Noise-induced excitability and associative memory in neuron-astrocyte networks

Lorenzo Amato
Sara Arias Blanco
Rodrigo Garcia Alcubilla
Paula Hernamperez Manso
Eleonora Nesterini

June 2020

Abstract

Investigation of the effects of associative memory in a neural network with a noise-induced excitability for artificial intelligence development by means of stochastic differential equation solving.

Contents

Chapter 1

Introduction

Understanding the neuron-astrocyte system behaviour seems to be a key factor in the path to improvement and development of artificial intelligence, as well as an interesting topic for nowadays neurological and physical research, since neural dynamics and noise-induced phenomena are an important field of the study of both the human brain and nonequilibrium systems.
The aim for this problem is the investigation of the effects of astrocytes in information processing for an induced-excitabilility neural system. So we will be experimenting matching up two different topics: noise-induced excitability and astrocyte organised associative memory.

On one hand it has been shown that multiplicative noise and coupling, if introduced in the lattice of the FitzHugh-Nagumo elements, can change a behaviour of the system from oscillatory to excitable [2].

On the other hand, it has been recently shown that astrocytes organise an associative memory if coupled to the neural network. Hence, it would be very interesting to investigate whether such an associative memory is possible in a neural network with a noise-induced excitability.

1.1 Problem Description

The problem will first consist on modelling a Noise-Induced Excitable system (NIE) by providing a code and a visualization and the transmission of plane or spiral waves through it. The second part of the problem will be adding astrocytes [3] and investigate whether astrocytes enable organization of an associative memory able to store, e.g., two patterns "1" and "0".

Chapter 2

Neurological background

In the mammalian brain neurons are arranged in layers and form specialized synaptic connections that make up different circuits.

2.1 What is a neuron?

A neuron is a nervous system cell whose role is the transmission of electrical and chemical impulses for information dissemination in mammal's bodies. In mammal brain, neurons are arranged in layers and connected to each other through synaptic sites, leading to different circuits.

2.2 What is an astrocyte?

An astrocyte is a nervous system type of cell. It is necessary for neuronal synapses to be formed and also crucial for synapsis regulation. For example they propagate intercellular Ca^{2+} waves in response to stimulation [1]. They have many receptors that are activated by adjacent neurons (mainly by elevating Calcium levels) although further investigation must be done. To sum up, astrocytes can detect the level of neuronal activity and release chemical transmitters to influence neuronal function. They also play an important role in regulation of neuronal excitability, so they have a bidirectional interaction with neurons [5]. And more importantly, astrocytes are thought to play a relevant role in memory processes so they are being studied for that purpose.

2.2.1 Problem goals

In this section the principles the problem is based on will be briefly exposed. The problem will focus on the importance of these two kind of cells and their interaction. For that end, two main models or suppositions will be implemented:

Neuron modelling neurons have been previously modeled as locally coupled elements following the FitzHugh-Nagumo model -FHN- that continuously oscillate unless there are certain thermal fluctuations. If that is the case, these lead to chemical reactions as well as electrical pulses that can be translated into multiplicative noise (further explained in section 3.1.Therefore modelling a Noise-Induced Excitable system will be the first step in our work.

Astrocyte & neuron interaction model neurons and astrocytes interaction has also been studied. We will use an interaction model that can be found in [1] which is based in a representation of the association of neurons by a network of networks. It would be interesting to see if one were to add the neuron modelling to the interaction model. That is why the second step in our work 4 will consist on making this junction.

Brief conclusions can be found within each chapter in the result sections.

Chapter 3

Modelling Noise-Induced Excitable systems

3.1 Modelling of the problem

We consider a system of coupled FitzHugh-Nagumo (FHN) elements in the oscillating state modeled by the following equations:

$$\begin{cases} \dot{u_i} = \dfrac{1}{\varepsilon}\left(F(u_i) - v_i\right) + D_u(\bar{u_i} - u_i) \\ \dot{v_i} = cu_i + d + v_i\xi_i + D_v(\bar{v_i} - v_i) \end{cases} \tag{3.1}$$

where

$$F(u) = \begin{cases} -1 - u + b & u \leq -\frac{1}{2}, \\ u + b & -\frac{1}{2} < u < \frac{1}{1+a}, \\ +1 - au + b & u \geq \frac{1}{1+a}. \end{cases} \tag{3.2}$$

In the neural context, $u(t)$ represents the neuron membrane potential, while $v(t)$ is related to the time-dependent conductance of the potassium channels in the membrane.

We consider a lattice of 40×40 neurons, so that the index i represents a single element, while $\bar{x}_i$ is the average of its neighbors' values. In particular, we assume that each neuron affects only its four neighbors with an intensity determined by the coupling constants: D_u and D_v.

ε is the time-scale ratio parameter; since it will be very small (≈ 0.01) it will determine the activation of $u(t)$ to be much faster than that of $v(t)$.

ξ represents the zero-main Gaussian noise: it appears in a multiplicative form in the equation of the inhibitor v .

Finally, we have constants whose values are: $a = 1$, $b = 2$, $c = 0.2$, $d = 0.075$.

3.2 Equation solving

3.2.1 Deterministic Lattice

First we will see the computation of the deterministic lattice **without noise** in time. Our goal is the depiction of the average field value $u(t)$ in time. The following considerations have to be accounted for to solve the deterministic lattice model:

- two variables following the equations in (3.1);
- a really small time step Δt shall be used in order to achieve system stability;
- initial conditions for the system are set to random numbers;
- periodic boundary conditions are chosen.

To solve the generic set of equations according to the Stratonovich interpretation:

$$\dot{x_i} = f(x_i) + \frac{D}{2d}\sum_j (x_j - x_i) + F(t) + g(x_i)\xi_i(t) + \zeta_i(t)$$

we are given the following iterative method:

$$\begin{aligned} x_n[i][j] =& x_{n-1}[i][j] + \Big(f(x_{n-1}[i][j]) + \frac{D}{d} * (x_{n-1}[i+1][j] + x_{n-1}[i-1][j] + \\ & x_{n-1}[i][j+1] + x_{n-1}[i][j-1] - d\, x_{n-1}[i][j]) + \\ & \sigma_\zeta^2/2 * g(x[i][j])\, g'(x_{n-1}[i][j]) + F(t)\Big) * \Delta t + \\ & sq_m\, g(x_{n-1}[i][j])\xi[i][j] + sq_a\, \zeta(i,j). \end{aligned}$$

In particular, ξ represents the multiplicative noise whose intensity is regulated by the parameter σ_ξ which appears in the determination of $sq_m = \sqrt{\sigma_\xi^2 \Delta t}$. In the same way σ_ζ determines the magnitude of the additive noise ζ through the parameter $sq_a = \sqrt{\sigma_\zeta^2 \Delta t}$. In our system σ_ζ will be always zero since we do not have additive noise; to study the deterministic lattice we impose for the moment that σ_ξ is zero too. According to this, the equation we will impose for both variables $u(t)$ and $v(t)$ will be:

$$\begin{aligned} x_n[i][j] =& x_{n-1}[i][j] + \Big(f(x_{n-1}[i][j]) + \frac{D}{d} * (x_{n-1}[i+1][j] + x_{n-1}[i-1][j] + \\ & x_{n-1}[i][j+1] + x_{n-1}[i][j-1] - d\, x_{n-1}[i][j])\Big) * \Delta t. \end{aligned} \tag{3.3}$$

Having set these, the resolution will follow the steps that can be summarized as follows.

Results

The figure 3.1 shows the oscillations of the system without noise: the oscillators are perfectly synchronized.

Algorithm 1: Deterministic Lattice Computation algorithm

initialize parameters;
set initial conditions;
for *each time step* **do**
    for *x dimension border members* **do**
        for *y dimension border members* **do**
            impose periodic boundary conditions for both variables
        end
    end
    start of the integration step;
    for *interior members in x dimension* **do**
        for *interior members in y dimension* **do**
            solve the equation (3.3) for both variables
        end
    end
    update values for next time iterationṡave time and average variable values for plot
end
plot average vs time

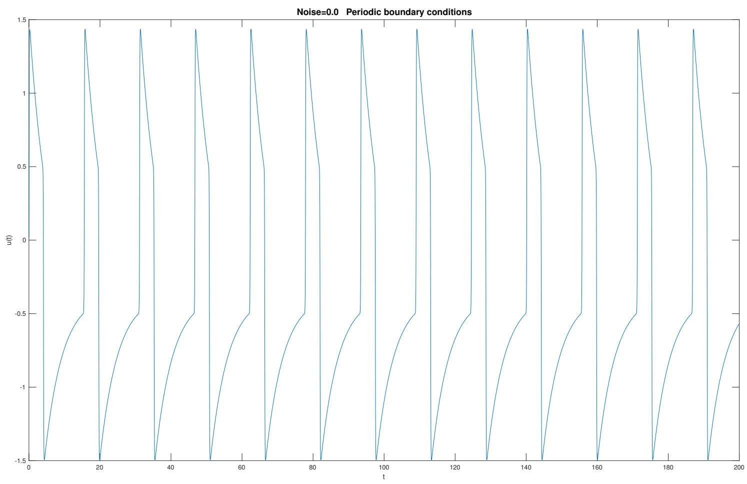


Figure 3.1: Mean field for a system of 1600 coupled elements, with the coupling strengths $D_u = 100$ and $D_v = 100$ and no noise.

3.2.2 Adding noise

Thermal fluctuations within the real system lead to chemical reactions and conductance fluctuations in ion channels which indeed cause some amount of noise in the neural system that is what we call multiplicative noise. So in order to better represent the real system this term has to be included. Multiplicative noise technically means there is some noise that is multiplied by the coordinate of the element we are integrating. This term is what makes our differential equation stochastic.

The multiplicative noise appears in the second equation of our system (3.1) and it is represented by ξ. To solve this differential stochastic equation we used the same iterative method as in the previous section; in our case the function $g(x)$ is just the identity, so we have:

$$\begin{aligned} x_n[i][j] =& x_{n-1}[i][j] + \Big(f(x_{n-1}[i][j]) + \frac{D}{d} * (x_{n-1}[i+1][j] + x_{n-1}[i-1][j] + \\ & x_{n-1}[i][j+1] + x_{n-1}[i][j-1] - d\, x_{n-1}[i][j]) + \sigma_\zeta^2/2 * (x[i][j]) \Big) * \Delta t \\ & + sq_m\, (x_{n-1}[i][j]) \xi[i][j]. \end{aligned} \tag{3.4}$$

In order to get a numerical solution of our problem, the steps we followed are the same of the ones in the case without noise, but, instead of the equation (3.3), we used (3.4).

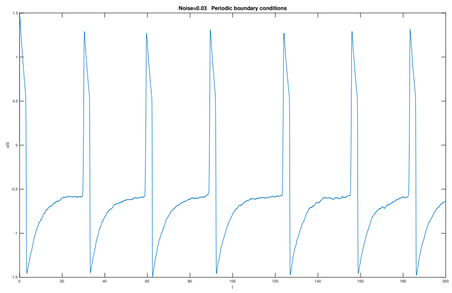
Results

Firstly, we generated a plot for the mean field values of neighboring neurons for different noise intensities.

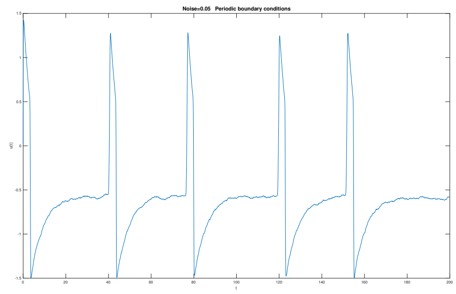
Increasing the noise intensity σ_ξ^2 leads to an increase and randomization of the time interval between consecutive spikes, as seen in figure 3.2. The solution changes from a periodical one to an irregular one: the spikes follow each other without a rule and the graphics appear less and less smooth. Finally, for large enough noise ($\sigma_\xi^2 = 0.08$) no spike appears, as we can see in the last figure in 3.2. This corresponds to an oscillation suppression due to multiplicative noise: the system stays at the noise-induced stable fixed point.

It is interesting to notice that changing the initial conditions only affects the state of the system at the beginning. If initial conditions are irregular each large propagation starts to propagate. Since the system has no memory the behaviour of the system tends to become the same with no dependence on the initial conditions, as shown in figure 3.3.

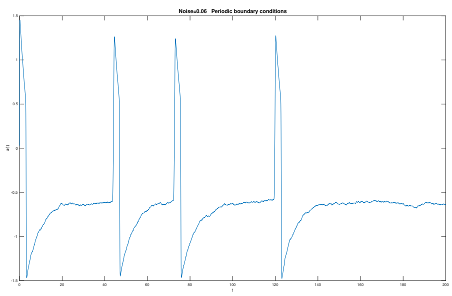
Furthermore, it is possible to notice that noise by itself is not able to suppress oscillations and bring stability. The fact that neurons are coupled is strictly necessary: the transition to excitability cannot be observed in isolated oscillators as shown in figure 3.5 where we have noise ($\sigma_\xi^2 = 0.08$) but no coupling ($D_u = D_v = 0$).



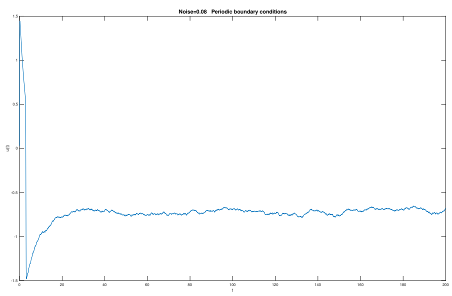
(a) $\sigma_\xi^2 = 0.03$



(b) $\sigma_\xi^2 = 0.05$



(c) $\sigma_\xi^2 = 0.06$



(d) $\sigma_\xi^2 = 0.08$

Figure 3.2: Mean field for a system of 1600 coupled elements, with the coupling strengths $D_u = 100$ and $D_v = 100$.

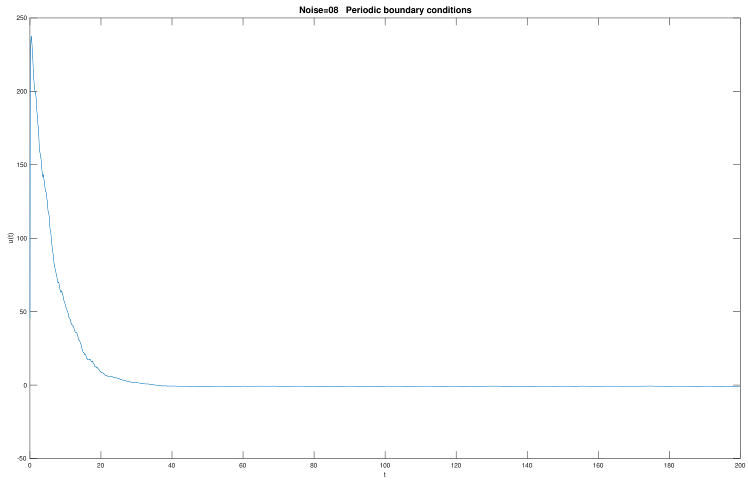


Figure 3.3: Noise $\sigma_\xi^2 = 0.08$ and different initial conditions.

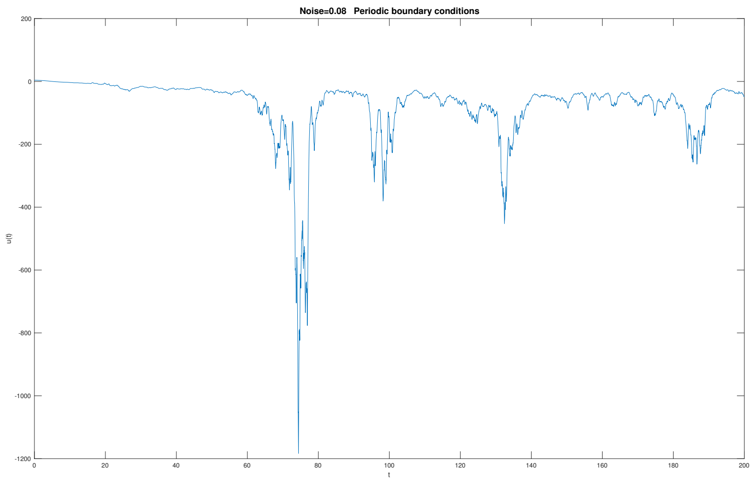


Figure 3.4: Noise $\sigma_\xi^2 = 0.08$ but no coupling ($D_u = D_v = 0$).

We can observe the phenomenon of spiral propagation of a signal with two different levels of noise. In the first case we observe, with a noise of $\sigma_\xi^2 = 0.001$, how the spiral is forming on a regular basis. IT is interesting to note how the initial condition, and with them the signal, loose their effect with time due to the noise and the system oscillates again as observed in the previous figures. In the second case we observe how a noise of $\sigma_\xi^2 = 0.1$ is too preponderant with respect to the signal and does not allow the correct formation of a spiral. Anyway even in this case the signal effect disappear after a couple of seconds.

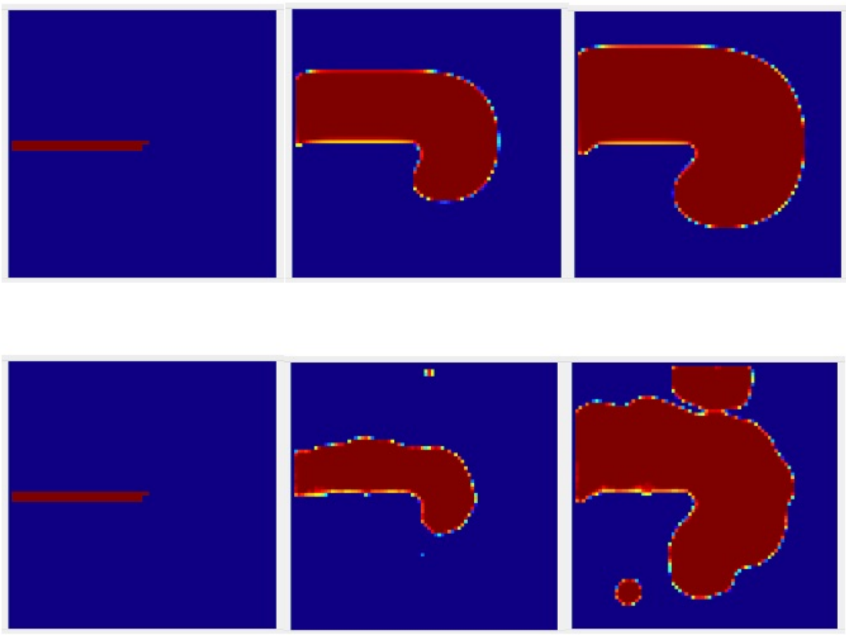


Figure 3.5: Propagation of a signal at times: 0 , 0.2, 0.4 in a 80×80 arrey. Parameters are: $\varepsilon = 0.01$, $a = 1.0$, $b = 2.0$, $c = 0.2$, $d = 0.075$, $g = 0.2$, $D_u = 100$, $D_v = 100$ and $\sigma_\xi^2 = 0.001$ in the first line and $\sigma_\xi^2 = 0.1$ in the second line

Chapter 4

Modelling Astrocytes Organized Associative Memory

4.1 Neuron-astrocyte network

The model of the neuron-astrocyte network (NAN) that is been considered consists of two layers, first layer of 40×40 neuron and a second layer of 13×13 astrocytes. In this NAN, we consider bidirectional neuron-astrocytic communication between layers, but the elements in each layer are no interconnected. The communication between layers is made so each astrocyte can communicate with 4×4 neurons with overlapping in one row (see Figure 4.1).

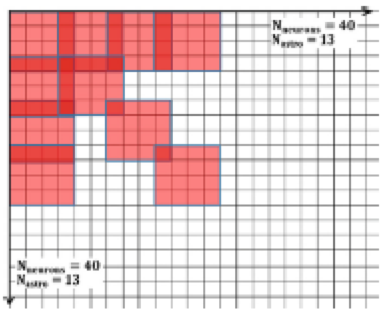


Figure 4.1: Structure of the neuron-astrocyte network. Input images 40x40 pixels size, each pixel representing a neuron. Red square correspond to the field of communication of each astrocyte, which overlap by one neuron wide layer. Taken from [3].

The model is designed so that when calcium level inside an astrocyte exceeds a threshold, there will be released a synaptic current to the connected neurons that will affect their activity. Similarly, when at least 50% of the neurons connected with an astrocyte reach some threshold, this astrocyte will be activated.

4.2 The model

This works is based on the results of [3] where the membrane potential of a single neuron is described by Izhikevich model, described by the equations 4.1.

$$\begin{cases} \frac{\dot{v}}{dt} = 0.04v^2 + 5v + 140 - u + I_{app} + I_{astro}, \\ \frac{\dot{u}}{dt} = a(bv - u). \end{cases} \tag{4.1}$$
$$If\ v \geq 30mV,\ then\ v \to c,\ u \to u + d.$$

The parameters values used where: $a = 0.1$, $b = 0.25$, $c = -65$, $d = 2$. The application currents I_{app} simulate imput signal ($I_{app} = 5$ if input signal is presented), I_{astro} represent the astracytic synaptic activity (setted as 30 if the Ca^{2+} level in astrocyte exceeds $0.15\mu M$ and more than 50% of neurons connected to the astrocyte are activated).

Calcium dynamics in astrocyte is described by the Li-Rinzel model. This model analyze the behavior of the IP_3 (IP_3 variable) and Ca^2 (Ca variable) concentration, and the fraction of activated IP_3 (h variable) receptors.

$$\begin{cases} \frac{dCa}{dt} = I_{er} - I_{pump} + I_{leak}, \\ \frac{dH}{dt} = \frac{H-h}{\tau_n}, \\ \frac{dIP_3}{dt} = (IP_{3s} - IP_3)\tau_r + I_{plc} + I_{neuro}. \end{cases} \tag{4.2}$$

$$I_{er} = c_1 v_1 \left(\frac{IP_3}{IP_3 + d_1}\right)^3 \left(\frac{Ca}{Ca + d_5}\right)^3 h^3 \left(\frac{c_0 - Ca}{c_1} - Ca\right),$$

$$I_{leak} = c_1 v_2 \left(\frac{c0 - Ca}{c_1}\right),$$

$$I_{pump} = v_3 \frac{Ca^2}{Ca_{2+k_3^2},}$$

$$H = \left(d_2 \frac{IP_3 + d_1}{IP_3 + d_3}\right) / \left(d_2 \frac{IP_3 + d_1}{IP_3 + d_3} + Ca\right),$$

$$\tau_n = 1/\left(a_2 \left(d_2 \frac{IP_3 + d_1}{IP_3 + d_3} + Ca\right)\right),$$

$$I_{plc} = v_4 \frac{Ca + (1 - \alpha)k_4}{Ca + k_4}.$$

The parameter's value of the model can be found in [3].

We have considered the neuron layer as a excitable system which is affected by noise as studied in chapter 3 of this. Our NAN astrocyte layer is modeled through the same equations as the previous one but we have change the neuron layer, integrating it with the NIE model. The new model evolves according to the equations 4.3 where F as in 3.1.

$$\begin{cases} \dot{u_i} = \frac{1}{\varepsilon}\left(F(u_i) - v_i\right) + D_u(\bar{u_i} - u_i) + \frac{2}{5}I_{app} + \frac{2}{82}I_{astro} \\ \dot{v_i} = cu_i + d + v_i\xi_i + D_v(\bar{v_i} - v_i) \end{cases} \tag{4.3}$$

$$If\ v \geq 2.4mV,\ then\ v \to 1.6.$$

$$If\ u \geq 1.4mV,\ then\ u \to -0.6.$$

We used the parameters value: $D_u = 100$, $D_v = 100$, $c = 0.2$, $d = 0.075$, $\varepsilon = 0.1$ and intensity noise $\sigma^2 = 0.08$. The reason we have setted the last conditions of the model is because F has two stable points but only one of them is of use.

The implementation of this model can be seen in appendix B. The results we have obtained will be shown in next chapter.

Chapter 5

Training the neuron-astrocyte network

5.1 Input data

The input of our network will be an image with the same size of the neuron network, 40 by 40. We will feed the network with images of 1's and 0's. Also, we will apply some salt and pepper noise, with different intensities, to those images. We apply the noise to slightly change the inputs, which and allows the model to generalize better in the test phase.

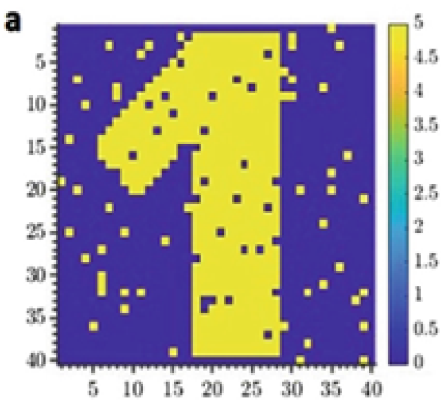


Figure 5.1: Input image with salt and pepper noise

5.2 Original network

As seen in the previous chapter our model is based on a previous neuron-astrocyte network. This model has the neurons modeled by the Izhikevich model and the astrocytes by the Li-Rinzel model.

The goal of the training for the astrocyte network is to learn the patterns of the input images and effectively store them. To do so, the input images are fed to the neural network within a short time interval. During that interval the neurons that have received an input are excited. An astrocyte will be excited if half of the 16 neurons that are connected with it are excited.

Figure 5.2: Left image: input. Middle image: neuron network. Right image: astrocyte network

In figure 5.2 it can be seen the training process where the neurons are excited by the input. However, the astrocyte network is not excited. This happens because the dynamics of the astrocyte model are remarkably slower and it takes some time for the astrocytes to get excited.

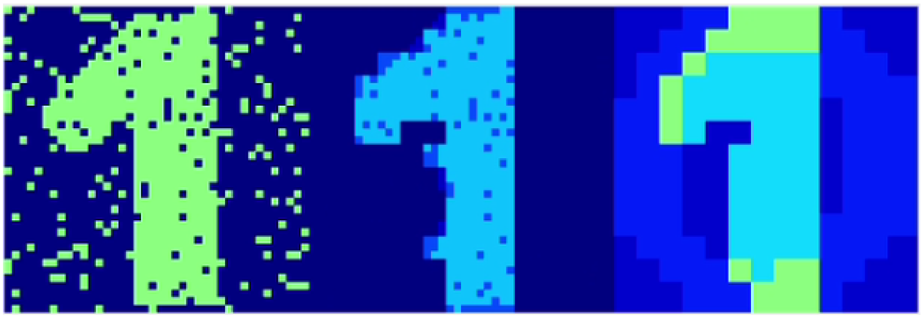


Figure 5.3: Test phase.

Once all the input images are fed to the network the training phase is over. In addition, some seconds without inputs are left for the astrocytes to stabilize, as their dynamics are slower. The final result is shown in the right part of figure 5.3. It can be seen how the 1 and 0 pattern has been stored in the astrocyte network. It is interesting to see than the noise has been filtered and is not stored in the astrocytes.

After the pause, we apply an input to see the response of the system once is trained. Now, the neurons remains excited much more time than in the training phase, as seen in figure 5.3. This happens because the neurons and the astrocytes have bidirectional communication and the 1 pattern is stored in the astrocytes.

5.3 Novel network

The challenge of this research is to use the FHN model, explained in Chapter 3, to model the neurons in the previous neuron-astrocyte model.

The code was modified to introduce the new neuron model. The astrocyte part and the training and test process remained untouched.

During the training phase the experiment started with the expected behaviour. The neurons were excited when the input was fed. The response was noisier than in the previous model. Though, this noisy response was expected because noise is inherent to the new neuron model. This can be seen in figure 5.4. However, while the experiment went on the response started to downgrade because of the noise. The neurons did not go back to the stable state before the next input. This caused a progressive increment of the noise that finished with the whole neuron network excited. When this happened the inputs had almost any effect on the network and the training was ineffective.

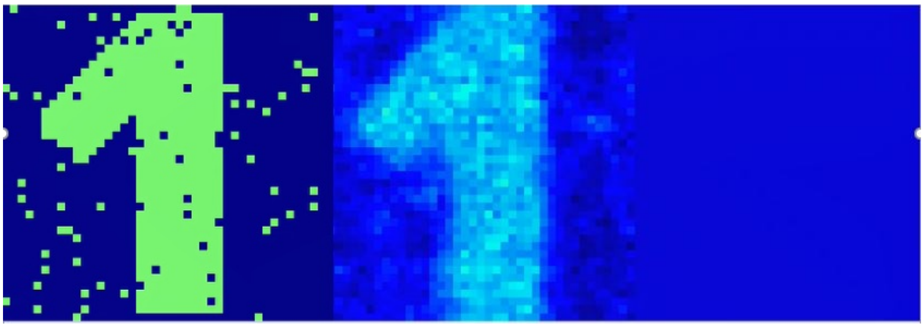


Figure 5.4: Train phase of the training.

The results of this experiments are shown in 5.5. The 1 pattern was stored by the astrocyte network but not the 0 pattern. This occurred because the 0 inputs were introduced when the noise was too high in the training phase and none effect was caused to the neurons. Also, when the test image was fed into the network the noise was too high to produce any response.

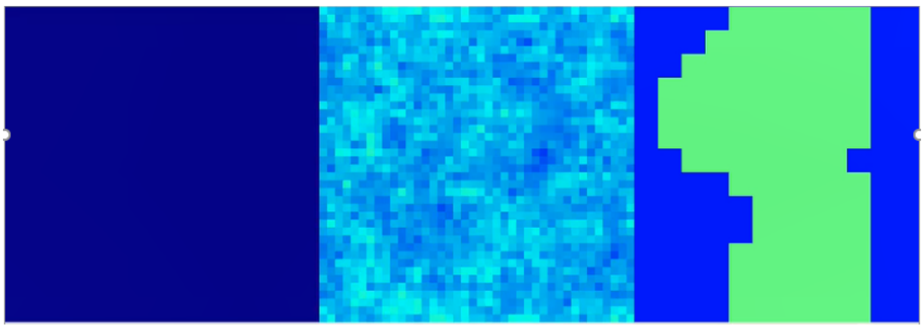


Figure 5.5: Test phase.

To conclude, some ideas emerged to improve the results of this experiment and to solve the noise problem.

First, the dynamics of the astrocyte and neurons can be studied and modified to give the possibility to the neurons to go back to the stable state before the new input is fed. Other option is to change the input intervals and give more time between inputs.

Second, the communication between the neurons and astrocytes are modeled by some parameters. The original neuron model had greater voltages and intensities than the new neuron model. We let the original parameters but those parameters that model the communication can be tunned to see if the response improves.

Appendix A

Noise-Induced Matlab code

```
% Initialization
% Stochastic initial conditions

t = 0; % initial time
d = 4;
N = 40; % lattice size
mu=0;
sigma=1;

u = 0.0001*normrnd(mu,sigma,N);
u_n = u;
v = 0.0001*normrnd(mu,sigma,N)+1.5;
v_n = v;


Du=0; %coupling
Dv=0;
c=0.2;
dd=0.075;
e= 0.08;
index_plot=0; %index for recording values for plot

Lim = 100000; % number of integration steps
tmp = Lim/(N^2); % size of each integration step
T = 200; % total time
dt = T/Lim; % time step

% Calculation of the new state of the system
sigma2=0.08;      % Noise intensity
sqm=sqrt(sigma2*dt);

for time = t:dt:T
    for i = 1:N

% Boundary conditions for the main step

         u_n(N,i) = u(2,i);
         u_n(1,i) = u(N-1,i);
         u_n(i,N) = u(i,2);
         u_n(i,1) = u(i,N-1);
```

```
        v_n(N,i) = v(2,i);
        v_n(1,i) = v(N-1,i);
        v_n(i,N) = v(i,2);
        v_n(i,1) = v(i,N-1);
    end

    chi=normrnd(0,1,N);

    %Integration step

    for i = 2:N-1
        for j = 2: N-1
            u_n(i,j) = u(i,j) + ( 1/e * (F(u(i,j))-v(i,j)) + Du/d *(...
                u(i+1,j) + u(i-1,j) + u(i,j+1) + u(i,j-1) - ...
                    d*u(i,j)))* dt;

            v_n(i,j) = v(i,j) +( c*u(i,j) + dd + Dv/d *(...
                v(i+1,j) + v(i-1,j) + v(i,j+1) + v(i,j-1) - ...
                    d*v(i,j)) ...
                 + sigma2/2*v(i,j)) * dt + sqm*v(i,j)*chi(i,j);
        end
    end
    v = v_n;
    u = u_n ;


    index_plot=index_plot+1;
    time_plot(index_plot) = time;
    average(index_plot)= sum ( sum(u) )/ (N^2);
    average_v(index_plot)= sum ( sum(v) )/ (N^2);

end
title('Evolution of average neuron membrane potential')
plot(time_plot,average)
ylabel('u(t)')
xlabel('t')
xlim([0,T])
title('Noise=08   Periodic boundary conditions','Fontsize',14)
```

Appendix B

Neuron-astrocyte Matlab code

B.1 Code for the neuron-astrocyte network training

```
close all;
clearvars;
u = 0.0001;
u2 = u/2;
u6 = u/6;
t = 1.65;
N = t/u;
Aneuro = 5;
Tneuro = 0.06;
PCa = 0.15; %Ca2+ level in astrocytes
Aastro = 82;
YW = 0;
h = 3;
%matrix neuro
mm = 40;
nn = 40;
v_tot = ones(mm,nn,N)*1.6;
u_tot = ones(mm,nn,N)*-0.6 ;
%matrix astro
m = 13;
n = 13;
masCa = zeros (m,n,N); % concentration of calcium
mash = zeros (m,n,N); % fraction of IP3 activated
masIP33 = zeros (m,n,N); % concentration of IP3
masCa(:,:,1) = 0.072495;
mash(:,:,1) = 0.886314;
masIP33(:,:,1) = 0.820204;
dispersion = 0.2;
Ivh = 0;
masIvh = zeros(mm,nn,N);

I1 = imread('one.jpg');
% add noise to the image
J1 = imnoise(I1,'salt & pepper',0.1);
```

```
J1 = J1(1:40, 1:40) > 127;
J2 = imnoise(I1,'salt & pepper',0.1);
J2 = J2(1:40, 1:40) > 127;
J3 = imnoise(I1,'salt & pepper',0.1);
J3 = J3(1:40, 1:40) > 127;
J4 = imnoise(I1,'salt & pepper',0.1);
J4 = J4(1:40, 1:40) > 127;
J5 = imnoise(I1,'salt & pepper',0.1);
J5 = J5(1:40, 1:40) > 127;
J6 = imnoise(I1,'salt & pepper',0.1);
J6 = J6(1:40, 1:40) > 127;
J7 = imnoise(I1,'salt & pepper',0.1);
J7 = J7(1:40, 1:40) > 127;
J8 = imnoise(I1,'salt & pepper',0.1);
J8 = J8(1:40, 1:40) > 127;
J9 = imnoise(I1,'salt & pepper',0.1);
J9 = J9(1:40, 1:40) > 127;
J10 = imnoise(I1,'salt & pepper',0.1);
J10 = J10(1:40, 1:40) > 127;
I11 = imread('zero.jpg');
J11 = imnoise(I11,'salt & pepper',0.1);
J11 = J11(1:40, 1:40) > 127;
J21 = imnoise(I11,'salt & pepper',0.1);
J21 = J21(1:40, 1:40) > 127;
J31 = imnoise(I11,'salt & pepper',0.1);
J31 = J31(1:40, 1:40) > 127;
J41 = imnoise(I11,'salt & pepper',0.1);
J41 = J41(1:40, 1:40) > 127;
J51 = imnoise(I11,'salt & pepper',0.1);
J51 = J51(1:40, 1:40) > 127;
J61 = imnoise(I11,'salt & pepper',0.1);
J61 = J61(1:40, 1:40) > 127;
J71 = imnoise(I11,'salt & pepper',0.1);
J71 = J71(1:40, 1:40) > 127;
J81 = imnoise(I11,'salt & pepper',0.1);
J81 = J81(1:40, 1:40) > 127;
J91 = imnoise(I11,'salt & pepper',0.1);
J91 = J91(1:40, 1:40) > 127;
J101 = imnoise(I11,'salt & pepper',0.1);
J101 = J101(1:40, 1:40) > 127;
II = imread('one.jpg');
JJ = imnoise(II,'salt & pepper',dispersion);
JJ = JJ(1:40, 1:40) > 127;
masMet = zeros(mm,nn,N);
F = zeros(m,n,N); % if neuron are activated in an instant time
aq = zeros(m,n,N);
aqq = zeros(m,n,N);
masIastro = zeros(mm,nn,N);
Iastroo = zeros(m,n,N); % corriente que modela la actividad ...
    sinaptica de los astrocitos
aQ = zeros(m,n,N);
t1 = [0]; %array t
%parameters neuron
aa = 0.1; %RZ value for potential V
b = 0.25; % in dU/dt
c = -65; % value for V, if V ≥ 30 mV => V -> V+c
d = 2; %value for U, if V≥ 30 mV -> U -> U+d
X = zeros(1, 3);
t11 = 0;
tic
for N1 = 1 : N % tiempo
    for m1 = 1 : 1 : m % 13: astrocyte
```

```
        for n1 = 1 : 1 : n %13: astrocytes
            YW = 0;
            if F(m1,n1,N1) > 8 % F : neuronas activas en un ...
                instante de tiempo
                aqq(m1,n1,N1) = Aneuro; % ponemos feedback ...
                    astrocitos - neurona
            end
            if N1 < (0.4/u)
                aq(m1,n1,N1) = 0;
            end
            if (N1 == (0.4 / u)) || (N1 == (0.9 / u))
                aQ = aqq(m1,n1,(N1+1-0.4/u):(N1-1));
                [aQ] = shiftdim(aQ);
                YW = sum(aQ == Aneuro);
                if YW > 50 %if more than 50% of neurons that ...
                    correspond to an
                    % astrocyte are activated
                    aq(m1,n1,N1 : N1 + Tneuro/u) = Aneuro;
                end
            end
            X(1) = masCa(m1,n1,N1);
            X(2) =  mash(m1,n1,N1);
            X(3) = masIP33(m1,n1,N1);
            Ineuro = aq(m1,n1,N1);

            % calculation of current value
            w1 = fun(0,X, Ineuro);
            w2 = fun(0,X + u2.*w1', Ineuro);
            w3 = fun(0,X + u2.*w2', Ineuro);
            w4 = fun(0,X + u .*w3', Ineuro);
            X = X + u6 .* (w1' + 2 .* w2' + 2 .* w3' + w4');
            Y = X;
            masCa(m1,n1,N1+1) = Y(1);
            masIP33(m1,n1,N1+1) = Y(3);
            mash(m1,n1,N1+1) = Y(2);
            if (masCa(m1,n1,N1) > PCa) && (F(m1,n1,N1) > 8)
                Iastroo(m1,n1,N1) = Aastro;
            else
                Iastroo(m1,n1,N1) = 0;
            end
        end
    end
    km=0;
    kmm=0;
    for j = 1 : h : (mm-3)
        kmm = 0;
        for jj = 1 : h : (mm-3)
            masIastro(j : j+h,jj : jj+h, N1) = Iastroo(j-km,jj-kmm,N1);
            kmm = kmm + 2;
        end
        km = km + 2;
    end

%     [u_tot(:,:,N1) , v_tot(:,:,N1)] = main_fun2(u_tot(:,:,N1), ...
    v_tot(:,:,N1) , u, 0.05) ;

    for mm1 = 1 : 1 : mm %40: neurons
        for nn1 = 1 : 1 : nn %40: neurons
%             v = v_tot(mm1,nn1,N1);
%             uu = u_tot(mm1,nn1,N1);
%             fired = find(v ≥ 30); %if the potencial is greater or ...
    equal than 30mV
```

```
%             v(fired) = c; %then V=> c
%             uu(fired) = uu(fired) + d; %and U => U+d
            if (((J1(mm1,nn1)==0)&& (t11≥ 0 && t11 ≤ ...
                0.004))||((J2(mm1,nn1)==0)&&(t11≥ 0.044 && t11 ≤ ...
                0.048))||...
                    ((J3(mm1,nn1)==0)&&(t11≥ 0.088 && t11 ≤ ...
                        0.092))||((J4(mm1,nn1)==0)&&(t11≥ 0.132 && ...
                        t11 ≤ 0.136))||...
                    ((J5(mm1,nn1)==0)&&(t11≥ 0.176 && t11 ≤ ...
                        0.180))||((J6(mm1,nn1)==0)&&(t11≥ 0.220 && ...
                        t11 ≤ 0.224))||...
                    ((J7(mm1,nn1)==0)&&(t11≥ 0.264 && t11 ≤ ...
                        0.268))||((J8(mm1,nn1)==0)&&(t11≥ 0.308 && ...
                        t11 ≤ 0.312))||...
                    ((J9(mm1,nn1)==0)&&(t11≥ 0.352 && t11 ≤ ...
                        0.356))||((J10(mm1,nn1)==0)&&(t11≥ 0.396 && ...
                        t11 ≤ 0.400))||...
                    ((J11(mm1,nn1)==0)&& (t11≥ 0.5 && t11 ≤ ...
                        0.504))||((J21(mm1,nn1)==0)&&(t11≥ 0.544 && ...
                        t11 ≤ 0.548))||...
                    ((J31(mm1,nn1)==0)&&(t11≥ 0.588 && t11 ≤ ...
                        0.592))||((J41(mm1,nn1)==0)&&(t11≥ 0.632 && ...
                        t11 ≤ 0.636))||...
                    ((J51(mm1,nn1)==0)&&(t11≥ 0.676 && t11 ≤ ...
                        0.68))||((J61(mm1,nn1)==0)&&(t11≥ 0.720 && ...
                        t11 ≤ 0.724))||...
                    ((J71(mm1,nn1)==0)&&(t11≥ 0.764 && t11 ≤ ...
                        0.768))||((J81(mm1,nn1)==0)&&(t11≥ 0.808 && ...
                        t11 ≤ 0.812))||...
                    ((J91(mm1,nn1)==0)&&(t11≥ 0.852 && t11 ≤ ...
                        0.856))||((J101(mm1,nn1)==0)&&(t11≥ 0.896 ...
                        && t11 ≤ 0.9))||...
                    ((JJ(mm1,nn1)==0)&&(t11≥ 1.6 && t11 ≤ 1.620)))
                Ivh = 5; % input signal is presented
            else
                Ivh = 0;
            end
            masIvh(mm1,nn1,N1) = Ivh;
            Iastro = masIastro(mm1,nn1,N1);

%           v = v + u * 1000 *(0.04 * v .^2 + 5 * v + 140 + Ivh + ...
    Iastro − uu);
%           uu = uu + u * 1000 * aa .*(b .*v − uu);
%
            [uu , v] = fun2(u_tot(:,:,N1), v_tot(:,:,N1), mm1 ,nn1 ...
                ,u, Ivh,Iastro);
            v_tot(mm1,nn1,N1+1) = v;
            u_tot(mm1,nn1,N1+1) = uu;

            if v > 2.4
                v_tot(mm1,nn1,N1+1) = 1.6;%stable state
            else
                v_tot(mm1,nn1,N1+1) = v;
            end

            if u > 1.4
                u_tot(mm1,nn1,N1+1) = −0.6;%stable state
            else
                u_tot(mm1,nn1,N1+1) = u;
            end
%             end
%             u_tot(mm1,nn1,N1+1) = uu;
```

```
            %if v > -63
            %if v > 2
            if u>0
                masMet(mm1,nn1,N1) = 1;
            else
                masMet(mm1,nn1,N1) = 0;
            end
        end
    end
    km = 0;
    kmm = 0;
    for j = 1 : 3 : (mm-3)
        kmm = 0;
        for jj = 1 : 3 :(mm-3)
            F(j-km,jj-kmm,N1+1) = sum(sum(masMet(j:j+h, jj:jj+h,N1)));
            kmm = kmm + 2;
        end
        km = km + 2;
    end
    if t11 < t
        t11 = t11 + u;
        t1 = [t1 t11];
    end
    if rem(N1, 500) == 0
        toc;
    end
end
km = 0;
kmm = 0;
masCa1 = zeros(mm,nn,N);
for qe = 1 : N
    for j = 1 : h :(mm-3)
        kmm = 0;
        for jj = 1 : h :(mm-3)
            masCa1(j:j+h,jj:jj+h,qe) = masCa(j-km,jj-kmm,qe);
            kmm = kmm + 2;
        end
        km = km + 2;
    end
    km = 0;
    kmm = 0;
end
masIvh = masIvh ./ 10;
u_tot = u_tot./1;
video = horzcat(masIvh,u_tot(:,:,1:end-1),masCa1);
handle = implay(video,10);
cmap = jet(256);
handle.Visual.ColorMap.Map = cmap;
handle.Visual.ColorMap.UserRangeMin = -0.8;
handle.Visual.ColorMap.UserRangeMax = 0.3;
```

B.2 Code for the astrocyte layer

```
function f=fun(t,X,Ineuro)

c0=2.0;
c1=0.185;
```

```
v1=6.0;
v2=0.11;
v3=2.2;
v4=0.3;
%v5=0.025;
v6=0.2;

k1=0.5;
k2=1.0;
k3=0.1;
k4=1.1;

d1=0.13;
d2=1.049;
d3=0.9434;
d5=0.082;

IP3s=0.16;

Tr=0.14;

a=0.8;
a2=0.14;


               M=X(3)/(X(3)+d1);
               NM=X(1)/(X(1)+d5);
               Ier=c1*v1*(M^3)*(NM^3)*(X(2)^3)*(((c0-X(1))/c1)-X(1));
               Ileak=c1*v2*(((c0-X(1))/c1)-X(1));
               Ipump=v3*(X(1)^2)/(X(1)^2+k3^2);
               Iin=v6*(X(3)^2/(k2^2+X(3)^2));
               Iout=k1*X(1);
               Q2=d2*((X(3)+d1)/(X(3)+d3));
               h=Q2/(Q2+X(1)); %receptor de moleculas activas
               Tn=1.0/(a2*(Q2+X(1)));
               Iplc=v4*((X(1)+(1.0-a)*k4)/(X(1)+k4));



f(1)=Ier-Ipump+Ileak+Iin-Iout; f(2)=(h-X(2))/Tn; ...
    f(3)=(IP3s-X(3))*Tr+Iplc+Ineuro;

%                M=X(1:m,1:n,3)./(X(1:m,1:n,3)+d1);
%                NM=X(1:m,1:n,1)./(X(1:m,1:n,1)+d5);
%                Ier=c1*v1*(M.^3)*(NM.^3)*(X(1:m,1:n,2).^3)*...
%                (((c0-X(1:m,1:n,1))/c1)-X(1:m,1:n,1));
%                Ileak=c1*v2*(((c0-X(1:m,1:n,1))/c1)-X(1:m,1:n,1));
%                Ipump=v3*(X(1:m,1:n,1).^2)/(X(1:m,1:n,1).^2+k3^2);
%                Iin=v6*(X(1:m,1:n,3)^2/(k2^2+X(1:m,1:n,3)^2));
%                Iout=k1*X(1:m,1:n,1);
%                Q2=d2*((X(1:m,1:n,3)+d1)/(X(1:m,1:n,3)+d3));
%                h=Q2/(Q2+X(1:m,1:n,1));
%                Tn=1.0/(a2*(Q2+X(1:m,1:n,1)));
%                Iplc=v4*((X(1:m,1:n,1)+(1.0-a)*k4)/(X(1:m,1:n,1)+k4));
%
%
%
% f(1)=Ier-Ipump+Ileak+Iin-Iout; f(2)=(h-X(1:m,1:n,2))/Tn; ...
    f(3)=(IP3s-X(1:m,1:n,3))*Tr+Iplc+Ineuro;

f=f';
```

B.3 Code for the neurons layer

```
function [u_n, v_n] = fun2(u, v, i , j , dt,Ivh,Iastro )

    % Input for function ASTRO:
    % u=u_tot(:,:,N1), v=v_tot(:,:,N1) , i=mm1, j=nn1 , chi=Ivh , ...
        dt=u (time step)

    N=size(u,2);
    % Parameters:

    d = 4; % 2-D lattice

    Du=100; %coupling
    Dv=100;
    c=0.2;
    dd=0.075;
    e= 0.01;

    % Noise:

    sigma2=0.08;
    sqm=sqrt(sigma2*dt);
    sqa=sqm;


    if i == 1
       u_n = u(N-1,j);
       v_n = v(N-1,j);

    elseif i == N
       u_n= u(2,j);
       v_n= v(2,j);

    elseif j == 1
        u_n = u (i, N-1);
        v_n = v (i , N-1);

    elseif j == N
        u_n = u (i,2);
        v_n = v(i,2);

    else

     %Integration step:

        u_n = u(i,j) + ( 1/e * (F(u(i,j))-v(i,j)) + Du/d *(...
          u(i+1,j) + u(i-1,j) + u(i,j+1) + u(i,j-1) - d*u(i,j)) + ...
               (Ivh/5)*2 + (Iastro/82)*2)* dt;

%    Moltiplicative noise :
        chi=normrnd(0,1);
        v_n = v(i,j) +( c*u(i,j) + dd + Dv/d *(...
            v(i+1,j) + v(i-1,j) + v(i,j+1) + v(i,j-1) - d*v(i,j)) ...
             + sigma2/2*v(i,j)) * dt + sqm*v(i,j)*chi ;

    end
end
```

Bibliography

[1] Farhy-Tselnicker I. and Allen N.A. (2018), *Astrocytes, neurons, synapses: a tripartite view on cortical circuit development*

[2] Ullner E., Zaikin A., Garcia-Ojalvo J. and Kurths J.(2003), *Noise-Induced Excitability in Oscillatory Media*

[3] Gordleeva S.Y., Lotareva Y.A., Krivonosov M.I., Zaikin A.A., Ivanchenko M.V. and Gorban A.N.(2020), *Astrocytes Organize Associative Memory*

[4] Boucsein C., Tetzlaff T., Meier R., Aertsen A. and Naundorf B. (2009) *Dynamical Response Properties of Neocortical*

[5] Fellin T. (2009), *Communication between neurons and astrocytes: relevance to the modulation of synaptic and network activity*

[6] Harada K., Kamiya T. and Tsuboi T. (2016), *Gliotransmitter release from astrocytes: functional, developmental and pathological implications in the brain*