Invited Paper

Spatial-temporal event detection via frameless cellular wave computing - a review

*András Horváth^{1a)}, Miklós Koller¹,
Attila Stubendek¹, and Tamás Roska¹*

¹ *Faculty of Information Technology and Bionics
Peter Pazmany Catholic University, Budapest, Hungary*

^{a)} *horvath.andras@itk.ppke.hu*

Received January 16, 2014; Published July 1, 2014

Abstract: The great majority of currently used computational models and devices are using discrete computation. Discrete in time, in value, in parameters. In nature, on the contrary, like in the mammalian retina, difficult tasks are solved with simplicity, low power and elegance, in a wavelike manner, without discretization. Spatial-temporal event detection is a prototype problem in many sophisticated problems of recognition, identification, associative memory, machine-vision, multimodal sensory systems, navigation, and control. In this paper we introduce how continuous spatial-temporal dynamics can be used as algorithmic tools for computation and show a few examples about their usage in practice. These techniques introduce a new kind of thinking also in the field on nonlinear dynamical systems, in general.

Key Words: frameless detection, cellular nonlinear networks, continuous computation

1. Introduction

The first modern computational models, containing continuous elements were introduced in the 1930s by Alan Turing. He has described the operation of standard digital computers, and involve looking at real numbers in terms of digits, and using discrete processes to generate these digits [1]. Such models inevitably handle computable reals only and can never do computations beyond those possible in ordinary discrete systems. This evolution can be continuous in values, but also continuous in time and in parameters. Several times since the 1940s it has been suggested that models of computation should be closer to traditional continuous mathematics, and should look at real numbers as a whole, not in terms of their digit or other representations. To set up any kind of continuous evolution, the most obvious approach is to use traditional mathematical ideas of calculus and differential equations. This leads to models in which possible computations are assumed, say, to correspond to combinations of differential equations - as in Claude Shannon's 1941 general-purpose analog computer. And if one assumes - as is usually implicitly done in traditional mathematics - that any solutions that exist to these equations can somehow always be found then at least in principle this allows computations impossible for discrete systems to be done [2].

As an example to analog computation, let us introduce a system, where the interaction of three different dynamics realizes the computation itself. This example will be described in details in Section 3.2, here we would like only to emphasize the joint evolution of the different continuous flows. Let us take a cellular array of computing elements (analog processors), where every computing element has an optical input from the environment. In addition to this, there are activation light sources located next to every light-sensing element. These light sources are responsible for the lighting of the environment (or the lighting of the measured object), while the optical sensors measure the reflected light from the environment. In this way we have a locally connected array where in every cell we have a processor with optical input, and an activating light source. The dynamics of the processor array can be characterized by the continuous state-equation of Cellular Neural Networks as the self-dynamics of this system. We can introduce other dynamics to the system through the light sources, which directly influences the perception of the environment, in this case the input flow. The interaction of these dynamics realizes the main computation itself, which can be quite efficient in some cases (e.g. spatial-temporal event detection).

In Section 2 we introduce nanoscale devices, spin-torque oscillators, where the computation happens via synchronization in continuous time.

In Section 3 we show examples how continuous dynamics can be used for detection of spatial temporal features.

In the additional part of Section 4 we show how other interesting phenomena can be observed by using continuous dynamics, like metastable rotating waves. In this Section we also show, how continuous dynamics can be used for optimization.

2. Computing via synchronization

Standard, CMOS based computational architectures seem to reach their capacities, according to the sustainability of the Moore's law.

In searching for the new nanoscale devices [4] that can be integrated, looking for alternatives beyond CMOS technology, and decreasing significantly the power dissipation, spin torque oscillator (STO) devices have significant potential [5]. The question is: can they be integrated and can they solve some significant problems with a physically realizable architecture?

Nanodevices of the following decades, like Spin Torque Oscillators (STOs), which are nano-magnets interacting through spin-waves support reducing the average data transmission distance and prefer local connectivity. The state of such an oscillatory-network is determined by a continuous-time, continuous-value differential equation system, which can provide solution to other practical problems. With this method complex computations are solved in one program cycle in order of milliseconds, while in case of standard CMOS architectures the evolution and solution of such a continuous and equivalent system requires orders of magnitude longer time. The oscillatory-network has the advantage of processing static and frameless dynamic inputs, making it feasible to solve spatial-temporal detection without cutting the input into still frames.

Oscillatory networks generally consists of several oscillators with given physical parameters: the oscillator-dynamics and the coupling interaction between the oscillators, these parameters are described in Section 2.0.1. In case of Spin Torque Oscillators the position and the physical properties of the oscillators are not variable during the computation, the interaction with the outer environment is possible only by excitation and the measurement of the actual magnetic state.

The program on an oscillatory-network is the function of the oscillator dynamics, the topology and the set of the input and the output nodes. The data is considered as the relative state of the oscillators: the unsynchronized oscillation, the partial synchronization, and if the oscillation is fully synchronized, then the phase-difference between the nodes. The phase-difference is given as the difference of consecutive oscillators in a predefined order, from what all differences can be computed. In case of static input the nodes of the oscillatory-network will synchronize after a given time. In case of continuous variable input in time the synchronization might not occur, or it might be disrupted, the presence of relatively constant synchronization also encodes information [3].

Writing a program on STO networks can be defined as finding the correct topology of oscillators, which requires a completely novel approach. To solve a given problem it is advisable to apply some restrictions on the topology, which adapt to the properties of the oscillatory-networks, and which were successfully used with other technologies, considering feasibility. Since the oscillatory networks are cellular, the interaction between closer nodes is stronger than between distant nodes, a CNN-type topology seems to be appropriate.

The oscillators in the Oscillatory CNN (OCNN) are placed on a grid. The rows of the grid are denoted as layers. Thus a program on an OCNN on a $n \times m$ grid is determined by $n + m - 2$ positive values. The nodes of the first (top) layers are denoted as input, and the last (bottom) layer as output. In case of $1 \times m$ chain network the input and the output nodes are the same.

2.0.1 Basic macro model of spin-torque oscillators

Spin torque oscillators (STOs) have gained interest as a possible component for beyond Moore's computing device. Local connectivity between the STOs is provided by physically local spin-wave interactions.

Spin torque oscillators (STOs) are magnetic multilayers, which generate microwave magnetization oscillations in response an impressed DC current. STOs can be made very compact (sub 50 nm in size or below) and their fabrication technology is compatible with standard CMOS processes. The magnetization oscillations can be straightforwardly read-out by monitoring the resistance of the STO. STO oscillation frequencies (10–100 GHz) match well to the typical frequencies involved in CMOS circuit operation and STOs are ‘good quality’ oscillators with high quality factor, low driving current (possibly < 1 mA) and small resistance [6]. This motivates our quest for using STOs as building blocks of electronic circuitry. Oscillators are not a primary building block in traditional signal processing and computing circuitry. However, there are recent proposals of oscillatory neural networks (ONNs) and oscillatory cellular nonlinear networks (O-CNNs) these circuits use primarily oscillators for computing functions (instead of Boolean logic gates) [7,8]. Such non-conventional signal processing concepts have the potential to outperform traditional (Boolean, digital) processing methods, especially if novel, efficient, nanoscale oscillators are employed.

Here we will introduce the basic model of spin torque oscillators which we used during our simulations [7,9]. In this part we have restricted ourselves to a simpler STO model where the oscillation happens ‘in plane,’ namely: the input current vector is the normal vector of the oscillation plane. First we have to describe some parameters to introduce this macro model:

For modelling STOs we use a macrospin model, similar to the one described in [6,9]. This is very approximate model, but reproduces the basic aspects of coupled STO dynamics.

- the geometry of the device is described by the vector $\mathbf{N}$ (i.e. the diagonal elements of the demagnetization tensor known from micromagnetics):

$$\mathbf{N} = (N_x, N_y, N_z)^T \quad (1)$$

The transpose of a vector is denoted by $(\cdot)^T$. The elements of vector $\mathbf{N}$ define the relative geometric length in the X , Y . and Z direction i.e. $N_x + N_y + N_z = 1$ has to be fulfilled. In the following we assume: $N_z > N_x$, $N_x = N_y$. This describes a cylindrical disc having $N_x = 0.2$ and $N_z = 0.6$ in our simulations. This geometry is the easiest to implement in practice and most commonly used [10].

- The polarization of the STO fixed layer is described by the vector $\mathbf{S} = (S_x, S_y, S_z)^T$, we use $\mathbf{S} = [0, 0, 1]^T$
- The magnetization is defined by the vector:

$$\mathbf{M}(t) = (M_x(t), M_y(t), M_z(t))^T \quad (2)$$

The magnitude of $\mathbf{M}(t)$ has to be equal to 1, i.e. the following relation holds

$$M_x^2(t) + M_y^2(t) + M_z^2(t) = 1, \quad \forall t \quad (3)$$

- the magnetic field $\mathbf{H}$ can be defined in terms of the vectors $\mathbf{M}$ and $\mathbf{N}$ as follows:

$$\mathbf{H} = -M_s \begin{pmatrix} N_x M_x \\ N_y M_y \\ N_z M_z \end{pmatrix} \quad (4)$$

- The interaction among STOs occurs through the magnetic field by spin waves. This can be modelled by modifying $\mathbf{H}$ given in (4) as follows:

$$\begin{aligned} \mathbf{H}_{eff} &= -\mathbf{H}_i + \mathbf{G}_j \\ &= -M_s \begin{pmatrix} N_x M_{x_i}(t) \\ N_y M_{y_i}(t) \\ N_z M_{z_i}(t) \end{pmatrix} + M_s \begin{pmatrix} C_x^j M_{x_j}(t - D_j) \\ C_y^j M_{y_j}(t - D_j) \\ C_z^j M_{z_j}(t - D_j) \end{pmatrix} \end{aligned} \quad (5)$$

where the first term is the magnetic field of the i^{th} STO (i.e. $\mathbf{H}_i = M_s(N_x M_{x_i}, N_y M_{y_i}, N_z M_{z_i})^T$) and the second addendum ($\mathbf{G}_j$) describes the influence of the j^{th} STO on the magnetic field of the i^{th} STO. This influence is defined by the vector of the coupling strength $\mathbf{C}^j = (C_x^j, C_y^j, C_z^j)^T$. D_j is delay caused by spin wave propagation. Both $\mathbf{C}^j$ and D_j are functions of the distance between the two oscillators.

- It is worth to point out that mathematically $\mathbf{H}$ is a helper variable only and the micromagnetic equations yield to a coupled system of ODEs.

The equation of the motion of the spin torque nano-oscillator is given by the well known Landau-Lifshitz-Gilbert (LLG) equation shown below:

$$\frac{d\mathbf{M}}{dt} = \gamma(\mathbf{M} \times \mathbf{H}_{eff}) - \gamma\alpha\mathbf{M} \times (\mathbf{M} \times \mathbf{H}_{eff}) - \gamma A\mathbf{M} \times (\mathbf{M} \times \mathbf{S}) \quad (6)$$

The physical parameters of the system are as follows: M_s is a parameter related to the saturation magnetization of the material. Permalloy is characterized by $M_s = 8.6 \cdot 10^5$.

$\times$ denotes the cross product between the vectors,

A is the normalized current¹. We have used this driving current as the input of an oscillator. The synchronization pattern of the oscillator depends on the distribution of this current.

γ (gyro-magnetic ratio) and α (Gilbert damping) are physical constants with values $\gamma = 2.21 \cdot 10^5$ and $\alpha = 8 \cdot 10^{-3}$, respectively [6].

2.1 Application of oscillatory CNN architectures

To show the capabilities of the OCN we have investigated the transformation between the input and the output data of a network. As input we used modified Projected Principal Edge Distribution (PPED) shape feature vectors [11] and HMAX feature vectors [12] generated from images databases. The feature vectors were put on the input, and the output, denoted as signature, was used for nearest neighborhood classification. The PPED is a general image descriptor examining image segments in a moving window. Although the PPED is implemented on dedicated CMOS architecture, the steps of the algorithm can be implemented on CNN.

Increased classification rate was achieved by a transformation where the test set elements get more separable, the regions of same classes get closer to each other and regions of different classes get further. By applying the proposed OCN preprossing, a reasonable separability measure is significantly increased.

The topology was determined by genetic computation. On shape set using modified PPED features the accuracy from 69.6% was increased to 87%, using two chains the result increased to 91.3%. On image set using HMAX features we also measured the ratio of average cross-group distances and in-group distances. The ingroup-crossgroup distance ratio expressing the separability measure of the classification data points increased from the average 5.58 to 78.75 [8].

¹This is a number proportional to the current of the oscillator and not the actual current itself (for instance $A = 100$ corresponds to 1 mA).

Fig. 1. Structure of an Oscillatory CNN. The input is present as a current flow in the top layer, the output are the phase differences between the last layer.

Fig. 2. The structure of the pyramid-like oscillatory network.

2.1.1 Classification using pyramidal cellular oscillatory network

The pyramidal CNN architecture is a binary classification tool on static and dynamic input including the preprocessing of the input data. The grid topology is modified in the way that the number of the oscillators in the layers are decreasing from top to the bottom, where two oscillators remain (Fig. 2). In each layer the distances between the oscillators are also varying. Hence all other constraints of the regular O-CNN are still satisfied: the oscillators are homogeneous, and the interaction between the node cells is depending only on the distances between them.

To demonstrate the functionality of the architecture we constructed two basic examples. To determine the optimal distances between the oscillators to solve the classification, a genetic algorithm was applied, where the fitness value is equal to the accuracy of the classification.

Fig. 3. Resulted topology of the Summation and threshold example.

2.1.2 Summation and threshold classification

The input is a vector of real numbers, the desired output is true, if the sum of the input is higher than a fix threshold, false otherwise (applicable also for AND and OR logical functions). With few steps of the genetic algorithm we reached 100% accuracy.

2.1.3 XOR-type classification

The input is a vector of random numbers from intervals representing logical values. The desired output is true, if the number of true inputs is odd, false otherwise. In hundred steps we could achieve 95% accuracy.

Fig. 4. Resulted topology of the XOR-type example.

Fig. 5. Example inputs for detection of the moving signal direction.

Fig. 6. The classification accuracy depending on the noise level. The vertical axis shows intervals of the noise level.

2.1.4 Spatial-temporal classification

In this example we utilize the memory in time represented by the oscillation to classify spatial-temporal inputs by the proposed architecture at the oscillator-level. We investigated a basic classification problem of detecting the direction of a moving signal with different noise ratio. Example inputs are shown on Fig. 5. The resulted oscillator network showed high classification accuracy on low noise and acceptable performance even on high noise level. The exact accuracy depending on the noise is shown on Fig. 6, the resulted topology on Fig. 7

3. Frameless detection

Spatial-temporal event detection is a crucial task in machine-vision and it is usually difficult to be handled efficiently with current algorithms and devices. The human vision system can solve extremely difficult tasks with simplicity and low power consumption. In this section we show how cellular wave computing algorithms via cellular nonlinear networks (CNNs) are capable of detecting certain spatial-temporal events. The detection is done by using continuous dynamics without cutting the input flow into frames. We can observe similar structures, the analogy of delayed type templates in the retina as well, which performs well and efficiently in processing tasks.

The detection of spatial features on still images is well developed and one can find many libraries for solving various image processing tasks [13]. These libraries and operations can applied for cellular neural networks (CNNs) as well.

Fig. 7. Topology of the pyramid ON with the highest fitness value resulted the accuracy on Fig. 6. The input row is at the bottom of the network, the output is computed as the difference os the phases of the top oscillators.

However it is almost impossible to find solutions for spatial-temporal detection without frames, not only for cellular wavecomputers but also for regular computers. Most commonly these tasks are solved via frame-by-frame algorithms, which usually handle differences between the consecutive processed frames. They are usually not identifying spatial-temporal dynamics, only spatial characteristics and temporal features are defined as the differences of the consecutive frames. One can find that nearly every practical solution depends on temporal frequencies (if the frequency of the input-sampling changes, the series of the consecutive frames can change as well an this can yield to a completely different results) and are usually not robust. In a more extreme situation if one of the frames is missing, dropped during image acquisition in the input-flow the results may change entirely. This shows that these methods are not identifying real spatial-temporal characteristics and are applied in practice because of the fact, that quantized samples can be handled in regular, transistor-based computers relatively efficiently.

3.1 Frameless detection of spatial-temporal events with delayed CNN templates

In the human vision system one can find some marvelous examples of spatial-temporal detections. The detection of looming objects, for example, takes place in the retina [14], and the operation of the retina is continuous, not frame-based. This detection is done by continuous dynamics and with extremely low power consumption.

Many similarities between the operation of the retina and cellular nonlinear networks have been shown [15]. However the template operations in standard CNNs happen with the same time constant in every direction and within the cell, meanwhile in the retina the communication time between the different cells (rod, cones and ganglion cells) and through the different ion channels are different, as well as electrical and chemical synapses. Considering this, we will examine certain type of delayed-CNNs and show how simple spatial temporal dynamics can be used to identify continuous dynamics on these delayed type CNNs. Delayed type templates can provide us with even more flexibilities and possibilities in new applications including frameless detection of motion features.

3.1.1 Delayed type CNNs

Great attention have been paid to cellular wave computers in recent years and many publications were presented regarding visual processing [16]. The applications clearly show that CNN algorithms can solve image and signal processing tasks efficiently and with low power consumption both for topographic and non-topographic tasks.

Although delayed type templates have been investigated [17, 19] most papers introduced and investigated stability criteria (with Ljapunov methods and linear matrix inequality techniques) [18, 20]. But the applicability of such architectures was not investigated extensively (except e.g. [18]).

To describe delayed type CNN dynamics we define an $M \times N$ grid, containing $n = MN$ the CNN elements: The behavior of the (i, j) th element ($i = 1, 2, 3 \dots N$ and $j = 1, 2, 3 \dots M$) is defined by the following delayed differential equation:

$$\begin{aligned} \frac{dx_{i,j}(t)}{dt} = & -\gamma_{i,j}x_{i,j}(t) + \sum_{k,l \in S_{i,j}} A_{i,j,k,l}y[x_{i,j,k,l}(t)] \\ & + \sum_{k,l \in S_{i,j}} A_{i,j,k,l}^D y[x_{i,j,k,l}(t - \tau_{A_{i,j,k,l}}(t))] \\ & + \sum_{k,l \in S_{i,j}} B_{i,j,k,l}u_{i,j,k,l}(t) + \sum_{k,l \in S_{i,j}} B_{i,j,k,l}^D u_{i,j,k,l}(t - \tau_{B_{i,j,k,l}}(t)) + Z_{i,j} \end{aligned} \quad (7)$$

The state of the CNN cells is defined by: $x_{i,j}(t) \in \mathbb{R}^{N \times M}$, the state vector of the system with initial conditions: $x_{i,j}(s)$, ($s \in [-\tau, 0]$, $i = 1, 2, 3 \dots N$, $j = 1, 2, 3 \dots M$) $\gamma_{i,j}$ are the state coefficients with $\gamma_{i,j} > 0$, representing the rate in which the cell will return to resting potential ($x_{i,j} = 0$) in isolated state. In most papers, also in this one, $\gamma_{i,j}$ are the same for all i, j . $y_{i,j}(t) \in \mathbb{R}^{N \times M}$ are the output values of the cells. $y_{i,j}(t)$ has to be continuous, bounded and monotonically increasing or decreasing on $\mathbb{R}$, satisfying the Lipschitz condition. In this case the nonlinearity is the following:

$$y_{i,j}(t) = \frac{1}{2} |x_{i,j}(t) + 1| - \frac{1}{2} |x_{i,j}(t) - 1| \quad (8)$$

$u_{i,j}(t)$ is the input of cell i, j in time t . $Z_{i,j}$ is the bias of cell i (considered constant). $A_{i,j,k,l}$ and $A_{i,j,k,l}^D$ are defining the feedback and delayed feedback templates for cell i, j, k, l . $B_{i,j,k,l}$ and $B_{i,j,k,l}^D$ are defining the feedforward, input and delayed input templates for cell i, j, k, l . $S_{i,j}$ is the neighborhood, the sphere of influence of cell i, j , $S_{i,j}$ contains the set of those cells which will directly influence the behavior of cell i, j . In case of space-invariant CNNs values of template A, A^D, B, B^D are not depending on their absolute position in the grid, they do not depend on i, j only on k, l . The transmission delays are defined as: $\tau_{A_{i,j,k,l}}(t)$ and $\tau_{B_{i,j,k,l}}(t)$. If the delays are not changing in time we can define them as $\tau_{A_{i,j,k,l}}$ and $\tau_{B_{i,j,k,l}}$.

To simplify our notation we can define two matrices A_c and A_d , where A_c will define the coupling strengths and A_d will define the coupling delays (τ).

Similarly we can do the same for the input templates: B_c and B_d will determine the coupling and the delay for the input-flow. In this case we do not have to use two different matrices, like feedback and delayed feedback template, because if element a_{ij} is non-zero in one matrix it has to be zero in the other, this way we can melt these two matrices into one.

$$\begin{aligned} \frac{dx_{i,j}(t)}{dt} = & -\gamma_{i,j}x_{i,j}(t) + \sum_{k,l \in S_{i,j}} A_{c,i,j,k,l}y[x_{i,j,k,l}(t - \tau_{A_{d,i,j,k,l}}(t))] \\ & + \sum_{k,l \in S_{i,j}} B_{c,i,j,k,l}u_{i,j,k,l}(t - \tau_{B_{d,i,j,k,l}}(t)) + Z_{i,j} \end{aligned} \quad (9)$$

Physical implementations of delayed CNN dynamics are made by coupling neighbouring cells in a cellular network. In general the cells can be electrical circuits [21], chemical substances [22], spin-torque oscillators [23] etc. If we consider our cellular network as a set of cells where the coupling strength and delay will depend on physical properties, like the distance between the elements in OCNNs, we can derive much simpler dynamics defining this subset of delayed type CNNs. These

networks are especially typical in the nano-scale using nano-magnets or spin torque oscillators as cells, because here the distance between the cells is the only parameter that will determine the connections and in the nano-scale only these type of connections can be implemented efficiently [24]. Also in biological neurons the time constants/delays of different ion channels in the synapses are actually different.

3.1.2 Template design for delayed type CNN templates

If the delays between the cells in both A and B templates are one directional, which means only one of the cells will affect the other and there are no retroactive coupling, the templates can be designed easily. In case of space-invariant CNNs this means that if a template value in A_d is non-zero, the centrally mirrored value has to be zero². Although this constraint may seem strict at first glance, we will see in section 3.1.4 that many of the practical tasks can be solved with templates fulfilling these conditions.

If the conditions above are fulfilled we can define separated time-planes for every cell and we can start template design from the cell which is not determined by any other cells and continue the design on the other, consecutive planes.

Similarly how we can calculate and define templates for certain functionalities with driving-point plots, we can use this methods in designing functionalities with delayed type CNN dynamics.

Although driving-point plots work at only one time instant, in case of delayed CNN dynamics, if we have K different delays in a network and we can define the previously described chain of the cells (acyclic graph) we can define a driving point plot for every cell in the graph and define the connections between the cells with different delays, between the different time instances, time-planes.

Although in theory this method is feasible, in practice the iterations define driving-point plots for every time-planes, which can mean in practice a few thousand different driving-point plots and template designs. In practice we can design the templates by computer programs which can easily handle a few thousand driving-point plots.

3.1.3 Detecting objects with a given speed and direction

Based on the method and prototype example defined in the previous section and with the aid of a computer simulator (we have used the MatCNN simulator³ [26]) we can easily calculate space-invariant CNN dynamics for the CNN universal machine, which will identify previously given trajectories only.

For our simulation setup we have recorded 64 different previously given trajectories (with different speed, path and object characteristics) and we have designed space-invariant CNNs which were able to distinguish between these trajectories. The trajectories are defined as chains of CNN cells and a cell can only become excited if the previous cell in the trajectory is already excited.

We can also use the feedback template to transfer the information from a previously detected object and we can design delayed-feedback templates, where only those objects will be detected, which were detected in the previous frames as well. We have to start our system from all black cells (true initial states) to ensure that all the objects with the given properties will be detected at the beginning of the operation.

The movement detection of an object can be detected by a feed-forward, B type templates only. Implementing a logical function of the state variable using the feedback, A type template we can create a template (combining the A and B templates) where not only the movement of the object is but the output of the detection is combined with logical functions of the previous state. This logical function can be implemented by using the A feedback template.

The movement detection by the B template can also be more complex identifying corners or edges, this way not only the movement but the movement together with some characteristic features (intensities, corners, edges) can be detected.

²In case of space-variant CNNs this condition is fulfilled if the connectivity graph of the cells (an edge is drawn from node i to node j , if i is coupled to j with a given non-zero τ) forms a directed acyclic graph.

³MatCNN can be downloaded from <http://cnn-technology.itk.ppke.hu/>

A delayed-feedback template detecting the objects which are contiguously moving on a trajectory with a given speed can be:

$$A_c = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad A_d = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B_c = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B_d = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$Z = -2$$

These simple example shows how delayed type CNN templates can be designed for given functionalities. A more detailed description of the template design and possible applications can be found in [25, 27, 28].

3.1.4 Prototype applications

In this section we a practical application: gesture detection, using frameless detection, where continuous dynamics are identified by continuous dynamics, none of the methods are sensitive to temporal quantization of the numerical integration of the simulator.

We have generated a setup, which is able to detect hand movement according to predefined trajectories. The different spatial movements and the velocity of the movement are identified by delayed type CNN architectures and it can translate it to stored commands. The system in its current state can distinguish between four different directions (the maximal component of the velocity vector) and three different speeds. This can define a user interface containing 12 different gestures identified by a space-invariant delayed type CNN dynamics.

The novelty of our method is frameless processing, which differs from common image processing techniques. We do not process spatial temporal signals via sequences of frames by discrete operations or processing individual frames and then combining or subtracting them from each other. Our detection is a result of interacting continuous wave dynamics combined with the dynamics of our device. The detection does not need temporal quantization and it can be applied in case of extremely fast events as well. We think this kind of detection can be applied in those cases where there are no inherent clock cycles, like in most of the currently researched nano devices.

3.2 Frameless detection of spatial-temporal events with lighting activation

In this section a hardware-software composition will be shown, where the frameless computation is done by the combination of three different dynamics. There is an active sensor-array, where in every cell we have an infrared LED and an infrared phototransistor. The phototransistors measure the LEDs' reflected activation light from the environment. This sensor-array is connected to a Cellular Wave Computer running as a simulator on a PC. This computational environment processes the raw measurement data coming from the sensor-array. (For the details of the system, the reader is kindly referred to [29].)

The first out of the three dynamics is the spatial-temporal input flow. This represents the environment or the object what we want to measure. It is completely independent from the two other dynamics. In the formulas (14)–(17) it is referred with u .

The second wave is the self-dynamics of the computing cellular array. From the CNN literature it is known as the inner state of the cells, and can be described with (10). Here $x_{i,j} \in \mathbb{R}$, $y_{k,l} \in \mathbb{R}$, $u_{k,l} \in \mathbb{R}$, $z \in \mathbb{R}$ are the state, the output, the input and the bias of cell (i, j) respectively. A and B are the feedback and the input synaptic operators; $rd \in \mathbb{N}$ is the radius of the sphere of influence of cell (i, j) . The dependence of the output on the state is described by (11) which is the standard piecewise-linear function. The form of the applied template-class is characterized by (12), and the parameter values are according to (13).

$$\dot{x}_{ij} = -x_{ij}(t) + \sum_{|k-i| \leq rd} \sum_{|l-j| \leq rd} A(i-k, j-l) y_{kl}(t) + \sum_{|k-i| \leq rd} \sum_{|l-j| \leq rd} B(i-k, j-l) u_{kl}(t) + z \quad (10)$$

$$y = f_{pwl} = \frac{1}{2}(|x+1| - |x-1|) \quad (11)$$

$$A = \begin{bmatrix} 0 & 0 & 0 \\ s & p & q \\ 0 & r & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 0 \end{bmatrix}, z = z \quad (12)$$

$$s = 1.1, \quad p = 1.0, \quad q = -1.1, \quad r = -0.7, \quad b = 1.0, \quad z = 0.0 \quad (13)$$

In the formulas (14)–(17) this self-dynamics of the computing array is defined by F , the inner states are referred with x .

The third wave out of the three is the dynamics of the activation light-sources. In formulas (14)–(17) it is defined by G_1 or G_2 , the inner states of them are referred with v .

We are interested in the interaction of these waves, which can happen in two different cases:

- *independent activation* case, as described by (14) and (15):

$$\dot{x} = F(x, f_1(x), u) \quad (14)$$

$$v = \text{const. or } \dot{v} = G_1(v, f_2(v)) \quad (15)$$

- *adaptive activation* case, as described by (16) and (17):

$$\dot{x} = F(x, f_1(x), u) \quad (16)$$

$$\dot{v} = G_2(v, f_2(v), f_1(x)) \quad (17)$$

In the formulas (14)–(17) f_1 usually means the piecewise linear function or a kind of a smooth sigmoid function, while f_2 stands for a kind of step function referring to the two main working modes of the activation light sources (ON or OFF). The difference between the two cases is in the activation dynamics: (15) can generate only an autonomous activation pattern (like constant strength activation light or some stroboscopic/periodic pattern), while (17) can facilitate in theory the evolution and processing of the input flow.

In the following subsection we will see an example from the simpler case, where the activation is constant throughout the whole measurement.

Detection of an oversized terrain feature

In this subsection we would like to present a measurement, where the difference between the frame-by-frame and frameless processing can be clearly seen, in the afore mentioned system. Figure 8 summarizes the measurement setup: the sensor array passes above a convex bump (terrain feature), which has approximately 3.5-times bigger size, than the sensor array itself. Throughout the movement we record the input flow to the system. Although these are discrete frames, we can decide whether we want to process them as individual measurements (frame-by-frame case), or we want to process them with one, continuous computation (frameless case). If we process them separately, we can observe different patterns and dynamics at the output of the system regardless the position of the sensor array. On the contrary, if we process them in the appropriate order in one transient, we can uniquely observe Pattern 1 in Region 1, Pattern 2 in Region 2 and Pattern 3 in Region 3. In this way we can identify the regions and also the whole complex bump on the basis of the output patterns.

A deeper study of this measurement and phenomenon can be found in [30]. As it turned out, the reason is the qualitative difference of the seemingly similar input pictures, even in the same regions.

4. Related topics

4.1 Long transient - metastable rotating waves

In this section we will review the behavior of a class of one-dimensional autonomous (inputless, $B = 0$) CNN array with periodic boundary condition. Between the nearest neighbors we have positive

measurement scene:

8x8 array responses:

Fig. 8. The upper part of the figure depicts the schematic drawing of the measurement setup: the convex bump with uphill (Region 1), plateau (Region 2), downhill (Region 3) parts; and the sensor array passing over the bump from left to right. The bottom part of the figure shows the three most important array responses (output) throughout the measurement. In frameless processing mode these patterns can be uniquely assigned to the different regions of the terrain. The figure is adopted from [30].

interconnection parameters (non-symmetric, see (18)), and the output function of the cells is the well-known three-segment piecewise linear function (19).

$$A = [\alpha \ 0 \ \beta], \alpha > 0, \beta > 0, \alpha \neq \beta, \quad B = [0 \ 0 \ 0], \quad z = 0 \quad (18)$$

$$y = f_{pwl}(x) = \frac{1}{2}(|x + 1| - |x - 1|) \quad (19)$$

According to the theory, an eventually strongly monotone (ESM) semiflow converges always toward an asymptotically stable equilibrium point in the long run ([31]). Although the afore mentioned system is only a monotone semiflow (not an ESM) due to the squashing effect of the output function ([32]), yet it enjoys Limit Set Dichotomy and convergence properties analogous to those of ESM semiflows ([33–36]) meaning that, it should also converge to an asymptotically stable equilibrium point in the long run.

According to software simulations and measurements on a discrete-component prototype circuit, this system can exhibit unexpectedly long oscillations in a wide set of interconnection parameters and with a wide range of initial conditions before converging to the asymptotically stable equilibrium point ([37–39]).

Figure 9 shows an oscilloscope measurement of the long transient metastable oscillation. The signal is measured on the first cell, the blue graph is the inner state of the cell, the red line is the output signal. On part a) of the figure we can see the first three cycles, while on part b) we can see the 57th - 59th cycles. One can see the distortion of the waveform as time goes on, but the qualitative outlook did not changed. This transient lasted less than 100 periods, but in other measurements even more than 100 cycles were observed.

With a bigger size of the array we can produce more complex waveforms, where sometimes a kind of metamorphosis happens. These seemingly permanent waveforms look as different stable states during

Fig. 9. Inner state (blue) and output (red) signal level of the first cell in the array during a long transient oscillation. It was measured with an oscilloscope in a circuit with 16 cells. The interconnection parameters were $\alpha = 3.5$, $\beta = 2.5$; the initial states were $\{+6V, +6V, +6V, +6V, +6V, +6V, +6V, +6V, -6V, -6V, -6V, -6V, -6V, -6V, -6V, -6V\}$ of the cells, respectively. The left subfigure depicts the first three periods of the oscillation, while the subfigure on the right shows the 57th, 58th and 59th cycles. This oscillation lasted less than 100 periods, but in other measurements even more than 100 cycles were observed.

the transient, but a sudden change occasionally transforms them into the direction of the simpler waveforms (lower frequency; less zero-crossings per a real period) up until the stable equilibrium point, which is the final station of these oscillations.

4.2 Global optimization

Boolean satisfiability problems (such as k-SAT) constitute one of the hardest classes of constraint satisfaction problems lying at the basis of many decision, scheduling, error-correction and bio-computational applications. Being an NP-complete problem increases the relevance of k-SAT, because every problem in NP can be transformed into this form in polynomial time. NP-complete problems are efficiently (polynomial time) checkable, but the worst-case complexity of finding a solution is exponential on Turing machines [40]. Since such problems are ubiquitous in science and technology, the development of efficient solvers for NP-complete problems becomes an extremely important task. It is for this reason that all new computational paradigms or proposed technologies are also tested and benchmarked on NP-complete problems.

In [41] provided a new and exact mapping of Boolean satisfiability into a set of ordinary differential equations (ODE) with a unique correspondence between its set of attractors and the k-SAT solutions. The parameters of the ODEs (set by the k-SAT problem) are considered as the input of the computation and the output results from the convergence of the dynamics to one of its attractors, which corresponds to a k-SAT solution.

The authors in [42] presented a continuous-time dynamical system (CTDS) designed with the simplest equations in mind providing a one-to-one correspondence between k-SAT solutions and the CTDS attractors. As a result, the system's auxiliary variables were unbounded, which, however, should preferably be avoided in physical implementations. As the system presented in [42] is not unique, we may search for other, implementation-friendly CTDS designs that could be used as k-SAT solvers.

In [43, 44] they present a novel CTDS for k-SAT that is similar to CNN models, with local interactions, continuous, bounded state values and continuous computational model and thus implementation friendly.

The equations are the following:

$$\dot{s}_i(t) = \frac{ds_i(t)}{dt} = -s_i(t) + Af(s_i(t)) + \sum_m c_{mi}g(a_m(t)) \quad (20)$$

$$\dot{a}_m(t) = \frac{da_m(t)}{dt} = -a_m(t) + Bg(a_m(t)) + \sum_i c_{mi}f(s_i(t)) + 1 - k \quad (21)$$

Where two different cells represent the state variables and the constraints, with different dynamics. The state variables are evolving according equation (20) where $s_i(t)$ is the cell of the i state variable. The constraints are noted as $a_m(t)$ and are evolving according to Eq. (21). The coupling is determined according to the constraints. If a constraint contains a state variable there will be a positive coupling between the two cells (with weight: +1), if the constraint contains a negated state variable the will be connected with weight -1. If a state variable is not used in a constraint, the two elements are not connected.

The nonlinearities in the equations are defined as:

$$f(s_i) = \frac{1}{2}(|s_i + 1| - |s_i - 1|) \quad (22)$$

for the state variables, and for the constraints the non-linearity can be written as:

$$g(a_m) = \frac{1}{2}(1 + |a_m| - |1 - a_m|) \quad (23)$$

4.2.1 A small k -SAT problem for CNN implementation

We will demonstrate a small k -SAT problem which can be solved using a CNN of 45 cells. In spite of its small size (only 10 variables and 35 constraints) it shows interesting behavior.

In this simple 3-SAT problem we have $N = 10$ variables and $M = 35$ constraints. In each constraint there are 3 logical variables out of the N possible. These variables can either be in their original, either in their negated forms (denoted by an over line). For example a constraint could look like $C_1 = x_4 \vee x_2 \vee x_9$. The exact problem can be formulated as:

$$\begin{aligned} & (x_6 \vee \bar{x}_9 \vee x_4) \wedge (\bar{x}_8 \vee x_{10} \vee x_2) \wedge (\bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_6) \wedge (\bar{x}_4 \vee x_6 \vee \bar{x}_5) \wedge \\ & (\bar{x}_3 \vee \bar{x}_4 \vee x_7) \wedge (x_1 \vee \bar{x}_7 \vee x_3) \wedge (x_2 \vee \bar{x}_3 \vee x_7) \wedge (\bar{x}_1 \vee \bar{x}_7 \vee x_{10}) \wedge \\ & (\bar{x}_6 \vee \bar{x}_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_6 \vee x_4) \wedge (\bar{x}_8 \vee x_5 \vee x_4) \wedge (\bar{x}_6 \vee \bar{x}_7 \vee x_{10}) \wedge \\ & (x_2 \vee \bar{x}_7 \vee x_6) \wedge (x_{10} \vee x_3 \vee x_4) \wedge (\bar{x}_1 \vee x_{10} \vee x_2) \wedge (x_6 \vee \bar{x}_1 \vee x_4) \wedge \\ & (x_1 \vee x_{10} \vee x_4) \wedge (\bar{x}_4 \vee \bar{x}_9 \vee x_2) \wedge (\bar{x}_4 \vee \bar{x}_8 \vee x_6) \wedge (\bar{x}_1 \vee \bar{x}_4 \vee \bar{x}_7) \wedge \\ & (\bar{x}_7 \vee \bar{x}_1 \vee x_2) \wedge (x_9 \vee x_{10} \vee x_4) \wedge (\bar{x}_3 \vee \bar{x}_9 \vee x_5) \wedge (x_6 \vee x_{10} \vee x_8) \wedge \\ & (\bar{x}_4 \vee \bar{x}_9 \vee x_{10}) \wedge (x_1 \vee x_{10} \vee x_5) \wedge (x_8 \vee \bar{x}_4 \vee x_9) \wedge (x_4 \vee x_6 \vee x_2) \wedge \\ & (x_3 \vee \bar{x}_4 \vee \bar{x}_6) \wedge (x_9 \vee \bar{x}_7 \vee x_{10}) \wedge (x_5 \vee \bar{x}_7 \vee \bar{x}_8) \wedge (\bar{x}_7 \vee \bar{x}_8 \vee \bar{x}_9) \wedge \\ & (x_4 \vee \bar{x}_7 \vee x_5) \wedge (\bar{x}_3 \vee \bar{x}_6 \vee \bar{x}_9) \end{aligned} \quad (24)$$

An example of the state transition can be seen on Fig. 10.

As it can be seen the states are converging to the solution of the formula, which clearly shows, how optimization problems can be solved by continuous, frameless dynamics.

5. Conclusions

In this paper we have reviewed some very recent simple methods and examples demonstrating, how frameless, continuous-time computational models can be used in practice for spatial-temporal event detection. The computational models are partially inspired by the human nervous system (especially the early visual system), where computation is done in a wave like continuous time domain. Although, spatial-temporal event detection is one of the most difficult challenges in information-technology, the examples clearly show that the computational model can be applied beyond these tasks, for example in global optimization and in the solution of NP-complete problems. One common denominator of the methods is the application of the general cellular wave computing principles. These methods can fit ideally to nanoscale devices, where analog signals are present (e.g. magnetic waves).

Fig. 10. The state transition of the 10 states and the 35 constraints can be seen on this plot in time (on the x-axis). The different states are marked with different colors. As it can be seen after the transition, the states converge to the solution of the formula and they are all converging to either true (with value 1.5) or to false (with value -1.5) values.

Acknowledgments

The support of the grants TÁMOP-4.2.1.B-11/2/KMR-2011-0002 and TÁMOP-4.2.2/B-10/1-2010-0014 are gratefully acknowledged.

References

- [1] A. Turing, “On computable numbers with an application to the entscheidungsproblem,” *Proceedings of the London Mathematical Society*, Ser. 2, vol. 42, 1937.
- [2] S. Wolfram, “A new kind of science,” *Wolfram Media*, 2002.
- [3] A. Horvath, A. Stubendek, and T. Roska, “One dimensional cellular array of in-plane spin torque oscillators,” *The 13th IEEE International Conference on Nanotechnology*, Beijing, China, 2013.
- [4] R.K. Cavin, V.V. Zhirnov, G. Bourianoff, J.A. Hutchby, D.J.C. Herr, H.H. Hosack, W.H. Joyner, and T.A. Wooldridge, “A long-term view of research targets in nanoelectronics,” *Journal of Nanoparticle Research*, vol. 7, pp. 573–586, 2005.
- [5] G. Bourianoff, “Towards a Bayesian processor implemented with oscillatory nanoelectronic arrays,” *Keynote lecture at IEEE International Workshop on Cellular Nanoscale Networks and Applications (CNNA 2012)*, Turin, Italy, 2011.
- [6] S.E. Russek, W.H. Rippard, T. Cecil, and R. Heindl, “Spin-transfer nano-oscillators,” *Handbook of Nanophysics: Functional Nanomaterials*, 2010.
- [7] A. Horvath, F. Corinto, Gy. Csaba, W. Porod, and T. Roska, “Synchronization in cellular spin torque oscillator arrays,” *13th IEEE International Workshop on Cellular Nanoscale Networks and Applications*, Turin, Italy, 2012.
- [8] T. Roska, A. Horvath, A. Stubendek, F. Corinto, Gy. Csaba, W. Porod, T. Shibata, and G. Bourianoff, “An associative memory with oscillatory CNN arrays using spin-torque oscillator cells and spin-wave interactions architecture and end-to-end simulator,” *13th IEEE International Workshop on Cellular Nanoscale Networks and Applications*, Turin, Italy, 2012.
- [9] Gy. Csaba, M. Pufall, D. Nikonov, G. Bourianoff, A. Horvath, T. Roska, and W. Porod, “Spin torque oscillator models for applications in associative memories,” *13th IEEE International Workshop on Cellular Nanoscale Networks and Applications*, Turin, Italy, 2012.
- [10] D.V. Berkov and N.L. Gorn, “Magnetization oscillations induced by a spin-polarized current in

- a point-contact geometry: Mode hopping and nonlinear damping effects,” *Physical Review B (Condensed Matter and Materials Physics)*, vol. 76, pp. 1–18, 2007.
- [11] M. Yagi and T. Shibata, “An image representation algorithm compatible with neural-associative-processor-based hardware recognition systems,” *IEEE Transactions on Neural Networks*, vol. 14, no. 5, pp. 1144–1161, September 2003.
 - [12] M. Riesenhuber and T. Poggio, “Hierarchical models of object recognition in cortex,” *Nature Neuroscience*, pp. 1019–1025, September 1999.
 - [13] D.A. Forsyth and J. Ponce, “Computer vision: A modern approach,” *Prentice Hall*, 2011.
 - [14] A. Zarandy and T. Fulop, “Approaching object detector mouse retina circuit model analysis and implementation on cellular sensor-processor array,” *International Journal of Circuit Theory and Applications*, vol. 40, no. 12, pp. 541–569, August 2012.
 - [15] F. Werblin, T. Roska, and L.O. Chua, “The analogic cellular neural network as a bionic eye,” *International Journal of Circuit Theory and Applications*, vol. 23, no. 6, pp. 541–569, November–December 1995.
 - [16] L.O. Chua and T. Roska, “Cellular neural networks and visual computing: Foundations and applications,” *University Press, Cambridge*, 2002.
 - [17] T. Roska, “Cellular neural networks with nonlinear and delay-type template elements,” *Proceedings of IEEE International Workshop on Cellular Neural Networks and Their Applications*, pp. 12–25, 1990.
 - [18] Z. Wu, X. Liao, and J. Yu, “Conditions of asymptotic stability for cellular neural networks with time delay,” *Journal of Electronics*.
 - [19] T. Roska, T. Boros, A. Radvanyi, P. Thiran, and L.O. Chua, “Detecting moving and standing objects using cellular neural networks,” *International Journal of Circuit Theory and Applications*, vol. 20, no. 5, pp. 613–628, 1992.
 - [20] A.I. Margaritis, “On the global stability of time delayed CNNs,” *Central European Journal of Computer Science*, vol. 1, no. 1, pp. 67–100, March 2011.
 - [21] A. Rodriguez-Vazquez, S. Espejo, R. Dominguez-Castron, J.L. Huertas, and E. Sanchez-Sinencio, “Current-mode techniques for the implementation of continuous- and discrete-time cellular neural networks,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, no. 3, pp. 132–146, March 1993.
 - [22] T. Asai, Y. Kanazawa, T. Hirose, and Y. Amemiya, “Analog reaction-diffusion chip imitating Belousov-Zhabotinsky reaction with hardware oreonator model,” *Int. Journal of Unconventional Computing*, vol. 1, pp. 123–147, 2005.
 - [23] Gy. Csaba, M. Pufall, D. Nikonov, G. Bourinaoff, A. Horvath, T. Roska, and W. Porod, “Spin torque oscillator models for applications in associative memories,” *13th IEEE International Workshop on Cellular Nanoscale Networks and Applications*, Turin, Italy, 2012.
 - [24] A. Adamatzky and C. Guanrong, “Chaos, CNN, memristors and beyond: A festschrift for Leon Chua,” *World Scientific*, 2013.
 - [25] A. Horvath and T. Roska, “Detection of spatial-temporal events with delayed CNN templates,” *International Symposium on Nonlinear Theory and Its Applications*, Santa Fe, USA, 2013.
 - [26] Cs. Rekeczky, “Analogic CNN simulation toolbox for MATLAB. Version 1.0,” *MTA SZTAKI*, 1997.
 - [27] M. Koller, A. Horvath, and T. Roska, “Frameless computing for spatial-temporal events,” *21st European Conference on Circuit Theory and Design*, Drezden, Germany, 2013.
 - [28] A. Horvath and T. Roska, “Frameless detection of spatial-temporal events with delayed CNN templates,” *Jedlik Laboratories, Pazmany Peter Catholic University, Budapest, Jedlik Laboratories Reports*, vol. 1, no. 2, pp. 1–22, December 2013.
 - [29] M. Koller and Gy. Cserey, “Spatial-temporal active wave computing using infrared proximity array,” *International Journal of Circuit Theory and Applications*, vol. 40, no. 12, pp. 1209–1218, December 2012.
 - [30] M. Koller and T. Roska, “Frameless spatial-temporal event detection via lighting activation,” *Jedlik Laboratories, Pazmany Peter Catholic University, Budapest, Jedlik Laboratories Reports*,

- vol. 1, no. 4, pp. 1–30, December 2013.
- [31] M. Hirsch and H. Smith, “Monotone dynamical systems,” in *Handbook of Differential Equations, Ordinary Differential Equations (second volume)*, A. Canada, P. Drabek, A. Fonda Eds., Elsevier, 2005, (available at: <http://math.la.asu.edu/halsmith/book.html>).
 - [32] M. Di Marco, M. Forti, M. Grazzini, and L. Pancioni, “The dichotomy of omega-limit sets fails for cooperative standard CNNs,” *12th IEEE International Workshop on Cellular Nanoscale Networks and Applications*, Berkeley, CA, February 2012.
 - [33] M. Di Marco, M. Forti, M. Grazzini, and L. Pancioni, “Limit set dichotomy and convergence of semiflows defined by cooperative standard CNNs,” *International Journal of Bifurcation and Chaos*, vol. 20, pp. 3549–3563, November 2010.
 - [34] M. Di Marco, M. Forti, M. Grazzini, and L. Pancioni, “Limit set dichotomy and convergence of cooperative piecewise linear neural networks,” *IEEE Transaction on Circuits and Systems I*, vol. 58, pp. 1052–1062, May 2011.
 - [35] M. Di Marco, M. Forti, M. Grazzini, and L. Pancioni, “Convergence of a class of cooperative standard cellular neural network arrays,” *IEEE Transaction on Circuits and Systems I*, vol. 59, pp. 772–783, April 2012.
 - [36] M. Di Marco, M. Forti, M. Grazzini, and L. Pancioni, “Further results on convergence of cooperative standard cellular neural networks,” *IEEE International Symposium on Circuits and Systems*, Rio de Janeiro, May 2011.
 - [37] M. Forti, B. Garay, M. Koller, and L. Pancioni, “An experimental study on long transient oscillations in cooperative CNN rings,” *13th International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA)*, Turin, Italy, 2012.
 - [38] M. Forti, B. Garay, M. Koller, and L. Pancioni, “Long transient oscillations in a class of cooperative cellular neural networks,” *International Journal of Circuit Theory and Applications*, November 2013.
 - [39] M. Di Marco, M. Forti, B. Garay, M. Koller, and L. Pancioni, “Multiple metastable rotating waves and long transients in cooperative CNN rings,” *21st European Conference on Circuit Theory and Design*, Drezden, Germany, 2013.
 - [40] M.R. Garey and D.S. Johnson, “Computers and intractability: A guide to the theory of NP-completeness,” *Series of Books in the Mathematical Sciences*, first edition ed. W.H. Freeman and Co Ltd, January 1979.
 - [41] B. Molnár, Z. Toroczkai, M. Ercsey-Ravasz, “Continuous-time Neural Networks Without Local Traps for Solving Boolean Satisfiability,” *13th International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA)*, Turin, Italy, 2012.
 - [42] M. Ercsey-Ravasz and Z. Toroczkai, “Optimization hardness as transient chaos in an analog approach to constraint satisfaction,” *Nature Physics*, vol. 7, pp. 966–970, October 2011.
 - [43] B. Molnár and M. Ercsey-Ravasz, “Asymmetric continuous-time neural networks without local traps for solving constraint satisfaction problems,” *PLoS ONE*, vol. 8, October 2013.
 - [44] M. Ercsey-Ravasz, Z. Neda, and T. Roska, “Cellular neural networks for NP-hard optimization,” *EURASIP Journal on Advances in Signal Processing*, vol. 2009, no. 2, January 2009.