1 Introduction

2 Methods

2.1 Leaky integrate and fire model

To model electric neural properties a conductance based leaky integrate and fire model was used:

$$\tau_m \frac{dV}{dt} = (-V + E_m) + R_m(-I_{K^+} - I_{syn} + I_{noise}) \tag{1}$$

If the membrane voltage V reaches the threshold V_{thresh}, V is being resetted to the resting potential and a spike is counted. To model the decreased probabilty of another spike shortly after a previous one, an absolute refractory period was introduced, which deactivates incoming inputs from other neurons.
Additonally, a relative refractory period was used, which describes the increase of the K^+ conductance after an action potential and the subsequent exponential decay:

$$\frac{dg_K}{dt} = -\frac{g_K}{\tau_K} \tag{2}$$

$$I_K = g_k(V - E_K) \tag{3}$$

The g_K conductance is manually increased after a postsnyaptic spike emerged.

$$g_K \rightarrow g_K + \tilde{g}_K \tag{4}$$

$$\frac{dI_{noise}}{dt} = \frac{I_\mu - I_{noise}}{\tau_{OUP}} + \sigma\eta \tag{5}$$

Used values for the mentioned parameters are as follows:

Δt	0.1 ms	Simulation time step	
E_K	-77 mV	K^+ equilibiurm potential	
V_{tresh}	-55 mV	Spiking threshold	
E_m	-70 mV	Membrane resting potential	
$\tilde{g}_K$	5 nS	Additive K-conductance	
R_m	10^7 Ω	Membrane resistance	
τ_{syn}	0.335 ms	Synaptic time constant	Membrane resistance
τ_{OUP}	5 ms	Noise time constant	
σ	10^{-7}	Standard deviatoin of noise current	
I_μ	0.886 pA	Mean noise current	

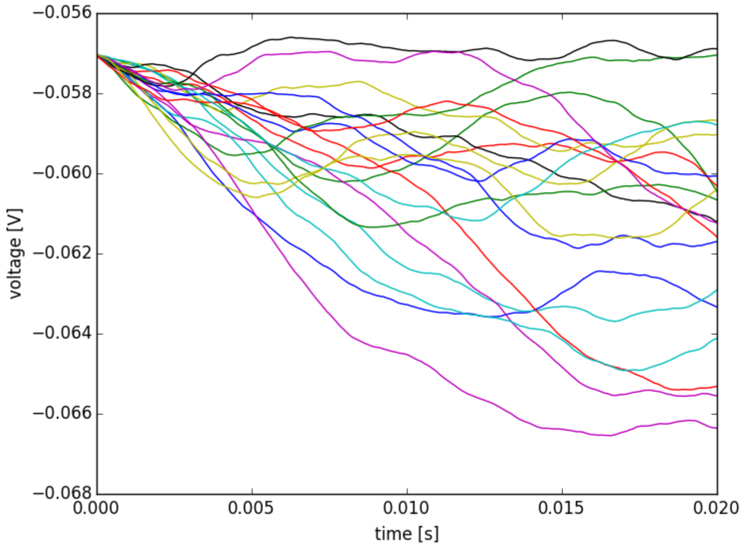

Figure 1

2.2 Implementation

The network is implemented using an object-oriented approach in Python 2.7. The two main classes are the model neuron and the model synapse.
Every model neuron has its own timing which updates instantaneously the conductance, current and voltage values. Every neuron creates a list of synapses, which all have the same strength. Every neuron from the previous group is connected to all neurons of the successive group, resulting in a fully connected feed forward network.
Background activity is introduced via a noise object, that computes Ornstein-Uhlenbeck-process-like current using the Euler-Maruyama method, for every neuron.

3 Resuneuronlts

4 Discussion

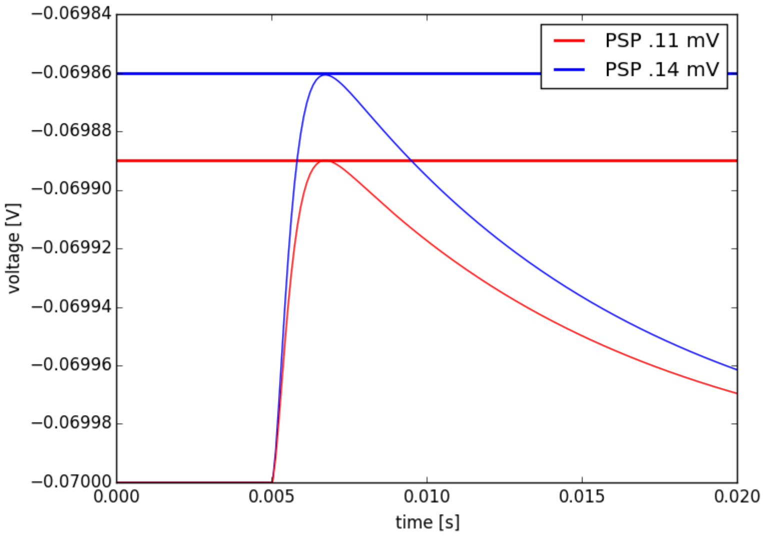

Figure 2

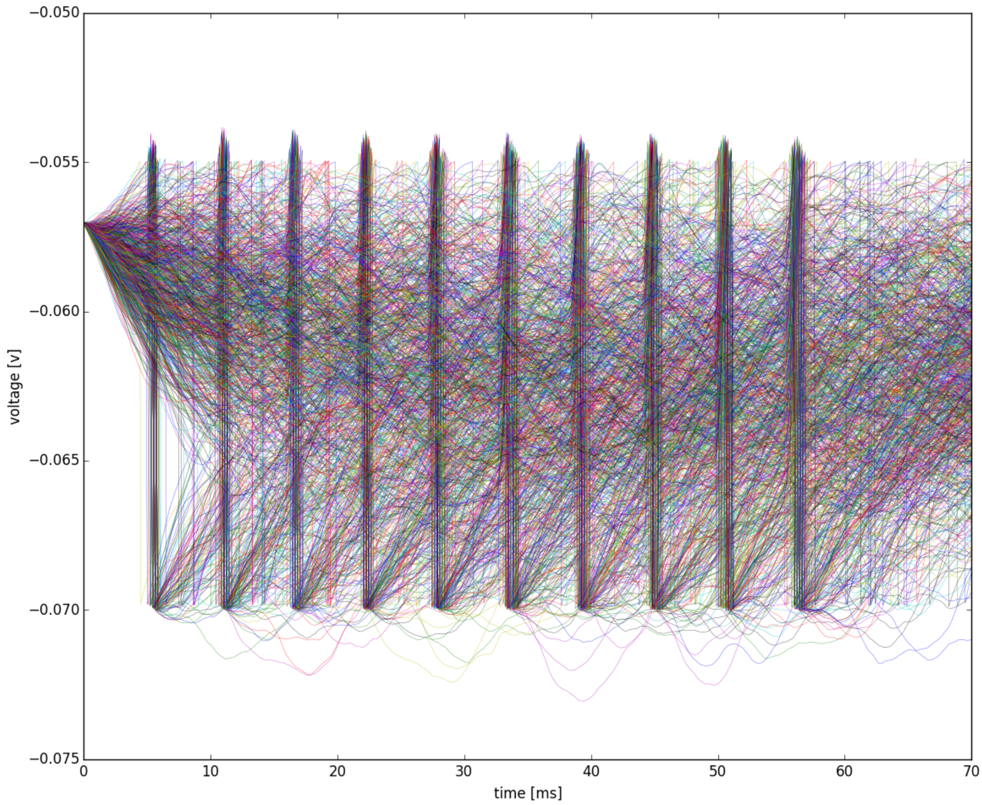

Figure 3

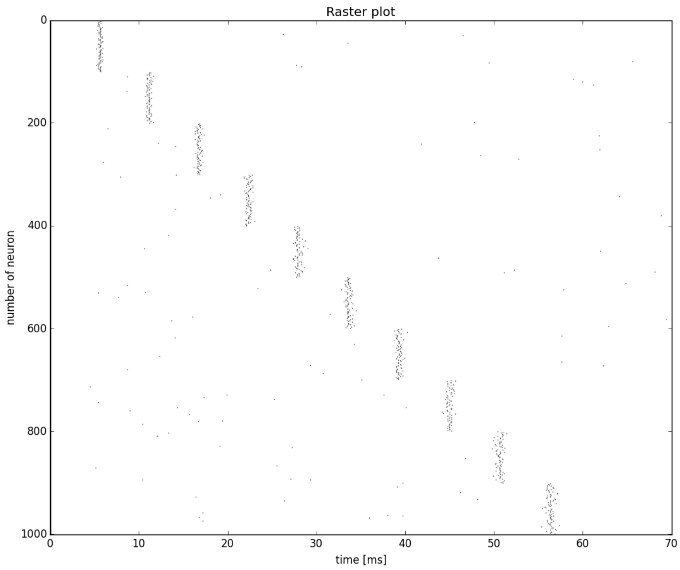
Raster plot
number of neuron
time [ms]
0
200
400
600
800
1000
0
10
20
30
40
50
60
70

Figure 4