# Modeling a Nonlinear Liquid Level System by Cellular Neural Networks

**Article** *in* International Journal of Modern Physics C · April 2010

DOI: 10.1142/S0129183110015245



Norberto Hernández Romero
Autonomous University of Hidalgo


Juan Carlos Seck Tuoh Mora
Autonomous University of Hidalgo


M. Gonzalez-Hernandez
Autonomous University of Hidalgo


Joselito Medina
Autonomous University of Hidalgo

International Journal of Modern Physics C
Vol. 21, No. 4 (2010) 489–501

DOI: 10.1142/S0129183110015245





# MODELING A NONLINEAR LIQUID LEVEL SYSTEM BY CELLULAR NEURAL NETWORKS

NORBERTO HERNANDEZ-ROMERO*, JUAN CARLOS SECK-TUOH-MORA†, MANUEL GONZALEZ-HERNANDEZ‡ and JOSELITO MEDINA-MARIN§

*Centro de Investigación Avanzada en Ingeniería Industrial*
*Universidad Autónoma del Estado de Hidalgo*
*Pachuca Hidalgo 42184, México*
**nhromero@uaeh.edu.mx*
*†jseck@uaeh.edu.mx*
*†juanseck@gmail.com*
*‡mghdez@uaeh.edu.mx*
*§jmedina@uaeh.edu.mx*

JUAN JOSE FLORES-ROMERO

*Facultad de Ingeniería Eléctrica*
*Universidad Michoacana de San Nicolás de Hidalgo*
*Morelia, Michoacán 58190, México*
*juanf@umich.mx*

Received 4 December 2009
Accepted 15 February 2010

This paper presents the analogue simulation of a nonlinear liquid level system composed by two tanks; the system is controlled using the methodology of exact linearization via state feedback by cellular neural networks (CNNs). The relevance of this manuscript is to show how a block diagram representing the analogue modeling and control of a nonlinear dynamical system, can be implemented and regulated by CNNs, whose cells may contain numerical values or arithmetic and control operations. In this way the dynamical system is modeled by a set of local-interacting elements without need of a central supervisor.

*Keywords*: Cellular neural networks; nonlinear systems; block diagrams.

PACS Nos.: 05.10.-a, 07.05.Tp, 87.16.-A, 87.18.Sn.

## 1. Introduction

In general, it is difficult and sometimes impossible to find an analytical (or even a numerical) solution for nonlinear systems modeled by differential equations in order to predict their dynamical behavior; moreover, this one often has a more complicated form when it is desired to include a nonlinear control. This situation yields the study of unconventional paradigms such as analogue computation for solving this kind of problems. This paper presents the application of cellular neural networks

(CNNs) for simulating a block diagram representing the analogue computation of a nonlinear liquid level system, including a closed-loop feedback to achieve a complete control of the system.

CNNs are hybrid models between artificial neural networks and cellular automata with continuous values, specified by a parallel interaction of cells where each has a simple behavior conditioned by the state of the neighboring cells. They were firstly conceived by Chua for image processing and pattern recognition.[1,2] CNNs have been widely investigated because they can produce complex behaviors in a natural way and have been used in an extensive number of applications such as signal processing,[3] simulation and control of chaotic systems,[4,5] circuit design,[6] robotic navigation control,[7] and modeling brain activity,[8] among others.

CNNs are appropriate for achieving analogue computation because they can easily simulate the behavior of block diagrams, which are tools classically employed in every engineering field for representing real dynamical systems by means of an analogue language. The original part of this paper is the implementation of a CNN for simulating block diagrams performing the analogue computation of a nonlinear liquid level system composed by two non-coupled tanks. This system is described by two nonlinear differential equations whose dynamics is regulated by a feedback control applying the exact linearization technique.[9,10] This paper represents an advance from previous results in Ref. 11 about modeling linear systems.

The rest of the paper is organized as follows: Section 2 presents the analogue modeling and control of the liquid level system by block diagrams. Section 3 explains the implementation, simulation and control of the liquid level system by CNNs. The final section gives the concluding remarks of the paper.

## 2. Modeling and Control of a Liquid Level System

The system examined is composed by two non-coupled tanks (see Fig. 1) where:

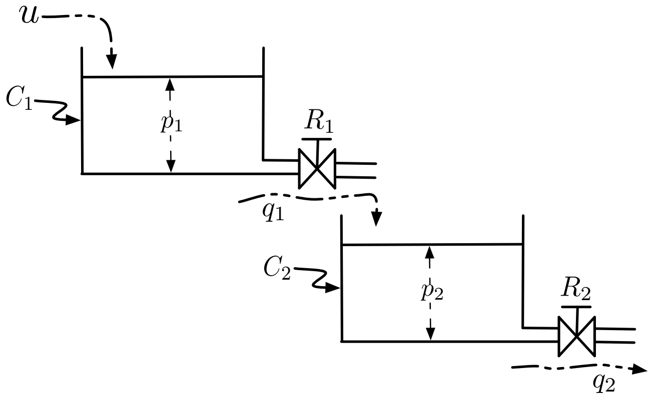


Fig. 1. Liquid level system formed by two non-coupled tanks.

- $u$ is the input flow in tank 1.
- $p_i$ is the hydrostatic pressure in tank $i$.
- $R_i$ is the resistance in valve $i$.
- $C_i$ is the capacitance in tank $i$.
- $q_i$ is the output liquid flow from tank $i$.

The nonlinearity of the system is given by the turbulent flow in the tanks which is generally modeled by:

$$q_i = R_i\sqrt{p_i}\,. \tag{1}$$

The volume of liquid in both tanks in a given time interval is determined by:

$$C_1 dp_1 = (u - q_1)dt\,, \quad C_2 dp_2 = (q_1 - q_2)dt\,. \tag{2}$$

It is desired to control $p_2$ applying an action control in the input flow $u$; hence it is suitable to let Eq. (2) be characterized in function of the hydrostatic pressure. Taking unitary values for $C_i$ and $R_i$ we have that $q_i = \sqrt{p_i}$, in this way:

$$\frac{dp_1}{dt} = u - \sqrt{p_1}\,, \quad \frac{dp_2}{dt} = \sqrt{p_1} - \sqrt{p_2}\,. \tag{3}$$

Taking the standard notation in state variables ($x_1 = p_1$ and $x_2 = p_2$) the output $y$ of the system is generated by $x_2$. This dynamical system can be simulated by analogue computation using block diagrams (see Ref. 11); the diagram associated to this particular case is presented in Fig. 2.

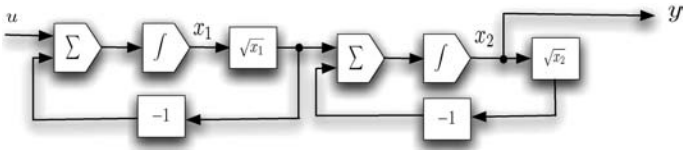


Fig. 2. Block diagram modeling the liquid level system.

It is described in Ref. 9 the exact linearization technique for this type of systems; briefly, this technique consists of expressing the nonlinear system as a linear one by means of a coordinate transformation, applying then a classical linear control and finally converting the result into a nonlinear control law. The coordinate transformation is defined as follows:

$$\begin{aligned} q_1 &= x_2 \\ q_2 &= \sqrt{x_1} - \sqrt{x_2} \end{aligned} \tag{4}$$

and the nonlinear control law is determined by:

$$u = \frac{x_1}{\sqrt{x_2}} + 2\sqrt{x_1}v\,. \tag{5}$$

The coordinate transformation in Eq. (4) yields that the system can be treated as a linear one which is controlled by state feedback and pole placement methods.

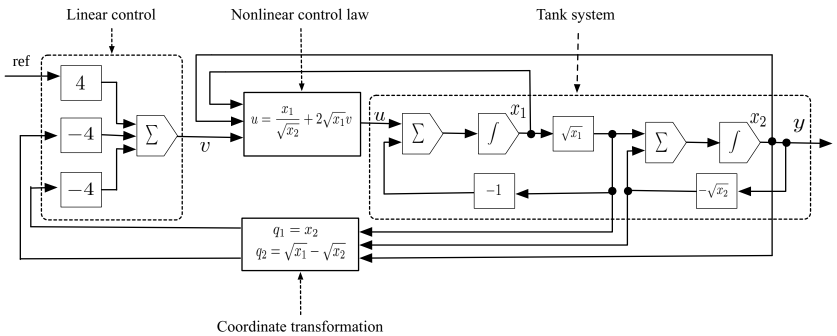


Fig. 3. Liquid level system controlled by the exact linearization technique.

The previous technique is implemented in analogue computation utilizing the block diagram in Fig. 3; in this one, the different parts of the system (liquid level system, coordinate transformation, nonlinear control law, linear control and state feedback) are performed in analogue way by blocks based on simple arithmetic operations; these blocks are connected by signals carrying out the information for solving the system.

It is evident that the analogue computation outlined by the block diagram in Fig. 3 is conformed by a set of locally-interacting simple parts, this agrees with the nature of a CNN, so it is possible for an implementation of the system in this environment.

## 3. Control of the Liquid Level System by CNNs

This paper applies the definition of one-dimensional CNNs based on a previous model presented in Ref. 11, where the set of states contains both real values and arithmetic operators; besides taking neighborhoods of different sizes in order to simulate the corresponding analogue computation.

A CNN consists of a set of states $\mathcal{S}$, where the set of finite sequences of states is described by $\mathcal{S}^*$. For every $w \in \mathcal{S}^*$, let $m_w$ be the number of states in $w$. These states are indexed from left to right, starting from position 0, thus $w_i$ is the state at position $i$ mod $m_w$ and $w_{[i,\ldots,j]}$ is the block of states from $i$ to $j$. The CNN has an initial condition (or configuration) $c^0 \in \mathcal{S}^*$; the superscript indicates the current time and will be omitted when it is understood. Let $\Phi \subset \mathcal{S}^*$ be a set of neighborhoods, where for every $w \in \Phi$ there is a mapping or evolution rule $\phi(w) = a \in \mathcal{S}$ executing a set of logical and arithmetic operations when the neighborhood appears in the current configuration. If $c^t_{[i,\ldots,j]} = w \in \Phi$ then $c^{t+1}_j = \phi(w)$; otherwise $c^{t+1}_j = c^t_j$.

Thus $\phi$ yields a new configuration $c^{t+1}$; periodic boundary conditions are applied to have complete neighborhoods for all the states in the configuration. We are using neighborhoods with right-sided evolutions for producing a shift of the information from left to right during the dynamics of the CNN.

For the nonlinear liquid level system, the associated block diagram has four arithmetic operators: sums, integrators, square roots, and products by constant values. They are executed using real values, so some cells of the CNN will keep real states described by $v \in \mathbb{R}$ and the previous operations shall be implemented by action states with some particular properties $\lambda$ defining the way in which the operations are executed to implement the block diagram.

With states in Table 1, Eq. (6) performs an integration using Euler's method.[a]

$$v_1 \mathrm{I} v_2 \mathrm{CS} v_3 \,. \tag{6}$$

In Eq. (6) state $v_2$ keeps the product of $v_1$ by an integration step 1; this result is accumulated in $v_3$ by CS after a predefined number of iterations. In order to model the system described by the block diagram in Fig. 3, this one is divided in four parts: liquid level system, coordinate transformation, nonlinear control law and linear control.

Let us take first the liquid level system in Fig. 2, all its arithmetic operators can be implemented using the action states in Table 1; with them, an initial configuration $c^0$ is defined in Fig. 4 showing how the block diagram is implemented.

The neighborhoods required for determining this configuration are in Table 2.

Table 1. Action states for implementing a block diagram.

| Operation | Symbol | $\lambda_1$ | $\lambda_2$ | Neighborhood | Evolution |
|---|---|---|---|---|---|
| Square Root | SR | NA | NA | $c_{i-1}\mathrm{SR}_i c_{i+1}$ | $c_{i+1} = \sqrt{c_{i-1}}$ |
| Multiplier | I | $r \in \mathbb{R}$ | $m \in \mathbb{Z}$ | $c_{i-m} \cdots \mathrm{I}_i c_{i+1}$ | $c_{i+1} = c_{i-m} * r$ |
| Copy | C | $m \in \mathbb{Z}$ | NA | $c_{i-m} \cdots \mathrm{C}_i c_{i+1}$ | $c_{i+1} = c_{i-m}$ |
| Product | PR | $m_1 \in \mathbb{Z}$ | $m_2 \in \mathbb{Z}$ | $c_{i-m_2} \cdots c_{i-m_1} \cdots \mathrm{P}_i c_{i+1}$ | $c_{i+1} = c_{i-m_1} * c_{i-m_2}$ |
| Inverse Product | $\mathrm{P}^*$ | $m_1 \in \mathbb{Z}$ | $m_2 \in \mathbb{Z}$ | $c_{i-m_2} \cdots c_{i-m_1} \cdots \mathrm{P}^*{}_i c_{i+1}$ | $c_{i+1} = \frac{1}{c_{i-m_1}} * c_{i-m_2}$ |
| Sum | SU | $m_1 \in \mathbb{Z}$ | $m_2 \in \mathbb{Z}$ | $c_{i-m_2} \cdots c_{i-m_1} \cdots \mathrm{S}_i c_{i+1}$ | $c_{i+1} = c_{i-m_1} + c_{i-m_2}$ |
| Conditional Sum | CS | $\min \in \mathbb{Z}$ | $\max \in \mathbb{Z}$ | $c_{i-1}\mathrm{CS}_i c_{i+1}$ | if (min < max)<br>min = min +1<br>else<br>min = 0<br>$c_{i+1} = c_{i+1} + c_{i-1}$ |

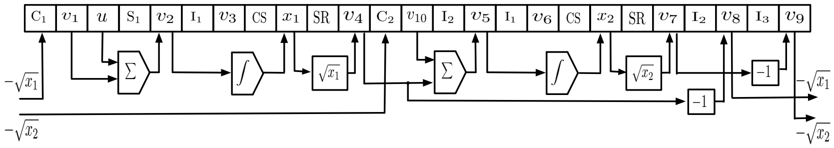


Fig. 4. Initial configuration of the CNN modeling the liquid level system.

[a]This method is chosen for simplicity and gives the sufficient accuracy for this case, of course, more efficient integration methods can be used.

Table 2. Neighborhoods modeling the liquid level system.

| Neighborhood | Evolution | Comment |
|---|---|---|
| $v_1 u \mathrm{S}_1 v_2$ | $v_2 = u + v_1$ | $v_1 = -\sqrt{x_1}$, where initially $x_1 = 0$ |
| $v_2 \mathrm{I}_1 v_3 \mathrm{CS} x_1$ | $x_1 = x_1 + v_3$ where $v_3 = v_2 * r$ | $r = 0.1$. |
| $x_1 \mathrm{SR} v_4$ | $v_4 = \sqrt{x_1}$ | |
| $v_4 \cdots v_{10} \mathrm{S}_2 v_5$ | $v_5 = v_4 + v_{10}$ | $v_{10} = -\sqrt{x_2}$, where initially $x_2 = 0$ |
| $v_5 \mathrm{I}_1 v_6 \mathrm{CS} x_2$ | $x_2 = x_2 + v_6$ where $v_6 = v_5 * r$ | $r = 0.1$ |
| $x_2 \mathrm{SR} v_7$ | $v_7 = \sqrt{x_2}$ | |
| $v_4 \cdots \mathrm{I}_2 v_8$ | $v_8 = v_4 * r$ | $r = -1$ |
| $v_7 \cdots \mathrm{I}_3 v_9$ | $v_9 = v_4 * r$ | $r = -1$ |
| $v_8 \cdots \mathrm{C}_1 v_1$ | $v_1 = v_8$ | |
| $v_9 \cdots \mathrm{C}_2 v_{10}$ | $v_{10} = v_9$ | |

| | | | | | | | | $c_9$ | | | | | | | | | | $c_{19}$ | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C1 | 0.000 | 1.000 | S1 | 0.000 | I1 | 0.000 | 0 | 0.000 | SR | 0.000 | C2 | 0.000 | S2 | 0.000 | I1 | 0.000 | 0 | 0.000 | SR | 0.000 | I2 | 0.000 | I3 | 0.000 |
| C1 | 0.000 | 1.000 | S1 | 1.000 | I1 | 0.000 | 0 | 0.000 | SR | 0.000 | C2 | 0.000 | S2 | 0.000 | I1 | 0.000 | 0 | 0.000 | SR | 0.000 | I2 | -0.000 | I3 | -0.000 |
| C1 | -0.000 | 1.000 | S1 | 1.000 | I1 | 0.100 | 1 | 0.000 | SR | 0.000 | C2 | -0.000 | S2 | 0.000 | I1 | 0.000 | 0 | 0.000 | SR | 0.000 | I2 | -0.000 | I3 | -0.000 |
| C1 | -0.000 | 1.000 | S1 | 1.000 | I1 | 0.100 | 0 | 0.100 | SR | 0.000 | C2 | -0.000 | S2 | 0.000 | I1 | 0.000 | 0 | 0.000 | SR | 0.000 | I2 | -0.000 | I3 | -0.000 |
| C1 | -0.000 | 1.000 | S1 | 1.000 | I1 | 0.100 | 0 | 0.100 | SR | 0.316 | C2 | -0.000 | S2 | 0.000 | I1 | 0.000 | 0 | 0.000 | SR | 0.000 | I2 | -0.000 | I3 | -0.000 |
| C1 | -0.000 | 1.000 | S1 | 1.000 | I1 | 0.100 | 0 | 0.100 | SR | 0.316 | C2 | -0.000 | S2 | 0.316 | I1 | 0.000 | 0 | 0.000 | SR | 0.000 | I2 | -0.316 | I3 | -0.000 |
| C1 | -0.316 | 1.000 | S1 | 1.000 | I1 | 0.100 | 0 | 0.100 | SR | 0.316 | C2 | -0.000 | S2 | 0.316 | I1 | 0.032 | 1 | 0.000 | SR | 0.000 | I2 | -0.316 | I3 | -0.000 |
| C1 | -0.316 | 1.000 | S1 | 0.684 | I1 | 0.100 | 0 | 0.100 | SR | 0.316 | C2 | -0.000 | S2 | 0.316 | I1 | 0.032 | 0 | 0.032 | SR | 0.000 | I2 | -0.316 | I3 | -0.000 |
| C1 | -0.316 | 1.000 | S1 | 0.684 | I1 | 0.068 | 0 | 0.100 | SR | 0.316 | C2 | -0.000 | S2 | 0.316 | I1 | 0.032 | 0 | 0.032 | SR | 0.178 | I2 | -0.316 | I3 | -0.000 |
| C1 | -0.316 | 1.000 | S1 | 0.684 | I1 | 0.068 | 0 | 0.100 | SR | 0.316 | C2 | -0.000 | S2 | 0.316 | I1 | 0.032 | 0 | 0.032 | SR | 0.178 | I2 | -0.316 | I3 | -0.178 |
| C1 | -0.316 | 1.000 | S1 | 0.684 | I1 | 0.068 | 0 | 0.100 | SR | 0.316 | C2 | -0.178 | S2 | 0.316 | I1 | 0.032 | 0 | 0.032 | SR | 0.178 | I2 | -0.316 | I3 | -0.178 |
| C1 | -0.316 | 1.000 | S1 | 0.684 | I1 | 0.068 | 0 | 0.100 | SR | 0.316 | C2 | -0.178 | S2 | 0.138 | I1 | 0.032 | 0 | 0.032 | SR | 0.178 | I2 | -0.316 | I3 | -0.178 |
| C1 | -0.316 | 1.000 | S1 | 0.684 | I1 | 0.068 | 0 | 0.100 | SR | 0.316 | C2 | -0.178 | S2 | 0.138 | I1 | 0.014 | 0 | 0.032 | SR | 0.178 | I2 | -0.316 | I3 | -0.178 |
| C1 | -0.316 | 1.000 | S1 | 0.684 | I1 | 0.068 | 1 | 0.100 | SR | 0.316 | C2 | -0.178 | S2 | 0.138 | I1 | 0.014 | 0 | 0.032 | SR | 0.178 | I2 | -0.316 | I3 | -0.178 |
| C1 | -0.316 | 1.000 | S1 | 0.684 | I1 | 0.068 | 0 | 0.168 | SR | 0.316 | C2 | -0.178 | S2 | 0.138 | I1 | 0.014 | 0 | 0.032 | SR | 0.178 | I2 | -0.316 | I3 | -0.178 |
| C1 | -0.316 | 1.000 | S1 | 0.684 | I1 | 0.068 | 0 | 0.168 | SR | 0.410 | C2 | -0.178 | S2 | 0.138 | I1 | 0.014 | 0 | 0.032 | SR | 0.178 | I2 | -0.316 | I3 | -0.178 |
| C1 | -0.316 | 1.000 | S1 | 0.684 | I1 | 0.068 | 0 | 0.168 | SR | 0.410 | C2 | -0.178 | S2 | 0.233 | I1 | 0.014 | 0 | 0.032 | SR | 0.178 | I2 | -0.410 | I3 | -0.178 |
| C1 | -0.410 | 1.000 | S1 | 0.684 | I1 | 0.068 | 0 | 0.168 | SR | 0.410 | C2 | -0.178 | S2 | 0.233 | I1 | 0.023 | 1 | 0.032 | SR | 0.178 | I2 | -0.410 | I3 | -0.178 |
| C1 | -0.410 | 1.000 | S1 | 0.590 | I1 | 0.068 | 0 | 0.168 | SR | 0.410 | C2 | -0.178 | S2 | 0.233 | I1 | 0.023 | 0 | 0.055 | SR | 0.178 | I2 | -0.410 | I3 | -0.178 |

Fig. 5. Evolution of the CNN modeling the liquid level system.

Figure 5 displays 19 evolutions of the CNN evolving from the initial configuration in Fig. 4 taking a unitary input $u$ in cell 3, in particular cells 9 and 19 contain the numerical solutions of the differential equation system. The evolution shows how the information goes from left to right and the feedback of the system is given after 12 timesteps. In the initial configuration, cell 8 is a conditional sum CS (a local control unit) used for obtaining a right computation, the initial parameters in $c_8^0$ are: min $= 10$ and max $= 11$; thus $c_8^1$ holds that min $=$ max, letting pass the information in the following timestep. Analogously, the initial parameters in $c_{18}^0$ are: min $= 6$ and max $= 11$; the same behavior is repeated every 11 steps.

The integration step defined in the CNN (as parameter of cell 6) is 0.1 secs., so we need 1100 evolutions for calculating 10 seconds of dynamical response. The values in cells 9 and 19 are taken every 11 steps and graphed in Fig. 6; we can see the expected exponential responses of $x_1$ and $x_2$ for the unitary input $u$.

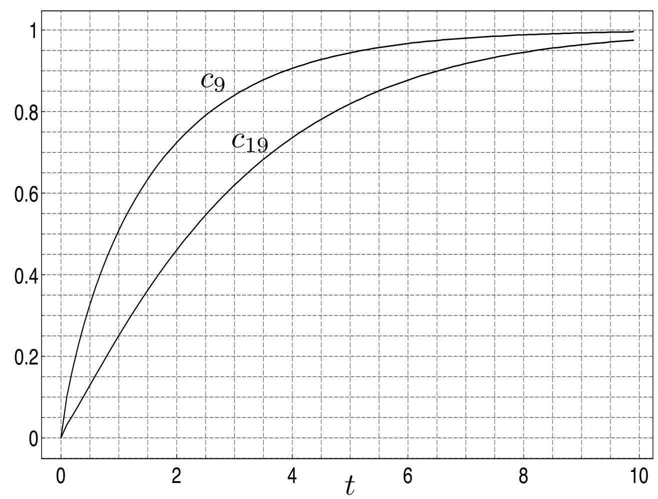


Fig. 6. Hydrostatic-pressure response simulated by the CNN.

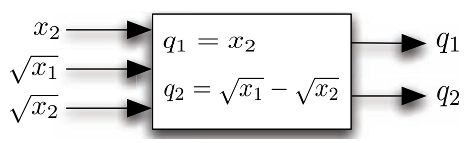


Fig. 7. Block diagram modeling the coordinate transformation.

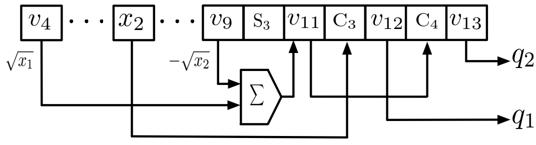


Fig. 8. CNN modeling the coordinate transformation.

Now, the remaining parts of the control for the liquid level system in Fig. 3 will be described by other CNNs; taking the values previously exposed in the cells of the CNN in Fig. 4. The analogue computation for the coordinate transformation is given in Fig. 7.

This diagram is performed by the initial condition of the CNN depicted in Fig. 8, where $\sqrt{x_1}$ and $-\sqrt{x_2}$ are taken from cells $v_4$ and $v_9$ in Fig. 4.

The arithmetic operations in Fig. 8 are implemented using the action states in Table 1; thus the needed neighborhoods are shown in Table 3.

The action states $\mathrm{C}_3$ and $\mathrm{C}_4$ in these neighborhoods are used to copy the values of $x_2$ and $\sqrt{x_1} - \sqrt{x_2}$ in order to have the representation of the block diagram in one piece for clarity; nevertheless it is not necessary for the correct analogue computation of the solution.

Table 3. Neighborhoods for the coordinate transformation.

| Neighborhood | Evolution | Comment |
|---|---|---|
| $v_4, \ldots, v_9 S_3 v_{11}$ | $v_{11} = v_4 + v_9$ | $v_4 = \sqrt{x_1}$ and $v_9 = -\sqrt{x_2}$ |
| $x_2, \ldots, \mathrm{C}_3 v_{12}$ | $v_{12} = q_1 = x_2$ | |
| $v_{11}, \ldots, \mathrm{C}_4 v_{13}$ | $v_{13} = q_2 = v_{11}$ | |

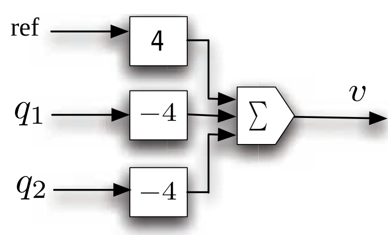


Fig. 9. Block diagram for the linear control based on state feedback.

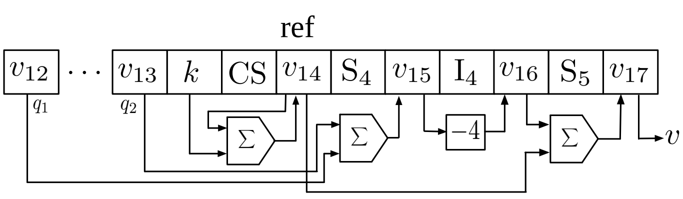


Fig. 10. CNN modeling the linear control by state feedback.

Figure 9 shows the analogue computation for the linear control where ref is the desired value in the output of the system, $q_1$ and $q_2$ are the signals in the feedback updated by their corresponding gains to calculate the control action $v$.

In order to observe the action of the nonlinear control in the overall system, the desired reference ref is periodically incremented to control the error between the output and the wished reference to 0. Figure 10 shows the CNN implementation of the block diagram in Fig. 9.

The neighborhoods composing the CNN in Fig. 10 are exposed in Table 4, values $v_{12}$ and $v_{13}$ are the output of the coordinate-transformation in Fig. 8 containing the estimations of $q_1$ and $q_2$ respectively. To calculate the reference change in the input of the system, the first neighborhood of Table 4 is utilized; in this neighborhood the value $v_{14}$ corresponds to the ref variable in Fig. 9. Cell $k$ is a fixed value which is added periodically to $v_{14}$ in a determined number of evolutions controlled by the parameters *min* and *max* of cell CS, these ones are explicitly specified when all the cells in the CNN are delineated to simulate the whole liquid level system.

Finally, the nonlinear control is fulfilled by the analogue computation in Fig. 11.

Figure 12 shows the initial configuration of the CNN representing the nonlinear control in Fig. 11 using the neighborhoods in Table 5. Cells $x_1$, $v_4$, $v_7$, $v$ and $u$ are

Table 4. Neighborhoods modeling the linear control by state feedback.

| Neighborhood | Evolution | Comment |
|---|---|---|
| $k\mathrm{CS}v_{14}$ | $v_{14} = v_{14} + k$ | Initially $v_{14} = 4$, $k = 4$ |
| $v_{12} \cdots v_{13} \cdots \mathrm{S}_4 v_{15}$ | $v_{15} = v_{12} + v_{13}$ | $v_{12} = q_1$, $v_{13} = q_2$ |
| $v_{15}\mathrm{I}_4 v_{16}$ | $v_{16} = v_{15} * r$ | $r = -4$ |
| $v_{14} \cdots v_{16}\mathrm{S}_5 v_{17}$ | $v_{17} = v_{14} + v_{16}$ | $v_{17}$ is the control action $v$ |

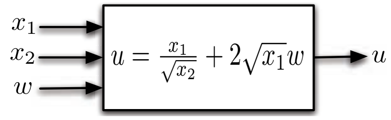


Fig. 11. Block diagram for the nonlinear control law.

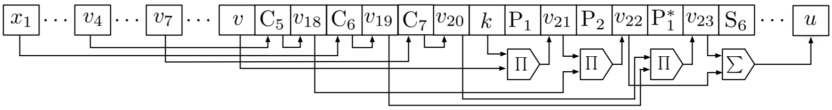


Fig. 12. CNN modeling the nonlinear control law.

elements of the preceding CNNs communicating the other parts of the liquid level system. Thus the nonlinear control law is implemented from cell $\mathrm{C}_5$ up to $\mathrm{S}_6$.

Since almost all values in the corresponding block diagram have been calculated in the previous CNNs; essentially the neighborhoods modeling the nonlinear control law are based on copies of earlier values established in the above initial configurations. These neighborhoods are presented in Table 5.

Figure 13 depicts in a compact form the way in which all the previous initial configurations (liquid level system, coordinate transformation, linear control, and nonlinear control law) are connected to specify the complete CNN modeling and control the liquid level system subjected to perturbations in the reference, every cell is indexed according to our computational implementation (which is not necessarily unique). The final CNN has 54 cells where cells $c_{36}$ and $c_{46}$ hold the values $x_1$ and $x_2$ respectively.

The first valid value of $x_1$ is calculated in 13 evolutions, this one is used to compute $x_2$ after 17 evolutions of the automaton. Therefore, once valid values for $x_1$ and $x_2$ are obtained, it takes other 17 timesteps to generate again valid values of these variables; in this way 17 evolutions are equivalent to one integration step of 0.1 s in the liquid level system. The computation of $x_1$ and $x_2$ is locally controlled by conditional-sum states in cells $c_{35}$ and $c_{45}$, where the initial parameter *min* in $c_{35}$ is 6 and for $c_{45}$ is 2 and, max $= 17$ in both cases.

It is desired to increase the reference of $p_2$ in one unit every five seconds for observing the action of the control system, so cell CS in the linear control part of

Table 5. Neighborhoods modeling the nonlinear control.

| Neighborhood | Evolution | Comment |
|---|---|---|
| $v_4\cdots \mathrm{C}_5 v_{18}$ | $v_{18}=v_4$ | where $v_4=\sqrt{x_1}$ |
| $x_1\cdots \mathrm{C}_6 v_{19}$ | $v_{19}=x_1$ | |
| $v_7\cdots \mathrm{C}_7 v_{20}$ | $v_{20}=v_7$ | where $v_7=\sqrt{x_2}$ |
| $v_{17}\cdots k\mathrm{P}_1 v_{21}$ | $v_{21}=v_{17}*k$ | where $v_{17}=w$ and $k=2$ |
| $v_{18}\cdots v_{21}\mathrm{P}_2 v_{22}$ | $v_{22}=v_{18}*v_{21}$ | $v_{22}=2\sqrt{x_1}w$ |
| $v_{19}\cdots v_{2}0\cdots \mathrm{P}^{*}{}_{1}v_{23}$ | $v_{23}=v_{19}/v_{20}$ | $v_{23}=x_1/\sqrt{x_2}$ |
| $v_{22}\cdots v_{23}\mathrm{S}_6 u$ | $u=v_{22}+v_{23}$ | $u=x_1/\sqrt{x_2}+2\sqrt{x_1}w$ |

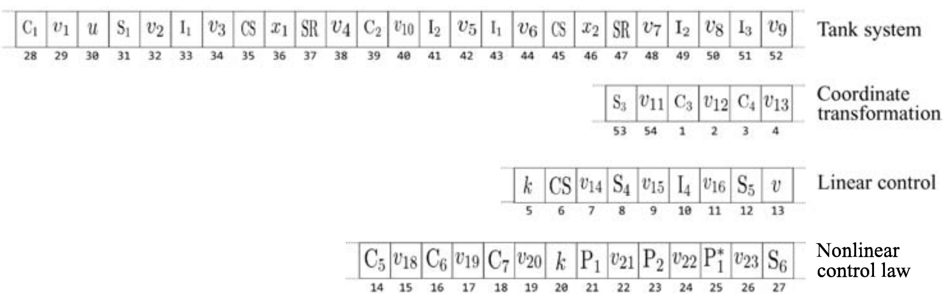


Fig. 13. Complete configuration of the CNN controlling the liquid level system.

the CNN has parameter min $=0$ and max $=850$ to yield the increment of ref. In this way, 3400 evolutions of the CNN are needed to produce 20 seconds of analogue simulation and control; where values in cells $c_{36}$ and $c_{46}$ can be taken every 17 evolutions to get the dynamics of the liquid level system; these ones are graphed in Fig. 14, proving that the system reaches the desired reference for $x_2$ every five seconds and Fig. 16 shows the evolution of the CNN, the first configuration has the initial conditions of the system, with $c_{36}=c_{46}=0.01$.

In order to determine the initial conditions of the system ($x_1=c_{36}$ and $x_2=c_{46}$), it is necessary to analyze the equation in Fig. 11; if $x_1=0$ the system is not controllable and does not matter if the value is $x_2$. If $x_1>0$ and $x_2=0$, the output of the controller is infinite and we do not have a correct operation. Therefore the valid initial conditions are provided by $x_1>0$ and $x_2>0$; as a consequence, in this paper we have fixed initial conditions near to 0 for observing a complete evolution of the nonlinear control law.

## 4. Comments about the Complexity and Parallelism of the CNN

Since the CNN is just implementing and simulating the block diagram associated with the liquid level system, the complexity of the CNN is in straightforward

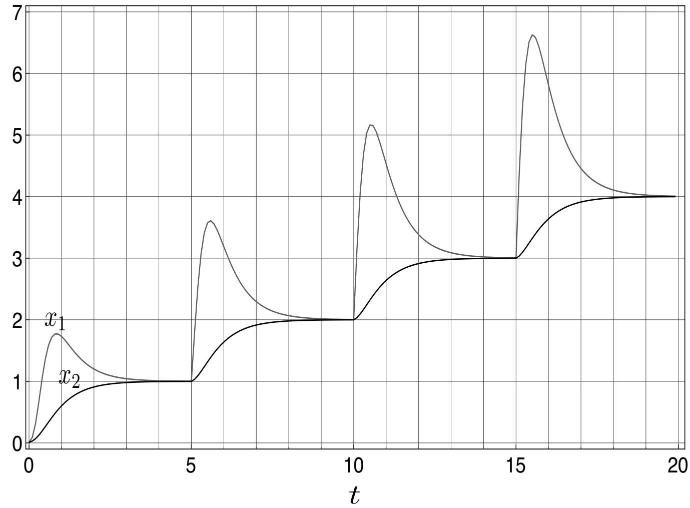


Fig. 14. Dynamical response of $x_1$ and $x_2$ for 3400 evolutions of the CNN.

relation with the one related to the block diagram. For a given block diagram, let $\Gamma$ be the number of basic operations[b] required for executing it.

For the block diagram in Fig. 3 and its four subdiagrams, let us associate $\gamma_1$ with the number of basic operations for accomplishing the linear control, in the same way let us tie in $\gamma_2$ with the nonlinear control law, $\gamma_3$ with the tank system and $\gamma_4$ with the coordinate transformation. Thus for $1 \leq i \leq 4$, one iteration of the liquid level system is achieved in $\Gamma = \sum_i \gamma_i$ operations. Let $m$ be the number of iterations required for controlling the system, hence its complexity is given by $m\Gamma$.

Every subdiagram is implemented by a part of the CNN which works in a serial way, so the correct calculation of the nonlinear control law depends on having the right value of the linear control and the same happens for the other parts. But the internal operations of each subdiagram can be parallelized in the CNN, of course, respecting the dependencies in the variable computation which is the common issue in parallel processing.

Let $\gamma'_i$ be the number of necessary operations for the adequate calculation of every subdiagram in the corresponding fragment of the CNN, therefore, since $\gamma_1$ consists of a single operation, hence $\gamma'_1 = \gamma_1$. In the case of the nonlinear control law, the root calculation and products are parallelized in the CNN, so $\gamma'_2 = \gamma_2 - 2$.

For the tank system and the coordinate transformation, in the associated block subdiagrams in Fig. 3 the output of the integrator block ($x_2$) is parallelized in the CNN for obtaining $q_2$, thus $\gamma'_3 + \gamma'_4 = \gamma_3 + \gamma_4 - 1$.

[b]In our case we have sum and product of $n$ variables, square root, scalar product, copy from one variable to an other and Euler integration.

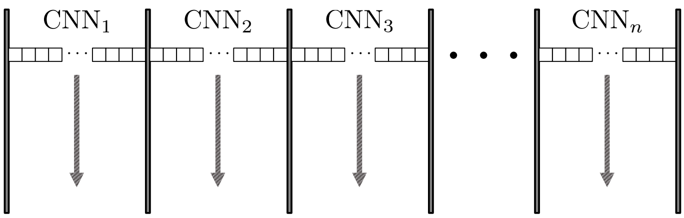


Fig. 15. Parallel evolutions of different CNNs each with distinct parameters.

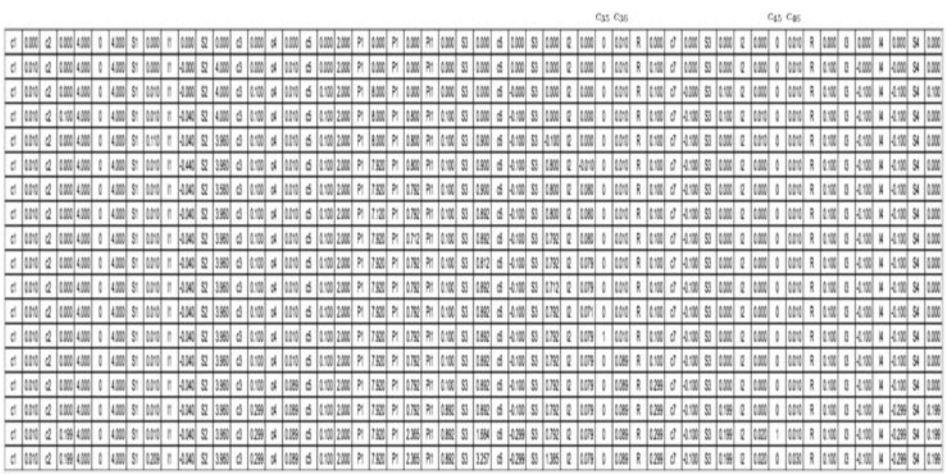

Fig. 16. Complete CNN modeling and controlling the liquid level system.



Finally, if $\Gamma'$ is the number of operations achieved for the CNN, we have that $\Gamma' = \Gamma - 3$. In this way for $m$ iterations of the system, the complexity of the CNN is given by $m(\Gamma - 3)$ which is almost the same obtained by the serial procedure.

Nevertheless, the relevance of the proposed method lies in the simplicity of the CNN and that the feedback of the system is produced by periodic boundary conditions, thus we may have $n$ copies of the CNN, each with distinct conditions (for instance: initial values, integration steps or perturbation times) and execute them in parallel (Fig. 15), in order to analyze different behaviors at the same time. Usually this parallelization is a heavy computational task with a classic implementation of block diagrams, but due to the local interactions and the lack of a central control in the case of CNNs, this task can be constructed in an easier way with the application of a quiescent state which serves as a barrier separating every CNN.

## 5. Concluding Remarks

The previous sections have established the analogue modeling and control by CNNs for the dynamics associated to a nonlinear system, using as link the corresponding block diagram. In this way, the cells of the CNN keep both data or arithmetic and logic operators to perform the same analogue task that the block diagram, and the evolution of the CNN provides the desired approximation to the solution of the problem; in this particular case, the control of a liquid level system. The advantage of using CNNs is that they can be easily implemented avoiding the necessity of a complex global control by their parallel nature.

Meanwhile the execution time of a single CNN and the serial realization of the block diagram is almost the same for a small number of iterations, the convenience of this procedure lies on managing in an easier way $n$ copies of the CNN, each with distinct parameters, for observing different behaviors of the system in parallel. Further research implies performing this kind of CNNs in parallel hardware as FPGAs and extending the analysis for nonlinear systems with multiple input and multiple output values.



## Acknowledgments

This work has been partially supported by PROMEP Collaboration Network "Modelado y análisis de sistemas complejos" and by CONACYT project CB2007-83554.

## References

1. L. Chua and L. Yang, *IEEE T. Circuits Syst.* **35**, 1257 (1988).
2. L. Chua and L. Yang, *IEEE T. Circuits Syst.* **35**, 1273 (1988).
3. L. Chua, L. Yang and K. Krieg, *J. VLSI Signal Proc.* **3**, 25 (1991).
4. S. Jankowski, A. Londei, A. Lozowski and C. Mazur, *Int. J. Circ. Theor. App.* **24**, 275 (1996).
5. M. Ercsey-Ravasz, T. Roska and Z. A. Neda, *Int. J. Mod. Phys. C* **17**, 909 (2006).
6. J. Nossek, G. Seiler, T. Roska and L. Chua, *Int. J. Circ. Theor. App.* **20**, 533 (1998).
7. A. Adamatzky, P. Arena, A. Basile, R. Carmona-Galan, B. Costello, L. Fortuna, M. Frasca and A. Rodríguez-Vázquez, *IEEE T. Circuits-I* **51**, 926 (2004).
8. F. Gollas and R. Tetzlaff, *Modeling Brain Electrical Activity in Epilepsy by Reaction-Diffusion Cellular Neural Networks, in Proc. SPIE*, Vol. 5839, eds. R. A. Carmona *et al.* (IEEE, 2005), p. 219.
9. S. Sastry, *Nonlinear Systems, Analysis, Stability and Control* (Springer-Verlag, Berlin, 1999).
10. A. Isidori, *Nonlinear Control Systems* (Springer-Verlag, Berlin, 1995).
11. J. C. Seck-Tuoh-Mora, M. Gonzalez-Hernandez, N. Hernandez-Romero, A. Rodriguez-Trejo and S. V. Chapa-Vergara, *Int. J. Mod. Phys. C* **18**, 833 (2007).