New Spatial-Temporal Patterns and The First Programmable On-Chip Bifurcation Test-Bed

István Petrás, *Student Member, IEEE,* Tamás Roska, *Member, IEEE* and Leon O. Chua, *Member, IEEE*

***Abstract* — In this paper we introduce a new experimental tool, a real-time programmable, spatial-temporal bifurcation test-bed. We present the experimental analysis of an antisymmetric template class. This class produces novel spatial-temporal patterns that exhibit complex dynamics. The character of these propagating patterns depends on the self-feedback and on the sign of the coupling below the self-feedback template element.**

***Index* — Cellular Nonlinear Networks, Chaos, Pattern formation**

I. Introduction

Numerical simulations of spatial-temporal chaotic systems require enormous digital computing power, but even so this is the usual analysis tool because it offers the advantage of easy experimentation via programming. Until now, the physically implemented chaotic circuits were “hard-coded”. The new analogic cellular computing paradigm [1][7][18] places the spatial-temporal dynamics into array computer architecture. Using the new ACE4K test-bed [1]–[3] it is possible for the first time to make programmable real-time experiments and uncover new complex dynamic behaviors in the Cellular Nonlinear Network.

The Cellular Nonlinear Network [4]–[8] is a locally coupled one, two or three-dimensional array of first or higher order nonlinear dynamical systems. The elements of the array are called *cells*. The couplings are called *templates*. They are written usually as a matrix, their elements represent the strength of the coupling between the neighboring cells. If we consider the case when the cells are organized into a 2D array, the *output* (or *state*) of the system is a two dimensional image in which each cell corresponds to one pixel of the image. The *input* of the CNN is an image which can also be considered as a constant bias to each cell. An image of the outputs of the cells at a given time instant may characterize the actual state of the whole system. Throughout this paper we consider the 2D-case except explicitly noted.

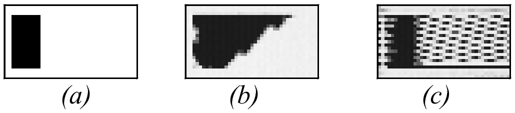


Fig. 1: Typical pattern classes. (a) The input and initial state, (b) snapshot of the output when the extra coupling is greater than zero, (c) snapshot of the output when the extra coupling is less than zero (Chip measurements)

The qualitative theory of nonsymmetric feedback (*A*) template were first exposed in [16]. Later, several papers studied the operation of the CNN with non-symmetric or sign-antisymmetric templates [9]–[11]. They described some necessary conditions under which propagation effects occur or the solution is periodic. Other works investigated the pattern formation properties of the CNN [12] or studied the complex behavior [13][14][15] of the CNN. However, only a few works dealt with the case when there is a constant input [17]. With constant input, we are able to localize the propagation effect into a certain region according to the extent of the input pattern. By using a constant input as a seed, different shapes can be generated depending on the properties of the template.

This work was supported by the Computer and Automation Research Institute of the Hungarian Academy of Sciences (SZTAKI), Office of Naval Research (ONR) Grant No.: N0014-00-C-0295 and the Hungarian National Foundation (OTKA) Grant No.: T026555.

I. Petrás is with the Analogical and Neural Computing Laboratory, Computer and Automation Research Institute, Hungarian Academy of Sciences, Kende u.13-17, Budapest, 1111 – Hungary (corresponding author to provide phone: +36 1 279-6181, fax: +36 1 209-5264, email: *petras@sztaki.hu).*

L.O. Chua is with the Electronics Research Laboratory, College of Engineering, University of California at Berkeley, Berkeley, CA 94720, USA

T. Roska is with both institutes above.

In this paper, in addition to *(i)* introducing the new experimental tool, a real-time programmable, spatial-temporal bifurcation test-bed [1][2][18][19], we show *(ii)* new spatial-temporal patterns. In the following we present some experimental analysis of a simple antisymmetric template in that case when we add only one extra coupling below the central element. We introduce the basic template class and show how the behavior of the CNN changes from stable to chaotic states at different values of the extra coupling. Fig. 1 shows two basic pattern classes that are generated with two different values of the key template element (the extra coupling is greater or less than zero). Throughout this paper, in the images *black color* means *"+1"* and *white color* means *"-1"*. The input is the same as the initial state in every measurements and examples. Boundary conditions are set to *zeroflux* in the simulator and to –1 in the chip measurements, except explicitly noted.

Section II describes the CNN *model* of the simulation and of the chip. Section III presents the basic *pattern classes*. Section IV and Section V describes the *effect* of the *self-feedback* and of the *input and initial state* respectively. Section VI presents an example of a *1D CNN* with first order cells that exhibits *chaotic* behavior. Section VII shows some *additional examples* of propagating pattern classes.

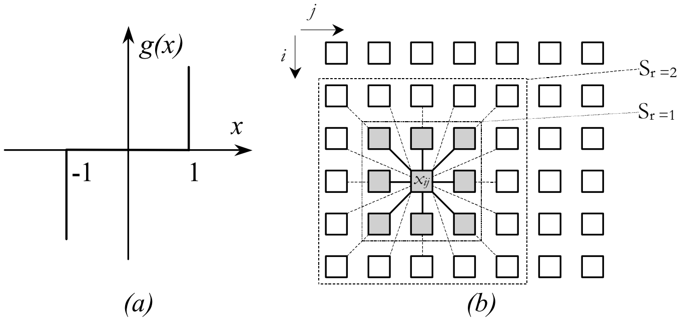


Fig. 2: *(a)* Hard nonlinearity *g(.)*, *(b)* The interpretation of S_r in the CNN array for *r=1* (3×3).

II. The Programmable CNN Model

Throughout this analysis we only consider CNNs with *space invariant* templates i.e. the same template matrix describes the local couplings for each cell. In the model (1) we have a nonzero input *u* that adds a constant value to each cell. This model is implemented as the main elementary instruction of the ACE4k chip, an implementation of the CNN Universal Machine (CNN-UM) [7] architecture. Its stored programmability, via software (in the ALADDIN System [19]) makes it a new and handy tool for us, as a test bed.

A. Simulation

The mathematical model of the simulation of the basic CNN dynamics is the following:

$$\dot{x}_{ij}(t) = -g(x_{ij}(t)) + \sum_{k=-r}^{r}\sum_{l=-r}^{r} A_{kl}x_{i+k,j+l}(t) + \sum_{k=-r}^{r}\sum_{l=-r}^{r} B_{kl}u_{i+k,j+l} + z \quad (1)$$

We use the so-called full range model [8]. In (1), $x_{i,j}$ denotes the state, A_{kl} is the feedback template matrix, B_{kl} is the control template matrix, that describes the effect of the constant input u_{kl} , and *z* is the offset or bias. The integration method throughout the simulation was an *implicit Euler method*. In all cases the boundary cells are set to *zeroflux*, except if explicitly noted differently.

B. Programmable chip measurements – the experimental test bed

The chip experiments were made on the new ACE4k test bed. The ACE4k chip is an analogic cellular microprocessor [18]. The model of the CNN-instruction is described in (2):

$$\dot{x}_{ij}(t) = -g'(x_{ij}(t)) + \sum_{k=-r}^{r}\sum_{l=-r}^{r} A_{kl}x_{i+k,j+l}(t) + \sum_{k=-r}^{r}\sum_{l=-r}^{r} B_{kl}u_{i+k,j+l} + z \quad (2)$$

On the chip, the ideal nonlinearity (See Fig. 2a) is approximated by a *"less hard"* nonlinearity (See Fig. 3).

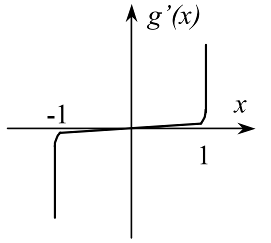


Fig. 3: "Less hard" nonlinearity *g'(.)*

III. THE EFFECT OF VERTICAL COUPLING

In the following we will show how an extra coupling added to a vertically uncoupled template changes the behavior of the system. At first, let us consider a one-dimensional "CCD-like" template of which solution is periodic (See Template 1 and Fig. 4).

$$\text{Template 1:}\ \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0.6 & 0.3 & -0.6 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 0.1$$

During the time transient, cells along the right hand side border of the constant input pattern act like oscillators. The oscillators are only coupled horizontally and the rows operate independently.

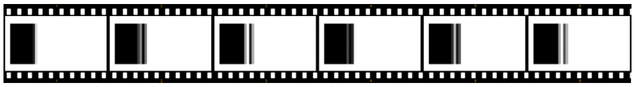


Fig. 4: The time evolution for Template 1, snapshots of the state. Observe that the propagation decays *spatially* after a few pixels, but the oscillation remains. The initial state is the same as the input. (Simulated results, size: 41×23)

The oscillation propagates to the right along the row starting from the triggering constant input (black pixels), and depending on the template values, it stops (dies) after a certain distance or endures until the edge of the array.

Template 2 shows the general form of the nonsymmetric template with an added vertical coupling. When $sq < 0$ the template is called *sign-antisymmetric*. This is shown in Template 3 with $s = -q$. By introducing an extra template element r below of the central one (denoted by p) the homogeneous propagation and oscillation disappears. We get some structured pattern. The character of the structure depends on the sign of the extra template element r. (See Fig. 1 and Table 1.)

$$\begin{array}{l}\text{Template 2:}\\ \textit{general form}\\ s \neq q\end{array} \quad \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ s & p & q \\ 0 & r & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{z} = z$$

$$\begin{array}{l}\text{Template 3:}\\ \textit{antisymmetric}\\ s = -q\end{array} \quad \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ s & p & -s \\ 0 & r & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{z} = z$$

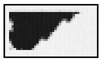
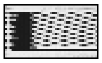
Coupling sign	*Propagating pattern*	*Snapshots*
positive: r > 0	solid inner part, oscillating border cells	
negative: r < 0	texture like oscillating pattern	

Table 1: Categorization of templates containing one extra coupling. The effect of the sign on the shape of the pattern.

A. Positive vertical coupling (r > 0)

If the extra coupling is positive, a pattern is formed which is solid inside, however its right border is oscillating. At the beginning, the input pattern propagates oscillating to the right until a certain extent, and then the global propagation stops and cells along the right border continue oscillating (at certain parameter setting the left border can also oscillate but does not propagate). A typical snapshot of the pattern is shown in Fig. 5, its corresponding template is Template 4. The ruffles along the right border of the pattern move up and right during the evolution – it is a periodic solution in time and space (See Fig. 8). Fig. 6 shows the result of the chip measurement and its corresponding template, Template 5. The exact values of the chip templates are different from the template of the simulation, however the phenomenon can be reproduced quite well. The differences between the simulation and the chip measurements are due to the effect of the AD/DA converters in the supporting circuitry. It is important to note that it is not trivial to get similar attractor to the simulated one on a real, noisy physical system.

$$\text{Template 4: } \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0.55 & 0.3 & -0.55 \\ 0 & 0.5 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 0.6$$

$$\text{Template 5: } \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0.9 & 0.4 & -0.9 \\ 0 & 0.5 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.3 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 2$$

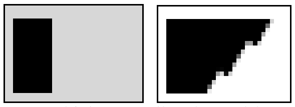


Fig. 5: The input & initial state and snapshot of the output pattern if the coupling is positive. (Simulated result)

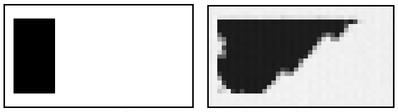


Fig. 6: The input & initial state and snapshot of the output pattern if the coupling is positive. (Chip measurement)

Along the right border of the shadow-like structure local oscillators operate which are coupled horizontally to the nearest neighbor and to the cell below them (see Fig. 7). Generally, there is no oscillation inside the structure, only along its border. Together these local oscillations form the main pattern which looks like the cross-section of waves on the surface of the water (see Fig. 8).

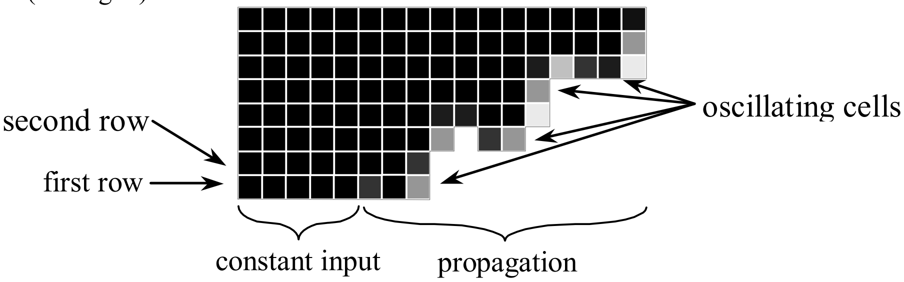


Fig. 7: Structure of the pattern when p is greater than zero.

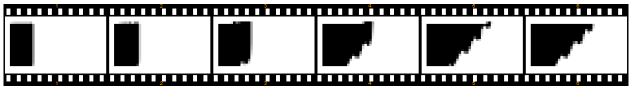


Fig. 8: Time evolution of the pattern when p is positive. Snapshots of the state (simulated results, size: 41×23). The last snapshot shows the largest spatial extent of the generated pattern. There is no global propagation after that moment, only the ruffles travel along the right border of the pattern.

B. Negative vertical coupling (r < 0)

When the vertical coupling is negative a texture-like oscillating pattern is formed. This pattern has a unique structure. It consists of propagating horizontal line segments which spread to the right (See Fig. 11). The spatial extent of the propagation to the right direction changes as we change the parameters but the main characteristics remain the same.

$$\text{Template 6: } \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1.1 & 0.4 & -1.1 \\ 0 & -0.5 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.3 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 0.4$$

$$\text{Template 7: } \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0.9 & 0.4 & -0.9 \\ 0 & -0.5 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.3 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 2$$

Let us define the bottom row of the input pattern (black pixels) as the "first" row (See Fig. 12). The template produces a straight line (a shadow) for this row over a large range of parameter values. The cells in this row have really simple dynamics: they are stable (saturated black pixels). The straight line serves as a constant driving for the cells in the next – "second" – row upward. The "second" row does not produce a propagating pattern. Instead, we find a few neighboring oscillating cells. The number of oscillating cells depends on the other elements of the template. The third row also gives rise to periodic signal but with a different waveform and propagating pattern. Typical snapshots of the pattern are shown in Fig. 9 and Fig. 10 for the simulation and for the chip measurements respectively. The corresponding generator templates are Template 6 and Template 7.

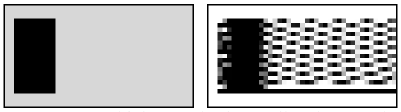


Fig. 9: The input & initial state and snapshot of the output pattern if the vertical coupling is negative. (Simulated result)

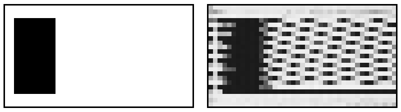


Fig. 10: The input & initial state and snapshot of the output pattern if the vertical coupling is negative. (Chip measurement)

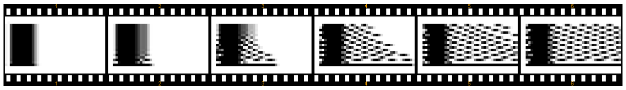


Fig. 11: Time evolution of the pattern when *r* is negative. Snapshots of the state (simulated results, size: 41×23).

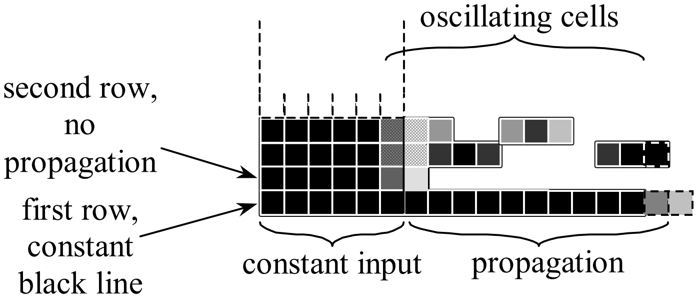


Fig. 12: The structure of the propagating pattern.

IV. The effect of the central template element

If we increase the central element (self feedback) the generated pattern becomes more and more irregular and it can become chaotic. Based on the simulations and the measurements it is possible to construct a partition of the *r*-*p* space. Fig. 13 shows the different dynamic regions of the system parameterized by *r* and *p* of Template 8.

$$\text{Template 8:}\quad \mathbf{A}=\begin{bmatrix}0 & 0 & 0\\ s & p & -s\\ 0 & r & 0\end{bmatrix}\quad \mathbf{B}=\begin{bmatrix}0 & 0 & 0\\ 0 & b & 0\\ 0 & 0 & 0\end{bmatrix}\quad z$$

A. Simulation

The *r*-*p* plane can be divided into *stable-periodic-chaotic* sub-regions (See Fig. 13).

<u>Stable region</u>

Around the periodic and chaotic region there is a stable region with various stable patterns. When *p* is high the effect of the input becomes dominant and therefore the output is almost the same as the input. When *p* is low (negative) the character of the system is diffusive. Between the two extreme values of *p* is a region where the effect of the input is less significant. Therefore the patterns are dominantly one-dimensional or there is no pattern at all.

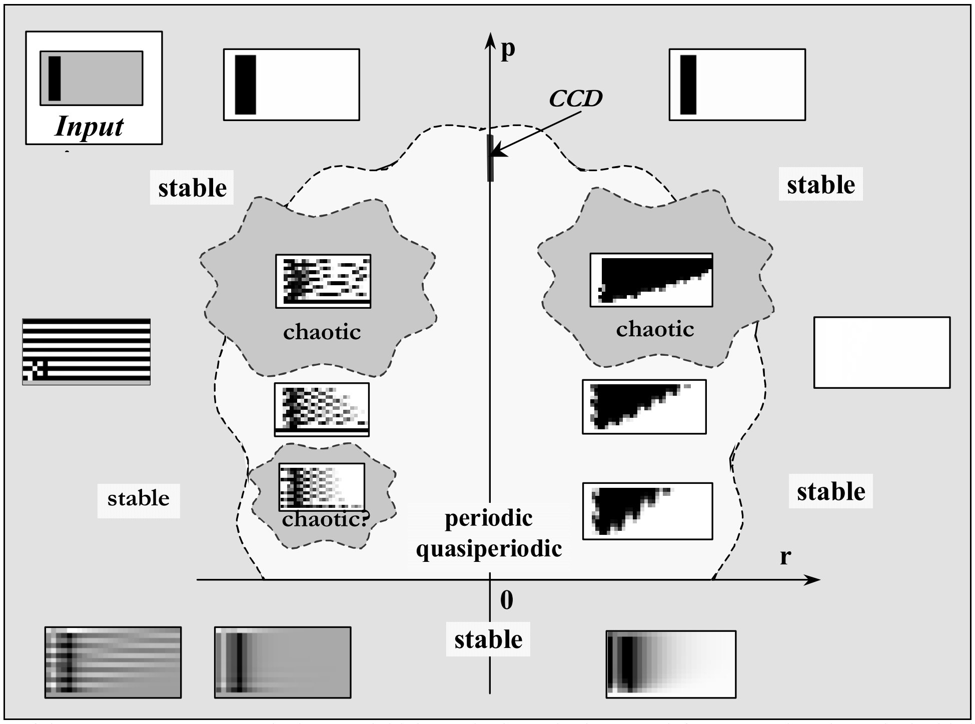


Fig. 13: Partitioning of the r-p parameter space. The input & initial state is shown in the upper left corner. It is a three-pixel wide bar. The pictures in the different regions show few typical snapshots of outputs belonging to that region. The arrangement and size of the different regions gives only qualitative information.

Periodic region

The patterns propagate periodically and have the form of *solid wave-like* and *texture,* as it is described previously in Section III.

Chaotic region

If p is large enough, the system can become chaotic. However, chaotic behavior may occur at smaller p, when r is less than zero. (See Fig. 13).

1) Positive coupling (r > 0)

This section contains results of simulations when r is greater than zero. Observe the transition from the simple to the more complex dynamics. Fig. 14 shows the zoomed structure of the pattern. Figures 15 - 17 show the snapshot of the generated pattern, the time evolution of one sampled state variable, the power spectrum of that variable and the trajectory of the sampled cell and its neighbor cell. The captions contain the actual value of parameter *p (p = [0.5, 0.87], r = 0.3).* The generator template for the figures is Template 9.

$$\text{Template 9:}\quad \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1.1 & p & -1.1 \\ 0 & 0.3 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 0.1$$

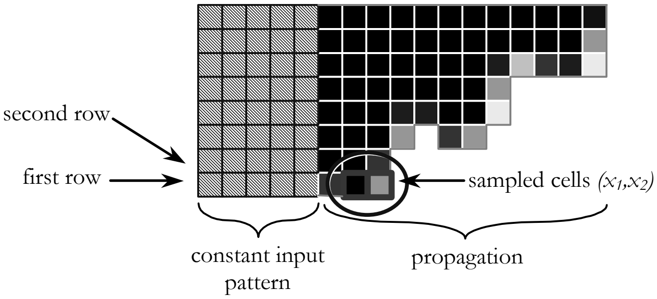


Fig. 14: In the first row, to the right from the edge of the black part of the constant input (denoted by blue striped boxes) the second and third cells were sampled as shown.

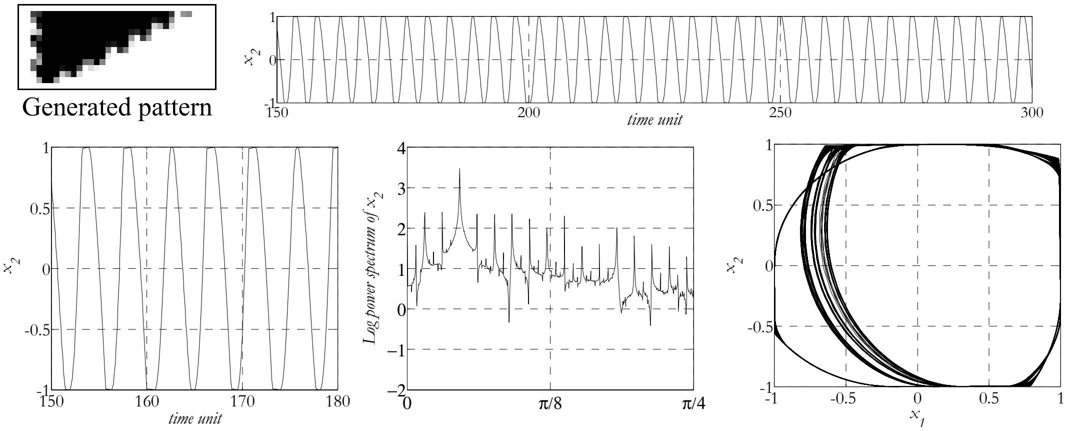


Fig. 15: Snapshot of the output, the time evolution of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row (p=0.5).

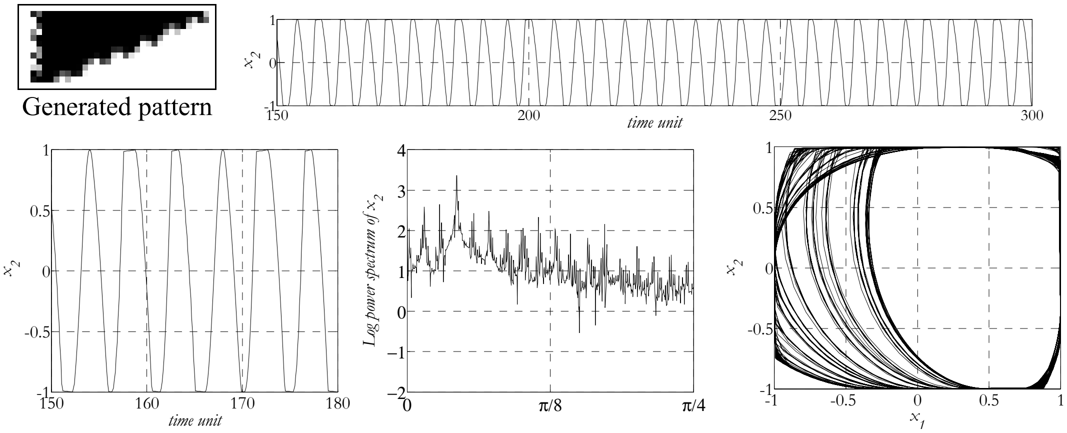


Fig. 16: Snapshot of the output, the time evolution of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row (p=0.6).

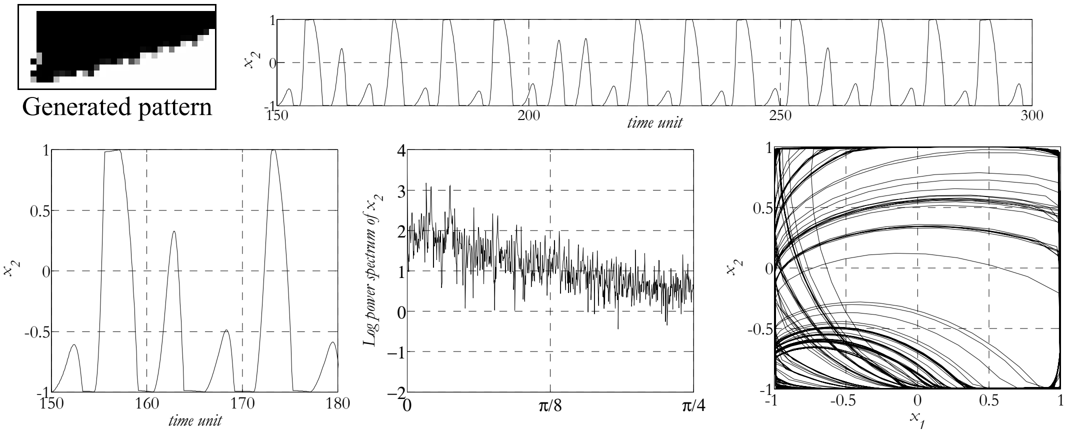


Fig. 17: Snapshot of the output, the time evolution of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row (p=0.87).

2) Negative coupling ($r < 0$)

When r is less than zero, a texture like oscillating pattern is formed. Fig. 18 shows the zoomed structure of the pattern. Observe the transition from the simpler to the more complex dynamics in Figures 19 - 21. The figure captions contain the actual value of parameter *p (p=[0.2, 0.7], r = – 0.3).* See the generator template, Template 10.

$$\text{Template 10:}\quad \mathbf{A}=\begin{bmatrix} 0 & 0 & 0 \\ 1.1 & p & -1.1 \\ 0 & -0.3 & 0 \end{bmatrix}\quad \mathbf{B}=[1.2]\quad z=0.1$$

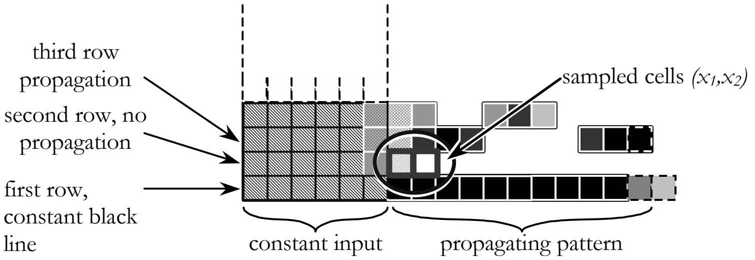

third row propagation
second row, no propagation
first row, constant black line
sampled cells (x_1,x_2)
constant input
propagating pattern

Fig. 18: In the second row, right from the edge of the black part of the constant input (denoted by blue striped boxes) the first and second cells were sampled as shown.

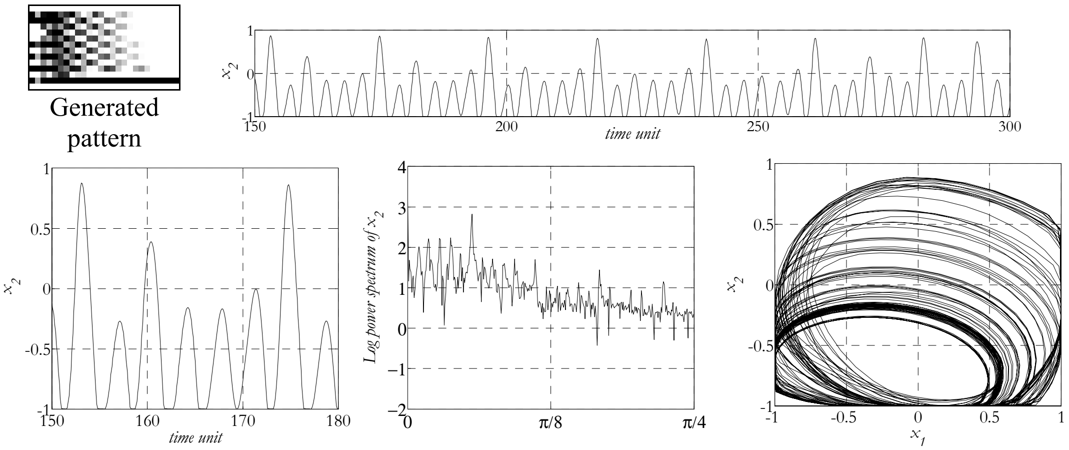


Fig. 19: Snapshot of the output, the time evolution of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row (p=0.2).

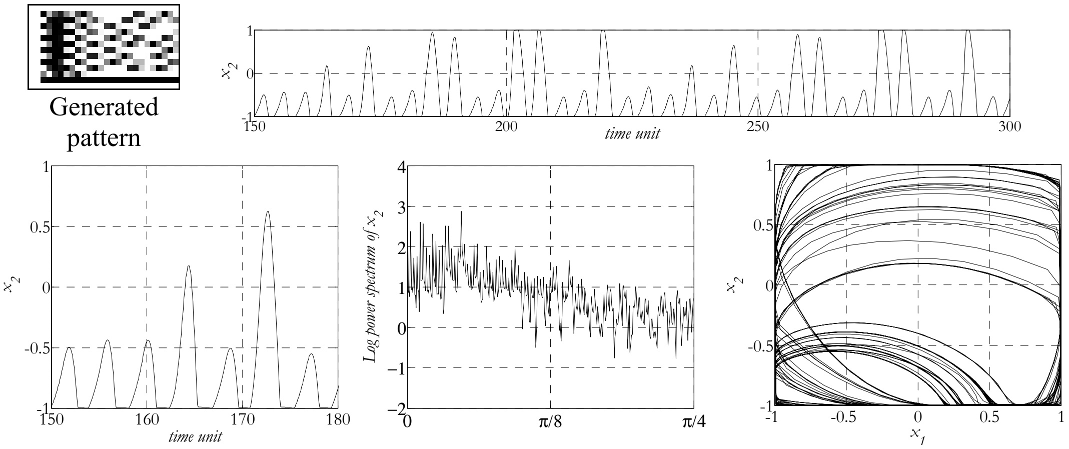


Fig. 20: Snapshot of the output, the time evolution of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row (p=0.6).

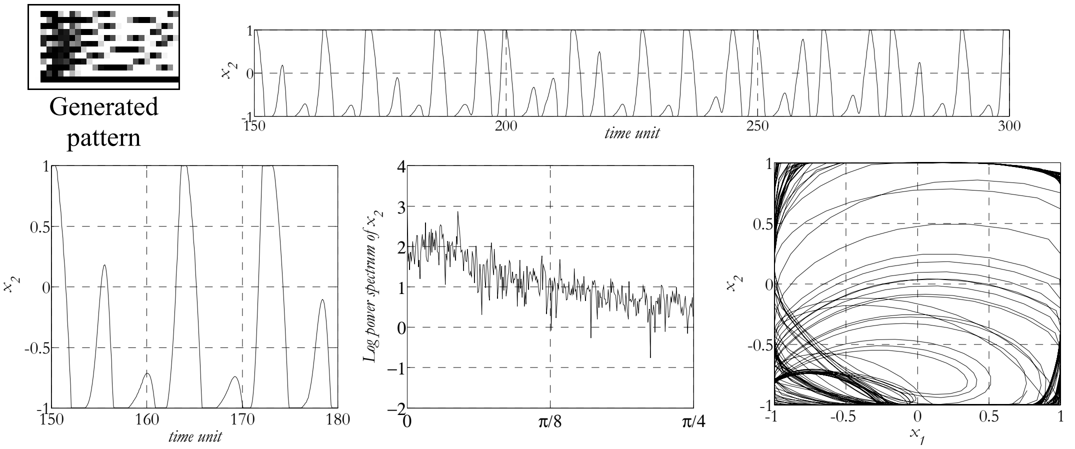


Fig. 21: Snapshot of the output, the time evolution of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row (p=0.7).

B. Programmed chip measurement

The following subsections contain programmed chip measurements at different values of *p* for positive and negative values of *r*. While the measured waveforms do not coincide completely with that of the simulation, the qualitative details of the phenomenon are the same. In the figures the trajectories are virtually scaled and the null points are virtually shifted compared to the simulation results. This is due to the AD/DA supporting circuitry of the chip.

When there is no stable equilibrium, such as in the case of periodic or chaotic steady state behavior, it is not trivial to find the equivalent settings for the chip to get the same result as that of the simulation. Since the chip is a real, physical, noisy system with high complexity and interconnection, the model that we simulate is inevitably different from the model of the chip. In spite of this it is still possible to get appropriate result.

1) Positive coupling (r > 0)

Cells from the first row were sampled (See Fig. 14). When *p* is small, the power spectrum contains dominant peaks according to the periodic signal. Later, when *p* is higher the peaks disappear or significantly decrease. Figures 22 - 24 show the measured time series, power spectrum and the trajectory of the two sampled cells. The sampling position is shown in Fig. 14. The generator template for the figures is Template 11. The captions contain the actual value of parameter *p (p = [0.51, 0.85], r = 0.24).*

$$\text{Template 11: } \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & p & -1 \\ 0 & 0.24 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.6 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 2.1$$

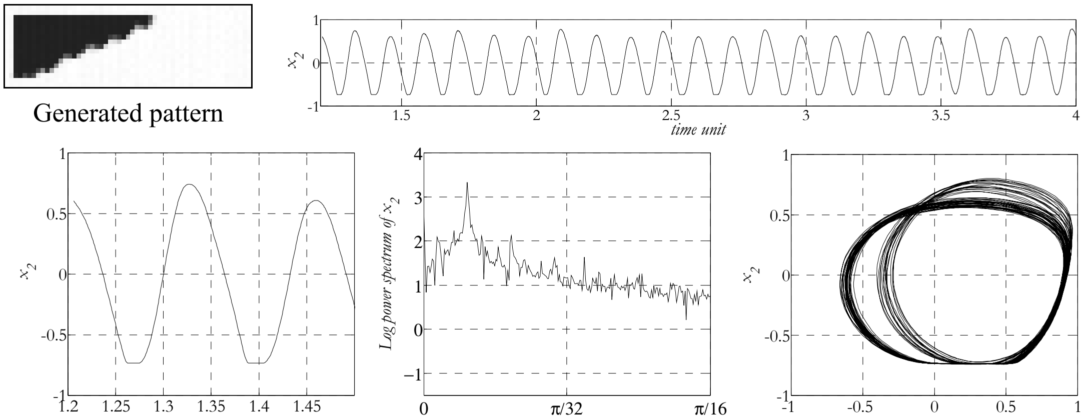


Fig. 22: Snapshot of the output, the time series of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighboring cell from the first row (p=0.51).

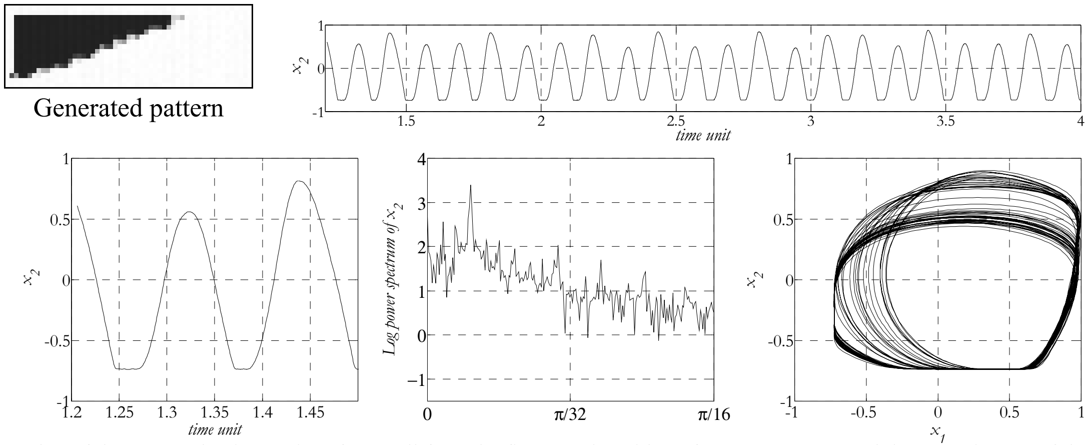


Fig. 23: Snapshot of the output, the time series of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighboring cell from the first row (p=0.61).

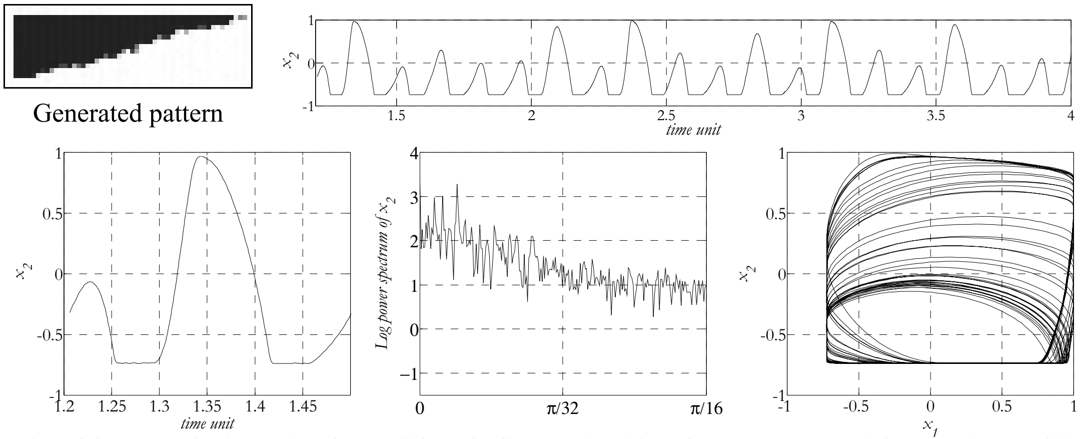


Fig. 24: Snapshot of the output, the time series of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighboring cell from the first row (p=0.85).

2) Negative coupling (r < 0)

In the negative coupling case we experiment with a similar phenomenon to that of the simulation. Figures 25 - 27 show the measured time series, power spectrum and the trajectory of the two sampled cells. Fig. 18 shows the sampling position. The generator template for the figures is Template 12. The captions contain the actual value of parameter *p (p = [0.42, 0.68], r = -0.4).*

$$\text{Template 12:}\quad \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1.2 & p & -1.2 \\ 0 & -0.4 & 0 \end{bmatrix} \quad \mathbf{B} = [1.4] \quad z = 2$$

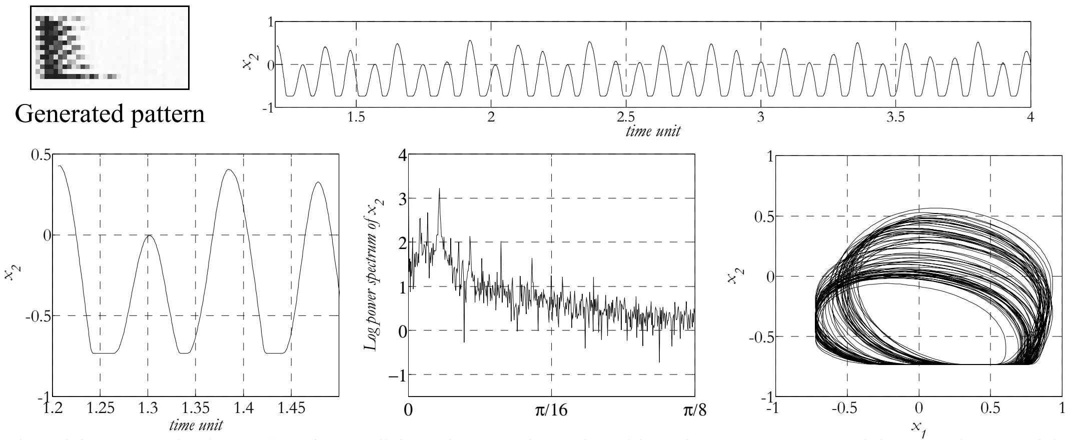


Fig. 25: Snapshot of the output, the time series of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighboring cell from the second row (p=0.42).

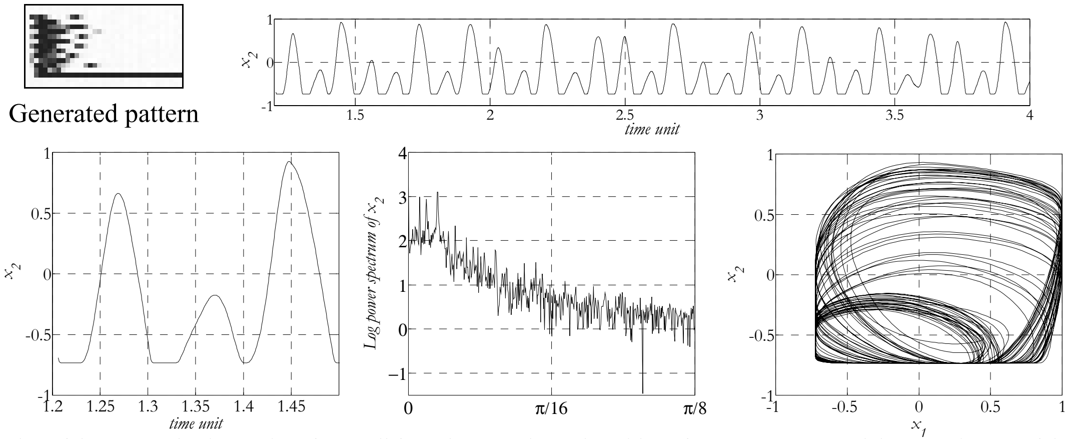


Fig. 26: Snapshot of the output, the time series of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighboring cell from the second row (p=0.6).

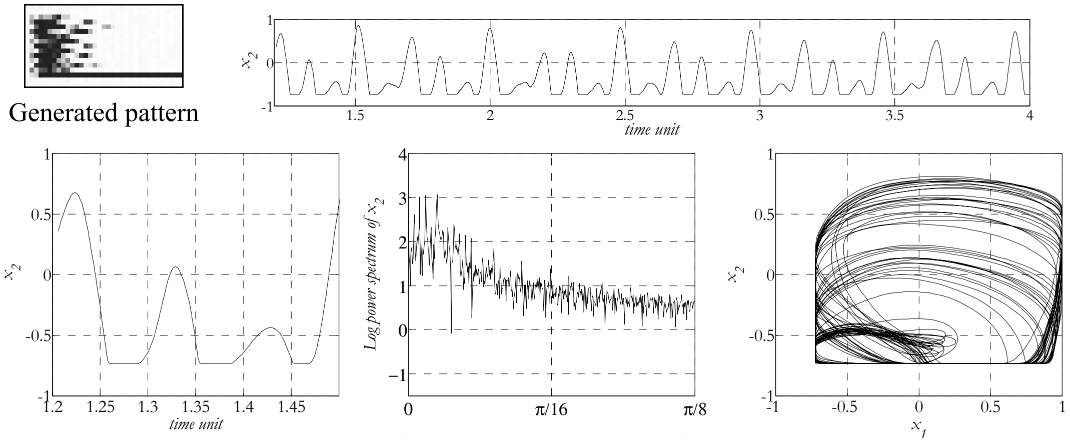


Fig. 27: Snapshot of the output, the time series of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighboring cell from the second row (p=0.68).

C. A. Visual signatures

As it can be seen from the results, there is a correlation between the time evolution of the selected cells of the CNN and the dynamic behavior of the whole array. The propagation preserves the history of the dynamics, therefore the output picture can be suitable for characterizing the system without cell data measurements. Observe that, due to the spatial-temporal patterns, it is difficult to find characteristic 2D snapshots. Table 2 shows the most characteristic signatures for the state of the CNN.

Table 2: Visual signatures for different values of ***r*** and ***p***.

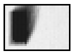
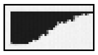
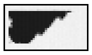
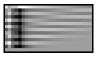
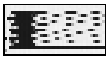
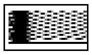
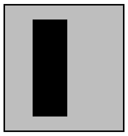
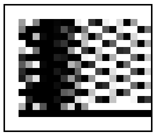
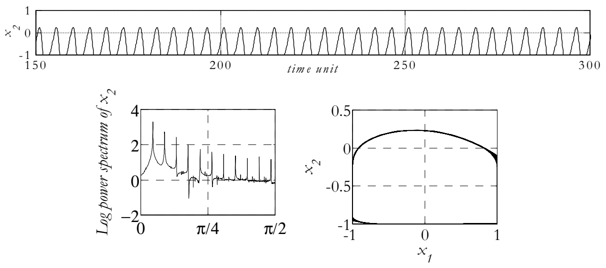
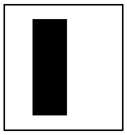
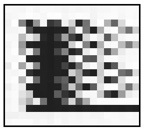
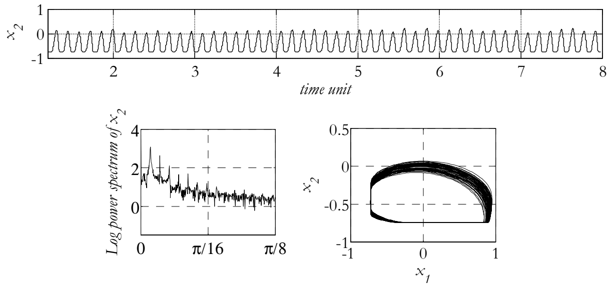
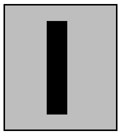
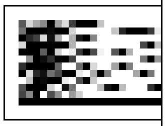
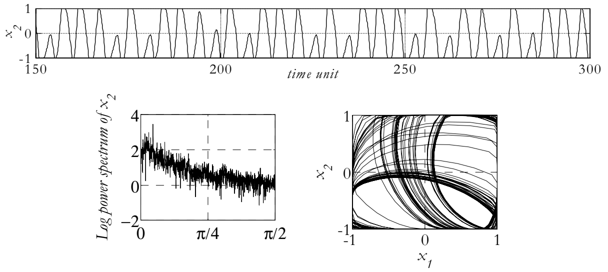
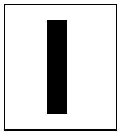
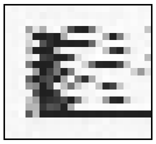
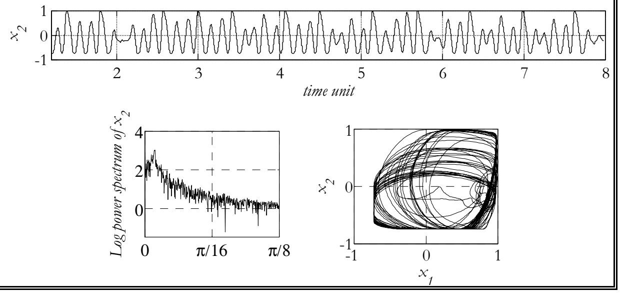
Stable		*Chaotic*		*Periodic*	
	*r > 0, p < 0*		*r > 0, p >> 0*		*r > 0, p > 0*
	*r < 0, p < 0*		*r < 0, p >> 0*		*r < 0, p > 0*

Table 3 Different dynamical behaviors with the same template but with different input and initial state. The time evolution of one cell from the second row, logarithm of power spectrum of the same cell and the 2D trajectory of the same cell and the neighbor are shown.

Periodic	*simulation*	 input	 snapshot of the output	
	chip measurement	 input	 snapshot of the output	
Chaotic	*simulation*	 input	 snapshot of the output	
	chip measurement	 input	 snapshot of the output	

V. THE EFFECT OF THE CONSTANT INPUT AND INITIAL STATE

An inherent property of the chaotic systems is the extreme sensitivity to the initial condition. In this section some results relating to this aspect are presented.

A. Periodic-chaotic transition

Table 3 shows the effect of the different input patterns in the case of simulation and of chip measurements. The applied templates are the same for the two different inputs, i.e. the different behavior of the system is due to the difference of the input pattern.

The first input is a five pixel wide black vertical bar and the other input is a three pixel wide black vertical bar. The input and the initial state are the same. Template 13 and 14 shows the templates for the simulation and for chip measurements respectively. The gray background color of the input image for the simulation is to mimic the property of the chip, namely, its gray level shifting property. Using these level-shifted images we get similar result to that of the chip.

Template 13:

$$\mathbf{A}=\begin{bmatrix}0 & 0 & 0\\ 1.1 & 0.7 & -1.1\\ 0 & -0.3 & 0\end{bmatrix}\quad \mathbf{B}=\begin{bmatrix}0 & 0 & 0\\ 0 & 1.2 & 0\\ 0 & 0 & 0\end{bmatrix}\quad \mathfrak{z}=0.1$$

Template 14:

$$\mathbf{A}=\begin{bmatrix}0 & 0 & 0\\ 0.9 & 0.5 & -0.9\\ 0 & -0.5 & 0\end{bmatrix}\quad \mathbf{B}=\begin{bmatrix}0 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}\quad \mathfrak{z}=2.1$$

The result shows that if the input is the five pixel wide bar then the transient of the cell is periodic. But if the input is a three pixel wide bar, the transient – and the propagating pattern too – is chaotic. The reason for the difference is that the local oscillators along the left and right border of the bar can influence each other. This can happen only if there is no stable (constant, saturated black) vertical column of cells along the center of the bar. If the input is the three pixel wide bar there is no column of saturated *(+1)* stable cells in the bar. In the other case there is at least a one pixel wide column of saturated stable cells horizontally. Thus, the oscillators along the left and right border of the bar are uncoupled. The cells were sampled in the second row.

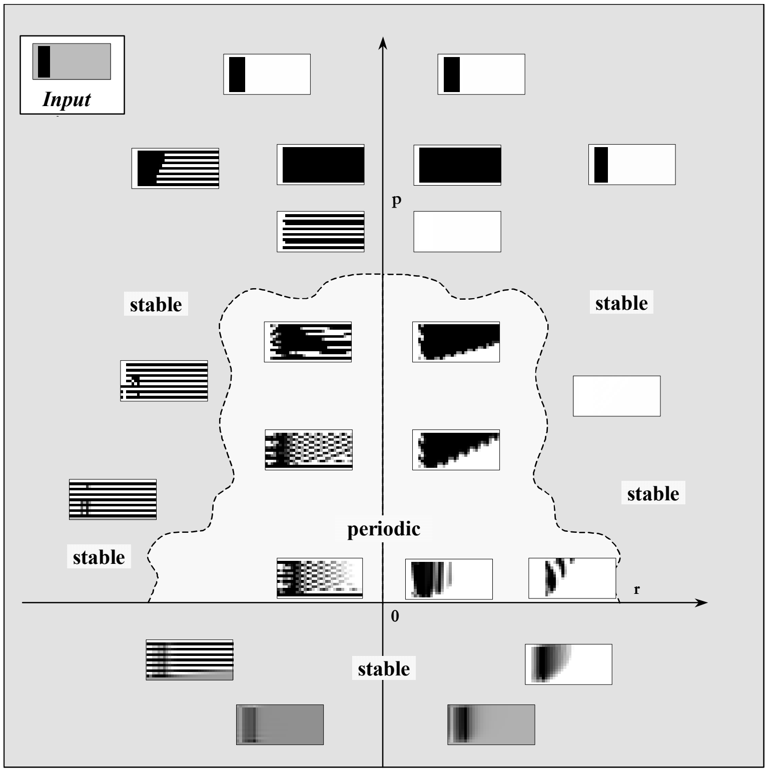


Fig. 28: Partitioning of the r-p parameter space when the input is a five pixels wide bar. The input picture is shown in the upper left corner. The pictures in the different regions show few typical snapshots of the output patterns belonging to that region. With this input no chaotic behavior has been found up to now. The arrangement and size of the different regions gives only qualitative information.

Fig. 28 shows the *r-p* diagram in that case when the input and initial state is changed to the five pixel wide vertical bar. No chaotic behavior has been found up to now. However it cannot be excluded that it is possible to find a certain parameter setting at which the system produces chaos. A necessary condition for this seems to be that the stable saturated cells along the center of the bar become unstable.

B. Stable-periodic transition

Fig. 29 illustrates that a single pixel perturbation can alter the general dynamic behavior of the system. The only difference between the three inputs, which are also the initial state, is that a single pixel is changed from black to gray (and from gray to black in the other case) in the middle of the right vertical edge of the bar. In the first case there is no propagation and no pattern. However the second and third input produces a periodic pattern (See Fig. 29). The boundary condition is periodic (torus-like left-right and bottom-up connections, this can also be programmed on the chip). The generator template is Template 15. This change of the dynamical state could be exploited to detect changes in the input image. It is probably possible to find a template that produces e.g. stable steady state for one input pattern and e.g. periodic for another one.

$$\text{Template 15: } \mathbf{A}=\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0.5 & -1 \\ 0 & -0.5 & 0 \end{bmatrix} \quad \mathbf{B}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.3 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z=0$$

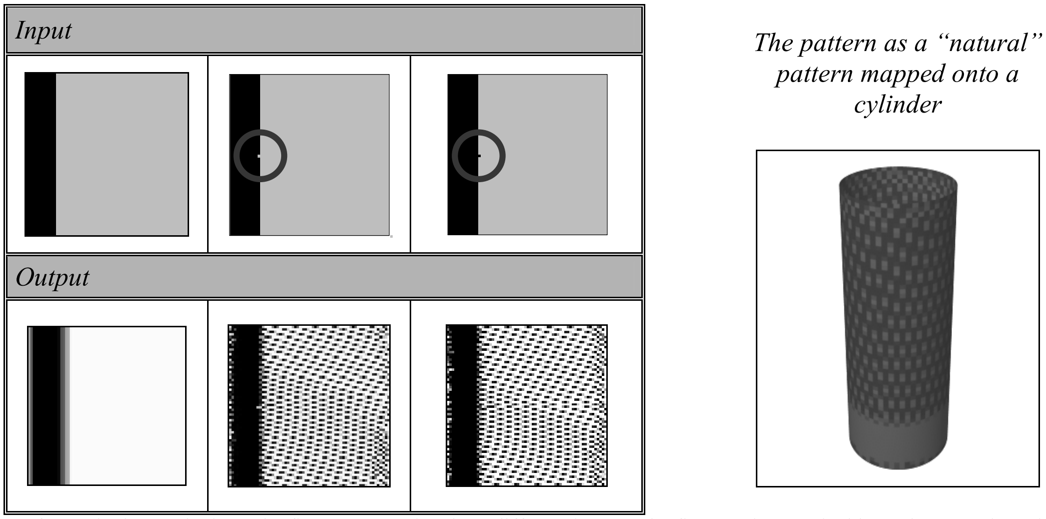
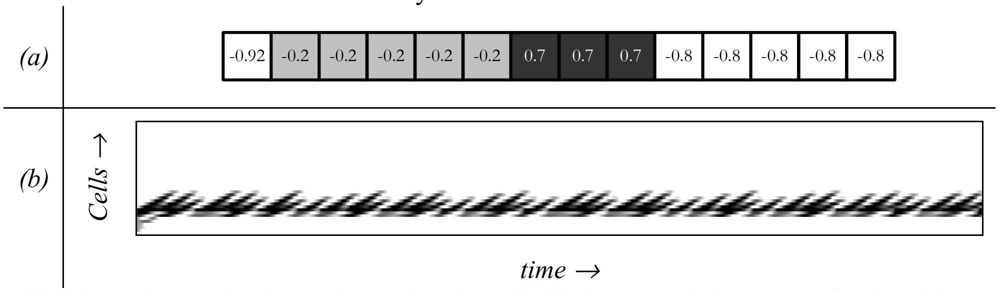


Fig. 29: Effect of one pixel perturbation. The first row contains three different inputs. The first one is a vertical bar. The second one is the same except one pixel: a pixel is *clipped off* from the middle of the right border of the pattern. In the third one a pixel is *added* to the middle of the right border of the pattern. The circles denote the location of the difference. The second row contains the corresponding snapshots of the output patterns. In the first column the solution is equilibrium. The second and third column shows periodic (in space and time) solution.

VI. 1D CHAOS

Based on the measurements and the simulations we can construct probably the simplest template (e.g. Template 16) for a 1D CNN that can exhibit chaos with an appropriate input shown in Fig 30. The minimal number of cells that is necessary to produce chaos has not been determined yet.



-0.92	-0.2	-0.2	-0.2	-0.2	-0.2	0.7	0.7	0.7	-0.8	-0.8	-0.8	-0.8	-0.8

Fig. 30: *(a)* Input and initial state for a 1D chaotic CNN that consists of 14 cells. *(b)* The generated 1d pattern as a function of time.

The general structure of the templates presented in this paper shows that a CNN cell in the array does not depend on the cells above it. In the cases above (Templates 13,14) we measured and simulated chaotic signals in the row which has a constant valued row below it. Therefore, if we integrate the effect of the constant valued row into the constant input it is possible to construct a 1D CNN (template, initial state and input) for which the system is chaotic (see Fig 30 and Fig 31).

All boundary cells were set to zero.

$$\text{Template 16:}\ \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1.1 & 0.39 & -1.1 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad z = 0.2$$

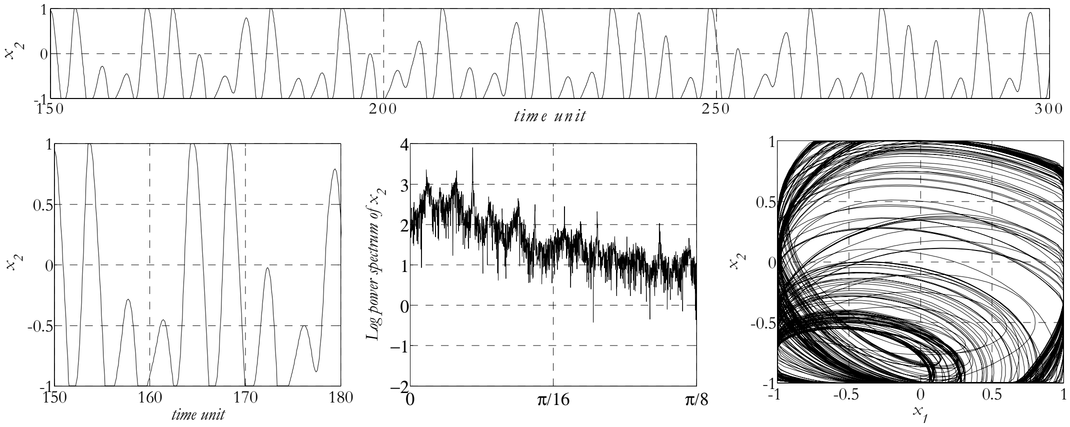
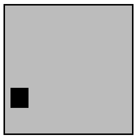
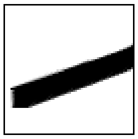
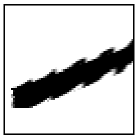
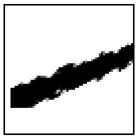
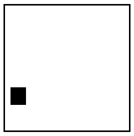
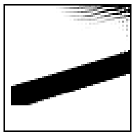
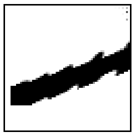
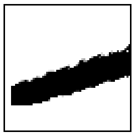


Fig. 31: The time evolution, the power spectra of one cell and the trajectory of two neighboring cells.

VII. Additional propagating pattern examples

This section shows some patterns which are generated with more complicated templates. The basic structure is the same i.e. the antisymmetry is preserved but some nonzero couplings are added. The corresponding templates matrices are shown below the figures.

Table 4: Snapshots of the simulation and of the chip measurements of the complex wave template. Input and initial state are the same. The gray background of the input for the simulation is to mimic the level shift property of the chip.

Simulation					
Snapshots of the output					
	Input	*p*	*0.05*	*0.3*	*0.7*
Chip measurement					
Snapshots of the output					
	Input	*p*	*0.15*	*0.41*	*0.94*

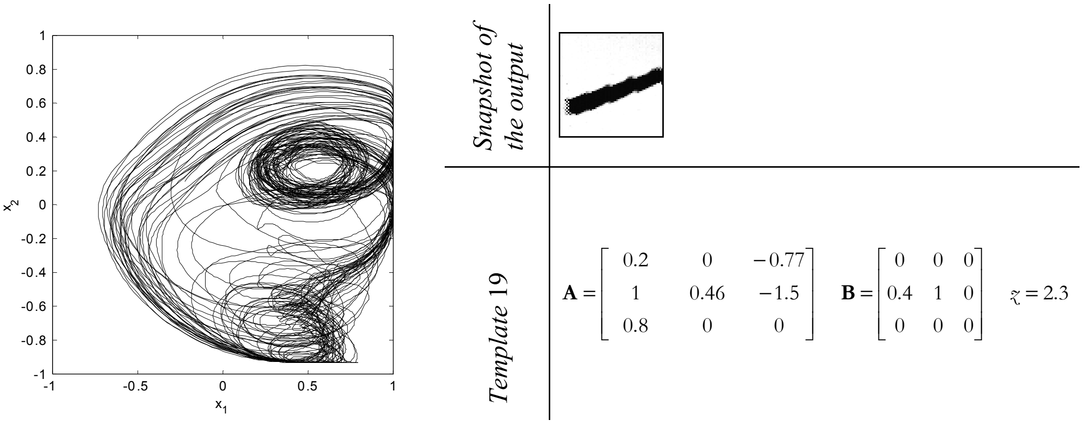
A. Wave shadow

This special shadow operator produces different patterns depending on the central element *p*. The main structure remains the same at different values of *p*, but the border of the shadow looks like traveling wrinkles. The phenomenon related to the change of *p* is similar to that of described in Section IV. Template 17 and 18 are the generator templates of the patterns in Table 4 for simulation and for chip measurements respectively.

Template 17: Template 18:

$$\mathbf{A}=\begin{bmatrix}0 & 0 & -0.6\\ 1 & P & -1\\ 0.8 & 0 & -0.1\end{bmatrix}\quad \mathbf{B}=\begin{bmatrix}0 & 0 & 0\\ 0 & 2 & 0\\ 0 & 0 & 0\end{bmatrix}\quad z=0.9\qquad \mathbf{A}=\begin{bmatrix}0 & 0 & -0.6\\ 1 & P & -1\\ 0.8 & 0 & -0.2\end{bmatrix}\quad \mathbf{B}=\begin{bmatrix}0 & 0 & 0\\ 0 & 3 & 0\\ 0 & 0 & 0\end{bmatrix}\quad z=4.2$$

With a different template setting (Template 19) a characteristic trajectory is measured on the chip (See Fig. 32 and Fig. 33).



$$\mathbf{A}=\begin{bmatrix}0.2 & 0 & -0.77\\ 1 & 0.46 & -1.5\\ 0.8 & 0 & 0\end{bmatrix}\quad \mathbf{B}=\begin{bmatrix}0 & 0 & 0\\ 0.4 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}\quad z=2.3$$

Fig. 32: The measured trajectory of the attractor and a snapshot of the propagating pattern. *(chip measurement)*.

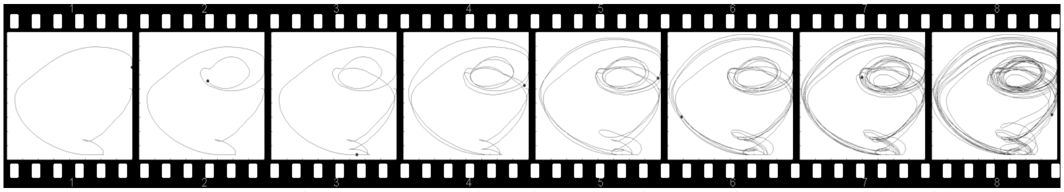


Fig. 33: The measured trajectory of the attractor. Snapshots of trajectory of two out of 4096 state variables of the CNN array *(chip measurement)*.

B. *"Four pixels" examples*

The patterns in this subsection were simulated using the full-range CNN model (**1**). The next examples show different patterns, all of which were generated from the same input. The "seed" of the patterns is a four-pixel wide horizontal line section. The initial state was the same as the input that is shown in Fig. 34.

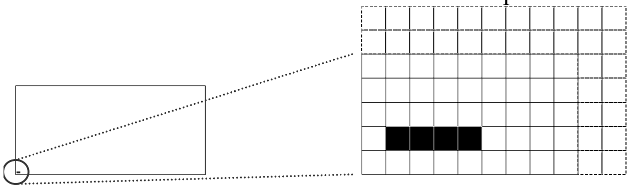


Fig. 34: Input and initial state for the propagating patterns.

1) Bird

This example shows a propagating pattern which makes an impression of distant flying birds. (See Fig. 35 and Template 20)

$$\text{Template 20:}\quad \mathbf{A}=\begin{bmatrix}0 & 0 & -0.8\\ 1.1 & 1 & -1\\ 0.7 & 0 & 0\end{bmatrix}\quad \mathbf{B}=\begin{bmatrix}0 & 0 & 0\\ 0 & 1.1 & 0\\ 0 & 0 & 0\end{bmatrix}\quad z=0.9$$

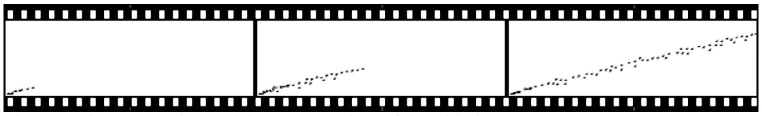


Fig. 35: Three snapshots of the propagation of the "bird" pattern.

2) Volcano eruption

This pattern "erupts" periodically during the propagation. It resembles fluid coming out of a pipe. (See Fig. 36 and Template 21)

Template 21: $\mathbf{A} = \begin{bmatrix} 0 & 0 & -1 \\ 1.1 & 1.3 & -1 \\ 1.1 & 0 & 0 \end{bmatrix}$ $\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ $z = 1.4$

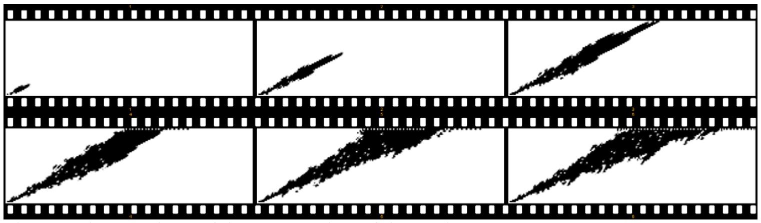


Fig. 36: Six snapshots of the propagation of the “volcano” pattern..

VIII. Simulation time vs. real-time measurements

A sophisticated simulation of chaotic systems takes several minutes (or even hours), especially when the dimension of the system is high. Using the new programmable ACE4K test bed it is possible to speed up the analysis process of the chaotic system by at least four orders of magnitude. Table 5 shows the comparison of a typical simulation and the real time chip measurement.

Table 5: Comparison of simulation and chip measurement time.

Simulation (500Mhz PC)	*Chip measurement (with sampling)*	*Chip measurement (without sampling)*
25 min = 1500 sec	*200 ms*	*5 ms*

IX. Conclusion

Various propagating patterns represent a new and interesting spatial-temporal pattern class. Long and sophisticated analysis and digital computer simulation are necessary to uncover the characteristics of the complex dynamics of this class. The new analogic programmable ACE4k topographic microprocessor test-bed is an ideal tool for experimental work, which makes it possible to speed up the analysis process by more than four orders of magnitude.

Both the simulations and the real-time experiments with the ACE4k test bed showed that if we add a single nonzero element *r* to a CCD-like antisymmetric template, the behavior of the system is significantly changed, complex dynamics occurs and characteristic patterns are formed. At certain parameter settings – especially when the self-feedback *p* is high – the output pattern showed a unique chaotic pattern. The layout of the *r-p* parameter space showed a characteristic structure. As it can be seen from the parameter space diagram, the dynamics of the system strongly depends on the self-feedback and on the initial state (and input). This sensitive dependence on the initial state can be suitable for morphological detection. We expect that future works discover additional interesting and useful properties of this template class and helps to understand the complex dynamics more. This would also lead to a new class of analogic spatial-temporal algorithms for the dynamic detection of exotic and complex events.

References

[1] L. Chua and T. Roska, *Cellular neural networks and visual computing - Foundations and applications*, Cambridge University Press, Cambridge ISBN: 0521652472, 2001

[2] T. Roska, Á. Zarándy, S. Zöld, P. Földesy and P. Szolgay, “The Computational Infrastructure of Analogic CNN Computing - Part I: The CNN-UM Chip Prototyping System”, *IEEE Trans. Circuits and Systems I:* Special Issue on Bio-Inspired Processors and Cellular Neural Networks for Vision, (CAS-I Special Issue), Vol. 46, No.2, pp. 261-268, 1999

[3] G. Liñan, S. Espejo, R. Domínguez-Castro, A. Rodríguez-Vázquez, “The CNNUC3: An Analog I/O 64x64 CNN Universal Machine Chip Prototype with 7-Bit Analog Accuracy”, in *Proceedings of IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, (CNNA'2000), Catania, 0-7803-6344-2, 2000, pp. 201-206.

[4] L.O. Chua and L. Yang, “Cellular Neural Networks: Theory*”, IEEE Trans. Circuits and Systems*, Vol.35. pp. 1257-1272, 1988.

[5] L.O. Chua and L. Yang, “Cellular neural networks: Applications”, *IEEE Trans. Circuits and Systems*, Vol.35. pp. 1273-1290, 1988.

[6] L.O. Chua and T. Roska, “The CNN paradigm”, *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, Vol.40, No. 3, pp. 147-156, 1993.

[7] T. Roska and L.O. Chua, The CNN universal machine: an analogic array computer, *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing* Vol. 40, No. 3, pp. 163-173, 1993.

[8] A. Rodriguez-Vázquez, S. Espejo, R. Dominguez-Castro, J.L. Huertas, and E.Sánchez-Sinencio, “Current-Mode Techniques for the Implementation of Continuous- and Discrete-Time Cellular Neural Networks”, *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, Vol.40. No.3. pp. 132-146, 1993

[9] F. Zou and J.A. Nossek, “Stability of cellular neural networks with opposite sign templates”, *IEEE Trans. Circuits and Systems*, (CAS), Vol. 38. pp. 675-677, 1991

[10] P. Thiran, G. Setti, and M. Hasler, “An approach to information propagation in 1-D cellular neural networks—Part I: Local diffusion”, *IEEE Trans. Circuits Systems* I, Vol.45, No.8, pp. 777–789, August 1998.

[11] G. Setti, P. Thiran, and C. Serpico, “An approach to information propagation in 1-D cellular neural networks—Part II: Global Propagation”, *IEEE Trans. Circuits Systems* I, Vol.45, No.8, pp. 790–811, August 1998.

[12] P. Thiran, K. R. Crounse, L. O. Chua, and M. Hasler, “Pattern formation properties of autonomous cellular neural networks,” *IEEE Trans. Circuits Systems* I, vol. 42, pp. 757–776, Oct. 1995.

[13] F. Zou, J.A. Nossek, “Bifurcation and Chaos in Cellular Neural Networks”, *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, (CAS-I), Vol.40, No.3, pp.166-173, 1993

[14] M. Biey. M. Gilli, and P. Checco, “Complex Dynamic Phenomena in Space-Invariant Cellular Neural Networks”, *IEEE Trans. Circuits and Systems I: Special Issue*, March 2000 *(in print)*.

[15] G. Manganaro, P. Arena, L. Fortuna “Cellular Neural Networks: Chaos, Complexity and VLSI Processing”, Springer Verlag, New York; ISBN: 3540652027, 1999

[16] L. O. Chua and T. Roska, “Stability of a Class of Nonreciprocal Cellular Neural Networks”, *IEEE Trans. Circuits and Systems*, Vol.37.pp. 1520-1527, 1990

[17] Cs. Rekeczky and L.O. Chua, “Computing with Front Propagation: Active Contour and Skeleton Models in Continuous-Time CNN”, *Journal of VLSI Signal Processing Special Issue:* Spatiotemporal Signal Processing with Analogic CNN Visual Microprocessors, (JVSP Special Issue), Vol. 23. No.2/3. pp. 373-402, guest editors: T. Roska and Á. Rodríguez-Vázquez, Kluwer, ISSN 0922-5773, 1999

[18] T. Roska and A. Rodríguez-Vázquez (editor), *Towards the Visual Microprocessor: VLSI Design and the Use of Cellular Network Universal Machines*, John Wiley & Sons; Chichester, ISBN: 0471956066, 2000

[19] The ALADDIN System, *http://www.analogic-computers.com/*

APPENDIX

The programming of the test-bed is possible through a high-level language interface called *Alpha*. Here you will find a typical *Alpha* language code for the simulation and measurement.

```
PROGRAM complex_dynamics (;);
CONSTANT
ZERO = 0.0;
ONE = 1;
BoundaryValue = ZEROFLUX;
TimeValue = 10;
TimeStepValue = 0.1;
ENDCONST;
A_CHIP
SCALARS
IMAGES
chip_p1: ANALOG;
chip_p2: ANALOG;
chip_p3: ANALOG;
chip_p4: ANALOG;
chip_pOut: ANALOG;
chip_InputValue:   ANALOG;
chip_StateValue:   ANALOG;
chip_BiasValue:    ANALOG;
ENDCHIP;
E_BOARD
SCALARS
IMAGES
board_p1:     BYTE;
board_p2:     BYTE;
board_p3:     BYTE;
output:       BYTE;
ENDBOARD;
FUNCTION func_complex;
USE ( COMPLEX );
  SwSetTimeStep (TimeStepValue); /* Set the timestep */
  chip_p1 := board_p1;
  chip_InputValue := ZERO;
  COMPLEX( chip_InputValue, chip_p1, chip_pOut, TimeValue, BoundaryValue); /* Computation of the
transient */
  output := chip_pOut;
ENDFUNCT;
PROCESS complex_dynamics; /* this is the main program */
USE ();
  HostGetImage(ONE, board_p1, A_GRAY);
        /* Acquire the input */
    func_COMPLEX; /* function call */
  HostDisplay(output, ONE); /* Display the result */
ENDPROCESS;
ENDPROG;
```

I. Petrás received his B.Sc and M.Sc degrees in information engineering from Bánki Donát Polytechnic and University of Veszprém in Hungary in 1994 and 1998, respectively. Since 1998 he has been Ph.D student in the Technical University of Budapest and the Analogical and Neural Computing Research Laboratory of the Computer and Automation Research Institute of the Hungarian Academy of Sciences.

From October 2000 to May 2001 he worked in the Nonlinear Research Laboratory, University of California, Berkeley as visiting scholar. From May 2002 to June 2002 he worked in the SISTA-COSIC laboratory, Department of Electrical Engineering, Katholieke Universiteit Leuven, Belgium. Now he is working toward his Ph.D. in the Analogical and Neural Computing Research Laboratory of the Computer and Automation Research Institute of the Hungarian Academy of Science. His research interests include new parallel image processing methods that can be implemented on Cellular Neural Networks; analysis and design of spatio-temporal chaotic pattern generation and analysis, spatio-temporal nonlinear wave filters for 2D signal processing tasks; somatosensory system modeling, ultra-high speed object recognition and tracking.

Tamás Roska (M'87-SM-90-F'93) received the Diploma in electrical engineering from the Technical University of Budapest, Budapest, Hungary, in 1964 and the Ph.D. and D.Sc. degrees in 1973 and 1982, respectively.

From 1964 to 1970 he was with the Measuring Instrument Research Institute, Budapest. From 1970 to 1982 he was with the Research Institute for Telecommunication, Budapest. Since 1982 he has been with the Computer and Automation Institute of the Hungarian Academy of Sciences, where he is Head of the Analogic and Neural Computing Research Laboratory. Since 1989 he has been a Visiting Scholar at the Department of Electrical

Engineering and Computer Sciences and the Electronics Research Laboratory and was recently a Visiting Research Professor at the Vision Research Laboratory of the University of California at Berkeley.

His main research areas are cellular neural networks, nonlinear circuits and systems, and analogic spatiotemporal supercomputing.

Since 1975 he has been a member of the Technical Committee on Nonlinear Circuits and Systems of the IEEE Circuits and Systems Society. Recently, he has served twice as Associate Editor of the IEEE Transactions on Circuits and Systems. He received the IEEE Fellow Award for contributions to the qualitative theory of nonlinear circuits and the theory and design of programmable cellular neural networks. In 1993 he was elected a Member of the Academia Europaea (European Academy of Sciences, London) and the Hungarian Academy of Sciences. In 1999 he became the Founding Chair of the Technical Committee on Cellular Neural Networks and Array Computing of the IEEE Circuits and Systems Society. His IEEE membership number is 08547077.

Leon O. Chua – is currently a Professor of electrical engineering and computer sciences at the University of California, Berkeley. His research interests are in the areas of general *nonlinear* network and system theory. He has been a consultant to various electronic industries in the areas of nonlinear network analysis, modeling, and computer-aided design. He is the author of *Introduction to Nonlinear Network Theory* (McGraw-Hill, 1969), *Cellular Neural Networks: a paradigm for complexity* (World Scientific, 1999), and a co-author of the books *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques* (Prentice-Hall, 1975), *Linear and Nonlinear Circuits* (McGrow-Hill, 1987), and *Practical Numerical Algorithms for Chaotic Systems* (Springer-Verlag, 1989). He has published more than 500 papers in the area of nonlinear networks and systems. He served as the editor of the *IEEE Transactions on Circuits and Systems* from 1973 to 1975 and as the president of the *IEEE Society on Circuits and Systems* in 1976. He is presently the editor of the *International Journal of Bifurcation and Chaos* and a deputy editor of the *International Journal of Circuit Theory and Applications*. His IEEE membership number is 01466135.