

## Research Article

# Simulation, Modeling, and Analysis of Soliton Waves Interaction and Propagation in CNN Transmission Lines for Innovative Data Communication and Processing

G. Borgese,<sup>1</sup> S. Vena,<sup>2</sup> P. Pantano,<sup>2</sup> C. Pace,<sup>1</sup> and E. Bilotta<sup>2</sup>

<sup>1</sup>*Dipartimento di Informatica, Modellistica, Elettronica e Sistemistica (DIMES), Università della Calabria, Via P. Bucci 42C, 87036 Arcavacata di Rende, Cosenza, Italy*

<sup>2</sup>*Dipartimento di Fisica, Università della Calabria, Via P. Bucci 31C, 87036 Arcavacata di Rende, Cosenza, Italy*

Correspondence should be addressed to G. Borgese; gianluca.borgese@dimes.unical.it

Received 14 February 2015; Revised 7 April 2015; Accepted 14 April 2015

Academic Editor: Lu Zhen

Copyright © 2015 G. Borgese et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present an innovative approach to study the interaction between oblique solitons, using nonlinear transmission lines, based on Cellular Neural Network (CNN) paradigm. A single transmission line consists of a 1D array of cells that interact with neighboring cells, through both linear and nonlinear connections. Each cell is controlled by a nonlinear Ordinary Differential Equation, in particular the Korteweg de Vries equation, which defines the cell status and behavior. Two typologies of CNN transmission lines are modelled: crisscross and ring lines. In order to solve KdV equations two different methods are used: 4th-order Runge-Kutta and Forward Euler methods. This is done to evaluate their accuracy and stability with the purpose of implementing CNN transmission lines on embedded systems such as FPGA and microcontrollers. Simulation/analysis Graphic User Interface platforms are designed to conduct numerical simulations and to display elaboration results. From this analysis it is possible both to identify the presence and the propagation of soliton waves on the transmission lines and to highlight the interaction between solitons and rich nonlinear dynamics. With this approach it is possible to simulate and develop the transmission and processing of information within large brain networks and high density sensor systems.

## 1. Introduction

In the scientific literature in the last 40 years, the presence of solitons in Nonlinear Transmission Lines (NLTL) has been extensively studied, both theoretically and experimentally [1–4]. In the continuous case, the Korteweg de Vries (KdV) equation [5] can be easily deduced, by using the reductive perturbation methods [6]. It is well known that this equation has solutions of soliton type, the behavior of which has been extensively studied both numerically [7, 8] and analytically [9, 10]. Also some interesting related negative-order integrable equations, including the negative-order KdV equation and some associated properties, were obtained, such as the results in [11–13]. Since the early work in NTLs [1, 14, 15], these solitons are still subject to studies [16–20]. Recently, the first robust electrical oscillator has been built, the dynamics of which is based on soliton waves [21, 22].

Recent works present a completely different approach [23–26], based on a Cellular Neural Network (CNN). Introduced in 1988 by Chua and Yang [27, 28], CNN consists of a lattice of individual cells, which interact with neighboring cells, usually to perform very advanced sensory recognition tasks. In general, the state of the CNN cells evolves in compliance with nonlinear Ordinary Differential Equations (ODEs), and it is well known that the behavior of such systems is very rich with striking manifestations of chaos. Nonlinear ODEs describe a set of physical and biological phenomena, fundamental in contemporary scientific research and dynamical systems theory.

The nonlinear dynamics of the CNN is extremely complex and is used in contexts of real-time image processing [29], to study the emergent behaviors of complex systems (e.g., Turing patterns), or as metamodel to simulate other mathematical systems, such as Cellular Automata (CA) [30, 31].

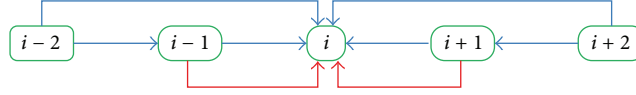


FIGURE 1: Each cell interacts in both linear (blue lines) and nonlinear fashion (red lines) with their adjacent cells but just in a linear way with neighboring not adjacent cells.

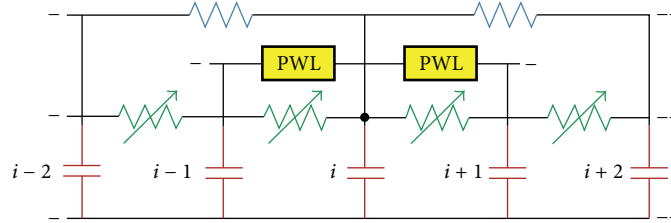


FIGURE 2: CNN circuits. The linear connections are made using active and passive resistances; whereas, non-linear connections are modeled using non-linear piece-wise functions.

CNNs can be easily implemented in hardware and, therefore, capable of processing large amounts of information very quickly and in parallel. They can potentially be used to build artificial organs, such as retinas or other sensory systems, to be embodied in artificial robots, to analyze vast amounts of data automatically coming from Magnetic Resonance Imaging (MRI), for patients with degenerative diseases [32].

In [33] CNN NLTL have been presented and implemented on FPGA. This FPGA system, called DCMARK (Distributed Computing Micro-ARCHitecture), allows solving complex differential equations using less time than other systems. DCMARK uses the 1D KdV equation just as a benchmark to evaluate and test the proper working of the system. Ultimately, it can produce a complex nonlinear dynamics and show soliton waves. One of the main problems in the soliton theory is the interaction between solitons in multiple dimensions [34–37], due to its connection with application. For example, this problem is of fundamental importance in optical computing [38].

This approach allows the introduction of crisscrossed transmission lines for studying the interaction among soliton waves. Unlike traditional models based on PDEs that do not permit studying the intersection of oblique solitons, this approach allows crossing the lines of propagation in a very simple way and observing the behavior of solitons, when they cross another line or when they collide with each other. Throughout this paper, we will see how this nonlinear dynamics is extremely rich and varied, as it presents unexpected and unpredictable behaviour. Such dynamics, never observed in these contexts, may be useful for engineering applications, specifically dedicated to physical systems for transmitting information. In this paper, we deal with the following: the general concept of CNN transmission and typologies of NLTL investigated in Section 2, the simulation/analysis software environments used in Section 3, the evaluation of simulation results in Section 4, and the conclusions in Section 5.

## 2. CNN Transmission Lines

As already introduced before, a CNN transmission line can be seen as a 1D cellular structure of  $N$  elements, called “cells,” which interact through both linear and nonlinear connection with their adjacent cells and through only linear connection with the other neighbours as in Figure 1.

Let us suppose that the dynamics of each cell can be modeled through the following ODE:

$$\dot{u}_i = P(u_{i=\pm 1}) + L(u_{i=\pm 2}), \quad (1)$$

where

$$P(x) = ax + bx^2 + \dots \quad (2)$$

is a polynomial function of  $x$  and  $L(x) = sx$  is a linear function of  $x$ . If we restrict our attention to polynomials of second degree, (1) can be written as

$$\dot{u}_i = \sum_{j=-2i \neq j}^{+2} \alpha_{i+j} u_{i+j} + \sum_{j=-1i \neq j}^{+1} \beta_{i+j} u_{i+j}^2, \quad (3)$$

where  $\alpha_{i+j}$  (with  $j = -2; -1; +1; +2$ ),  $\beta_{i+j}$  (with  $j = -1; +1$ ) are coefficients. In (3),  $\alpha_i = \beta_i = 0$ . Assuming that  $\alpha_{i+j}$  and  $\beta_{i+j}$  do not depend on the  $i$ th cell, that is, we consider homogeneous networks, then (3) depends on 6 parameters  $\alpha_{-2}$ ,  $\alpha_{-1}$ ,  $\alpha_1$ ,  $\alpha_2$ ,  $\beta_{-1}$ , and  $\beta_1$ , representing the genome of the network. Equation (3) differs from the standard CNN, because of the presence of a reaction-diffusion term and because it is a state-controlled network.

These networks can be easily implemented in hardware Figure 2, using single-cell capacitors and resistors for active and passive connections, respectively. Nonlinear connections can be modeled through Piecewise Linear (PWL) functions.

Considering  $\alpha_{-2} = 1/2h^3$ ,  $\alpha_{-1} = -1/h^3$ ,  $\alpha_1 = 1/h^3$ ,  $\alpha_2 = -1/2h^3$ ,  $\beta_{-1} = 1/2h$ , and  $\beta_1 = -1/2h$ , where  $h$  is a constant

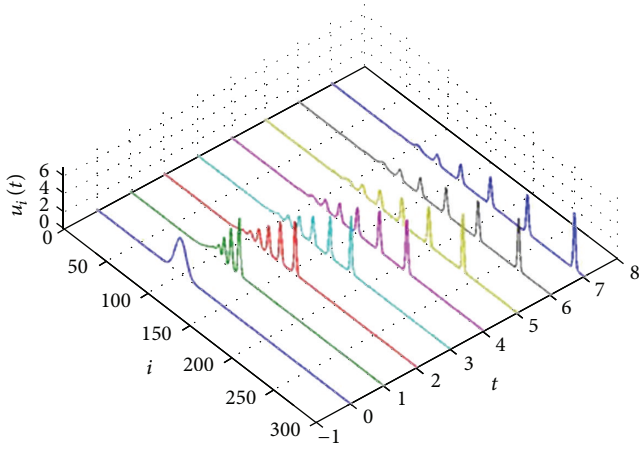


FIGURE 3: The initial Gaussian data decomposed into a series of localized travelling waves.

parameter that depends on the number of cells and on the spatial interval, (3) becomes

$$\begin{aligned} \dot{u}_i = & \frac{1}{2h^3} [u_{i-2} - 2u_{i-1} + 2u_{i+1} - u_{i+2}] \\ & - \frac{3}{2h} [u_{i+1}^2 - u_{i-1}^2]. \end{aligned} \quad (4)$$

This is the spatial discretized version of 1D KdV equation. In fact, (4) in the continuum limit  $h \rightarrow 0$  approximates to the following 1D KdV equation:

$$\frac{\partial u(t, x)}{\partial t} = -3 \frac{\partial [u(t, x)]^2}{\partial x} - \frac{\partial^3 u(t, x)}{\partial x^3}. \quad (5)$$

Equation (4) is getting solving of Kirchoff's law as in [25].

Simulating (4) and assuming an initial function such as a Gaussian function, we observe the emergence of traveling waves as shown in Figure 3.

Simulating a CNN transmission line of 300 cells and fixing properly the boundary conditions, traveling soliton waves are shown. In fact, as the classical solitons of the KdV equation, they have very well-defined profiles, with a particular relationship between amplitude, width, and velocity of propagation. The taller and slimmer solitons travel faster than the lower and wider ones, and they maintain their distinctive character in the interaction with the others, as it will be explained later.

If we consider  $\alpha_i < 0$  in (4), a new linear damping term is added. In this case, this equation becomes a modified KdV equation, due to the presence of inhomogeneity. If  $\alpha_i = 1/2t$ , (4) becomes, in the continuous case, just the cylindrical KdV equation.

Model (1), whose corresponding patterns are plotted in Figure 3, gives rise to a rich environment, in which the nonlinear dynamics can be studied.

On the basis of the same reasoning, with some modifications, it is possible to simulate the crisscross of two 1D CNN transmission lines analysing the interaction between

solitons belonging to different lines. Even if the interaction of oblique solitons is a classic problem, it is still far from being solved. The main obstacle to its solution is that the equations at its basis are rare, and no approximation seems to be convincing. The interest of researchers in this area is very high, especially in the context of optical computing, because these studies may provide a significant input for new forms of computation. Now, numerical models based on CNN transmission lines can provide new and natural approaches to solve this long-standing problem.

Considering two transmission lines that intersect in a given cell  $(a, b)$ , called cross cell, as in Figure 4, the state of each cell, except for cross cell  $(a, b)$ , is governed by (4). Not only does the neighborhood of cross cell  $(a, b)$  consist of the two cells at its left and right, respectively but also it is made up of its upper and lower cells. As before, the neighboring cells, up and down, have a linear excitatory synaptic connection, whereas the other connection is nonlinear. Upper or lower cells have rather inhibitory synaptic connections.

From a circuitual point of view, this leads to a configuration similar to that in Figure 5.

All the cells of the two CNN transmission lines have (4) as spatial discretized state equation; just at the point of intersection between the two lines, cross cell  $(a, b)$  is ruled by the following spatial discretized state equation:

$$\begin{aligned} \dot{u}_{a,b} = & \frac{1}{2h^3} (u_{a-2,b} - u_{a+2,b} - 2u_{a-1,b} + 2u_{a+1,b}) \\ & - \frac{3}{2h} (u_{a+1,b}^2 - u_{a-1,b}^2) \\ & + \frac{1}{2h^3} (u_{a,b-2} - u_{a,b+2} - 2u_{a,b-1} + 2u_{a,b+1}) \\ & - \frac{3}{2h} (u_{a,b+1}^2 - u_{a,b-1}^2). \end{aligned} \quad (6)$$

This equation that governs the behavior of the cross cell could be obtained as done for (4) in [25], solving Kirchoff's law for the circuit in Figure 5 and finding the expression of current through the capacitor  $(a, b)$ . The state  $u(a, b)$  of the cross cell and the voltage across the capacitor is represented by  $R$ , the inhibitory synaptic connection,  $R'$ , the excitatory synaptic connection, and  $R''$ , the coefficient of the nonlinear term. Then the equation of the state of the cross cell  $(a, b)$  becomes

$$\begin{aligned} & \frac{u_{a-2,b} - u_{a,b}}{R} - \frac{u_{a-1,b} - u_{a,b}}{R'} + \frac{u_{a-1,b}^2 - u_{a,b}^2}{R''} \\ & + \frac{u_{a,b-2} - u_{a,b}}{R} - \frac{u_{a,b-1} - u_{a,b}}{R'} + \frac{u_{a,b-1}^2 - u_{a,b}^2}{R''} \\ = & \frac{u_{a+2,b} - u_{a,b}}{R} - \frac{u_{a+1,b} - u_{a,b}}{R'} + \frac{u_{a+1,b}^2 - u_{a,b}^2}{R''} \\ & + \frac{u_{a,b+2} - u_{a,b}}{R} - \frac{u_{a,b+1} - u_{a,b}}{R'} + \frac{u_{a,b+1}^2 - u_{a,b}^2}{R''} \\ & + C \frac{du_{a,b}}{dt} \end{aligned} \quad (7)$$

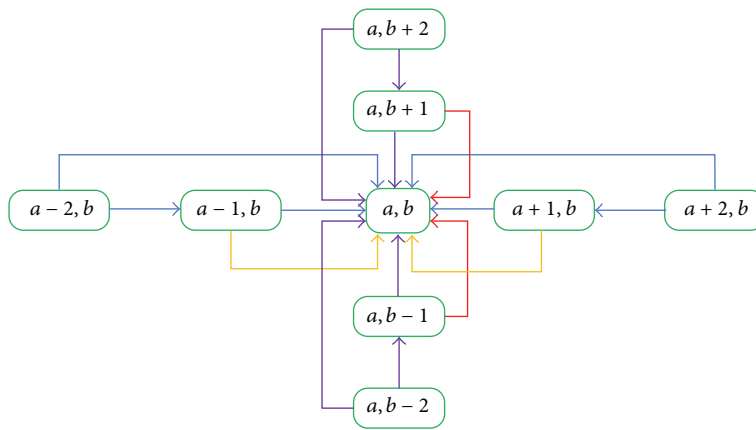


FIGURE 4: Block diagram representation of the crisscrossed lines structure at the point of intersection between the two lines (red and yellow lines for nonlinear interactions, blue and violet lines for linear interactions).

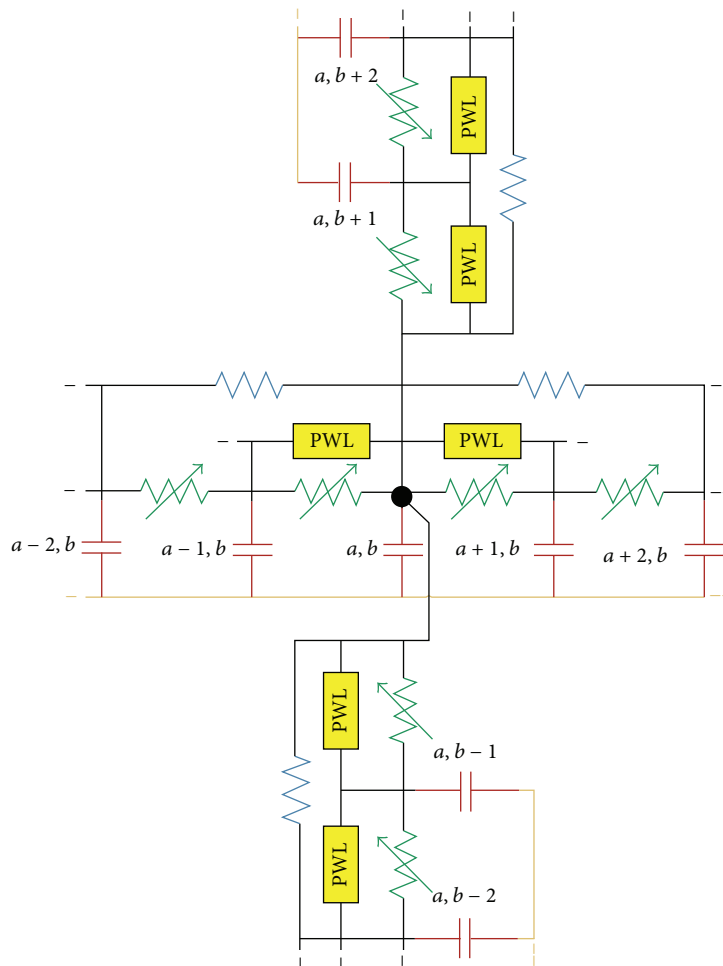


FIGURE 5: Circuitual representation of the crisscrossed lines structure at the point of intersection.

from which

$$C \frac{du_{a,b}}{dt} = \frac{u_{a+2,b} - u_{a-2,b}}{R} - \frac{u_{a+1,b} - u_{a-1,b}}{R'} + \frac{u_{a+1,b}^2 - u_{a-1,b}^2}{R''} + \frac{u_{a,b+2} - u_{a,b-2}}{R} - \frac{u_{a,b+1} - u_{a,b-1}}{R'} + \frac{u_{a,b+1}^2 - u_{a,b-1}^2}{R''} \quad (8)$$

taking  $R' = R/2$ ,  $RC = 2h^2$ , and  $R''C = 2h/3$ , where  $h = \text{cost}$ , and grouping together equation terms state equation (6) is obtained.  $h$  is a normalization parameter by which it is possible to tune the system model to improve convergence and stability.

The dynamics of cross cell is regulated by the state of eight neighboring cells, with radius 2. Neighboring cells, with radius 1, interact with the state of the cell in both linear and nonlinear ways.

Hence the soliton interaction is studied in two different scenarios: a ring line on which there are a single soliton propagating and a crisscross of two lines with a soliton on both lines. The motivation for the choice of these two topological setups sprang from facility for analysis and detection of soliton dynamics. In particular, the second one (relating to crisscross setup) is the best for analyzing the interaction between oblique solitons, very useful for highlighting typical dynamical behaviors of interaction, without any complications related to topology.

In order to simulate the presence of a soliton, we can use, as an initial condition, a function such as a Gaussian function or a square hyperbolic secant function with different magnitudes.

Considering a square hyperbolic secant function as initial condition:

$$s(x) = A \cdot \text{sech}^2 x. \quad (9)$$

This function avoids divergence integration problems, thanks to its zero-tangent envelope for  $x \rightarrow \pm\infty$ .

According to [7] if one considers the following initial condition:

$$u(x, 0) = p(p+1) \cdot \text{sech}^2 x \quad \text{with } p > 0, \quad (10)$$

the eigenvalues of Schrodinger equation spectrum to which every Korteweg de Vries solution is related are

$$\lambda_n = (p-n)^2 \quad (11)$$

with  $n = 0, 1, \dots, N$  (index of eigenvalue).

So, if  $p$  is an integer number, there will be the birth of  $p$  solitons from the initial state, otherwise there will be  $p$  solitons with a radiation tail coming after.

The magnitude of solitons is

$$A_n = 2\lambda_n \quad (12)$$

and the velocity of propagation is

$$c_n = 4\lambda_n = 2A_n. \quad (13)$$

As already introduced, amplitude, width, and velocity are connected. For example, amplitude is directly proportional to velocity of propagation. Higher amplitude corresponds to higher velocity of propagation.

In order to increase slightly the stability and accuracy of Forward Euler method which will be firstly validated using a custom MATLAB tool and then implemented on embedded platforms, a particular handling is done on discretized state equations.

Starting from (4) and sorting the equation terms,  $N$  single equations [39] are obtained from the KdV equation:

$$\frac{\partial u_i}{\partial t} = \frac{1}{2\Delta x^3} [u_{i-2} - u_{i+2} + 2(u_{i+1} - u_{i-1})] + \frac{3}{2\Delta x} [u_{i-1}^2 - u_{i+1}^2], \quad (14)$$

where  $i = 0, \dots, N$  is the space iteration index and  $\Delta x$  is the spatial step of the discrete grid. This numerical discretization of spatial derivative terms of (5) has been done using a space-centered finite difference method [40].

For the time derivative term of (5), just for the first iteration, we used a forward-time finite difference method as in [7, 41] because there is no preceding value at the first step of numerical integration process, obtaining (13). Hence, for the other iterations, we used a centered-time finite difference method, obtaining (14). We set  $K_{i1} = 1/2\Delta x^3$ ,  $K_{i2} = 3/2\Delta x$  and  $K_1 = 1/\Delta x^3$ ,  $K_2 = 3/\Delta x$ :

$$u_i^{k+1} = u_i^k + \Delta t \left\{ K_{i1} [u_{i-2}^k - u_{i+2}^k + 2(u_{i+1}^k - u_{i-1}^k)] + K_{i2} (u_{i-1}^{k2} - u_{i+1}^{k2}) \right\}, \quad (15)$$

$$u_i^{k+1} = u_i^{k-1} + \Delta t \left\{ K_1 [u_{i-2}^k - u_{i+2}^k + 2(u_{i+1}^k - u_{i-1}^k)] + K_2 [u_{i-1}^{k2} - u_{i+1}^{k2} + u_i^k (u_{i-1}^k - u_{i+1}^k)] \right\},$$

where  $k = 0, \dots, M$  are the time iteration index,  $i = 0, \dots, N$  are the spatial iteration index, and  $\Delta t$  is the integration time.

The same considerations about the spatial and time discretization methods have also been done for the crisscrossed lines scenario as well as the ring line scenario. Starting from (6) and sorting the equation terms, the spatial discretized state equation for the cross cell of intersection between the two lines is obtained:

$$\frac{\partial u_{a,b}}{\partial t} = \frac{1}{2\Delta x^3} [u_{a-2,b} - u_{a+2,b} + u_{a,b-2} - u_{a,b+2} + 2(u_{a+1,b} - u_{a-1,b} + u_{a,b+1} - u_{a,b-1})] + \frac{3}{2\Delta x} [u_{a-1,b}^2 - u_{a+1,b}^2 + u_{a,b-1}^2 - u_{a,b+1}^2], \quad (16)$$

where  $a, b = 0, \dots, N$  are the spatial iteration index for the two lines and  $\Delta x$  is the space step of the discrete grid. Equation (16) is the equation in which the forward-time finite

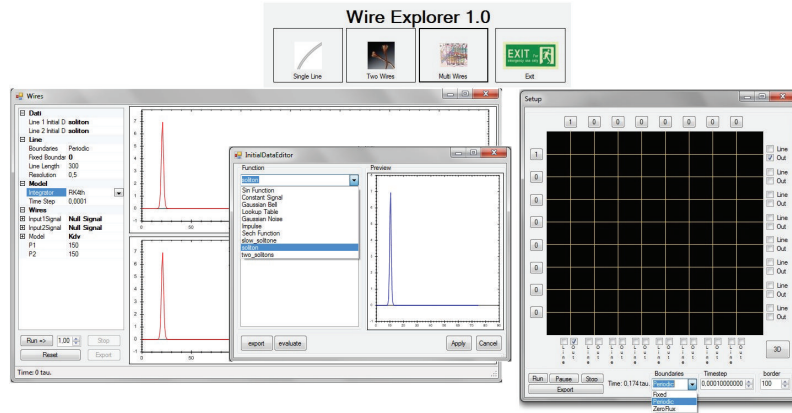


FIGURE 6: C++ SIMANSOL GUI tool for simulation and analysis.

difference method is applied while in (17) the centred-time difference method is used, for time discretization:

$$u_{a,b}^{k+1} = u_{a,b}^k + \Delta t \left\{ K_{i1} \left[ u_{a-2,b}^k - u_{a+2,b}^k + u_{a,b-2}^k - u_{a,b+2}^k + 2 \left( u_{a+1,b}^k - u_{a-1,b}^k + u_{a,b+1}^k - u_{a,b-1}^k \right) \right] + K_{i2} \left[ u_{a-1,b}^{k2} - u_{a+1,b}^{k2} + u_{a,b-1}^{k2} - u_{a,b+1}^{k2} \right] \right\}, \quad (17)$$

$$u_{a,b}^{k+1} = u_{a,b}^{k-1} + \Delta t \left\{ K_1 \left[ u_{a-2,b}^k - u_{a+2,b}^k + u_{a,b-2}^k - u_{a,b+2}^k + 2 \left( u_{a+1,b}^k - u_{a-1,b}^k + u_{a,b+1}^k - u_{a,b-1}^k \right) \right] + K_2 \left[ u_{a-1,b}^{k2} - u_{a+1,b}^{k2} + u_{a,b-1}^{k2} - u_{a,b+1}^{k2} + u_{a,b}^k \left( u_{a-1,b}^k - u_{a+1,b}^k + u_{a,b-1}^k - u_{a,b+1}^k \right) \right] \right\}, \quad (18)$$

where  $k = 0, \dots, M$  is the time iteration index,  $a, b = 0, \dots, N$  are the spatial iteration index for the two lines, and  $\Delta t$  is the integration time.

Using this combined approach a stable propagation of a soliton through all cells for all time cycles is obtained. This kind of discretization is less accurate than other types, but it is also the best technique in terms of implementation easiness and resources saving on embedded systems.

### 3. Simulation/Analysis Settings and Environment

In the literature, there are many numerical methods to solve state ODEs. In this work, two methods are used for two different aims: a 4th-order Runge-Kutta (RK4) method in order to conduct accurate high level analysis and a simpler Forward Euler (FE) method to be implemented, after a validation phase, on embedded platforms such as FPGAs, microcontrollers, DSP, or ASIC. FE method is chosen in order to relax the computing load of embedded platforms to the detriment of accuracy and stability.

With the goal of conducting extensive simulations and analysis, two Graphic User Interface (GUI) environments

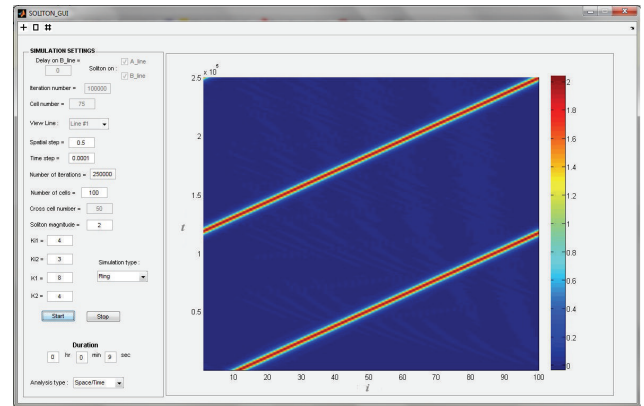


FIGURE 7: MATLAB SIMANSOL GUI tool for simulation and analysis.

have been designed (SIMULATION-ANalysis-SOLiton tools). The first, called C++ SIMANSOL, has been designed using C++ language (Figure 6); the latter, called ML SIMANSOL [42], instead has been designed in MATLAB environment (Figure 7). C++ SIMANSOL tool has been used mainly for fast and accurate analysis while ML-SIMANSOL tool has been used for validation and investigation in order to verify the efficiency, accuracy, and stability of FE method. MATLAB is preferred to other high level languages because of its optimal aptitude to handle matrices and vectors easily. Having these two high level environments allows both to compare their respective simulation results and to validate the ODE solving process. A comparison between Forward Euler method and Runge-Kutta method is not to show an evident difference between these two methods trivially but just for verifying if the stability and accuracy of Forward Euler method are acceptable using a limited number of CNN cells with respect to Runge-Kutta method. The impossibility of implementing on FPGA one 1D CNN, formed by a huge number of computing processors (one for each cell), causes problems of equation solution instability and divergence. So, it was very important to understand the minimum number of processors to implement on FPGA in order to minimize

TABLE 1: Ring line simulation settings.

ML-SIMANSOL	1st test	2nd test	3rd test
Number of cells ( $N$ )	100	400	400
Amplitude ( $A$ )	2	6	12
Time step ( $\Delta t$ )	0.01 s	0.00001 s	0.00001 s
Spatial step ( $\Delta x$ )	0.5	0.05	0.05
Number of iterations	10000	400000	400000

these problems. In addition, implementing a Forward Euler method for numerical integration permits saving a lot of FPGA resources guaranteeing the possibility to implement larger and larger Cellular Neural Network on the FPGA [26], allowing facing more complex dynamics problems.

In the C++ SIMANSOL tool (Figure 6), there are three different windows: initial input function window, simulation setup window, and multiwires grid setup window. It is possible to control all simulation parameters such as integration method (Euler, RK4th, RK2th, RK45, and RKAdaptive), boundaries (periodic, zero flux, and fixed), number of cells, number of iterations, initial input function (Gaussian, hyperbolic secant, sin, soliton, impulse, etc.), time step, spatial step, and simulation scenario (single line, crisscrossed lines, and grid network).

Instead MATLAB SIMANSOL tool (Figure 7) consists of only one main window from which it is possible to control several simulation parameters such as number of cells, number of iterations, soliton amplitude, time step, spatial step, analysis type (time, spatial or time/spatial), simulation scenario (ring line or crisscross lines), and plot simulation results. This tool, as already said, implements the FE integration method.

Using these tools it is possible to give a complete vision and to understand complex dynamics phenomena. Three types of graphs can be displayed: a state cell variable versus time graph, in which the time evolution of the state variable associated with a certain cell of the ring line is shown; a state cell variable versus space graph, where the value of all state cell variable associated with all cells is displayed for a well-defined time step; a time/space graph, in which the time and space evolution of the state cell variable of all cells is highlighted. This last graph is very interesting because it allows getting a global vision about the propagation of solitons and their interaction.

## 4. Simulation Setup and Results

**4.1. Ring Line Analysis.** In the first scenario, a ring line of 100–400 cells is considered according to what was said previously. The ring-fashion feature is obtained simply by imposed periodic boundary conditions. Three MATLAB simulations using the MATLAB (ML) SIMANSOL tool have been conducted under the following conditions (Table 1).

**4.1.1. 1st Test: Soliton with Magnitude 2.** In Figure 8 the state variable of 10th cell in function of time (number of iterations) is shown. A single soliton with magnitude 2 propagates

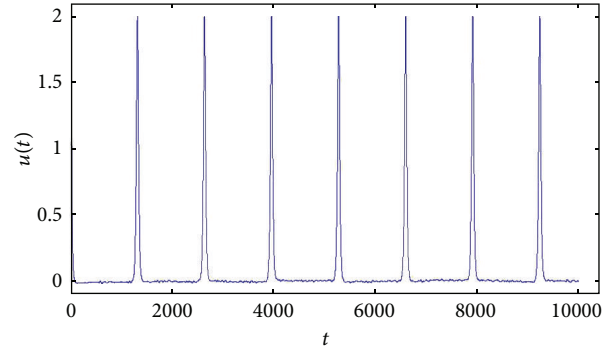


FIGURE 8: A single soliton with magnitude 2 which propagates through the ring line in correspondence with the 10th cell.

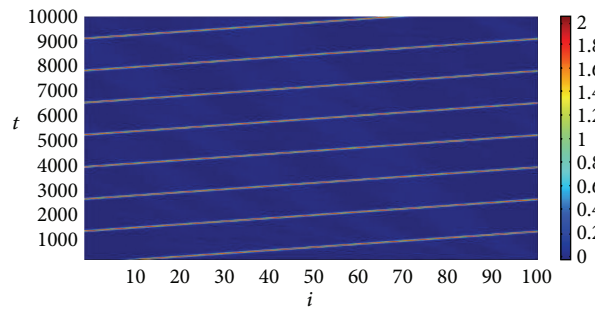


FIGURE 9: A Space/Time graph in which a single soliton with magnitude 2 propagates through the ring line.

through the ring line performing several loops. This soliton has a velocity of propagation of 4 according to (9), completing a single loop in about 1250 iterations or rather 12.5 s. There are not any numerical instability or divergence problems during simulations. This simulation allows verifying the quality of Forward Euler method to be used for implementation of soliton ring line on embedded devices [26, 33].

As shown in Figure 9, the narrow parallel lines with slope of 4 represent the single soliton which performs loops through the ring.

As already explained in Section 2, with an amplitude of 2, according to [7], just a single soliton is foreseen.

**4.1.2. 2nd Test: Soliton with Magnitude 6.** This time the state cell variable of 200th cell is analysed (Figure 10). The soliton now having an amplitude of 6 generates two solitons. Hence at the 150000th iteration, it is possible to see the interaction between the two solitons with magnitudes 8 and 2, respectively. These two solitons come from a single soliton with initial magnitude 6 which propagates through the ring line. They have a velocity of propagation of 16 and 4, respectively. This interaction is possible because of different velocity of two solitons.

From Figure 11 it is possible to understand the velocity of solitons on the basis of straight lines. The narrow parallel lines with slope of 16 represent the faster soliton with magnitude of 8 while the large line with slope of 4 represents the soliton with magnitude of 2. In this graph, the soliton interaction

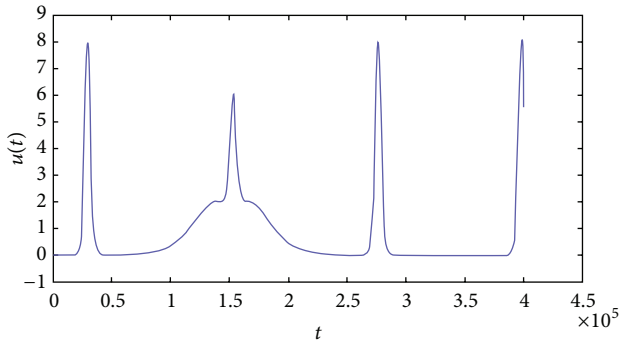


FIGURE 10: The two solitons which propagate and interact through the ring line in correspondence with the 200th cell.

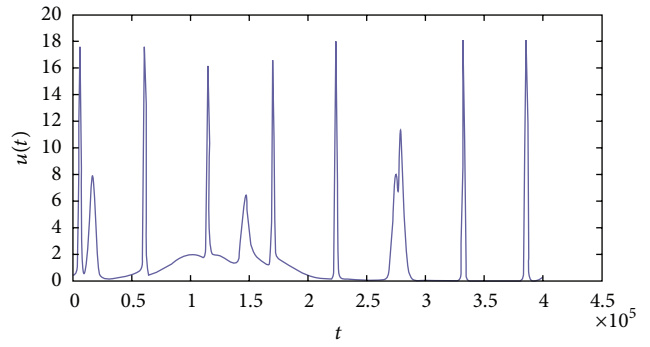


FIGURE 13: The three solitons which propagate and interact through the ring line in correspondence with the 150th cell.

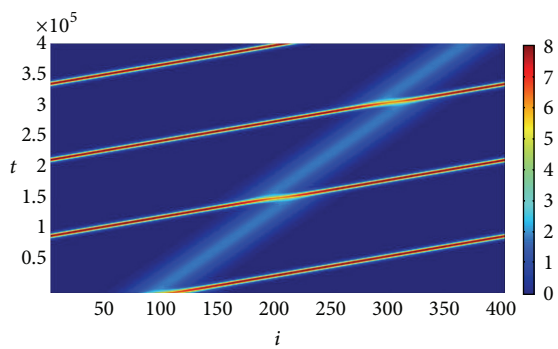


FIGURE 11: A Space/Time graph in which a single soliton with magnitude 6 propagates through the ring line and is divided into two solitons.

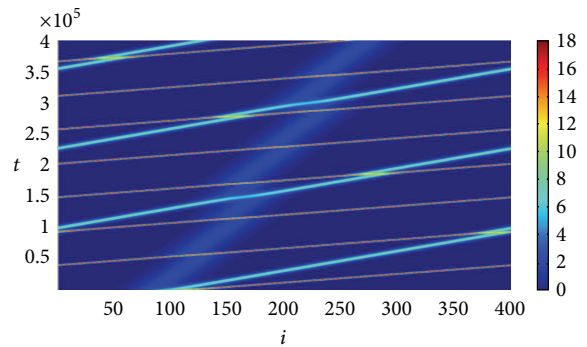


FIGURE 14: A Space/Time graph in which a single soliton with magnitude 12 propagates through the ring line and is divided into three solitons.

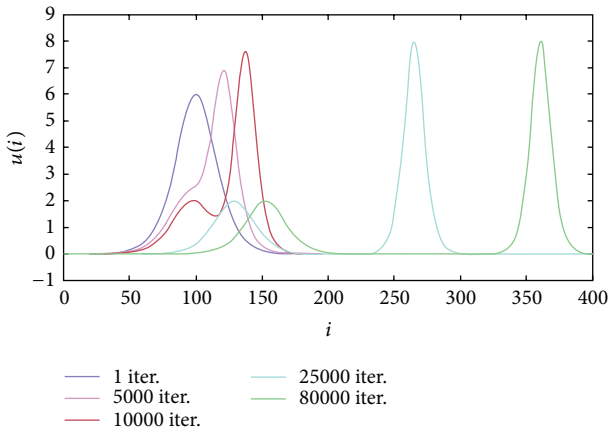


FIGURE 12: The generation and propagation through the ring line of two solitons from a single soliton with magnitude 6.

points are evident. When there is an interaction between solitons, it is possible to see a phase shift that is shown in the graph as a shift of lines. The slow soliton has a phase shift larger than fast soliton, and the direction of phase shift is positive for slow soliton.

It is also interesting to see in Figure 12 the generation and propagation through the ring line of the two solitons coming

from a single soliton (blue) with magnitude 6. It is possible to see the soliton generation at five different time iterations.

**4.1.3. 3th Test: Soliton with Magnitude 12.** In this case, following the time evolution of state cell variable of 150th cell more interactions between solitons can be seen (Figure 13). At the 110000th iteration, there is an interaction between the soliton with magnitude 18 and the soliton with magnitude 2; at the 150000th iteration, it is possible to see the interaction between the two solitons with magnitudes 8 and 2, respectively, while at the 270000th iteration we find an interaction between solitons with magnitudes 18 and 8, respectively. These three solitons come from a single soliton with initial magnitude 12 which propagates through the ring line. These solitons have a velocity of propagation of 36, 16, and 4, respectively.

Also in this case from Figure 14 it is interesting to see the different velocity of solitons and the soliton interaction points. The narrow parallel lines with slope of 36 represent the faster soliton with magnitude of 18 while the other lines with slope of 16 and 4 represent the solitons with magnitude of 2 and 8, respectively. In this graph, the soliton interaction points are evident. When there is an interaction between solitons, it is possible to see a phase shift that is shown in the graph as a shift of lines as in the other cases seen previously.

From Figure 15 the generation and propagation through the ring line of three solitons coming from a single soliton



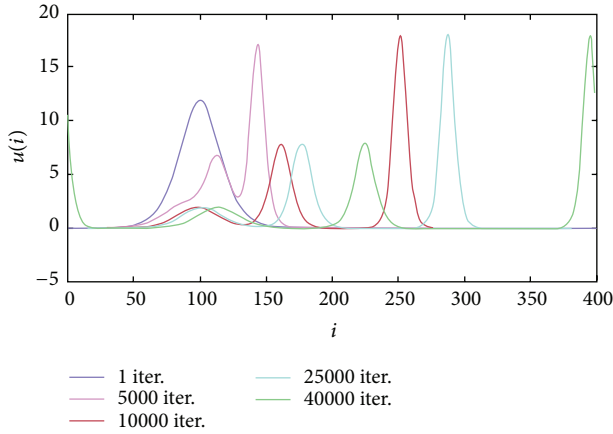


FIGURE 15: The generation and propagation through the ring line of three solitons from a single soliton with magnitude 12.

TABLE 2: Simulation settings: crisscross scenario.

C++ SIMANSOL	1st test	2nd test	3rd test
Line <i>a</i> input	Soliton	None	Soliton
Line <i>b</i> input	Soliton	Soliton	Soliton (delayed)
Time step ( $\Delta t$ )	0.0001 s	0.0001 s	0.0001 s
Spatial step ( $\Delta x$ )	0.5	0.5	0.5
Number of cells on <i>a/b</i> lines	300	300	300
Soliton magn. ( <i>A</i> )	1	1	1
ML-SIMANSOL	1st test	2nd test	3rd test
Line <i>a</i> input	Soliton	Soliton	Soliton
Line <i>b</i> input	Soliton	None	Soliton (delayed)
Time step ( $\Delta t$ )	0.0001 s	0.0001 s	0.0001 s
Spatial step ( $\Delta x$ )	0.5	0.5	0.5
Number of cells on <i>a/b</i> lines	100	100	100
Soliton magn. ( <i>A</i> )	2	2	2

(blue) with magnitude 12 are shown. It is possible to see the soliton evolution at five different time iterations as in the previous cases.

**4.2. Crisscross Lines Analysis.** In this section some simulations of crisscross transmission lines are shown. Also for these simulations the SIMANSOL tools are used. In both tools periodic boundary conditions are chosen, building a sort of crossed rings. Three different types of test are conducted. The initial conditions for these simulations are summarized in Table 2.

With the C++ SIMANSOL tool, solitons that propagate along the lines before and after the crossing point are shown on the graph while using ML-SIMANSOL tool the Space/Time graphs for lines *a* and *b* are plotted.

**4.2.1. 1st Test: One Soliton on Both Lines *a* and *b*.** As it can be seen in Figures 16 and 17, two solitons belonging to different lines meet in the crossing point and separate again,

propagating on their own line as if this interaction never happened. After the interaction, they preserve their distinctiveness. The only observed phenomenon is the presence of a small dispersive track and reflected solitons, linked to the periodic boundary conditions, rather than to the interaction between the two solitons.

**4.2.2. 2nd Test: Just One Soliton on Line *a*.** The second experiment involves the presence of a single soliton passing through the points of intersection between the two lines. When the soliton reaches the point of intersection, it slows down, while on the two lines disturbances occur. After crossing the interaction point, two new solitons emerge, one for each line; it is possible to see two wave trains as well, one for each line, but traveling in the opposite direction compared to solitons, as shown in Figures 18 and 19.

There is also a decreasing of magnitude of the two solitons; on line *a*, after the crossing point, soliton has magnitudes 0.6 on line *a* and 0.5 on line *b* for simulation with C++ SIMANSOL tool, while 1.3 and 0.5, respectively, on average for simulation with ML-SIMANSOL tool. The larger soliton is the one that resides on the same line as the one of the incoming soliton. In addition, it is possible to see phenomena of dispersive tracks and reflected solitons too.

**4.2.3. 3th Test: One Soliton on Line *a* and One Delayed Soliton on Line *b*.** The third experiment concerns the interaction between two solitons of the same amplitude but phase-shifted. The first soliton reaches the intersection cell before the other one and undergoes a process of division, similar to that described in the previous experiment. When the second soliton reaches the point of intersection, it moves in an unstable environment. The second soliton undergoes a process of division and, at the end of the interaction, a total of 4 new solitons emerge. Figures 20 and 21 show the second soliton while going through the intersection cell. Again, we note that there is no symmetry between the two pairs of solitons. On the line of the soliton, which first crosses the intersection cell, an emerging pair is generated, which has a width slightly larger than the second pair.

The generation of two small solitons and dispersive tails on both lines is also evident.

The results that emerge from this series of experiments are extremely interesting. Using both simulation tools, it is possible to see the same results, even if there are numerical mismatches because of the two different integration methods. They show that the interaction of oblique solitons is very rich and diverse. In Table 3, the results of the experiments are shown.

In the first case, two identical solitons interact and emerge unchanged after crossing. In the second case, a soliton is split into two smaller and different solitons. In the third case, two identical but phase-shifted solitons splits into four solitons, slightly different from each other.

This kind of results, even if it is still under research, could give, also, some ideas about a possible utilization of these CNN transmission lines in terms of digital logic gates, making the most of their nonlinear and innovative features.

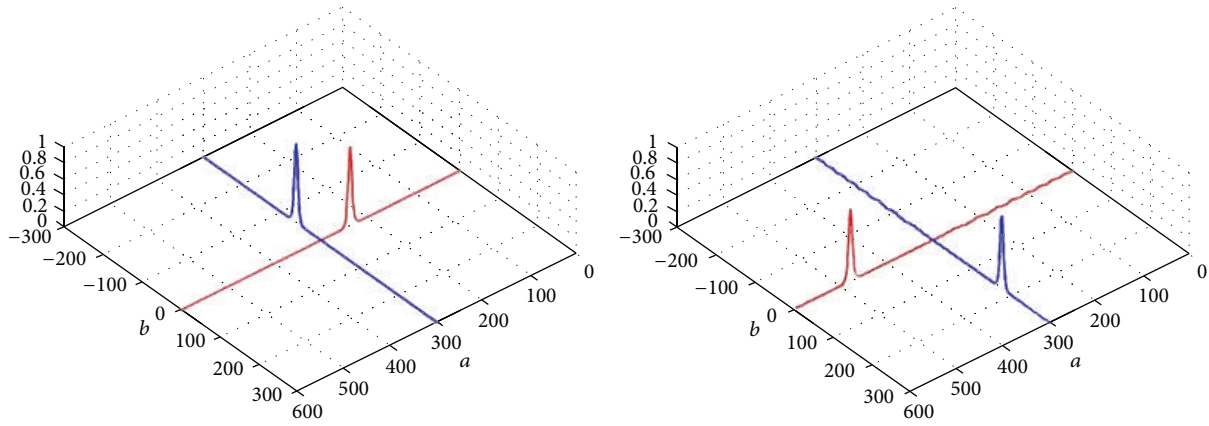


FIGURE 16: Simulation with C++ SIMANSOL tool: two identical solitons meet and cross with no interaction.

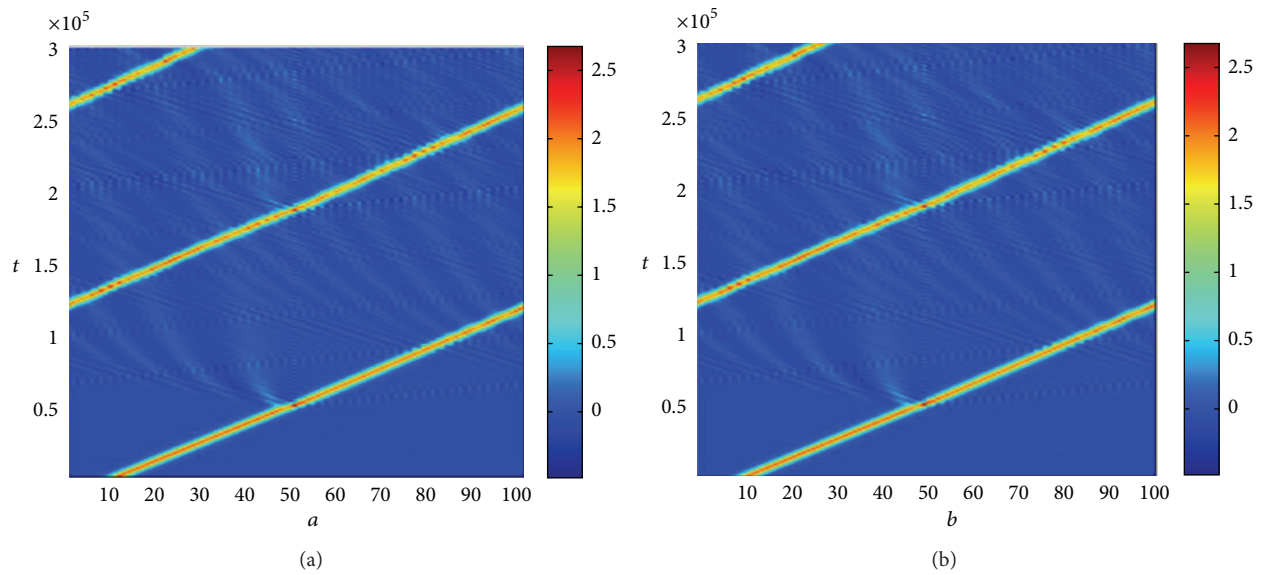


FIGURE 17: Simulation with ML-SIMANSOL tool: Space/Time graphs for line  $a$  (left) and line  $b$  (right). One soliton on each line.

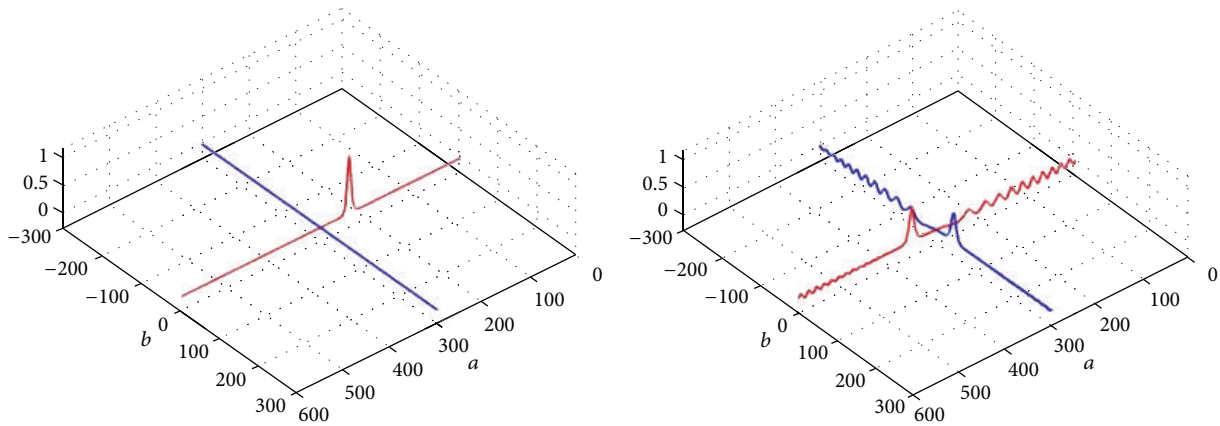


FIGURE 18: Simulation with C++ SIMANSOL tool: a soliton, crossing an intersection point between two lines and divides itself into two smaller solitons.

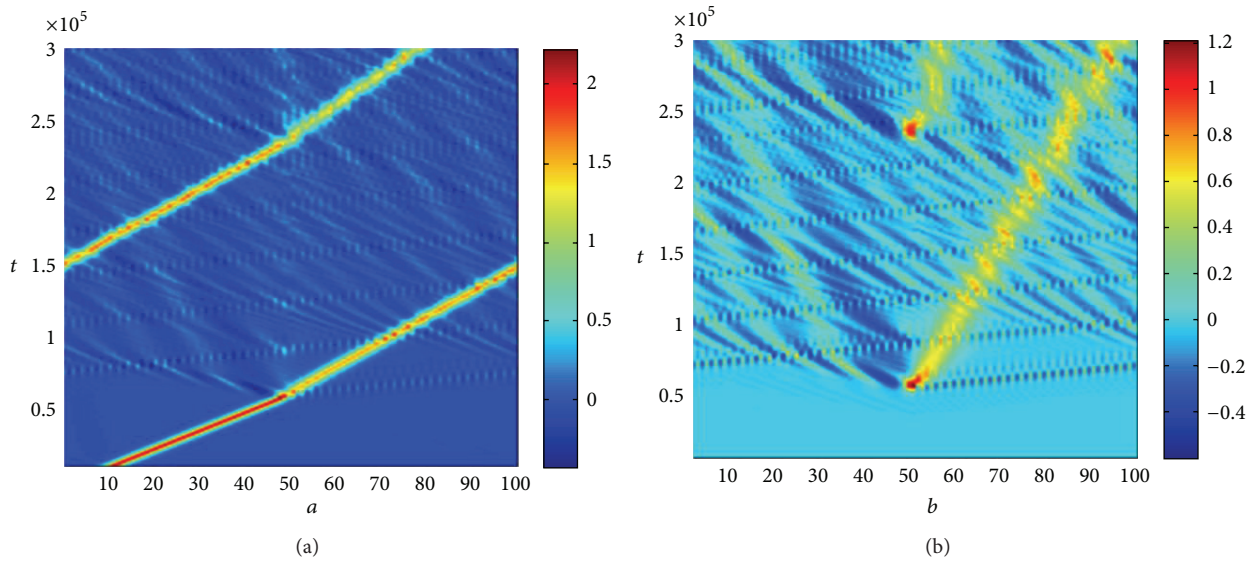


FIGURE 19: Simulation with ML-SIMANSOL tool: Space/Time graphs for line  $a$  (left) and line  $b$  (right). Only one soliton on line  $a$ .

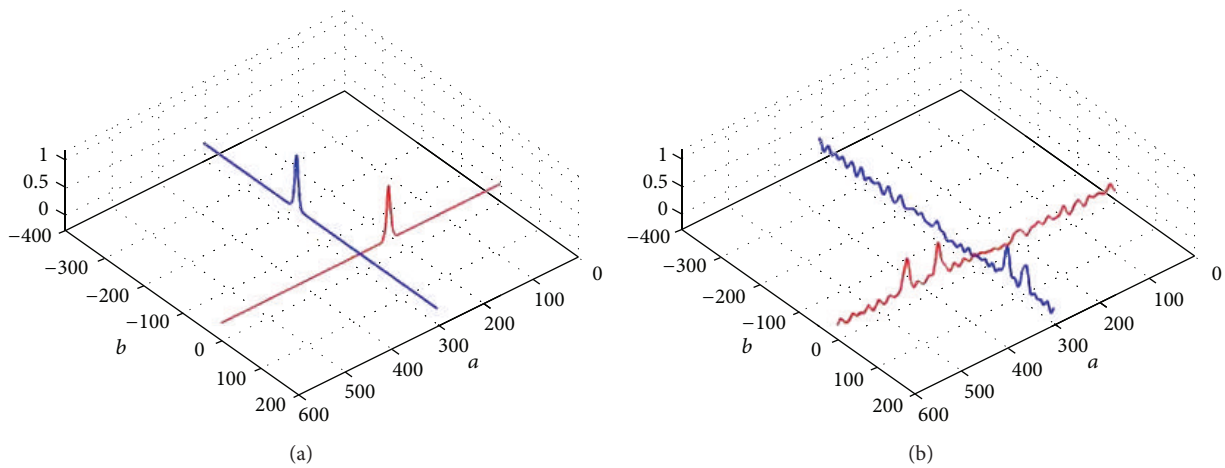


FIGURE 20: Simulation with C++ SIMANSOL tool: solitons (one on line  $b$  with a negative delay) before and after they crossed the intersection cell.

TABLE 3: Results table: crisscross scenario.

C++ SIMANSOL	1st test	2nd test	3rd test
Line $a$ input	Soliton ( $A = 1$ )	None	Soliton ( $A = 1$ )
Line $b$ input	Soliton ( $A = 1$ )	Soliton ( $A = 1$ )	Soliton (delayed)
Line $a$ output	Soliton ( $A = 1$ )	Soliton ( $A = 0.5$ )	Two solitons ( $A = 0.4, 0.3$ )
Line $b$ output	Soliton ( $A = 1$ )	Soliton ( $A = 0.6$ )	Two solitons ( $A = 0.4, 0.3$ )
ML-SIMANSOL	1st test	2nd test	3rd test
Line $a$ input	Soliton ( $A = 2$ )	Soliton ( $A = 2$ )	Soliton ( $A = 2$ )
Line $b$ input	Soliton ( $A = 2$ )	None	Soliton (delayed)
Line $a$ output	Soliton ( $A = 2$ )	Soliton ( $A = 1.3$ )	Two solitons ( $A = 1.4, 0.7$ )
Line $b$ output	Soliton ( $A = 2$ )	Soliton ( $A = 0.5$ )	Two solitons ( $A = 1.4, 0.7$ )

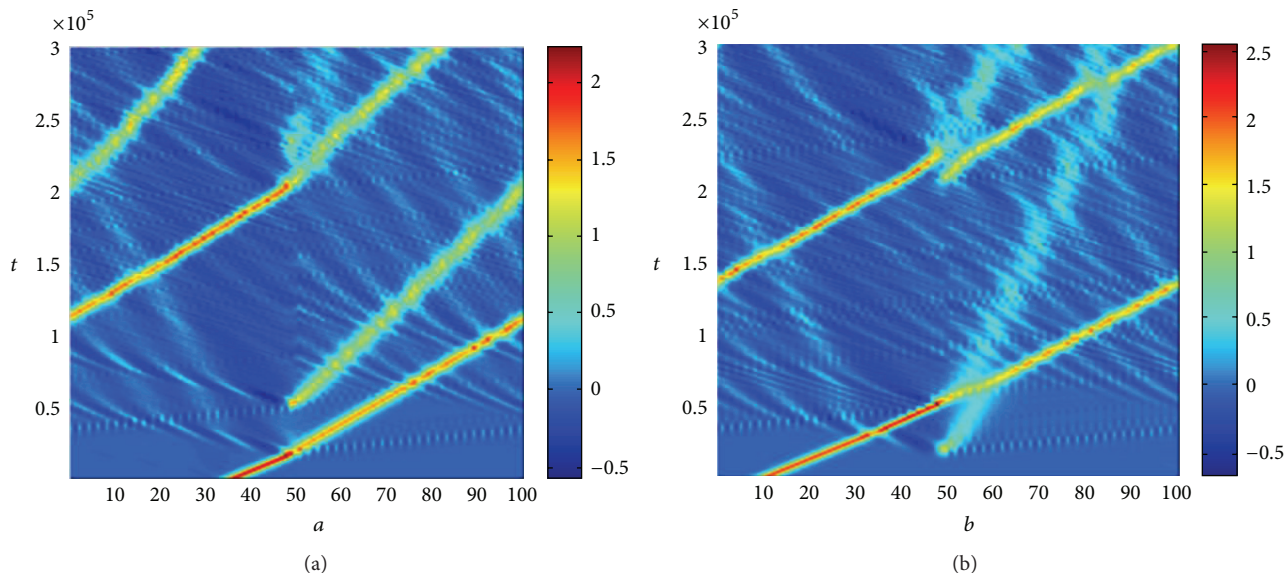


FIGURE 21: Simulation with ML-SIMANSOL tool: Space/Time graphs for line  $a$  (left) and line  $b$  (right). One soliton on line  $a$  and one soliton with a negative spatial delay of 25 cells on line  $b$ .

## 5. Conclusions

In this work, a new approach to the transmission lines based on nonlinear CNN is presented. After providing the basic equation that governs the status of each cell, the presence and the interaction of solitary waves have been detected. These waves interact with each other, while maintaining their distinctiveness. This behavior is not surprising, because in the case where the network is continuous the equation of state just becomes the well-known KdV equation. The introduction of nonhomogeneous terms provides a new approach to the study of wave propagation in multidimensional nonlinear and dispersive media. Furthermore, this approach allows dealing very effectively with the problems related to interaction between oblique solitons. Several simulations in two different scenarios (ring line and crisscross lines) have been conducted with the help of two custom simulation tools. The possibility to implement a simple Forward Euler method for solving KdV ODE on embedded systems has been validated. The results have great potential applications, especially in the construction of information transmission lines, embodied into other artificial intelligent media, such as automatic expert systems and robotic agents. There are many advantages using CNNs for designing transmission lines. CNNs can be easily implemented in silicon performing multiple processes in parallel, at very high speed. They constitute a very flexible simulation environment by which we investigate nonlinear dynamics and carry out a huge number of trials with different initial data. Using traditional systems, the direct observation of solitons behavior could be very complicated. Such an approach may guarantee a technological environment for developing new and unconventional forms of computation, which exploit fully nonlinear dynamics, opening interesting perspectives for the study of multidimensional solitons, with reference to optical computing applications. In addition, this

innovative approach allows simulating neuromorphic systems, creating new simulation hardware for the development of brain-like physical agents.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

- [1] R. Hirota and K. Suzuki, "Theoretical and experimental studies of lattice solitons in nonlinear lumped networks," *Proceedings of the IEEE*, vol. 61, no. 10, pp. 1483–1491, 1973.
- [2] M. Toda, *Theory of Nonlinear Lattices*, Original Japanese Edition, Springer, Berlin, Germany, 1978.
- [3] K. Fukushima, M. Wadati, T. Kotera, K. Sawada, and Y. Narahara, "Experimental and theoretical study of the recurrence phenomena in nonlinear transmission line," *Journal of the Physical Society of Japan*, vol. 48, no. 3, pp. 1029–1035, 1980.
- [4] S. Giambò, P. Pantano, and P. Tucci, "An electrical model for Korteweg-de Vries equation," *American Journal of Physics*, vol. 52, pp. 238–243, 1984.
- [5] D. J. Korteweg and G. de Vries, "XLI. On the change of form of long waves advancing in a rectangular canal, and on a new type of long stationary waves," *Philosophical Magazine Series 5*, vol. 39, no. 240, pp. 422–443, 1895.
- [6] A. M. Anile, J. Hunter, P. Pantano, and G. Russo, *Ray Methods for Nonlinear Waves in Fluids and Plasmas*, Longman Scientific and Technics, 1993.
- [7] N. J. Zabusky and M. D. Kruskal, "Interaction of 'solitons' in a collisionless plasma and the recurrence of initial states," *Physical Review Letters*, vol. 15, no. 6, pp. 240–243, 1965.

- [8] F. Nijhoff and H. Capel, "The discrete Korteweg-de Vries equation," *Acta Applicandae Mathematicae*, vol. 39, no. 1–3, pp. 133–158, 1995.
- [9] C. S. Gardner, J. M. Greene, M. D. Kruskal, and R. M. Miura, "Method for solving the Korteweg-deVries equation," *Physical Review Letters*, vol. 19, no. 19, pp. 1095–1097, 1967.
- [10] C. S. Gardner, J. M. Greene, M. D. Kruskal, and R. M. Miura, "Korteweg-devries equation and generalizations. VI. methods for exact solution," *Communications on Pure and Applied Mathematics*, vol. 27, no. 1, pp. 97–133, 1974.
- [11] Z. Qiao and J. Li, "Negative-order KdV equation with both solitons and kink wave solutions," *Europhysics Letters*, vol. 94, no. 5, Article ID 50003, 2011.
- [12] Z. Qiao and E. Fan, "Negative-order Korteweg-de Vries equations," *Physical Review E*, vol. 86, no. 1, Article ID 016601, 2012.
- [13] B. Feng and Y. Zhang, "Two kinds of new integrable couplings of the negative-order Korteweg-de Vries equation," *Advances in Mathematical Physics*, vol. 2015, Article ID 154915, 8 pages, 2015.
- [14] J. Kolosick, D. L. Landt, H. C. S. Hsuan, and K. E. Lonngren, "Experimental study of solitary waves in a nonlinear transmission line," *Applied Physics*, vol. 2, no. 3, pp. 129–131, 1973.
- [15] R. H. Freeman and A. E. Karbowiak, "An investigation of nonlinear transmission lines and shock waves," *Journal of Physics D: Applied Physics*, vol. 10, no. 5, pp. 633–643, 1977.
- [16] W.-S. Duan, "Nonlinear waves propagating in the electrical transmission line," *Europhysics Letters*, vol. 66, no. 2, pp. 192–197, 2004.
- [17] L. Guardado, F. Beníte, E. Kaikina, F. Ruiz, and M. Hernández, "An asymptotic solution to non-linear transmission lines," *Nonlinear Analysis: Real World Applications*, vol. 8, no. 3, pp. 715–724, 2007.
- [18] F. I. I. Ndzana, A. Mohamadou, T. C. Kofané, and L. Q. English, "Modulated waves and pattern formation in coupled discrete nonlinear LC transmission lines," *Physical Review E*, vol. 78, Article ID 016606, pp. 1–13, 2008.
- [19] S. I. Mostafa, "Analytical study for the ability of nonlinear transmission lines to generate solitons," *Chaos, Solitons & Fractals*, vol. 39, no. 5, pp. 2125–2132, 2009.
- [20] E. Kengne and R. Vaillancourt, "Propagation of solitary waves on lossy nonlinear transmission lines," *International Journal of Modern Physics B*, vol. 23, no. 1, pp. 1–18, 2009.
- [21] D. S. Ricketts, X. Li, and D. Ham, "Electrical soliton oscillator," *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 1, pp. 373–382, 2006.
- [22] T. H. Lee, "Device physics: electrical solitons come of age," *Nature*, vol. 440, no. 7080, pp. 36–37, 2006.
- [23] T. Roska, L. O. Chua, D. Wolf, T. Kozek, R. Tetzlaff, and F. Puffer, "Simulating nonlinear waves and partial differential equations via CNN. I. Basic techniques," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 42, no. 10, pp. 807–815, 1995.
- [24] L. Fortuna, A. Rizzo, and M. G. Xibilia, "Modeling complex dynamics via extended PWL-based CNNs," *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, vol. 13, no. 11, pp. 3273–3286, 2003.
- [25] E. Bilotta and P. Pantano, "Cellular nonlinear networks meet KdV equation: a newparadigm," *International Journal of Bifurcation and Chaos*, vol. 23, no. 1, Article ID 1330003, 2013.
- [26] G. Borgese, C. Pace, P. Pantano, and E. Bilotta, "FPGA-based distributed computing microarchitecture for complex physical dynamics investigation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 9, pp. 1390–1399, 2013.
- [27] L. O. Chua and L. Yang, "Cellular neural networks: theory," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 10, pp. 1257–1272, 1988.
- [28] L. O. Chua, *A Paradigm for Complexity*, World Scientific Series on Nonlinear Science, World Scientific, Singapore, 1996.
- [29] L. O. Chua and T. Roska, *Cellular Neural Networks and Visual Computing: Foundations and Applications*, Cambridge University Press, Cambridge, UK, 2004.
- [30] E. Bilotta, P. Pantano, and S. Vena, "Artificial micro-worlds. Part I: a new approach for studying life-like phenomena," *International Journal of Bifurcation and Chaos*, vol. 21, no. 2, pp. 373–398, 2011.
- [31] E. Bilotta and P. Pantano, "Artificial micro-worlds part II: cellular automata growth dynamics," *International Journal of Bifurcation and Chaos*, vol. 21, no. 3, pp. 619–645, 2011.
- [32] A. Cerasa, E. Bilotta, A. Augimeri et al., "A cellular neural network methodology for the automated segmentation of multiple sclerosis lesions," *Journal of Neuroscience Methods*, vol. 203, no. 1, pp. 193–199, 2012.
- [33] G. Borgese, C. Pace, P. Pantano, and E. Bilotta, "Reconfigurable implementation of a CNN-UM platform for fast dynamical systems simulation," in *Applications in Electronics Pervading Industry, Environment and Society*, vol. 289 of *Lecture Notes in Electrical Engineering*, Springer, Berlin, Germany, 2014.
- [34] F. Kako and N. Yajima, "Interaction of ion-acoustic solitons in two-dimensional space," *Journal of the Physical Society of Japan*, vol. 49, no. 5, pp. 2063–2071, 1980.
- [35] J. N. Dinkel, C. Setzer, S. Rawal, and K. E. Lonngren, "Soliton propagation and interaction on a two-dimensional nonlinear transmission line," *Chaos, Solitons and Fractals*, vol. 12, no. 1, pp. 91–96, 2001.
- [36] A. A. Alexeyev, "A multidimensional superposition principle: classical solitons," *Physics Letters. A*, vol. 335, no. 2–3, pp. 197–206, 2005.
- [37] T. Soomere, "Solitons interactions," in *Encyclopedia of Complexity and Systems Science*, pp. 8479–8505, Springer, New York, NY, USA, 2009.
- [38] B. A. Malomed, D. Mihalache, F. Wise, and L. Torner, "Spatiotemporal optical solitons," *Journal of Optics B: Quantum and Semiclassical Optics*, vol. 7, no. 5, pp. R53–R72, 2005.
- [39] L. Fortuna, M. Frasca, and A. Rizzo, "Generating solitons in lattices of nonlinear circuits," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '01)*, vol. 2, pp. 680–683, Sydney, Australia, May 2001.
- [40] A. C. Vliegthart, "On finite-difference methods for the Korteweg-de Vries equation," *Journal of Engineering Mathematics*, vol. 5, no. 2, pp. 137–155, 1971.
- [41] R. Courant, K. Friedrichs, and H. Lewy, "On the partial difference equations of mathematical physics," *IBM Journal of Research and Development*, vol. 11, pp. 215–234, 1967.
- [42] G. Borgese, "MATLAB SIMANSOL GUI," [http://esg.unical.it/share/MSIMANSOL\\_GUI.rar](http://esg.unical.it/share/MSIMANSOL_GUI.rar).



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

