Some image processing features of the second order Hysteretic Cellular Neural Networks

Tiberiu Teodorescu, Bvd. Copou, 11, 6600, Iasi, Romania, t-teodor@etc.tuiasi.ro
Andrei Maiorescu, Bvd. Copou, 11, 6600, Iasi, Romania, mandrei@etc.tuiasi.ro

Keywords: hysteresis, cellular neural networks, dispersion curve

Abstract

This article briefly presents some image processing features of the second order cellular neural networks with hysteresis. The results can be used for simulating the differences between the elementary cells (i.e. the situation when the network is not perfectly homogeneous from this point of view) as well.

Introduction

The image processing tasks are usually accomplished with networks made with first order cells [1, 2, 3]. Networks made with cells of second order were used for producing patterns [4, 5]. Their dynamics was studied (as one single entity and integrated into the network) in the linear part of their characteristic. As a consequence of that, linear processing (linear spatial filtering) was the main application for these systems. The minimum degree for the Chua cell capable of producing patterns is two and it is given by the two capacitors

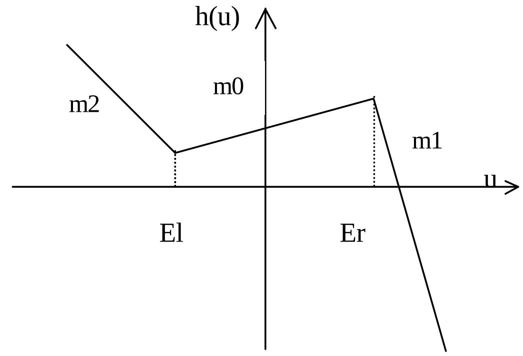
By using a hysteretic element as the nonlinear part in the Chua cell One can obtain a plus ½ degree for the Chua circuit and hence its dynamics is enriched. Consequently, the "non-linear part" contributes more to the final pattern (image).

The cell

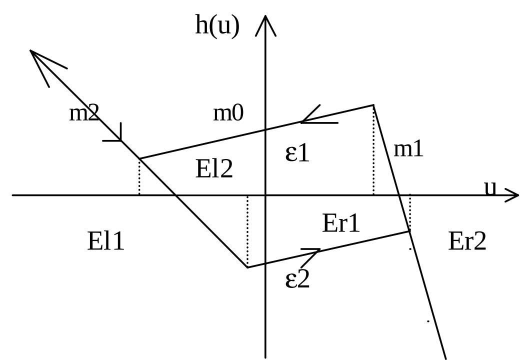
We have started from the idea of using the results of the previous work [4, 5] and to extent the capabilities of the classical cell. The system of differential equations that describes the 2D network is given bellow:

$$\begin{cases} \dfrac{du_{i,j}}{dt} = \gamma f(u_{i,j}, v_{i,j}) + D_u \Delta u_{i,j} \\ \dfrac{dv_{i,j}}{dt} = \gamma g(u_{i,j}, v_{i,j}) + D_v \Delta v_{i,j} \end{cases}$$

The non-linear part is located in the u dependence of f(u,v) function. The classical non-linear part is represented in the figure bellow:



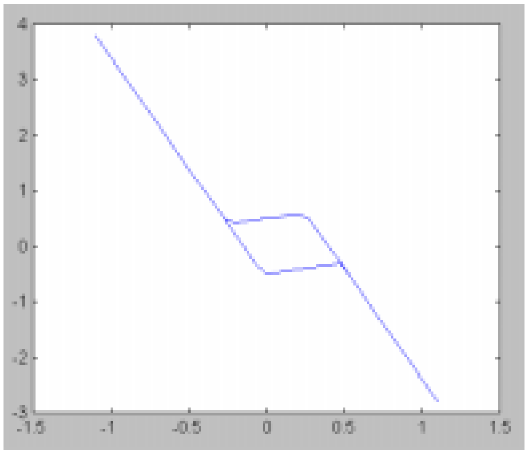
The new non-linear part is represented in the next picture:



The evolution of the cell versus time is: from the left to the right we go on the slope m_2, then on the first (upper) slope m_0 and last on the m_1 slope. From right to left, we will go on the m_1 slope, then on the second (lower) m_0, and finally on the m_2 slope.

Let us observe that we can describe the characteristic using the initial shape of the non-linear characteristic and its version moved to the right and to the bottom suitable for matching the slopes. This is the way the system was software described and one can

implement the system easier using this approach (one can use the initial circuit topology with minimum change).
We have verified the simulated hysteresis with a Matlab function for a sinusoidal input:



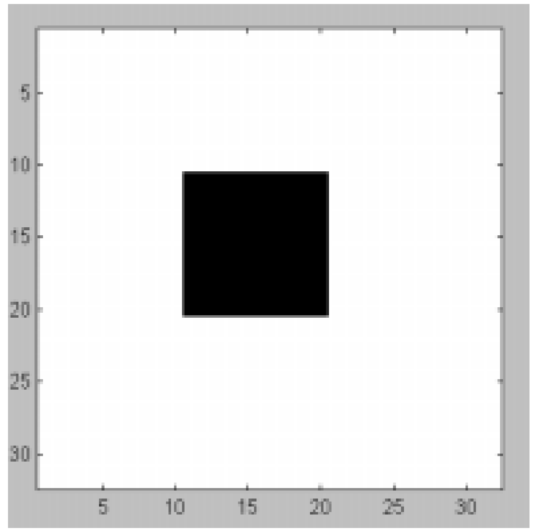
We will use the image as the initial condition for the autonomous system. The network will be seeded with a signal of small amplitude. Taking into account that we have an extra "discrete" state the initialization can be made either on the first or on the second slope around the origin.
Let us analyze some of the possibilities we have:

- starting with all cells from the upper slope;
- starting with all cells from the lower slope;
- starting with some of the cells on a slope and with the other on the other.

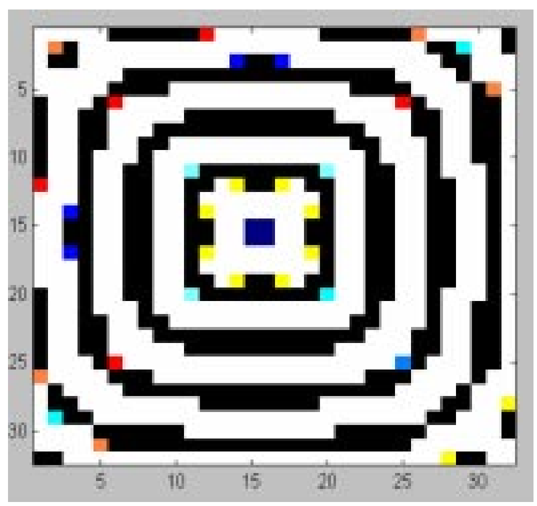
From the dynamics' point of view, it is interesting to start with the cells corresponding to "positive" voltages from the lower m_0 slope and with the cells corresponding to "negative" voltages from the upper m_0 slope. Let us note this situation with A.
Another interesting initialization is starting with the cells corresponding to "negative" pixels from the upper m0 slope and with the cells corresponding to "positive" pixels from the lower m0 slope. Let us note this starting situation with B.

Simulation results

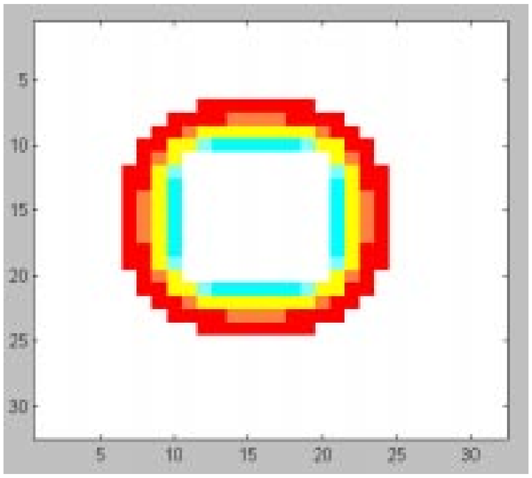
As a methodology for comparing the classical network with the network made with hysteretic cells we will simulate first classical and then the new ones using the same image.
The first example, which bring into our attention the difference between the cell without hysteresis and the cell with hysteresis uses a cell with the following parameters: m0=0.1, m1=-1, m2=-1, gu=0.1, gv=-0.2, Est1=-1, Edr1=1, Eps1=0; Eps2=-0.5. The network has: Du=1, Dv=45, Gama=50, M=N=32.
The input image for these simulations is given in the figure bellow:



By using the classic system we have obtained the following image processing (in fact, there is a only a spatial filtering []):



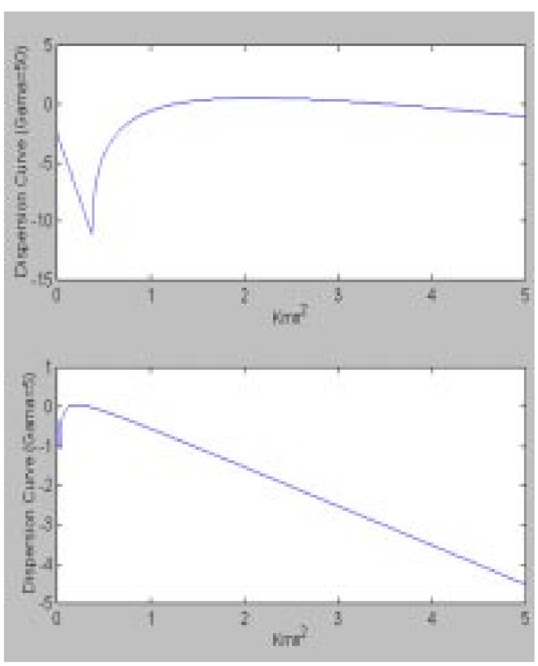
The same result is obtained by seeding all hysterons both on the lower and upper parts of region with m_0 slope. It is interesting to study the result of image processing when starting with the A situation explained above.



The image processing performed by the network is an edge extraction combined with a “smoothing” of rectangular curves. So, there is a classical image processing (edge extraction) which can be obtained with first order cellular neural networks [3] and a nonlinear image processing (smoothing the rectangular curves).

"Smoothness" and dispersion curve

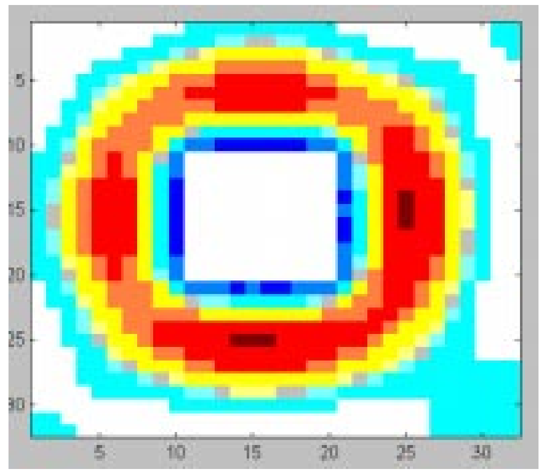
Another interesting aspect is establishing a connection between the “smoothness” degree and the number and the position of modes



inside the dispersion curve.

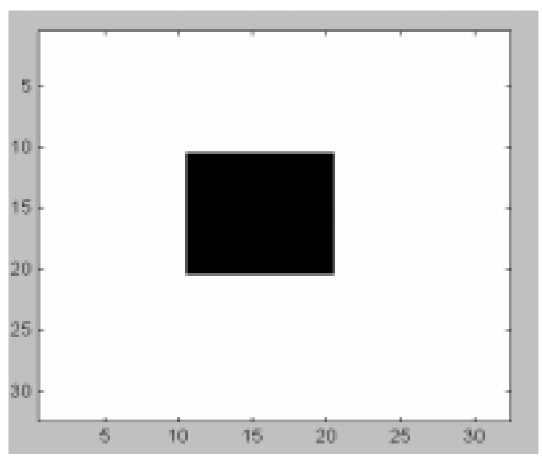
We have displayed above a comparison between the dispersion curve with Gama = 50, and respectively with Gama = 5.

There is a known fact that the decrease of Gama roughly moves the dispersion curve to the left. So, we will keep all the parameters set above and change only Gama: making it 5. The dispersion curve will move to the left and we expect to see an edge extraction and a larger "smoothing" for the input image:

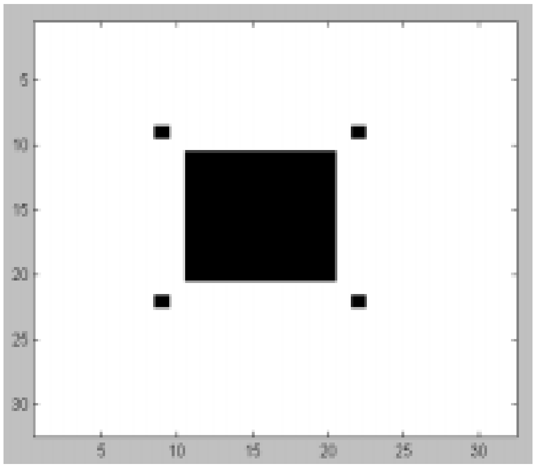


Another "residual" image processing

We will start the simulation with the B situation. It is interesting to see in what way the slope m_0 (upper and respectively lower) is changing through the simulation. The slope-change itself is an image processing. In the next figure we represent the initial "slope seeding":



Black is assigned to negative voltages and thus to the upper part of the hysteresis and white is for positive voltages, thus for the lower part of the hysteresis. At the end of the simulation, this situation changes: some of the cells on the lower part of the hysteresis go through hysteresis curve on the upper part of it, and the final distribution of the cells is given in the figure bellow:



We will see that this image processing is a corner extraction. We must emphasize the fact that this type of image processing is possible using the network made with first order cells too.
We can easily make an exclusive OR between the input image and the output image for obtaining only the corners.

Conclusions and further work

A new cell was proposed and implemented in a software simulator. The main idea in using this cell is to "enrich" the degree of the basic circuit and to combine the benefits of linear filtering with the one of non-linear filtering. The next step is to physically implement the circuit. We can use the idea of shifting to the right and to the bottom the original Chua non-linearity and perhaps using two non-linear elements of the same type, but having different parameters.
The circuit appears to do two different image processing tasks simultaneously, one on the state variables, and the other one on the third discrete state variable introduced here.
In addition, we can use the dispersion curve to "fine tune" the behavior of the network.

Bibliography

[1] L. O. Chua, L. Yang, Cellular Neural Networks: Theory, IEEE Transactions on Circuits and Systems, no. 10, Oct. 1988
[2] L. O. Chua, L. Yang, Cellular Neural Networks: Applications, IEEE Transactions on Circuits and Systems, no. 10, Oct. 1988
[3] K. Crounse, L. O. Chua, Methods for Image Processing and Pattern Formation in Cellular Neural Networks: A Tutorial, IEEE Transactions on Circuits and Systems, no. 10, Oct. 1995
[4] L. Goras, L. O. Chua, Turing Patterns in CNN's – Part I: Once Over Lightly, IEEE Transactions on Circuits and Systems, no. 10, Oct. 1995
[5] L. Goras, L. O. Chua, Turing Patterns in CNN's – Part II: Equations and Behaviors, IEEE Transactions on Circuits and Systems, no. 10, Oct. 1995