Implementaion of the Potjans Diesmann microcircuit model using the classical Galves-Löcherbach neuron model

Nilton Liuji Kamiji, Christophe Pouzat, Aline Duarte, Antonio Roque, Antonio Galves

July 27, 2022

Contents

List of Figures

1 Introduction

Implementation of the Potjans and Diesmann **?** microcircuit model using the classical Galves-Löcherbach neuron model **?** in discrete time.

2 Model definition

2.1 The original PD model

The PD model (Potjans and Diesmann, 2014) consists of 8 neuronal populations, representing excitatory and inhibitory populations of 4 cortical layers, namely, layers 2/3 (L23e/i), 4 (L4e/i), 5 (L5e/i) and 6 (L6e/i), where e denotes excitatory and i inhibitory. The number of neurons of each population is shown in table 1, which totalizes 77169 neurons.

Table 1: Number of neurons per cortical layer

L23e	L23i	L4e	L4i	L5e	L5i	L6e	L6i
20683	5834	21915	5479	4850	1065	14395	2948

2.1.1 Membrane potential dynamics

All neurons are described with the same LIF (leaky integrate-and-fire) dynamics (eq. (1))

$$\begin{aligned} \frac{dV_m(t)}{dt} &= \begin{cases} 0 & \text{if neuron is refractory} \\ -\frac{V_m(t)}{\tau_m} + \frac{S^i_{syn}(t)}{C_m} & \text{otherwise} \end{cases} \\ \frac{dI_{syn}(t)}{dt} &= \begin{cases} 0 & \text{if neuron is refractory} \\ -\frac{I_{syn}(t)}{\tau_{syn}} + \sum_j W_j & \text{otherwise} \end{cases} \\ \text{if } V^i_m(t) \geq V_{th} & \quad \text{neuron spiked and refractory} \end{aligned} \tag{1}$$

where $V_m(t)$ is the membrane potential, $\tau_m = 10$ ms is the membrane time constant representing the leakage effect, $C_m = 250$ pF the membrane capacitance, $V_{th} = 15$ mV the threshold potential in which a neuron fires when its membrane potential exceeds this value, $I_{syn}(t)$ the membrane current elicited by synaptic inputs from pre-synaptic neurons j (W_j), and $\tau_{syn} = 0.5$ ms the synaptic time constant indicating the dynamics of synaptic receptor channel. The refractory period (τ_{ref}), the period in which a neuron stays silent after emmiting a spike, was 2 ms.

2.1.2 The graph of interaction

Neurons are connected following the connection probability between neuronal populations shown in table 2. The resulting graph of interaction is a directed graph with nearly 300 million edges.

Table 2: Connection probability between neuronal populations

		from							
		L23e	L23i	L4e	L4i	L5e	L5i	L6e	L6i
	L23e	0.101	0.169	0.044	0.082	0.032	0.0	0.008	0.0
	L23i	0.135	0.137	0.032	0.052	0.075	0.0	0.004	0.0
	L4e	0.008	0.006	0.050	0.135	0.007	0.0003	0.045	0.0
to	L4i	0.069	0.003	0.079	0.160	0.003	0.0	0.106	0.0
	L5e	0.100	0.062	0.051	0.006	0.083	0.373	0.020	0.0
	L5i	0.055	0.027	0.026	0.002	0.060	0.316	0.009	0.0
	L6e	0.016	0.007	0.021	0.017	0.057	0.020	0.040	0.225
	L6i	0.036	0.001	0.003	0.001	0.028	0.008	0.066	0.144

Moreover, synaptic connections between neurons j and i are described by the synaptic weights ($w_{j\to i}$) and transmission delay ($d_{j\to i}$), drawn from a clipped normal distribution, and are different for excitatory and inhibitory synapses as shown in table 3. dt is the simulation time step, indicating that synapse transmission requires at least one time step.

Table 3: Synaptic weight and delay. Synaptic weights are clipped at 0, and synaptic delays are clipped at simulation step ($d_t = 0.1$ ms)

excitatory	w_e	$\mathcal{N}(\mu = 87.8, \sigma = 8.8)$ pA; $w_e > 0$
	d_e	$\mathcal{N}(\mu = 1.5, \sigma = 0.75)$ ms; $d_e \geq dt$
inhibitory	w_i	$\mathcal{N}(\mu = -351.2, \sigma = 35.1)$ pA; $w_i < 0$
	d_i	$\mathcal{N}(\mu = 0.8, \sigma = 0.4)$ ms; $d_i \geq dt$

Remark 1 A synaptic weight of 87.8 pA will elicit a maximum membrane potential change of approximately 0.15 mV.

Remark 2 The number of synapses (K) between any two populations are calculated from:

$$K = \frac{\log(1 - C_a)}{\log(1 - 1/(N_{pre}N_{post}))}, \tag{2}$$

where C_a is the connection probability taken from table 2, N_{pre} and N_{post} the number of neurons in the pre- and post-synaptic population, respectively, taken from table 1. The K pairs are then randomly chosen, so that multiple synapses with different parameter values for any pair can occur. This phenomenon is know as multapses.

Remark 3 The connection weight between neurons of L4e to L23e is doubled, that is, taken from a clipped normal distribution of $\mathcal{N}(\mu = 175.6, \sigma = 17.6)$ pA

2.1.3 Background input

The network is driven by a constant external input that can be described as Poissonian spike trains ($rate = 8$ Hz) or constant current. The number of external inputs is layer dependent as shown in table 4.

Table 4: Number of external inputs onto each cortical layer

L23e	L23i	L4e	L4i	L5e	L5i	L6e	L6i
1600	1500	2100	1900	2000	1900	2900	2100

Remark 1 In the case of Poissonian spike trains input, every neuron on each layer receives independent Poissonian spike trains with $rate = 8 \times n$, where n is the number of external inputs, which is the same as generating n Poissonian spike trains of 8 Hz and applying all of it to the neuron. Note that all input has fixed weight and delay values of 87.8 pA and 1.5 ms, respectively.

2.2 The GL model

2.2.1 Membrane Potential Process

The the dynamics (*membrane potential process* - **MPP**) of the neuron is described as:

$$V_t(i) = \sum_j W_{j\to i} \sum_{s=L_t^i}^{t-1} \exp\left(-\frac{t-s-1}{\tau_i}\right) X_{s-d_{j\to i}+1}(j), \tag{3}$$

where:

j The *presynaptic neuron*.

i The *postsynaptic neuron*.

$W_{j\to i} \in \mathbb{R}$ The *synaptic weight*, if $w_{j\to i} < 0$ the synapse is *inhibitory*, if $w_{j\to i} > 0$ it is *exitatory*.

$d_{j\to i} \in \mathbb{R}^+$ The *synaptic delay*, this accounts for the *propagation time* of the spike from the soma of neuron j to the the synaptic terminal on neuron i, as well as the time for the presynaptic vesicles to fuse upon spike arrival, the transmitter to diffuse and bind to the postsynaptic receptors and the latter to open.

L_t^i The *last spike time* of neuron i prior to time t.

τ_i The decay time constat of the leakage effect of the neuron.

$X_t(i)$ The *spike train* of neuron i taking value in $\{0,1\}$, if $X_t(i) = 1$ means neuron i fired at time t.

Note that eq. (3) can be described as a Markovian system as:

$$\begin{aligned} V_{t+1}(i) &= e^{-1/\tau_i} V_t(i) + \sum_j W_{j\to i} X_{s-d_{j\to i}+1}(j) \\ X_{t+1}(i) &= \begin{cases} 1 & \text{if neuron spiked; set } V_{t+1}(i) = 0. \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{4}$$

2.2.2 The firing probability function $\Phi(V_t(i))$

The firing probability function ($\Phi(V_t(i))$) was determined so that the firing rate of the neuron to constant current input (FI-curve) reproduces that of the LIF neuron of the original PD model.

We consider the following function for $\Phi(V)$

$$\Phi(V) = \begin{cases} 0 & \text{if } V \leq V_{rheo} \\ [\gamma(V - V_{rheo})]^r & \text{if } V_{rheo} < V < V_{sat} \\ 1 & \text{if } V \geq V_{sat} = V_{rheo} + 1/\gamma \end{cases} \tag{5}$$

Fig. 1 represents the FI-curve of the original LIF neuron (red) and that of the GL model (blue). The FI-curve for the GL neuron was obtained with $\gamma = 0.1$ mV^{-1} (i.e. there is a window of 10 mV for $0 < \Phi(V) < 1$), $r = 0.4$ and $V_{rheo} = 15$ mV (-50 mV in Fig. 2; will be corrected in future version), which is represented in Fig. 2.

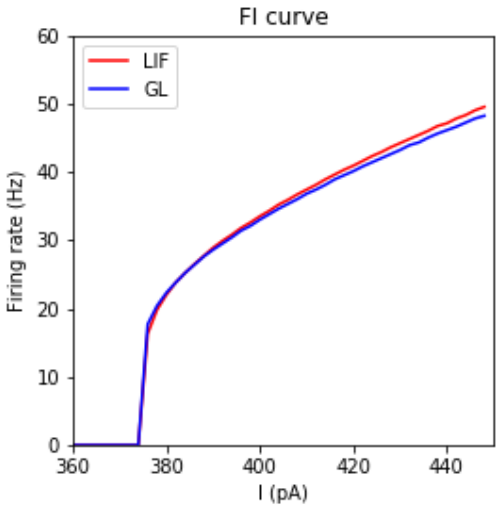


Figure 1: FI curve of LIF (red) and GL (blue) neuron model.

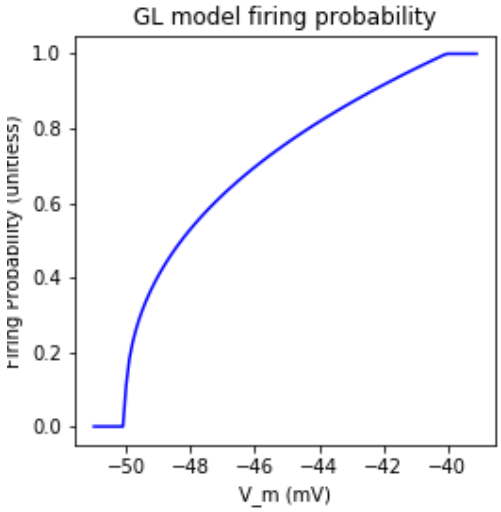


Figure 2: Firing rate ($\Phi(V)$) of the GL model.

In practice a sequence of Bernoulli random variable is used at each time step: $\{U_t(i)\}_{i \in I}$, $U_t(i) \overset{IID}{\sim} \mathcal{B}$ and the $X_{t+1}(i)$ are given by:

$$X_{t+1}(i) = \begin{cases} 1 & \text{if } U_t(i) \leq \phi_i\left(V_t(i)\right) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

3 C++ implementation

TODO: write about the code

3.1 Compile

```
unzip dGLPD_delta.zip
mkdir bin
cd bin
cmake ../src
make
```

3.2 Execution time

Execution time was measured using time command. I have also measured simulation time of the original NEST implementation, however, only NEST 2.16.0, which was released on 2018-08-21, allowed to execute using a single process. All other NEST versions has restriction on the number of connections per virtual-process, requiring at least 3 virtual processes to simulate the PD model in full scale. When using a single process, our code was much faster than NEST 2.16.0. On the other hand, NEST 2.20.1 had a great improvement when using 4 virtual processes (OpenMP) when compared to NEST 2.16.0. However, if we compare user time, our code is comparable to that of NEST 2.20.1, suggesting that if similar approach is used to paralelize the code, we may obtain even better performance.

3.2.1 Simulation time step of 1.0 ms

```
time ./dGLPD_delta 11111 1000 1.0
```

```
Random Number Generator Engine: xoroshiro128+
seed: 11111, Simulation time: 1000 ms, Simulation time step: 1 ms
prepare: 372.762 s
simulate(1000): 155.432 s

real    8m49.235s
user    8m26.601s
sys     0m21.554s
```

3.2.2 Simulation time step of 0.1 ms

```
time ./dGLPD_delta 11111 1000 0.1
```

```
Random Number Generator Engine: xoroshiro128+
seed: 11111, Simulation time: 1000 ms, Simulation time step: 0.1 ms
prepare: 362.266 s
simulate(1000): 269.677 s

real    10m32.913s
user    10m11.589s
sys     0m20.644s
```

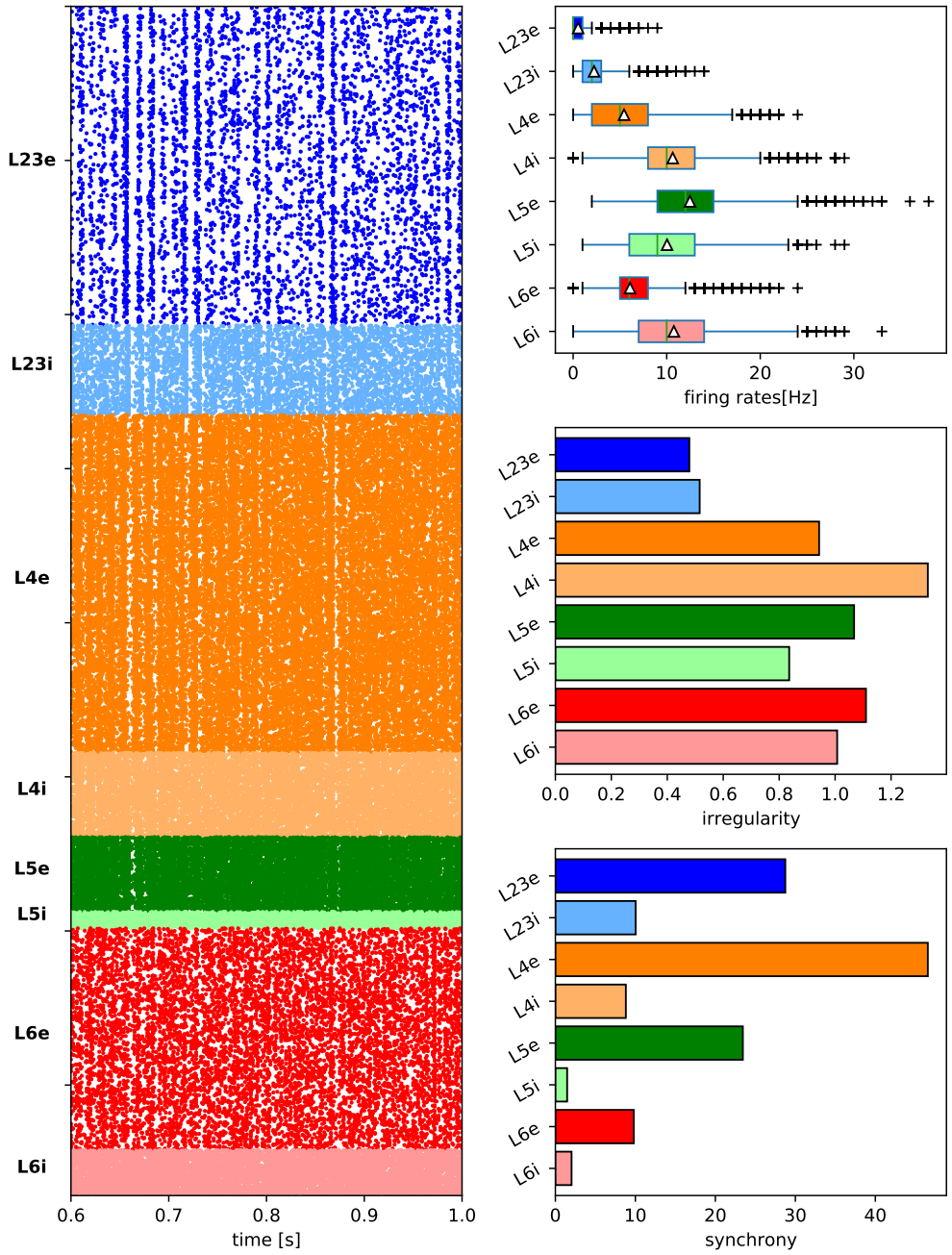


Figure 3: Measures obtained from 1 sec of simulation (dt = 1 ms). The first 100 ms was discarded due to its transient bvehavihour. Dark colors indicate excitatory population, and light colors indicate inhibitory population. **Left:** raster plot of the last 400 ms. **Top right:** mean firing rate per cortical layer. **Center right:** Irregularity measured as the coefficient of variation (CV) of the interspike interval (ISI). **Bottom right:** The synchrony index is computed as the variance of the spike count histogram divided by its mean.

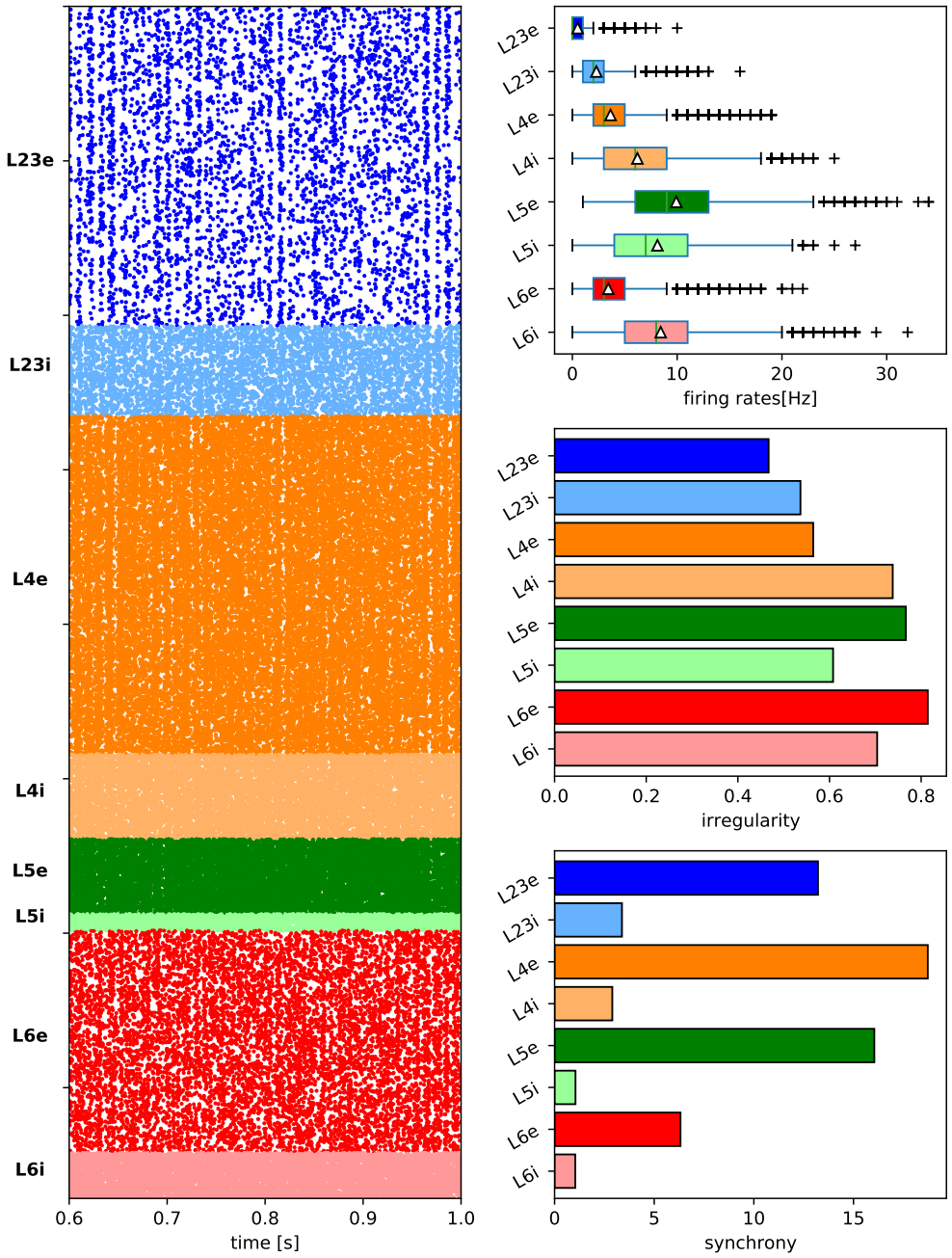


Figure 4: Measures obtained from 1 sec of simulation ($dt = 0.1$ ms). The first 100 ms was discarded due to its transient bvehavihour. Dark colors indicate excitatory population, and light colors indicate inhibitory population. **Left:** raster plot of the last 400 ms. **Top right:** mean firing rate per cortical layer. **Center right:** Irregularity measured as the coefficient of variation (CV) of the interspike interval (ISI). **Bottom right:** The synchrony index is computed as the variance of the spike count histogram divided by its mean.

3.2.3 NEST 2.16, simulation time step 0.1 ms

```
module load py36-nest/2.16.0
time python3 example.py
```

```
Spike detectors created
Thalamic input not provided
Poisson background input created
Recurrent connections are established
Poisson background input is connected
Spike detector connected
Time to create the connections: 224.70 s

May 04 09:26:07 NodeManager::prepare_nodes [Info]:
    Preparing 77185 nodes for simulation.

May 04 09:27:13 SimulationManager::start_updating_ [Info]:
    Number of local nodes: 77185
    Simulation time (ms): 1000
    Number of OpenMP threads: 1
    Number of MPI processes: 1

100 %: network time: 1000.0 ms, realtime factor: 0.0002

May 04 11:13:29 SimulationManager::resume [Info]:
    Simulation finished.
Time to simulate: 6442.45 s

real        111m18.500s
user        107m4.390s
sys        3m33.969s
```

3.2.4 NEST 2.20.1, simulation time step 0.1 ms

```
module load py36-nest/2.20.1
time python3 example.py
```

```
Spike detectors created
Thalamic input not provided
Poisson background input created
Recurrent connections are established
Poisson background input is connected
Spike detector connected
Time to create the connections: 58.93 s

May 04 09:14:18 NodeManager::prepare_nodes [Info]:
    Preparing 77233 nodes for simulation.

May 04 09:14:43 SimulationManager::start_updating_ [Info]:
    Number of local nodes: 77233
    Simulation time (ms): 1000
    Number of OpenMP threads: 4
```

```
    Number of MPI processes: 1

100 %: network time: 1000.0 ms, realtime factor: 0.0109

May 04 09:16:17 SimulationManager::run [Info]:
    Simulation finished.
Time to simulate: 119.92 s

real        3m2.829s
user        10m21.236s
sys        0m32.026s
```

3.3 Memory usage

For measuring memory usage, the code was compiled without optimization (-O0) with debug code (-g), and profiled with valgrind.

3.3.1 Simulation time step of 1.0 ms

```
cmake -DCMAKE_CXX_FLAGS=-g -Dwith-optimize=-O0 ../src
make
valgrind --tool=massif ./dGLPD_delta 1111 1000 1.0
ms_print ./massif.out.18029
```

```
--------------------------------------------------------------------------------
Command:            ./dGLPD_delta 12345 1000 1.0
Massif arguments:   (none)
ms_print arguments: ./massif.out.18029
--------------------------------------------------------------------------------


    GB
13.67^                                                                     :
     |                                                                 @:@@#############:
     |                                                                 @ @@#          :
     |                                                              @@::@ @@#          :
     |                                                           :@@@@ :@ @@#          :
     |                                                       ::::::@@@@ :@ @@#          :
     |                                                 ::::  :::@@@@ :@ @@#          :
     |                                        :::::::: ::  :::@@@@ :@ @@#          :
     |                                     @:    ::: ::  :::@@@@ :@ @@#          :
     |                                @@@@@:    ::: ::  :::@@@@ :@ @@#          :
     |                           :@:@@  @:    ::: ::  :::@@@@ :@ @@#          :
     |                        ::::@ @@  @:    ::: ::  :::@@@@ :@ @@#          :
     |                    ::::: ::@ @@  @:    ::: ::  :::@@@@ :@ @@#          :
     |                @:@::: : ::@ @@  @:    ::: ::  :::@@@@ :@ @@#          :
     |                @:@::: : ::@ @@  @:    ::: ::  :::@@@@ :@ @@#          :
     |                @:@::: : ::@ @@  @:    ::: ::  :::@@@@ :@ @@#          :
     |                @:@::: : ::@ @@  @:    ::: ::  :::@@@@ :@ @@#          :
     |          :::::::@:@::: : ::@ @@  @:    ::: ::  :::@@@@ :@ @@#          :
     |        @:     :@:@::: : ::@ @@  @:    ::: ::  :::@@@@ :@ @@#          :
     |  :@@:::@:     :@:@::: : ::@ @@  @:    ::: ::  :::@@@@ :@ @@#          :
   0 +-------------------------------------------------------------------------->Ti
```

```
  0                                                                    2.649
```

3.3.2 Simulation time step of 0.1 ms

```
cmake -DCMAKE_CXX_FLAGS=-g -Dwith-optimize=-O0 ../src
make
valgrind --tool=massif ./dGLPD_delta 12345 1000 0.1
ms_print ./massif.out.30417
```

```
--------------------------------------------------------------------------------
Command:            ./dGLPD_delta 12345 1000 0.1
Massif arguments:   (none)
ms_print arguments: ./massif.out.30417
--------------------------------------------------------------------------------


    GB
13.73^                                                                   :
     |                                :@@#########################################:
     |                                @@@#                                    :
     |                              ::@@@#                                    :
     |                            :@::@@@#                                    :
     |                          ::::@::@@@#                                   :
     |                        :: ::@::@@@#                                    :
     |                    :::::: ::@::@@@#                                    :
     |                    @  ::: ::@::@@@#                                    :
     |                 :::@  ::: ::@::@@@#                                    :
     |                :: :@  ::: ::@::@@@#                                    :
     |              :::: :@  ::: ::@::@@@#                                    :
     |             ::::: :@  ::: ::@::@@@#                                    :
     |           @:::::: :@  ::: ::@::@@@#                                    :
     |           @:::::: :@  ::: ::@::@@@#                                    :
     |          @@:::::: :@  ::: ::@::@@@#                                    :
     |          @@:::::: :@  ::: ::@::@@@#                                    :
     |    @::::@@:::::: :@  ::: ::@::@@@#                                    :
     |    @:  :@@:::::: :@  ::: ::@::@@@#                                    :
     | :::@:  :@@:::::: :@  ::: ::@::@@@#                                    :
   0 +-------------------------------------------------------------------------->Ti
     0                                                                   4.535
```

3.4 Comparison with the original (NEST) implementation

The measures shown in figs. 3 and 4 was obtained from 1s of simulation data. On the other hand, the measures on Fig. 6 of the original work **?** (fig. 5a) is based on simulation data of 60 s. For a direct comparison, 60 s. was simulated for both conditions ($dt = 1$ and 0.1 ms). Fig. 5b shows a re-simulation of the NEST code. Please note that synchrony is about 10 folds different to that of the orinal paper (fig. 5a), and I am working on to find out the reason. In addition, the synchrony using the implemented GLPD model (figs. 5c and 5d) was qualitatively similar to that of fig. 5b, and may have quantitatively differed

due to the slightly different synaptic dynamics (the original NEST implementation uses exponentially dacaying synaptic current, whereas GLPD uses delta synapses).

3.4.1 Simulation time step of 1 ms

```
time ./dGLPD_delta 98765 60000 1.0
```

```
Random Number Generator Engine: xoroshiro128+
seed: 98765, Simulation time: 60000 ms, Simulation time step: 1 ms
prepare: 370.132 s
simulate(60000): 12721.2 s

real        55m39.426s
user        52m58.152s
sys         1m4.037s
```

3.4.2 Simulation time step of 0.1 ms

```
time ./dGLPD_delta 98765 60000 0.1
```

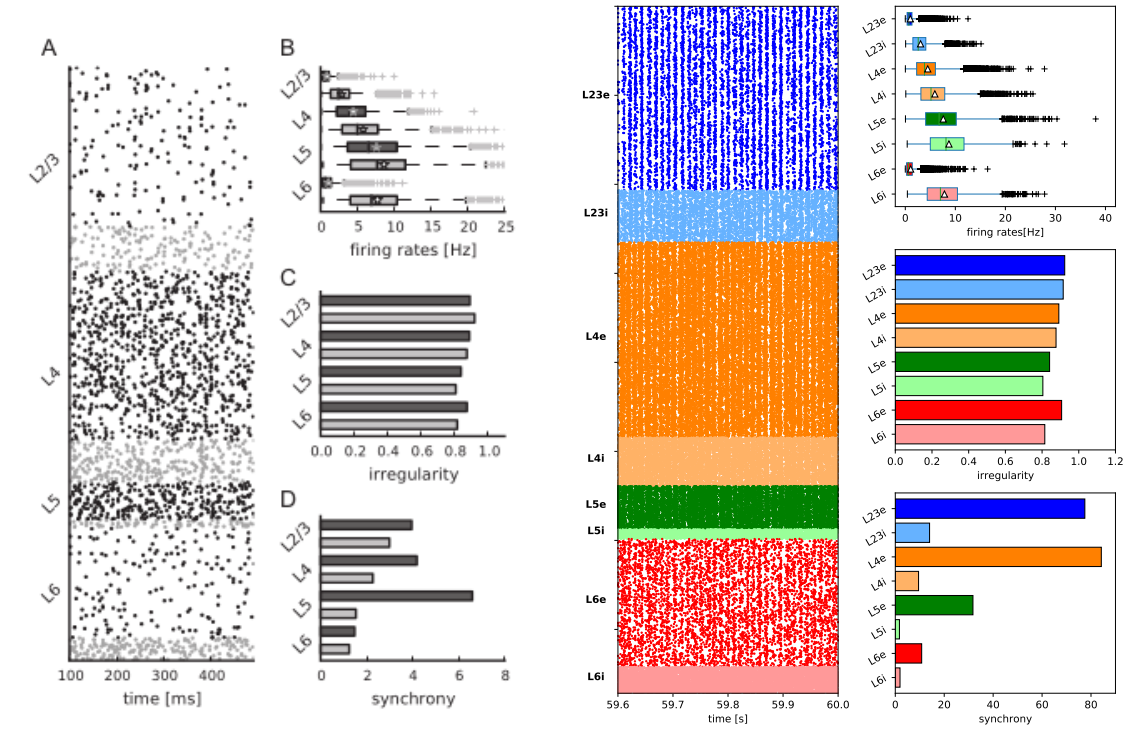
```
Random Number Generator Engine: xoroshiro128+
seed: 98765, Simulation time: 60000 ms, Simulation time step: 0.1 ms
prepare: 367.555 s
simulate(60000): 12721.2 s

real        218m9.308s
user        214m54.626s
sys         2m59.228s
```

3.5 Storage of V_m every 1.0 ms

By changing the last argument in line 81 of `./src/main_dGLPD_delta.cpp`, one can enable storage analog data such as $V_t(i)$ of every neuron at every 1.0 ms interval. In such case, the simulation time for $dt = 0.1$ and $dt = 1.0$ are both about 15 min. This is expected as writing data to file is a time-consuming process.

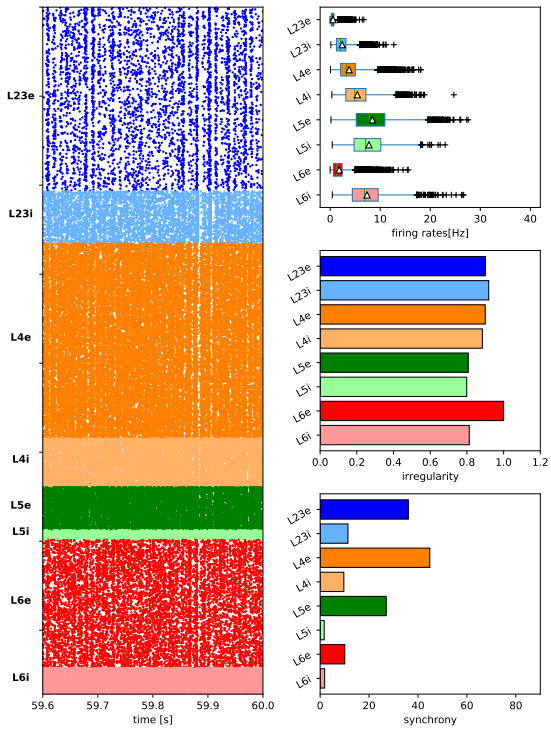
3.6 TODO List of future tasks

- cross-correlogram
- effect of considering the delay by a fixed delay of 1 time step for simulations with 1ms time step.

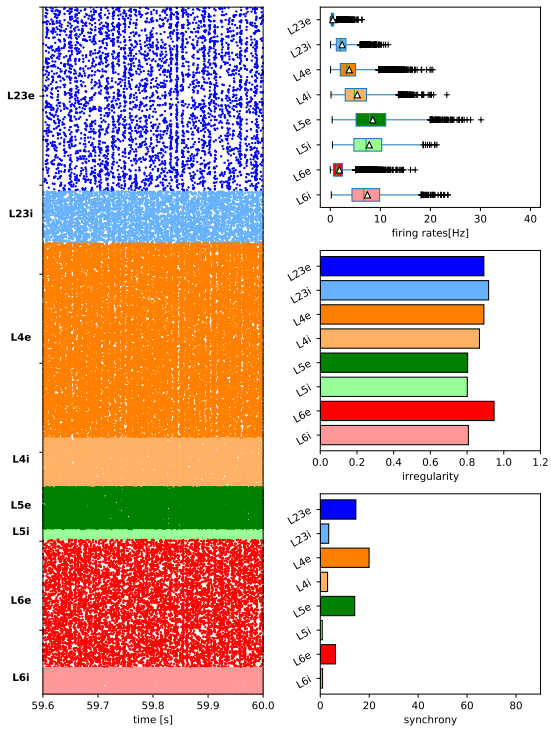


(a) NEST paper

(b) NEST sim



(c) $dt = 1$ ms



(d) $dt = 0.1$ ms

Figure 5: Measures obtained from 60 sec of simulation ($dt = 0.1$ ms). Dark colors indicate excitatory population, and light colors indicate inhibitory population. **Left:** raster plot of the last 400 ms. **Top right:** mean firing rate per cortical layer. **Center right:** Irregularity measured as the coefficient of variation (CV) of the interspike interval (ISI). **Bottom right:** The synchrony index is computed as the variance of the spike count histogram divided by its mean.