

**New spatial-temporal patterns and
the first programmable
on-chip bifurcation test-bed**

I. PETRÁS, T. ROSKA, AND L.O. CHUA

DNS-6-2001

COMPUTER AND AUTOMATION RESEARCH INSTITUTE
BUDAPEST
2001

1	INTRODUCTION.....	3
2	THE PROGRAMMABLE CNN MODEL	4
2.1	SIMULATION.....	4
2.2	PROGRAMMABLE CHIP MEASUREMENTS – THE EXPERIMENTAL TEST BED	4
3	THE EFFECT OF VERTICAL COUPLING	5
3.1	POSITIVE VERTICAL COUPLING ($r > 0$).....	6
3.2	NEGATIVE VERTICAL COUPLING ($r < 0$).....	7
4	THE EFFECT OF THE CENTRAL TEMPLATE ELEMENT	8
4.1	SIMULATION.....	8
4.1.1.	<i>Positive coupling ($r > 0$)</i>	9
4.1.2.	<i>Negative coupling ($r < 0$)</i>	11
4.2	PROGRAMMED CHIP MEASUREMENT.....	14
4.2.1.	<i>Positive coupling ($r > 0$)</i>	14
4.2.2.	<i>Negative coupling ($r < 0$)</i>	17
4.3	VISUAL SIGNATURES.....	20
5	THE EFFECT OF THE CONSTANT INPUT AND INITIAL STATE.....	20
5.1	PERIODIC-CHAOTIC TRANSITION.....	20
5.2	STABLE-PERIODIC TRANSITION.....	22
6	1D CHAOS.....	23
7	ADDITIONAL PROPAGATING PATTERN EXAMPLES	24
7.1	WAVE SHADOW	24
7.2	“FOUR PIXELS” EXAMPLES.....	25
7.2.1.	<i>Bird</i>	26
7.2.2.	<i>Volcano eruption</i>	26
7.2.3.	<i>Flower</i>	26
8	SIMULATION TIME VS. REAL-TIME MEASUREMENTS	26
9	CONCLUSION	27
	ACKNOWLEDGEMENTS	27
	REFERENCES.....	27
	APPENDIX.....	29

1 Introduction

Numerical simulations of spatial-temporal chaotic systems require huge digital computing power, but still that is the usual analysis tool because it offers the advantage of easy experimentation via programming. Till 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 2-3 dimensional array of n -th 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. The *output* (or *state*) of the system is considered as a two dimensional image where 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.

The qualitative theory of nonsymmetric feedback (\mathcal{A}) templates had been first exposed in [16]. Later, several papers have studied the operation of the CNN with nonsymmetric 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.

In this paper, besides (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 addition, we show (iii) how to use these exotic patterns for morphological detection. In the following we present an experimental analysis of a simple antisymmetric template in that case when we add only one extra coupling bellow the central element. We will introduce the basic template class and show how the behavior of the CNN changes from stable to chaotic states at different values of that coupling. Figure 1 shows two basic pattern classes that are generated by 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”.

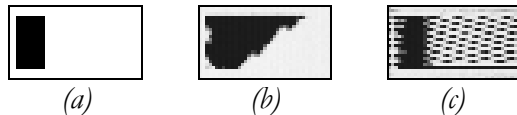


Figure 1: Typical pattern classes. (a) The input and initial state, (b) snapshot of the output when the extra coupling is greater than zero, (c) snapshot of the output when the extra coupling is less than zero (*Chip measurements*)

Section 2 describes the CNN *model* of the simulation and of the chip. Section 3 presents the basic *pattern classes*. Section 4 and Section 5 describes the *effect* of the *self-feedback* and of the *input and initial state* respectively. Section 6 presents an example of a *1D CNN* with first order cells that exhibits *chaotic* behavior. Section 7 shows some *additional examples* of propagating pattern classes.

2 The Programmable CNN Model

Throughout this analysis we will only consider CNNs with *space invariant* templates i.e. the same template matrix describes the local couplings for each cell. In the model, (Equation 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 ideal tool for us, as a test bed.

2.1 Simulation

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

$$\dot{x}_{ij}(t) = -g(x_{ij}(t)) + \sum_{kl \in S_r} A_{kl} x_{ij,kl}(t) + \sum_{kl \in S_r} B_{kl} u_{kl} + \xi \quad (1)$$

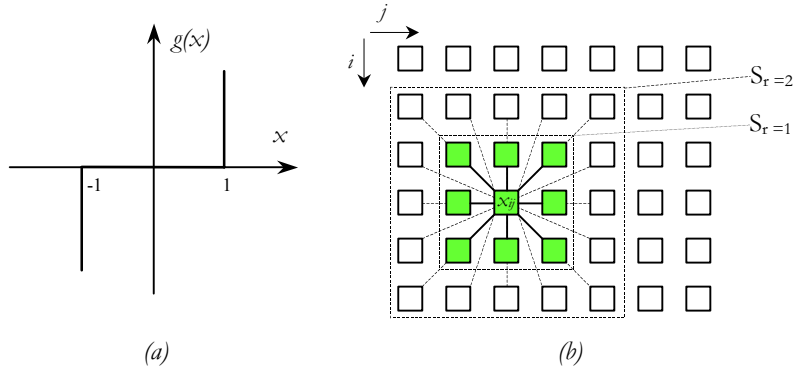


Figure 2: (a) Hard nonlinearity $g(\cdot)$, (b) The interpretation of S_r in the CNN array for $r=1$ (3×3).

We use a so called full range model [8]. In Equation 1, x_{ij} denotes the state, A_{kl} is the feedback template matrix, B_{kl} is the control template matrix, it describes the effect of the constant input u_{kl} , and ξ is the offset or bias. The integration method throughout the simulation was an *implicit Euler method*. In all cases the boundary cells are constant, except if explicitly noted differently.

2.2 Programmable chip measurements – the experimental test bed

The chip experiments had been made on the new ACE4k test bed. The ACE4k chip is an analogic cellular microprocessor [18]. The model of the CNN-instruction is described by Equation 2:

$$\dot{x}_{ij}(t) = -g'(x_{ij}(t)) + \sum_{kl \in S_r} A_{kl} x_{ij,kl}(t) + \sum_{kl \in S_r} B_{kl} u_{kl} + \xi \quad (2)$$

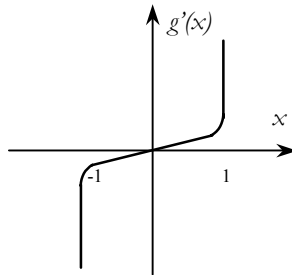


Figure 3: "Less hard" nonlinearity $g'(\cdot)$

On the chip, the ideal nonlinearity (Figure 2a) is approximated by a “less hard” nonlinearity (Figure 3).

When the state variable $x_j(t)$ reaches the saturation limits $(+1, -1)$, the $g'(x)$ term works against the increase or decrease and keeps the value of the state within the limits.

3 The effect of vertical coupling

In the following we will show how extra coupling added to an uncoupled template changes the behavior of the system. Let us consider the following one-dimensional “CCD-like” template of which solution is periodic (see Figure 4).

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

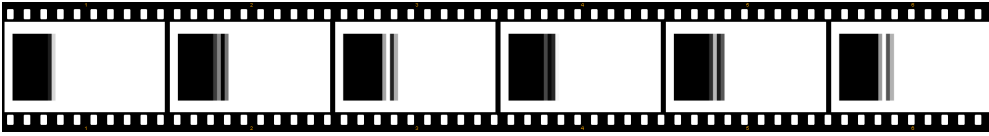


Figure 4: The time evolution for Template 1 and snapshots of the state. Observe that the propagation decays *spatially* after a few pixels, but the oscillation remains. The initial state is the same as the input. (Simulated results, size: 41x23)

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. 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 key template element r . (See Figure 1 and Table 1.)

Template 2: <i>general form</i> $s \neq q$	$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ s & p & q \\ 0 & r & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{z} = \xi$
Template 3: <i>antisymmetric</i> $s = -q$	$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ s & p & -s \\ 0 & r & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{z} = \xi$

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

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

Different types of propagation occur according to the sign of the coupling r . Local oscillators operate in each row due to the antisymmetric property ($s = -q$).

3.1 Positive vertical coupling ($r > 0$)

If the value 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 not propagates). Typical snapshot of the pattern is shown in Figure 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 Figure 8). Figure 6 shows the result of the chip measurement and its corresponding template, Template 5. The exact values of the chip templates are slightly different from the template of the simulation, however the phenomenon can be reproduced quite well.

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

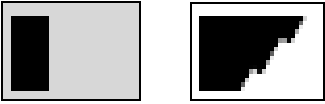


Figure 5: The input & initial state and snapshot of the output pattern if the coupling is positive. (Simulated result)

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

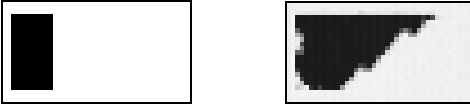


Figure 6: The input & initial state and snapshot of the output pattern if the coupling is positive. (Chip measurement)

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 of them (see Figure 7). Generally, there is no oscillation inside of the structure, only along its border. These local oscillations together form the main pattern which looks like the cross-section of waves on the surface of the water (see Figure 8).

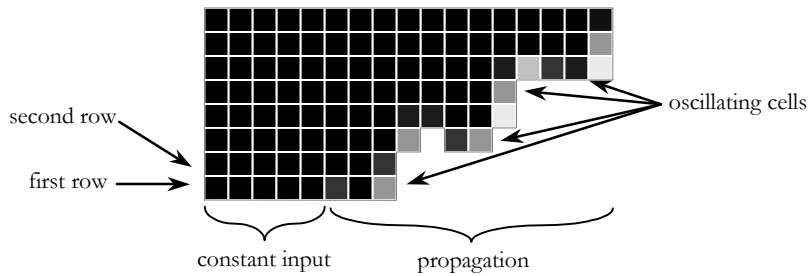


Figure 7: Structure of the pattern when p is greater than zero.

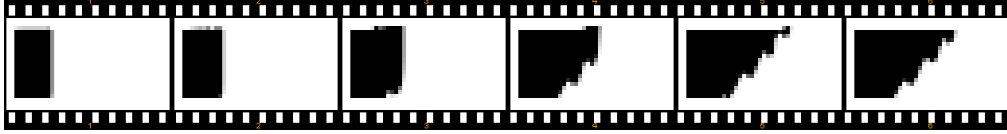


Figure 8: Time evolution of the pattern when p is positive. Snapshots of the state (simulated results, size: 41×23). The last snapshot shows the largest spatial extent of the generated pattern. There is no global propagation after this moment, only the ruffles travel along the right border of the pattern.

3.2 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 Figure 11). The spatial extent of the propagation to the right direction changes as we change the parameters but the main characteristics remain the same.

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



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

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

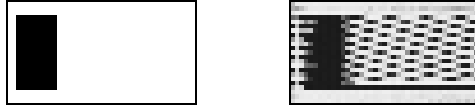


Figure 10: The input & initial state and snapshot of the output pattern if the vertical coupling is negative. (Chip measurement)

Let us define the bottommost row of the input pattern (black pixels) as the “first” row (Figure 12). The template produces a straight line (a shadow) for this row over a large domain of parameters 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 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.

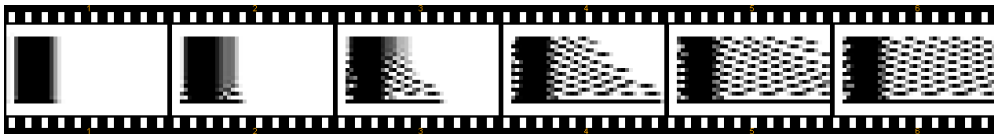


Figure 11: Time evolution of the pattern when r is negative. Snapshots of the state (simulated results, size: 41×23).

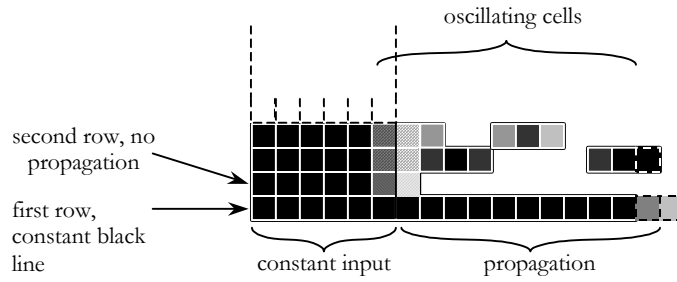


Figure 12: The structure of the propagating pattern.

4 The effect of the central template element

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

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

4.1 Simulation

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

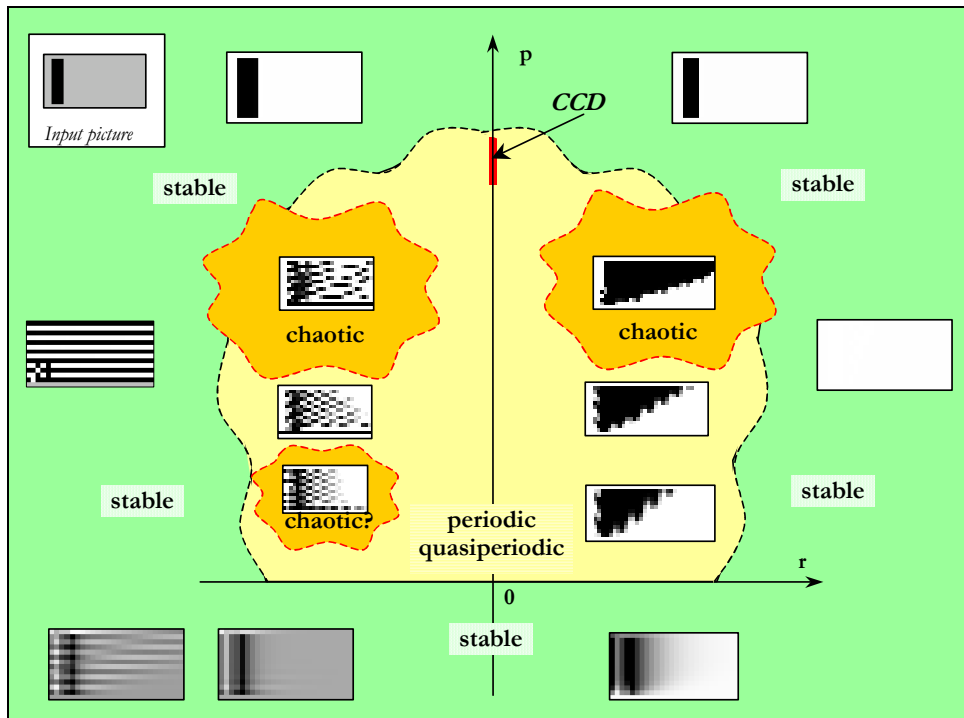


Figure 13: Partitioning of the r - p parameter space. The input & initial state is shown in the upper left corner. It is a three-pixel wide bar. The pictures in the different regions show few typical snapshots of outputs belonging to that region.

Stable region

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 out-

put 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.

Periodic region

The patterns propagate periodically and they have the form of *solid wave-like* and *texture*, as it is described previously in Section 3.

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. The reason for this can be that the constant black line (the first row) becomes oscillating (See Figure 13).

4.1.1. Positive coupling ($r > 0$)

This section contains results of simulations when r is greater than zero. Observe the transition from the simpler to the more complex dynamics. Figures 15-20 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 . ($p = [0.3 .. 0.87]$, $r = 0.3$)

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

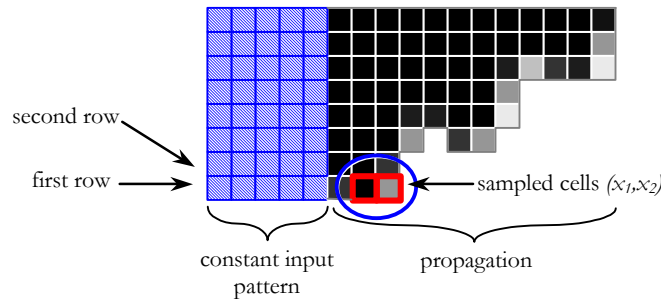


Figure 14: In the first row, to the right from the edge of the black part of the constant input (denoted by blue striped boxes) the second and third cells were sampled as shown in the figure.

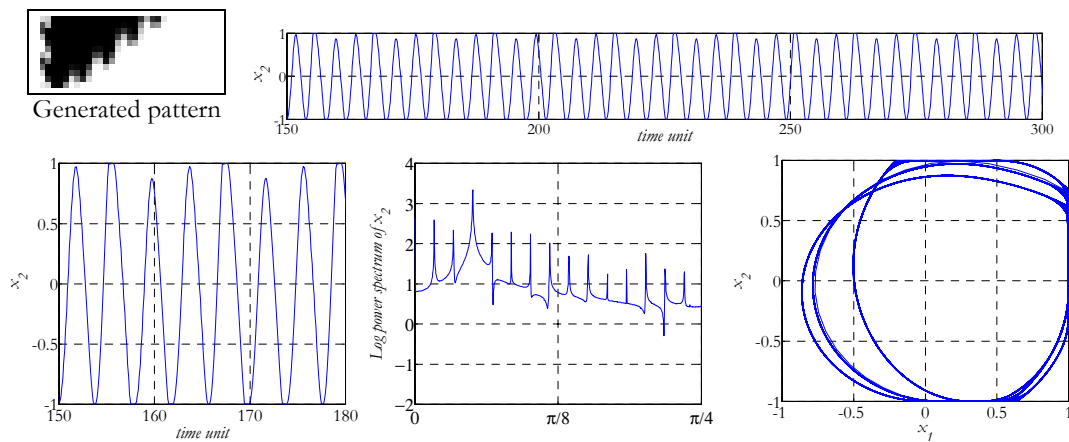


Figure 15: Snapshot of the output, the time evolution of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row ($p=0.3$).

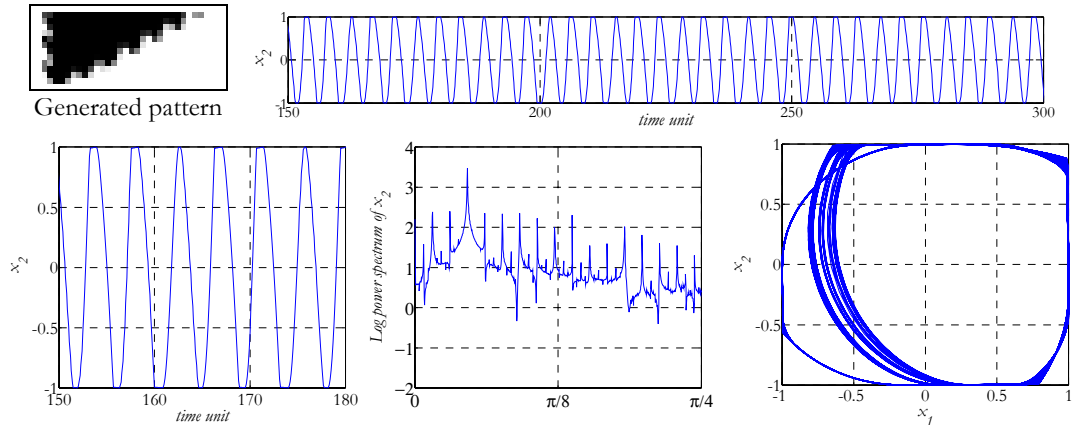


Figure 16: Snapshot of the output, the time evolution of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row ($p=0.5$).

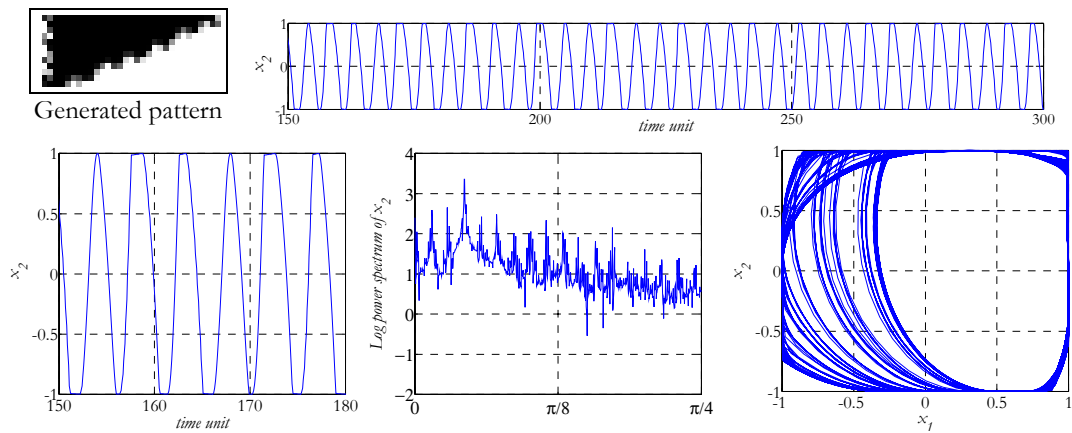


Figure 17: Snapshot of the output, the time evolution of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row ($p=0.6$).

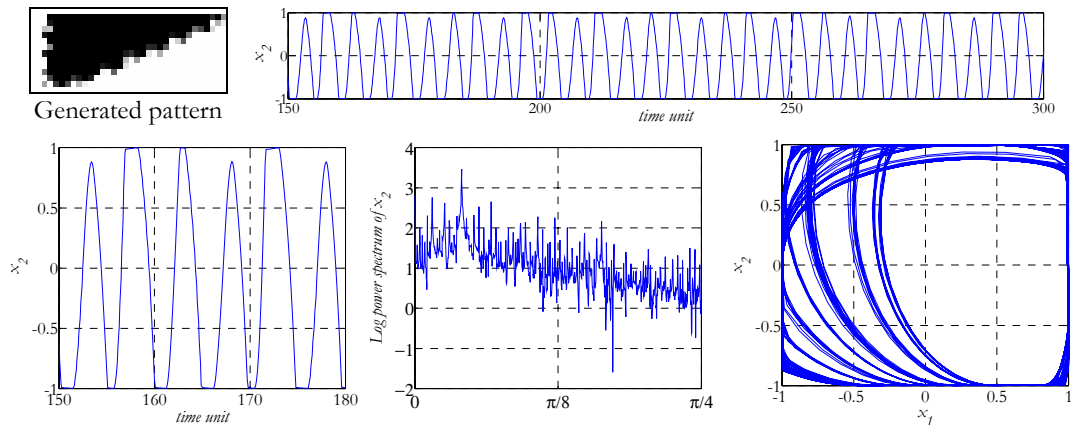


Figure 18: Snapshot of the output, the time evolution of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row ($p=0.7$).

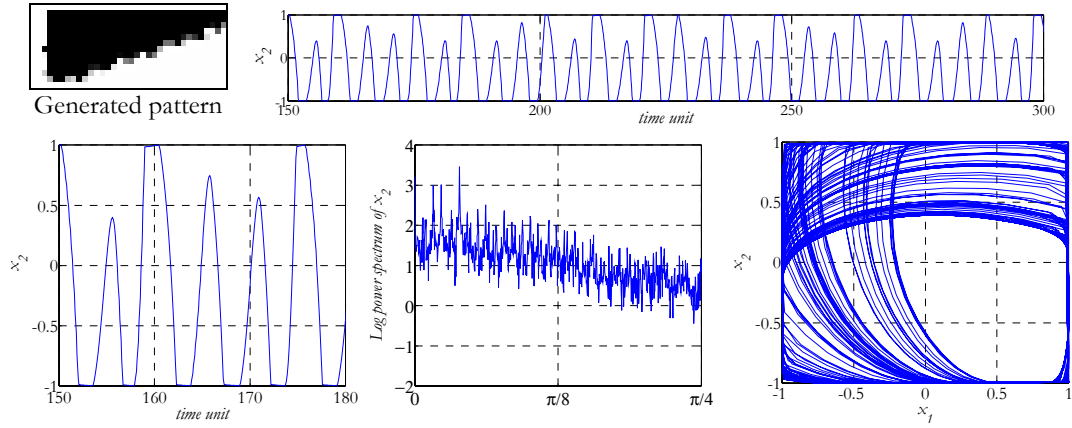


Figure 19: Snapshot of the output, the time evolution of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row ($p=0.8$).

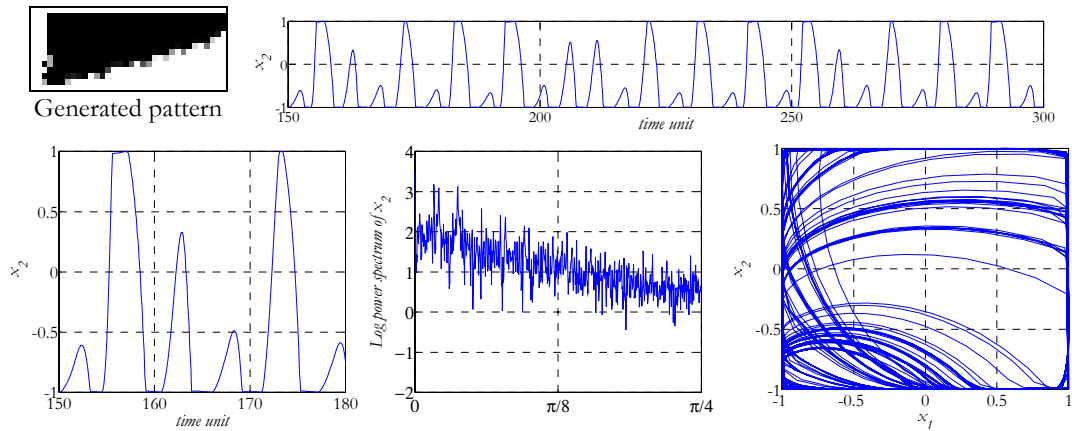


Figure 20: Snapshot of the output, the time evolution of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row ($p=0.87$).

4.1.2. Negative coupling ($r < 0$)

When r is less than zero, texture like oscillating pattern is formed. In Figures 22 - 28 observe the transition from the simpler to the more complex dynamics. The figure captions contain the actual value of parameter p . ($p \in [0.1 .. 0.7]$, $r = -0.3$) See Template 10.

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

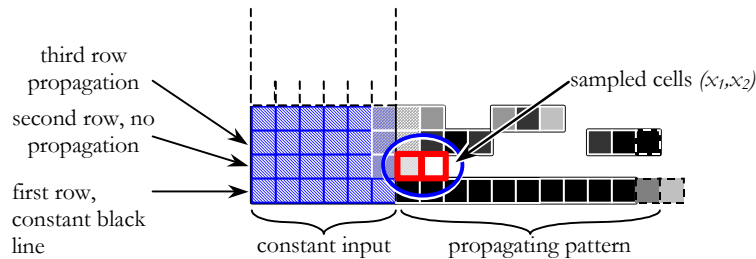


Figure 21: In the second row, right from the edge of the black part of the constant input (denoted by blue striped boxes) the first and second cells were sampled as shown in the figure.

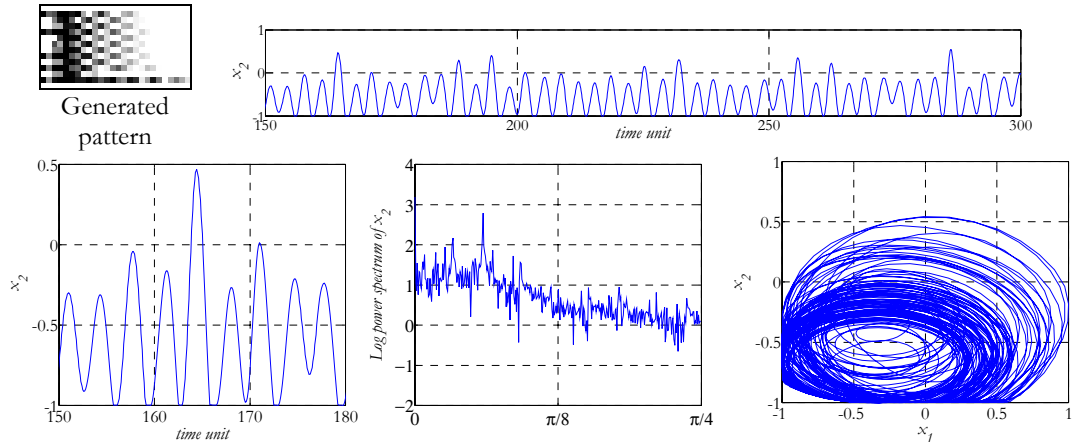


Figure 22: Snapshot of the output, the time evolution of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row ($\rho=0.1$).

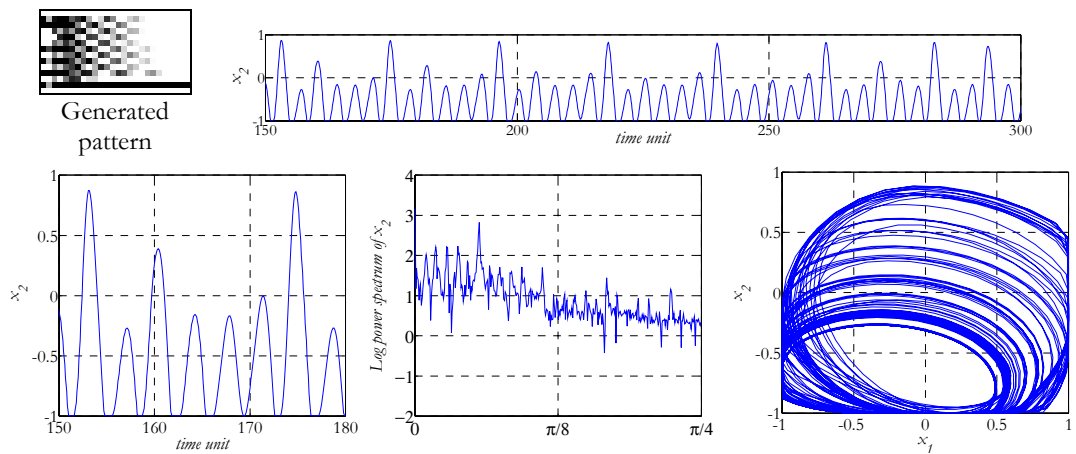


Figure 23: Snapshot of the output, the time evolution of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row ($\rho=0.2$).

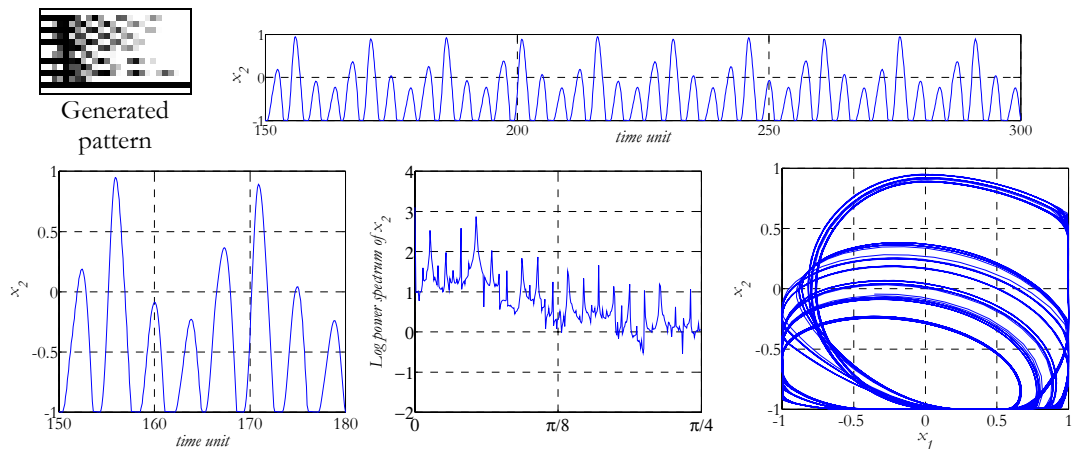


Figure 24: Snapshot of the output, the time evolution of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row ($\rho=0.3$).

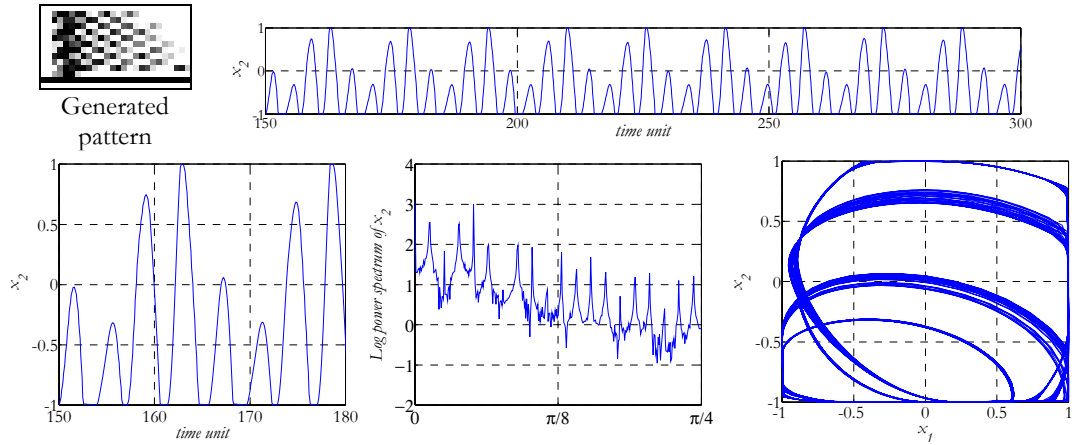


Figure 25: Snapshot of the output, the time evolution of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row ($\rho=0.4$).

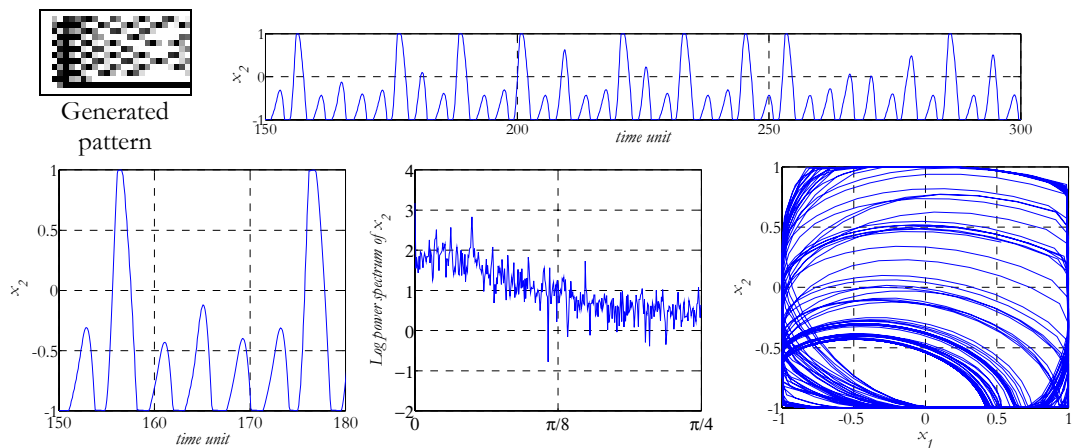


Figure 26: Snapshot of the output, the time evolution of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row ($\rho=0.5$).

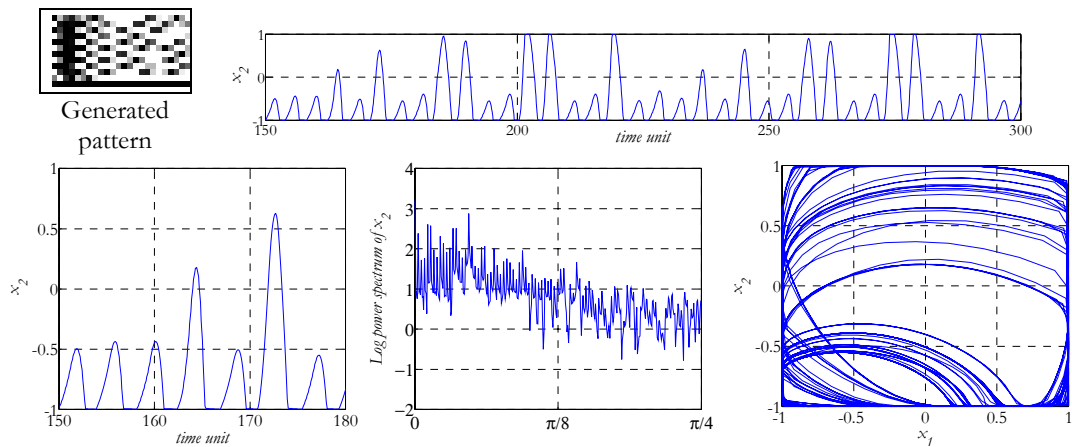


Figure 27: Snapshot of the output, the time evolution of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row ($\rho=0.6$).

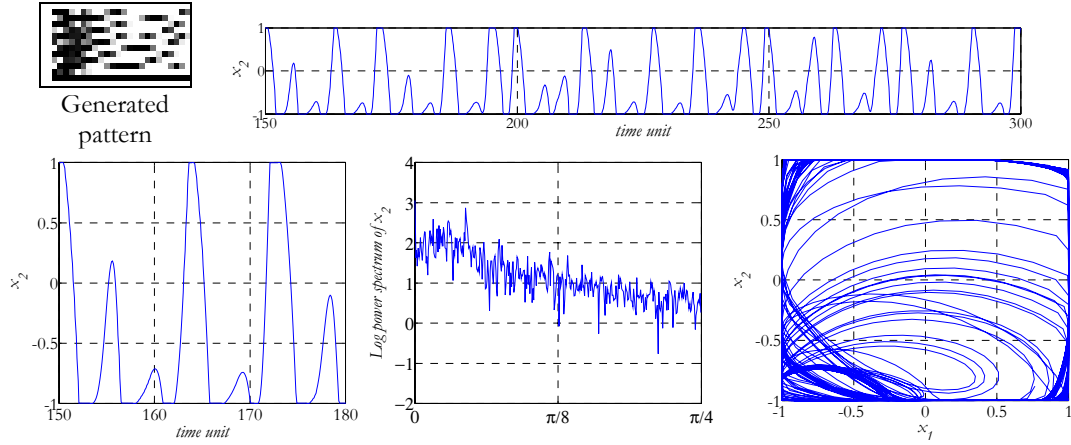


Figure 28: Snapshot of the output, the time evolution of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row ($p=0.7$).

4.2 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.

4.2.1. Positive coupling ($r>0$)

Cells from the first row were sampled (Figure 29). 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 29 - 37 show the measured time series, power spectrum and the trajectory of the two sampled cells. The sampling position is shown in Figure 29. The generator template for the figures is Template 11.

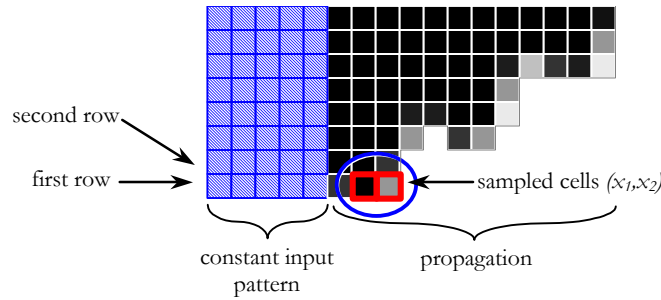


Figure 29: In the first row, to the right from the edge of the black part of the constant input (denoted by blue striped boxes) the second and third cells were sampled as shown in the figure.

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

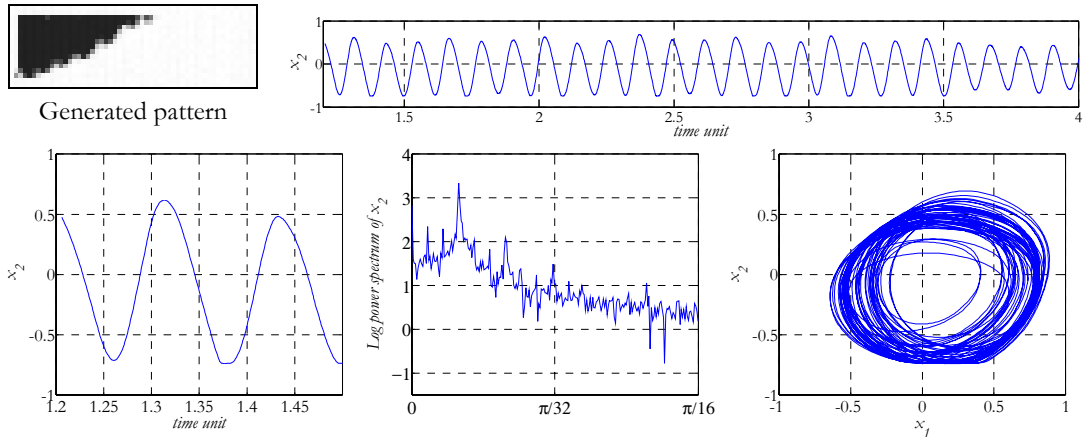


Figure 30: Snapshot of the output, the time series of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row ($p=0.45$).

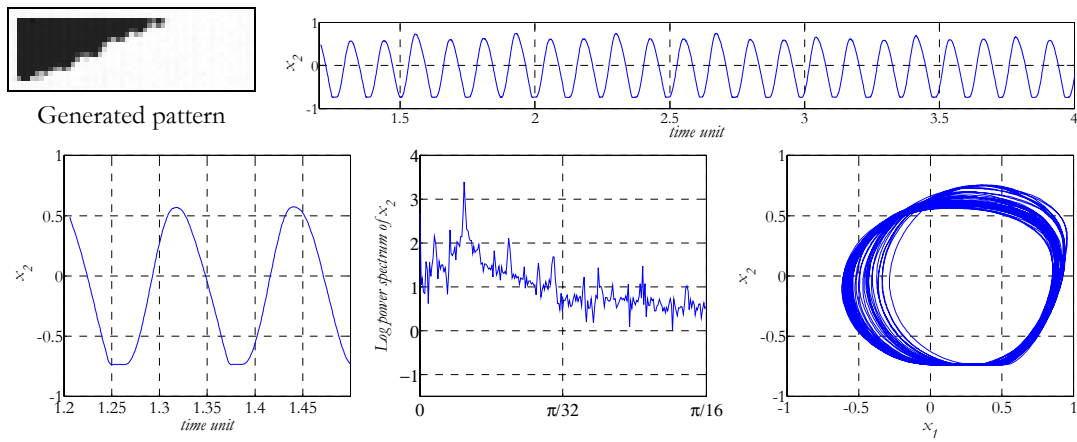


Figure 31: Snapshot of the output, the time series of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row ($p=0.49$).

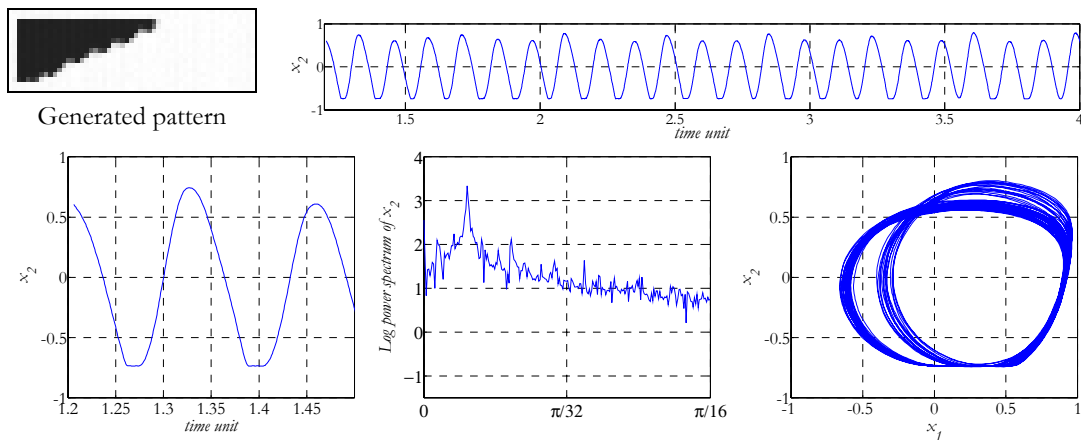


Figure 32: Snapshot of the output, the time series of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row ($p=0.51$).

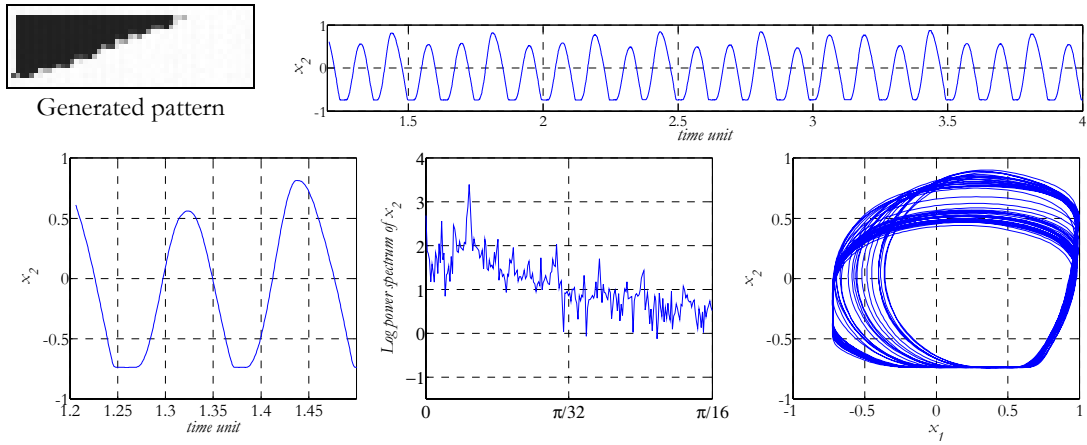


Figure 33: Snapshot of the output, the time series of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row ($p=0.61$).

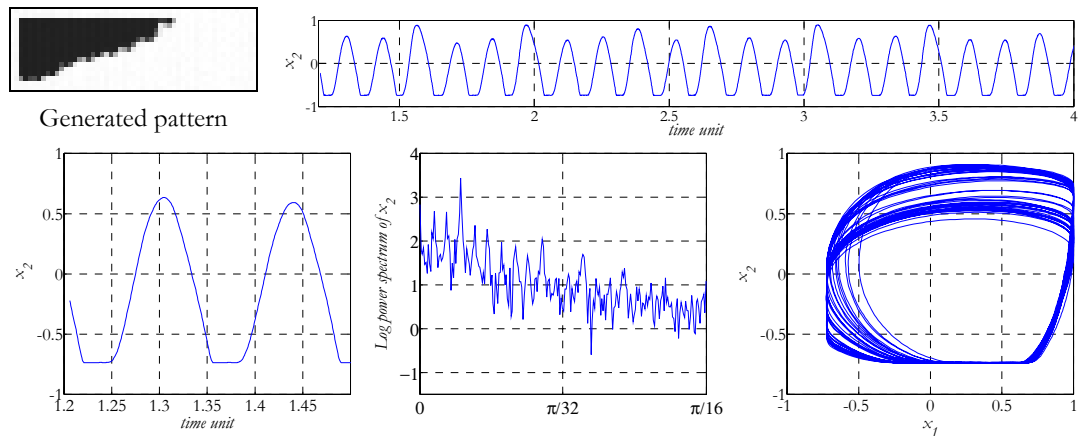


Figure 34: Snapshot of the output, the time series of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row ($p=0.69$).

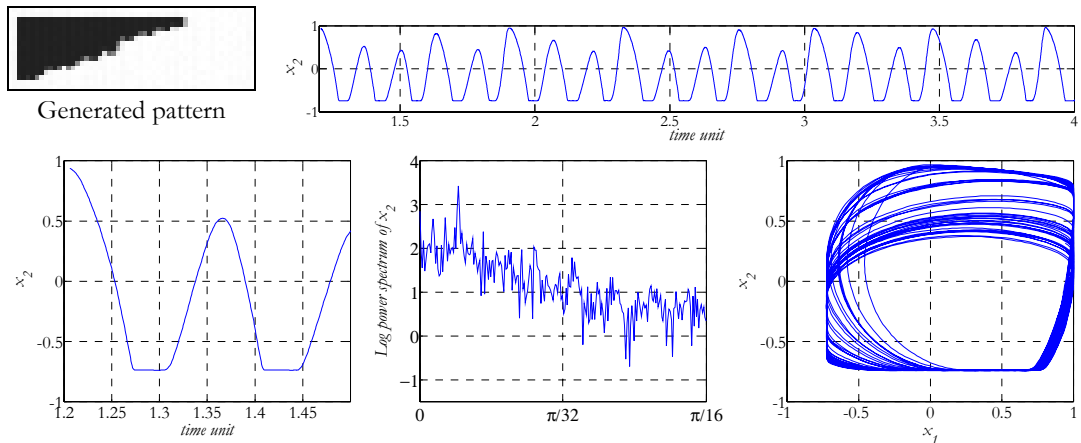


Figure 35: Snapshot of the output, the time series of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row ($p=0.75$).

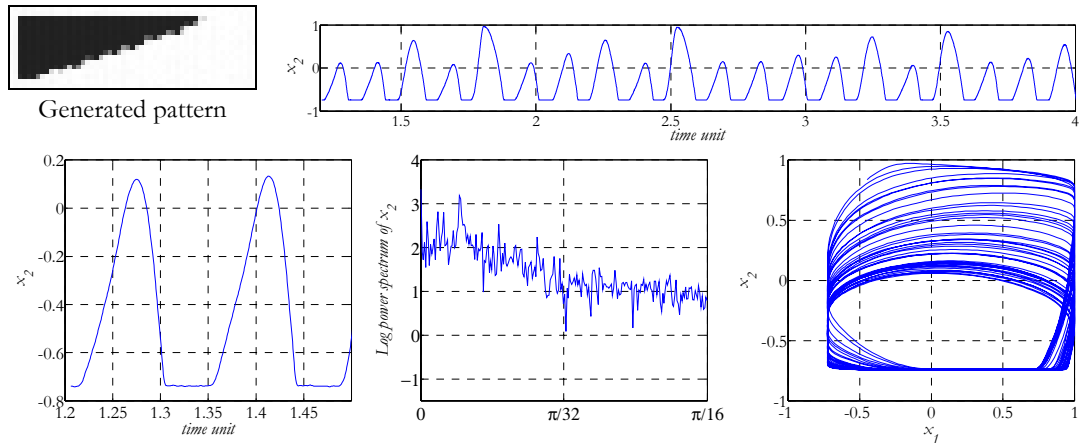


Figure 36: Snapshot of the output, the time series of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row ($p=0.79$).

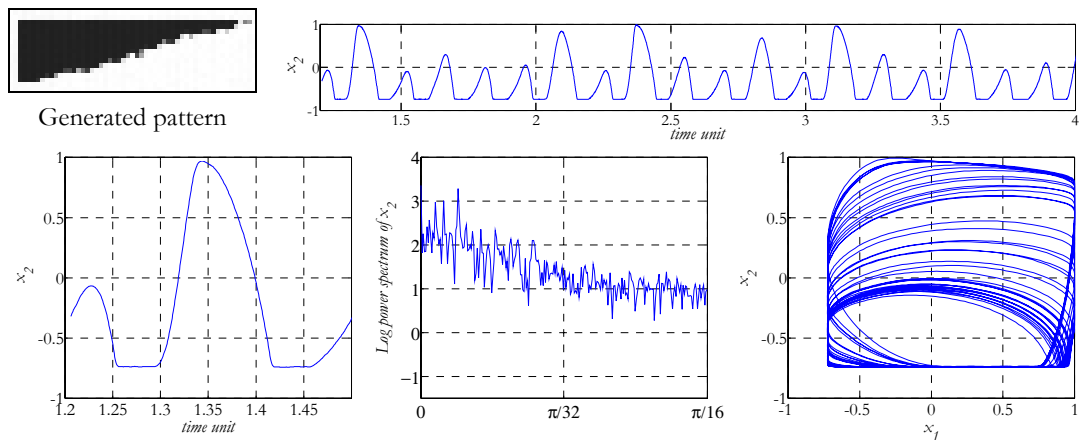


Figure 37: Snapshot of the output, the time series of one cell from the first row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the first row ($p=0.85$).

4.2.2. Negative coupling ($r < 0$)

In the negative coupling case we experiment similar phenomenon to that of the simulation.

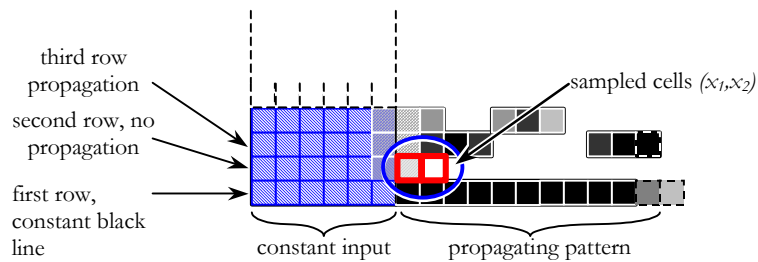


Figure 38: In the second row, right from the edge of the black part of the constant input (denoted by blue striped boxes) the first and second cells were sampled as shown in the figure.

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

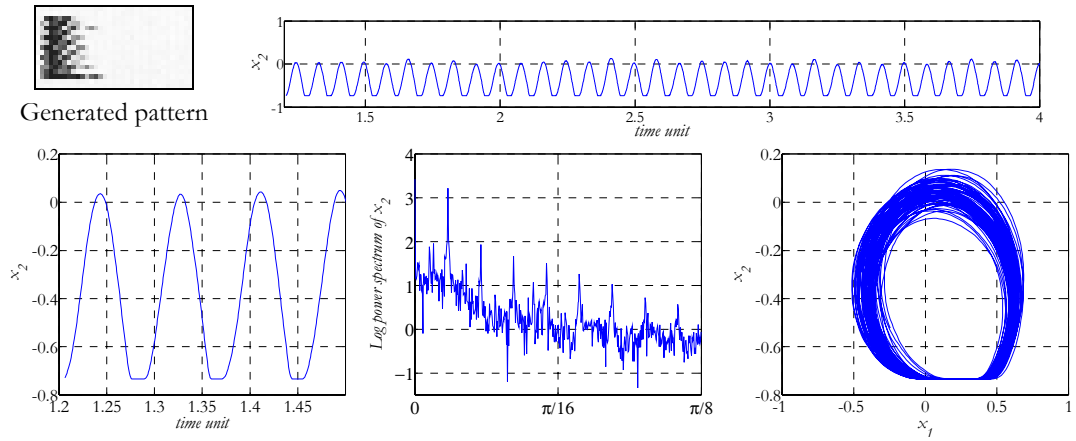


Figure 39: Snapshot of the output, the time series of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row ($p=0.3$).

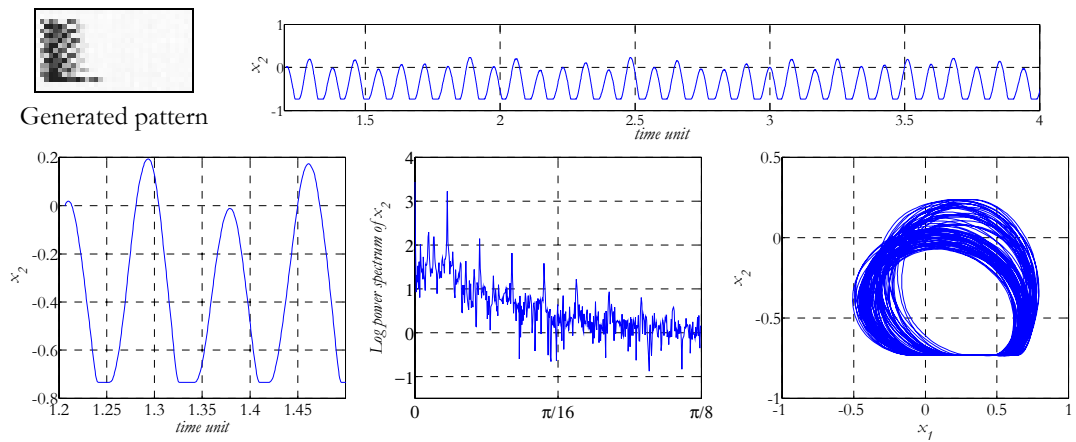


Figure 40: Snapshot of the output, the time series of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row ($p=0.34$).

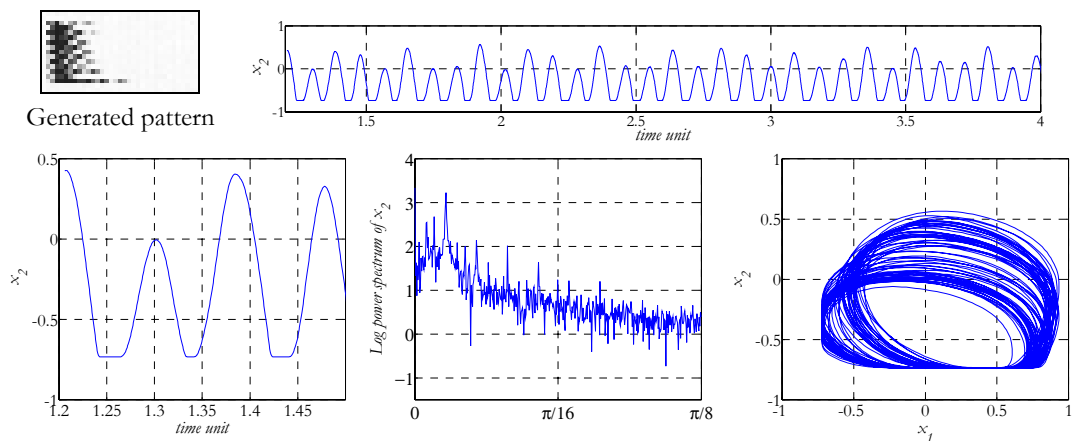


Figure 41: Snapshot of the output, the time series of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row ($p=0.42$).

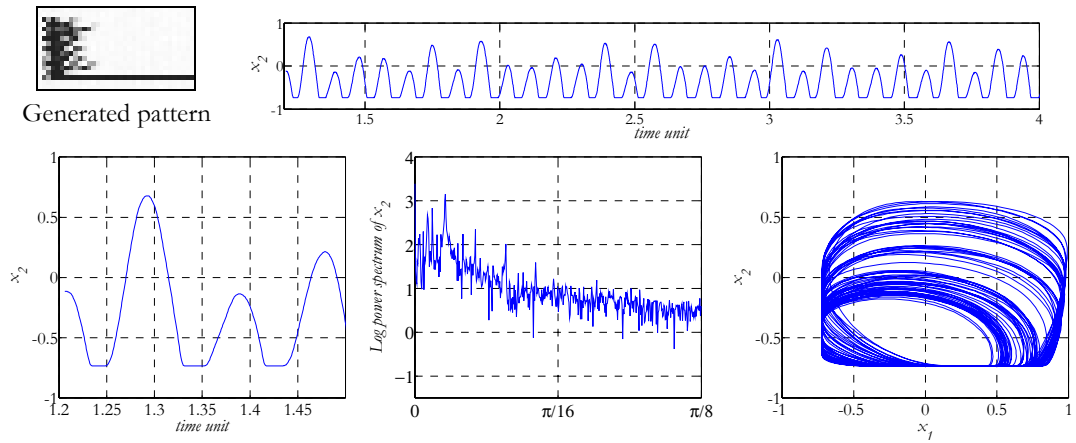


Figure 42: Snapshot of the output, the time series of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row ($p=0.5$).

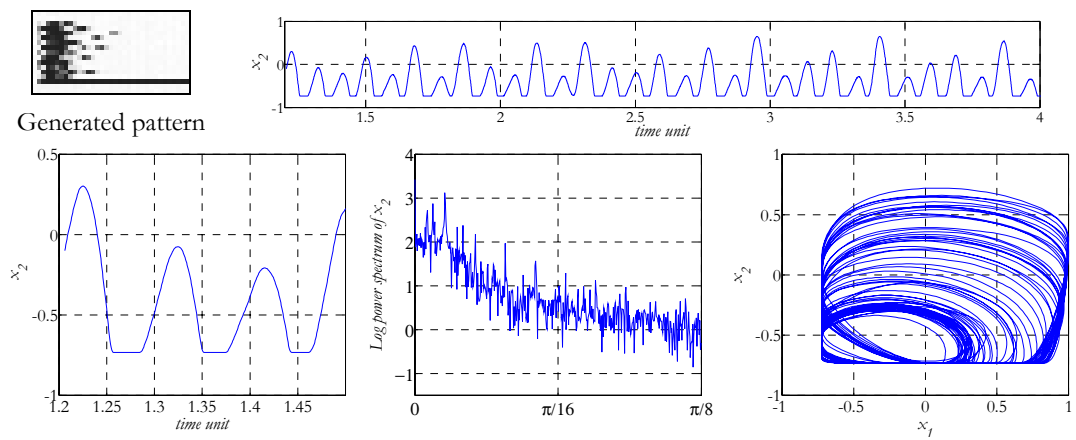


Figure 43: Snapshot of the output, the time series of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row ($p=0.56$).

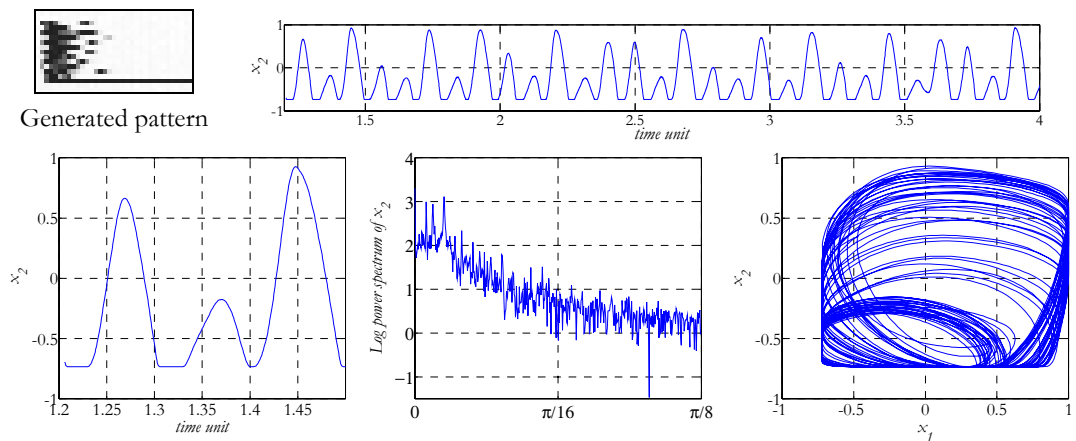


Figure 44: Snapshot of the output, the time series of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row ($p=0.6$).

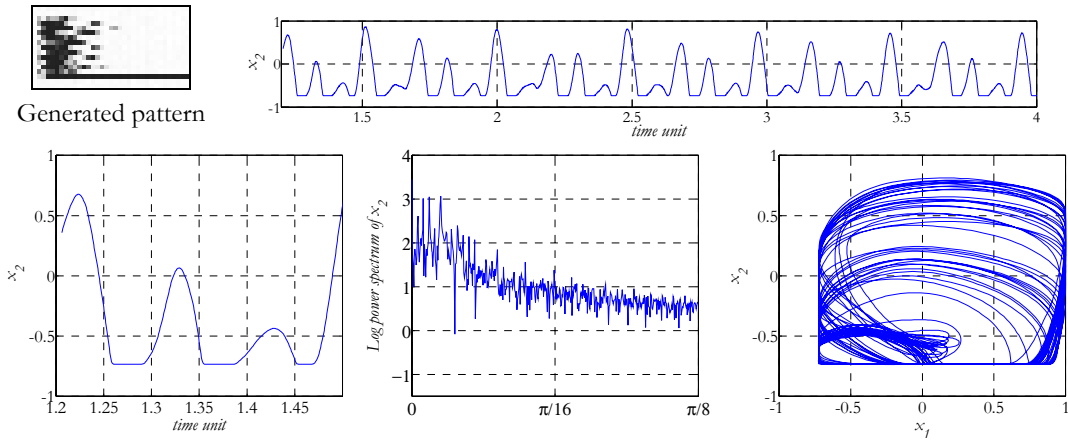


Figure 45: Snapshot of the output, the time series of one cell from the second row, logarithm of power spectrum and the 2D trajectory of the same cell and the neighbor cell from the second row ($p=0.68$).

4.3 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.

<i>Stable</i>		<i>Chaotic</i>	
	$r > 0, p < 0$		$r > 0, p \gg 0$
	$r < 0, p < 0$		$r < 0, p \gg 0$
<i>Periodic</i>			
	$r > 0, p > 0$		
	$r < 0, p > 0$		

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

5 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 are presented related to this aspect.

5.1 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 vertical bar and the other input is a three pixel wide vertical bar. The input and the initial state are the same.

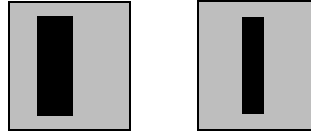


Figure 46: The five- and the three-pixel-wide inputs.

periodic	simulation		
	chip measurement		
chaotic	simulation		
	chip measurement		

Table 3 Different dynamical behaviors with the same template but with different input and initial state.

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 horizontally at least one pixel wide column of saturated stable cells. Thus, the oscillators along the left and right border of the bar are uncoupled. The cells were sampled in the second row.

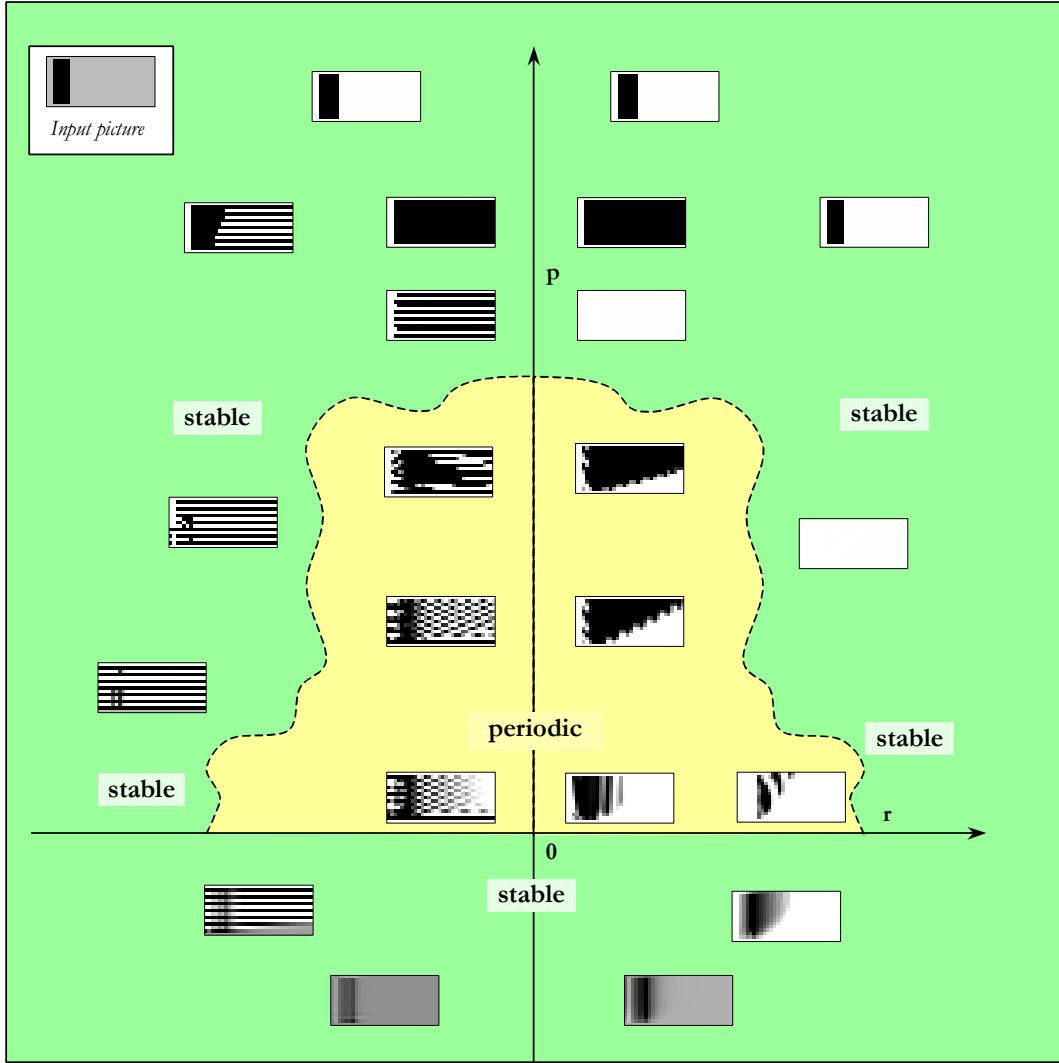


Figure 47: Partitioning of the r - p parameter space when the input is a five pixels wide bar. The input picture is shown in the upper left corner. The pictures in the different regions show few typical snapshot of the output patterns belonging to that region. With this input no chaotic behavior has been found up to now.

Template 13: <i>Template for the simulation</i>	Template 14: <i>Template for the chip measurement</i>
$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1.1 & 0.7 & -1.1 \\ 0 & -0.3 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \varepsilon = 0.1$	$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0.9 & 0.5 & -0.9 \\ 0 & -0.5 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \varepsilon = 2.1 \quad (3)$

Figure 47 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. Necessary condition for this seems to be that the stable saturated cells along the center of the bar becomes unstable.

5.2 Stable-periodic transition

The following table 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. The second and third input however produces a periodic pattern. The boundary condition is periodic (torus-like left-right and bottom-up connections, this can be programmed, as well, on the chip).

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

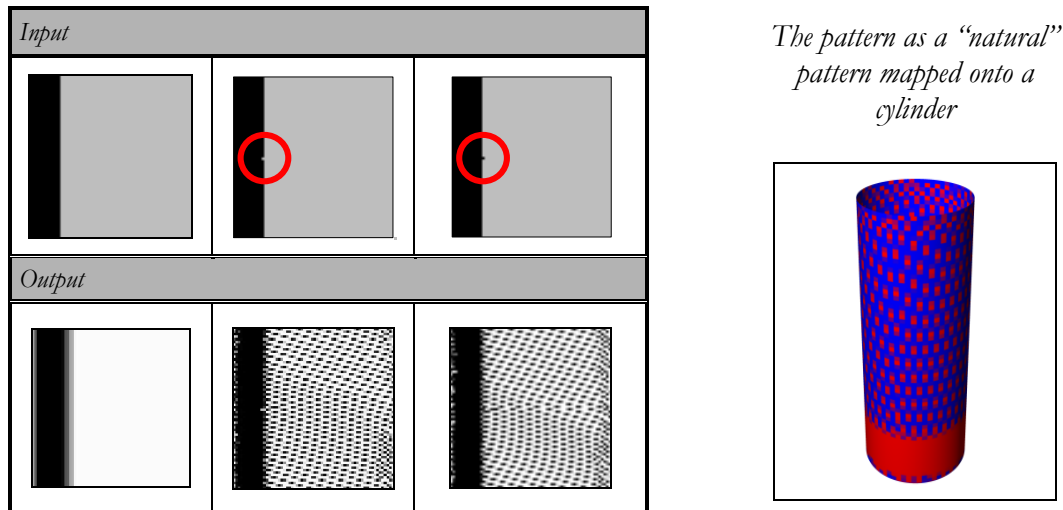


Table 4: Effect of one pixel perturbation. The first row contains three different inputs. The first one is a vertical bar. The second one is the same except one pixel: a pixel is *clipped off* from the middle of the right border of the pattern. In the third one a pixel is *added* to the middle of the 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.

6 1D chaos

Based on the measurements and the simulations we can construct the possibly simplest template that can exhibit chaos.

The general structure of the templates presented in this paper shows that a CNN cell does not depend on the cells above it. In the cases above (Templates 13,14) we measured and simulated chaotic signals in that 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 Figure 48 and Figure 49). All boundary cells were set to zero.

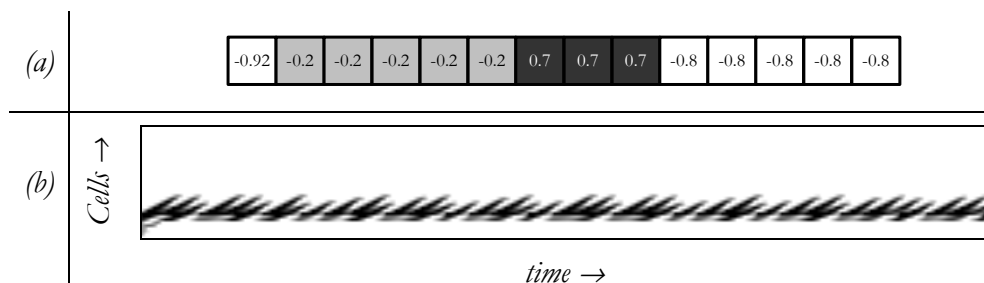


Figure 48: (a) Input for the 1D chaotic CNN. (b) The generated 1d pattern as a function of time.

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

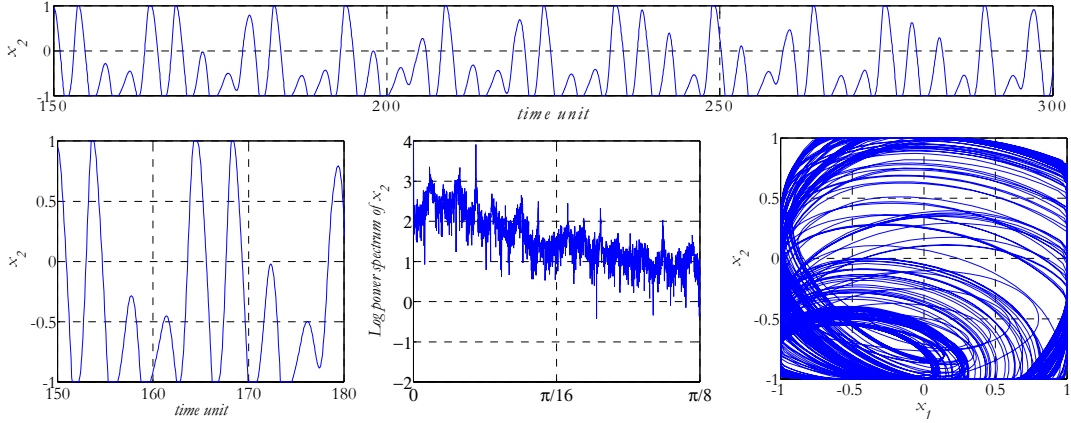


Figure 49: The time evolution, the power spectra of one cell and the trajectory of two neighboring cells.

7 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 non-zero couplings are added. The corresponding templates matrices are shown below of the figures.

7.1 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 3.

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

Table 5: Snapshots of the simulation and of the chip measurements of the complex wave template.

Template 17: <i>Template for the simulation</i>	Template 18: <i>Template for the chip</i>
$\mathbf{A} = \begin{bmatrix} 0 & 0 & -0.6 \\ 1 & P & -1 \\ 0.8 & 0 & -0.1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \zeta = 0.9$	$\mathbf{A} = \begin{bmatrix} 0 & 0 & -0.6 \\ 1 & P & -1 \\ 0.8 & 0 & -0.2 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \zeta = 4.2 \quad (4)$

With a different template setting a characteristic trajectory is measured on the chip.

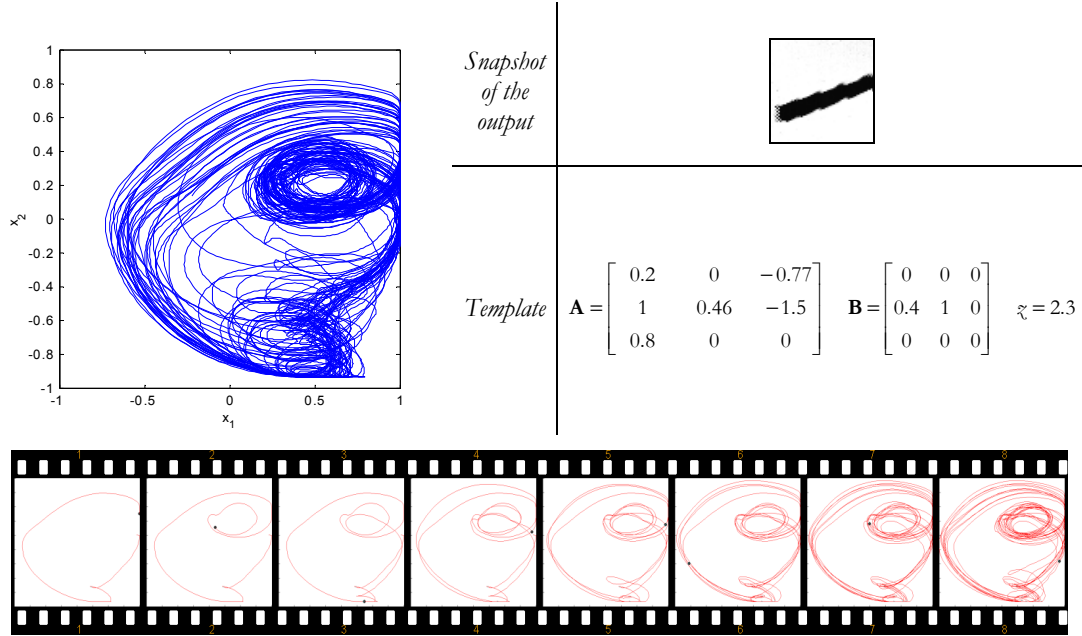


Figure 50: Snapshots of trajectory of two out of 4096 state variables of the CNN array (*chip measurement*).

7.2 “Four pixels” examples

The following patterns are simulated using the standard CNN model [4]:

$$\begin{aligned} \dot{x}_{ij}(t) &= -x_{ij}(t) + \sum_{kl \in S_r} A_{kl} y_{ij,kl}(t) + \sum_{kl \in S_r} B_{kl} u_{kl} + \zeta \\ y_{ij}(t) &= f(x_{ij}(t)) \end{aligned} \quad (5)$$

The nonlinearity $f(x)$ is shown in Figure 51. The next examples show different patterns, all of which are generated from the same input. The “seed” of the patterns is a four-pixel wide horizontal line section. The initial state is the same as the input.

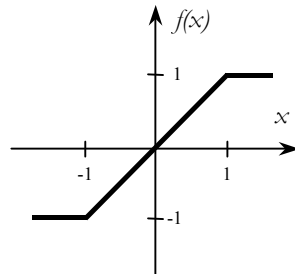


Figure 51: The nonlinearity $f(x)$

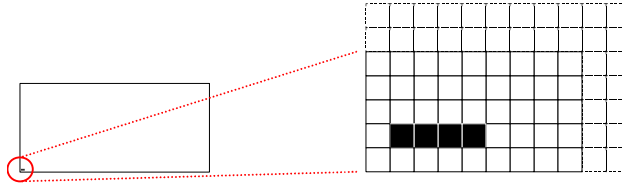


Figure 52: Input and initial state for the propagating patterns.

7.2.1. Bird

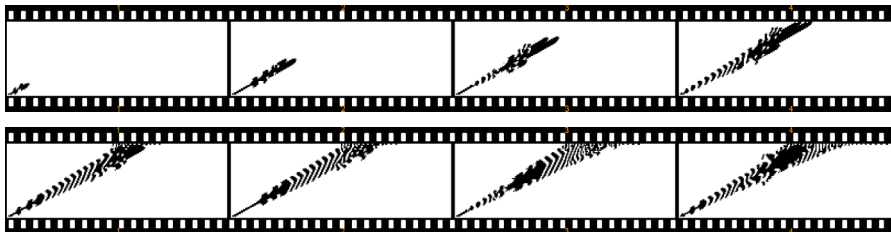
This example shows a propagating pattern which makes the impression of distant flying birds.



$$\mathbf{A} = \begin{bmatrix} 0 & 0 & -0.8 \\ 1.1 & 2 & -1 \\ 0.7 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.09 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \zeta = 0.9$$

7.2.2. Volcano eruption

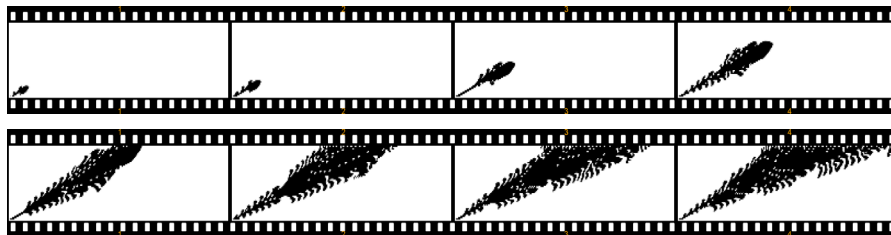
This pattern “erupts” periodically during the propagation. It looks like some fluid coming out of a pipe.



$$\mathbf{A} = \begin{bmatrix} 0 & 0 & -0.7 \\ 0.7 & 2 & -0.7 \\ 0.7 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.9 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \zeta = 1$$

7.2.3. Flower

This pattern grows like a plant.



$$\mathbf{A} = \begin{bmatrix} 0 & 0 & -1 \\ 1.1 & 2.4 & -1 \\ 1.1 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \zeta = 1.4$$

8 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 6 shows the comparison of a typical simulation and the real time chip measurement.

<i>Simulation (500MHz PC)</i>	<i>Chip measurement (with sampling)</i>	<i>Chip measurement (without sampling)</i>
<i>25 min = 1500 sec</i>	<i>200 ms</i>	<i>5 ms</i>

Table 6: Comparison of simulation and chip measurement time.

9 Conclusion

Various propagating patterns represent a new and interesting spatial-temporal pattern class. Long and sophisticated analysis and digital computer simulation is 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 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 more the complex dynamics. It would also lead to a new class of analogic spatial-temporal algorithms for the dynamic detection of exotic and complex events.

Acknowledgements

This research was supported by the Computer and Automation Research Institute of the Hungarian Academy of Sciences (SZTAKI), Office of Naval Research (ONR) Grant No.: N00014-00 1 0429 and the Hungarian National Foundation (OTKA) Grant No.: T O26 555.

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. on 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”, *Proceedings of IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, (CNNA'2000), pp. 201-206, Catania, 0-7803-6344-2, 2000

- [4] L.O. Chua and L. Yang, "Cellular Neural Networks: Theory", *IEEE Trans. on Circuits and Systems*, Vol.35. pp. 1257-1272, 1988.
- [5] L.O. Chua and L. Yang, "Cellular neural networks: Applications", *IEEE Trans. on Circuits and Systems*, Vol.35. pp. 1273-1290, 1988.
- [6] L.O. Chua and T. Roska, "The CNN paradigm", *IEEE Trans. on 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. on Circuits and Systems II: Analog and Digital Signal Processing* Vol. 40, No. 3, pp. 163-173, 1993.
- [8] A. Rodríguez-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. on 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. on 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. Crouse, 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. on 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. on 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. on 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 = 0;
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;
board_p4: 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;
```