ExFR.txt and …InFR.txt, while topographic data for V_m are in files …_V_m_top.txt (if selected). Time series and topographic data for other recordables can also be saved (to files ..._gAMPA.txt, ..._gAMPA.top etc.; see below). In addition, ‘snapshots’ of settable parameters and V_m for each layer can be saved to ...snap_t=#s.txt files. The files ..._ ExIsynSum.txt contain the sum of synaptic currents for excitatory cells. The latter are used to calculate three proxies of LFP as described by Mazzoni, Linden et al. (2015): (1) $-I_{GABA} = -I_{GABA\_A}-I_{GABA\_B}$; (2) $\Delta I = I_{GABA\_A}-I_{GABA\_B}- I_{AMPA}+I_{NMDA}$; (3) $RWS = \alpha(I_{GABA\_A}+I_{GABA\_B})- I_{AMPA}(t-\tau)+I_{NMDA}(t-\tau)$, where $\tau = 6$ ms and $\alpha = 1.65$ (different values for τ and α may be set in network_parameters_3.1.txt). [The signs of the currents are reversed compared with Mazzoni, Lindén et al. (2015) because they used the opposite sign convention.] These proxies are saved to ...LFP.txt files as centered data, i.e. $y - \bar{y}$ where y is the proxy. Average firing rates are in output\Pooled_firing_rates.txt and some miscellaneous data will be found in output\output.txt. Connection dictionaries can be saved to output\connection_dictionaries.txt.

Model input

Various input files are required for the model to work. They may be given arbitrary names (see below) but here we use their default names. A model parameters file such as model_parameters_3.1.tx is needed to define parameter values for model neurons, synaptic bombardment and synaptic depression; parameter names must not be changed. Another file (e.g. neuron_layers_3.1.txt) defines the neuron layers, specifying the number of rows and columns and the number of each type of neuron model to be placed at a node. Where the latter is a noninteger value, it is interpreted as a proportion of the nodes, chosen randomly. A connection file such as network_connections_3.1.txt defines connectivities for excitatory (E) and inhibitory (I) cells in the various layers and areas. Connections are made probabilistically within a mask of radius *Radius*, using a Gaussian kernel with maximum probability *Pmax*. The Gaussian standard deviation is given by *sigma* = *sigma*0 + *beta***Radius*, where *sigma*0 is set in the file network_parameters_3.1.txt. Synaptic weights (*Weight*) are also given in network_connections_3.1.txt, together with mean transmission delays in ms (*Delay(ms)*) and their standard deviations (*SD(delay)*). You may also add a *SD(weight)* column if you wish; then, like delays, weights will be drawn from a Gaussian distribution.

Global values of *Radius*, *Pmax*, *beta*, *Weight*, *SD(weight)*, *Delay(ms)*, and *SD(delay)*. may also be set in network_parameters_3.1.txt, but any values set in network_connections_3.1.txt take precedence. A multiplier for *Radius* may also be set in network_parameters_3.1.txt, together with some other network parameters. In particular, the fraction of layer 5/6 cells that are intrinsic bursters (IB) and the fraction of these cells that are intrinsically active network drivers (ND) may be set (Lőrincz, Gunner et al. 2015). Parameters for IB and ND cells are set in the model_parameters_3.1.txt file as usual, but they are not NEST neuron models in their own right (you will not find them at any node). Rather their parameters are used to adjust those of certain ExCort# neurons in L56 (selected at random). A global coefficient of variation for intrinsic and synaptic conductances may also be set. Finally parameters describing an initial triangular V_m distribution can be set in network_parameters_3.1.txt. Initial V_m values for all the cells in the network will be chosen at random from this distribution.

Controlling simulations

The file simulation_parameters_3.1.txt may be used to control various simulation parameters:

```
Threads: 4                                      (number of local threads)
Array_job_env_variable: SLURM_ARRAY_TASK_ID     (for array jobs on HPCs)
RNG_seed: -123456                               (negative to select a 'random' seed)
Independent_simulations_(yes/no): yes           (yes = different RNG seeds)
Resolution(ms): 0.25                            (iteration step size)
Sampling_interval(ms): 1.0                      (for recording data)
Simulation_interval(ms): 100.0                  (argument for nest.Simulate())
Between_saves_interval_(ms): 100.0              (interval between data saves)
Rebuild_network(yes/no): yes                    (for successive simulations; see below)
Reset_network(yes/no): yes                      (at the start of each simulation)
Store_connection_descriptors_(yes/no): no       (see User_functions...3.1.py)
Settling_time(s): 3.0                           (no data recorded during this time)
Zero_synaptic_conductances_during_settle_(yes/no): no (yes = V→RMP)
Run_time(s): 9.0                                (settling time excluded)
Record_all_spike_times: yes                     ("yes" for ..Ex_sp.txt and ...In_sp.txt files)
Time_series_for_variables: V_m                  (NEST ht_neuron model recordables*)
Number_of_cells_for_time_series: 3              (for ..._V_m.txt, …_g_AMPA  etc. files)
Measure_of_location_(mean/median): median
Measure_of_dispersion_(sd/quartiles): quartiles
```

```
Save_topographic_data_(yes/no): no (generate …_ top.txt files)
Topographics_for_variables: V_m       (NEST ht_neuron model recordables*)
Topographics_for_layers: LC2L23 LC2L4 LC2L56
Topographic_sampling_interval(ms): 10.0
Average_over_sampling_interval_(yes/no): no
Take_snapshots_at_times_(s): 0 9      (blank = none)
TMS(yes/no): no                       (toggle TMS)
TMS_target: LC1L23 LC1L4 LC1L56       (target layers for TMS pulse)
TMS_activates_interneurons(yes/no): yes
Number_of_TMS_runs: 30                (TMS pulses will be evenly distributed
On_time(s): 3.1                       between On_time and Off_time. These
Off_time(s): 6.0                      times also used for pooled firing rates.)
Antialias_filter(yes/no): no          (Filter at less than half the sampling rate
Filter_type(butter/bessel): butter    to avoid aliasing (if this is a problem) but at
Filter_cutoff_frequency_(Hz): 300     the cost of increased memory usage.)
Input_folder:                         (default = current working folder)
Prefix_for_input_files:               (optional for each of the following 7 files)
Model_parameters_file_(.txt): wake_model_parameters_3.1
Network_parameters_file_(.txt): network_parameters_3.1
Neuron_layers_file_(.txt): neuron_layers_3.1
Network_connections_file_(.txt): wake_network_connections_3.1
Neuron_parameters_file_(.txt): wake-sleep_neuron_parameters_3.1
Synapse_parameters_file_(.txt): wake-sleep_synapse_parameters_3.1
User_functions_file_(.py): User_functions_multiarea_HT_network_3.1
Output_folder_(relative_to_input_folder): (default = "output")
Save_connection_dictionaries_to_output_folder(yes/no): yes
Copy_input_files_to_output_folder(yes/no): yes
Copy_python_scripts_to_output_folder(yes/no): yes
Delete_output_folder_if_already_present(yes/no): yes
Compress_output_folder(yes/no): no
```

*V_m g_AMPA g_NMDA g_GABA_A g_GABA_B I_NaP I_KNa I_h I_T theta
(space-delimited as shown here; or choose "all").

The input folder may be specified relative to the current working folder or as an absolute path. It may also be passed as a command line argument to Run_multiarea_HT_network_3.1.py (without quotes). In the latter case simulation_parameters_3.1.txt will be imported from the input folder; otherwise it is always loaded from the current working folder. Alternatively give the name of a file ending with .txt containing a list of input folders (one per line) to perform several simulations consecutively (or as an array job if available; leave `Array_job_env_variable` blank if an array job is not desired).

If TMS is selected, TMS pulse times will be saved to output\TMS_times.txt and recorded data will be saved in subfolders output_1, output_2... A TMS pulse is simulated by the simultaneous firing of cells in the target area (Pashut, Magidov et al. 2014). Regardless of whether TMS is active or not, pooled firing rates for each neuronal layer will be calculated for time intervals 0-On_time(s), On_time(s)-Off_time(s), and Off_time(s)-Run_time(s) and stored in output\pooled_firing_rates.txt. Firing-rate statistics are not be calculated but you can do that yourself by recording all spike times.

If antialiasing is selected, a sampling interval of `Resolution(ms)` will be used. Then at the end of a simulation interval the resulting time series will be low pass filtered and down sampled to a sampling interval of `Sampling_interval(ms)`. Because of increased memory usage, it may be necessary to reduce `Between_saves_interval(ms)` when using an antialias filter.

Modifying model parameters

The file neuron_parameters_3.1.txt is used to change neuron model parameters during a simulation. Here is an example (supplied as wake-sleep_neuron_parameters_3.1.txt):

```
Layer    Model     Parameter     Times(s): 3        4        9
*        *         g_KL            1.05    1.05     1.6      1.6
LC       ND        g_KL            1.19    1.19     1.6      1.6
RC       ND        g_KL            1.19    1.19     1.6      1.6
T        Ex        g_KL            1.04    1.04     1.6      1.6
T        In        g_KL            1.049   1.049    1.6      1.6
LR       In        g_KL            1.044   1.044    0.85     0.85
RR       In        g_KL            1.044   1.044    0.85     0.85
LC       *         g_peak_NaP      0.5     0.5      0.95     0.95
RC       *         g_peak_NaP      0.5     0.5      0.95     0.95
LC       IB        g_peak_NaP      0.5     0.5      1.2      1.2
RC       IB        g_peak_NaP      0.5     0.5      1.2      1.2
LC       ND        g_peak_NaP      0.5     0.5      1.0      1.0
RC       ND        g_peak_NaP      0.5     0.5      1.0      1.0
T        Ex        g_peak_NaP      0.35    0.35     1.0      1.0
T        In        g_peak_NaP      0.5     0.5      1.0      1.0
LC       *         g_peak_AMPA     0.025   0.025    0.0375   0.0375
RC       *         g_peak_AMPA     0.025   0.025    0.0375   0.0375
LC       *         g_peak_KNa      0.5     0.5      2.0      2.0
RC       *         g_peak_KNa      0.5     0.5      2.0      2.0
C        Ex        CEx_bomb_rate   80.0    80.0     0.0      0.0
C        In        CIn_bomb_rate   40.0    40.0     0.0      0.0
T        Ex        TEx_bomb_rate   300.0   300.0    0.0      0.0
T        In        TIn_bomb_rate   100.0   100.0    0.0      0.0
LR       In        Ret_bomb_rate   250.0   250.0    0.0      0.0
RR       In        Ret_bomb_rate   250.0   250.0    0.0      0.0
```

The first line must be present as shown. “Parameter” may be an ht_neuron settable (such as those in model_parameters_3.1.txt) or a synaptic bombardment rate in Hz (CEx_bomb_rate, CIn_bomb_rate, TEx_bomb_rate, TIn_bomb_rate and Ret_bomb_rate, where C = cortex, T = thalamus and Ret = reticular nucleus). “Layer” must be a string that identifies one or more of the layers in neuron_layers_3.1.txt. Similarly “Model” must be a string that identifies one or more of the models in model_parameters_3.1.txt. You may also enter “*” if all layers or models are to be targeted for parameter changes. If a layer or model is specified more than once, the last specification takes precedence. The entries in the “Layer” and “Model” columns have no effect for synaptic bombardment rates. “Times(s)” may contain one value (v_0) or several (v_0 v_1 v_2 ...). v_0 sets the parameter value at the start of the simulation (t = 0). If additional values are present, they must be accompanied by times (t_1 t_2 ...) following “Times(s):”. The parameter will be ramped from v_i to v_{i+1} over the time interval t_i to t_{i+1} (actually ‘staircased’ with a time step equal to `Between_saves_interval_(ms)`). The above example causes a transition from wake to sleep during the interval 3-4 s (compare parameter values in wake_model_parameters_3.1.txt and sleep_model_parameters_3.1.txt). In this particular example no synaptic input is needed in sleep because activity is sustained by IB and ND cells. The size of these L56 burster populations may be set in network_parameters.txt.

Similarly, synaptic parameters (weight, delay) may be set in the file synapse_parameters_3.1.txt. Here is an example (supplied as wake-sleep_synapse_parameters_3.1.txt) which must accompany the above listing of wake-sleep_neuron_parameters_3.1.txt in order to get a correct wake/sleep transition (for clarity the listing has been split into two, but in reality there are only five lines in the file):

```
SourceLayer   SourceModel   TargetLayer   TargetModel   Transmitter
TC            Ex            L4            *             AMPA
TC            Ex            L56           *             AMPA
TM            Ex            L23           *             AMPA
TM            Ex            L56           *             AMPA

Parameter   Times(s):   3       4      9
weight      13.5        13.5    9      9
weight      3.375       3.375   2.25   2.25
weight      2.25        2.25    1.5    1.5
weight      2.25        2.25    1.5    1.5
```

These adjustments to thalamocortical AMPA synaptic weights are necessary to compensate for the increase in the cortical AMPA synaptic conductance (see the wake-sleep_neuron_parameters_3.1.txt listing above). The result is no change in the effective AMPA conductance for thalamocortical synapses. The parameter values in wake-sleep_neuron_parameters_3.1.txt and wake-sleep_synapse_parameters_3.1.txt are meant to reflect the action of the neuromodulator acetylcholine (McCormick 1992, Gil, Connors et al. 1997, Mittmann and Alzheimer 1998, Kimura, Fukuda et al. 1999, Hsieh, Cruikshank et al. 2000, Scammell, Arrigoni et al. 2017).

User functions

The file User_functions_multiarea_HT_network_3.1.py contains functions that, as supplied, do nothing. Of course the user may insert his/her own code to change parameters and/or analyse results. Multiple simulations may also be run using different parameter values. The use of these functions is described in User_functions_multiarea_HT_network_3.1.py, and some examples are given in User_functions_multiarea_HT_network_3.1_examples.py. Code from the latter file should be copied into User_functions_multiarea_HT_network_3.1.py in order to run these examples. To speed up simulations for purposes of illustration, the row/column divisor in neuron_parameters_3.1.txt can be set to 3. Of course `Run_time(s)` and `Settling_time(s)` can also be reduced, but be sure that

`Run_time(s)` is consistent with the times given in wake-sleep_neuron_parameters_3.1.txt and wake-sleep_synapse_parameters_3.1.txt) when running simulations of wake → sleep transitions. Note that the function `user_models()`, in conjunction with a small value of `Simulation_interval(ms)`, may be used to make rapid parameter changes during a run. This is preferable to reducing `Between_saves_interval_(ms)`, because if the latter is made less than about 30 ms, an increasingly heavy price will be paid in terms of execution speed.

Acknowledgements

This project was conceived by Prof. Johan F. Storm, director of the Brain Signaling Group at the University of Oslo (UiO), Norway, as part of the Human Brain Project (HBP). The model was developed on various computers using one node with at least 32 GB RAM and 12 or 16 cores: (1) the Abel supercomputer at UiO (now defunct); (2) the INDACO supercomputer at the University of Milan (thanks to Thierry Nieus); (3) the Saga supercomputer (UNINETT Sigma2 - the National Infrastructure for High Performance Computing and Data Storage in Norway); and (4) an *hp* Z640 workstation. The principal developer of the present model was the author, but the development would not have been possible without the help of Hans Ekkehard Plesser of the NEST Initiative (Norwegian University of Life Sciences and Jülich Research Centre). I am also grateful to the following people for helpful discussions and advice: Andre S. Nilsen (UiO), Bjørn E. Juel (UiO), Alessandro Arena (UiO), Sean Hill (Krembil Centre for Neuroinformatics), Thierry Nieus (University of Milan), Tom Bugnon (University of Wisconsin–Madison) and Johan F. Storm (UiO). Andre also maintained the *hp* Z640, and his implementation of an earlier two-area version of the model can be found at https://github.com/andresni/NYP. Thanks also to Sofia Anderholm Strand (UiO) for guidance about reporting progress on model development within HBP. This project has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 785907 (Human Brain Project SGA2).

References

Douglas, R. J. and K. A. C. Martin (2004). Neuronal circuits of the neocortex. *Annual Review of Neuroscience* **27**: 419-451.
Esser, S. K., S. Hill and G. Tononi (2009). Breakdown of Effective Connectivity During Slow Wave Sleep: Investigating the Mechanism Underlying a Cortical Gate Using Large-Scale Modeling. *Journal of Neurophysiology* **102**: 2096-2111.
Fardet, T. e. a. (2020). *NEST 2.20.0*.
Gil, Z., B. W. Connors and Y. Amitai (1997). Differential Regulation of Neocortical Synapses by Neuromodulators and Activity. *Neuron* **19**: 679-686.
Harris, K. D. and G. M. G. Shepherd (2015). The neocortical circuit: themes and variations. *Nature Neuroscience* **18**: 170-181.
Hill, S. and G. Tononi (2005). Modeling, sleep and wakefulness in the thelamocortical system. *Journal of Neurophysiology* **93**: 1671-1698.
Hsieh, C. Y., S. J. Cruikshank and R. Metherate (2000). Differential modulation of auditory thalamocortical and intracortical synaptic transmission by cholinergic agonist. *Brain Research* **880**: 51-64.

Kimura, F., M. Fukuda and T. Tsumoto (1999). Acetylcholine suppresses the spread of excitation in the visual cortex revealed by optical recording: possible differential effect depending on the source of input. *European Journal of Neuroscience* **11**: 3597-3609.
Lőrincz, M. L., D. Gunner, Y. Bao, W. M. Connelly, J. T. R. Isaac, S. W. Hughes and V. Crunelli (2015). A Distinct Class of Slow (~0.2–2 Hz) Intrinsically Bursting Layer 5 Pyramidal Neurons Determines UP/DOWN State Dynamics in the Neocortex. *The Journal of Neuroscience* **35**: 5442-5458.
Mazzoni, A., H. Linden, H. Cuntz, A. Lansner, S. Panzeri and G. T. Einevoll (2015). Computing the Local Field Potential (LFP) from Integrate-and-Fire Network Models. *Plos Computational Biology* **11**.
Mazzoni, A., H. Lindén, H. Cuntz, A. Lansner, S. Panzeri and G. T. Einevoll (2015) Computing the Local Field Potential (LFP) from Integrate-and-Fire Network Models. *Plos Computational Biology* **11**, e1004584 DOI: 10.1371/journal.pcbi.1004584.
McCormick, D. A. (1992). Neurotransmitter actions in the thalamus and cerebral cortex and their role in neuromodulation of thalamocortical activity. *Progress in Neurobiology* **39**: 337-388.
Mittmann, T. and C. Alzheimer (1998). Muscarinic Inhibition of Persistent Na+ Current in Rat Neocortical Pyramidal Neurons. *Journal of Neurophysiology* **79**: 1579-1582.
Pashut, T., D. Magidov, H. Ben-Porat, S. Wolfus, A. Friedman, E. Perel, M. Lavidor, I. Bar-Gad, Y. Yeshurun and A. Korngreen (2014). Patch-clamp recordings of rat neurons from acute brain slices of the somatosensory cortex during magnetic stimulation. *Frontiers in Cellular Neuroscience* **8**: Article 145.
Petreanu, L., D. Huber, A. Sobczyk and K. Svoboda (2007). Channelrhodopsin-2-assisted circuit mapping of long-range callosal projections. *Nature Neuroscience* **10**: 663-668.
Scammell, T. E., E. Arrigoni and J. O. Lipton (2017). Neural Circuitry of Wakefulness and Sleep. *Neuron* **93**: 747-765.
Wise, S. P. and E. G. Jones (1976). The organization and postnatal development of the commissural projection of the rat somatic sensory cortex. *Journal of Comparative Neurology* **168**: 313-343.