Virtual system for dynamical systems simulation and artificial Neural Networks Design

Juan Zambrano-Ramírez* Gloria Mousalli-Kayat Francklin Rivas-Echeverría*, Addison Ríos-Bolívar***

Universidad de Los Andes, Mérida, Venezuela 5101

*Laboratorio de Sistemas Inteligentes

**Dpto. de Medición y Evaluación

Abstract:- In this work it is presented a computational tool design for dynamical systems simulation and Artificial Neural Networks (ANN) training and implanting. The simulation is been handle using an Graphic User Interface (GUI), that can be easily used for describing linear or nonlinear dynamical systems using space state equations. It is used a Script connection for linking NI-LabVIEW™ and MatLab® Neural Networks Toolbox.

Key Words:- LabVIEW™, Applications Integration, Dynamical systems simulation, Artificial Neural Networks, MatLab®.

1 Introduction

The area of simulation has been constituted in a powerful tool for systems design, analysis and optimization in industrial processes. The availability of more powerful computers every day, with cheapest costs and easy for use, accompanied by software or application programs and highly flexible programming languages, has allowed the wide use of different simulation techniques in processes control [1].

Another area that has had an important peak in the last years is Neuro-computing, also called Artificial Neural Networks (ANN). At the end of the 80's, the interest for the ANN, had a quickly development. This has motivated to professionals of diverse fields, such as: engineering, philosophy, physiology and psychology for searching applications inside their disciplines based on the potential offered by this technique.

Nowadays, it seems possible to apply the computer science in domains that were restricted to the human intelligence. This, have make possible to construct machines that can learn and remember in a way that has remarkable likeness to the human mental processes [2].

ANN are inspired in biological neural networks: they are made up of elements that act similar to most of the elementary functions of the biological neuron. These elements are organized in such a form that can be related to the anatomy of the brain. For example, they can learn from the experience, they can generalize from previous collected information, and they can extract essential characteristics from input data.

In this work it is presented the LabVIEW™ program for developing a tool that allows, in an easy and effective way, to the user the simulation of dynamical systems and training ANN. The work is organized as follows: In section 2 it will be presented some basements concerning Simulation of Dynamic Systems. Then, in section 3 it will be introduced the Artificial Neural Networks (ANN). Section 4 will contain the Description of the designed Tool and section 5 will present the corresponding Conclusions.

2 Dynamical Systems Simulation

The systems simulation is defined as the technique used for solving problems related to dynamical model of a system considering the time changes [3,7].

Given a mathematical model for a real system, it is sometimes possible to obtain information related to the system using analytic methods. However, most of the real systems are too complex to admit analytical models, and they should be studied by means of simulation. Therefore, the simulation models are usually more complex and near to the real systems than the analytic models since it are not necessary to make so many simplifications to obtain results. In the simulation process, the computer is used for evaluating numerically the model; and the obtained solutions are used to estimate the real characteristics of the model [4].

The systems simulation has become a powerful tool for decision making that allows, among other things:

- To predict the result of the actions that have been taken on the process or control system.
- To understand the reasons for events occurrence.
- To identify conflictive areas before the system installation.
- To explore the effects of some modifications given to the system.
- To evaluate ideas and their viability, and to identify their problems.
- To stimulate the creative and to train personnel.
- To optimize processes (energy savings, bottle necks, results improvements, etc.).

The programming languages used for systems simulation are, generally, of high complexity, because for generating the appropriate code to simulate a system, sometimes, it requires of wide knowledge in programming logic; this fact, restricts the use of the simulators for people with few computational knowledge. This was the main motivation for this work is to design a graphic tool for dynamic systems simulation that is effective and easy for the user.

3 Artificial Neural Networks

Artificial Neural Networks (ANN) constitutes a branch of the Artificial Intelligence which tries to emulate the operation and capabilities of biological neural networks related to learning and information processing [5]. Its development has represented a favorable impact for the computer science and its technological applications, and also for other areas like physiology and neurology, with which a very profitable interrelation has been created for the use of cerebral operation models and for the processes interpretation associated with the learning capacities. ANN are based on biological behavior, and scientists are concerned in the brain organization, when they consider algorithms and configurations of ANN, but the total knowledge about the operation of the brain is so limited. For this reason, the designers of ANN should consider the current biological knowledge, looking for structures that execute useful functions. In many cases, this assumptions discards biological possibilities or produce networks models that are organically impossible to find or that require a great amount of suppositions about the anatomy and operation of the brain [6].

Due to the advance in computer science, ANN can be trained using big lots of data, improve their performance and self organization capabilities, fault tolerance and real time operation. This have made that the field of application of the ANN has increased, being used in different tasks as: systems modeling and identification, simulation, processes control, prediction, fault handling, patterns recognition, medical diagnosis, virtual sensors design, etc.

There are many applications where the data generated by a simulation process are used for obtaining non lineal processes modeling and identification models. Due to this, it was created an easy access user interface that can be used for simulating non linear systems and for training ANN.

4 Designed Computational Tool Descriptions

The designed tool consists of two graphic user interfaces that allow the simulation of dynamic systems and the training and testing of ANN.

Figures 1 depicts a scheme concerning the tool functionality. It can be seen the relationship between each of the user interfaces and the generated data; also can be observed the interaction with data generated by other applications.

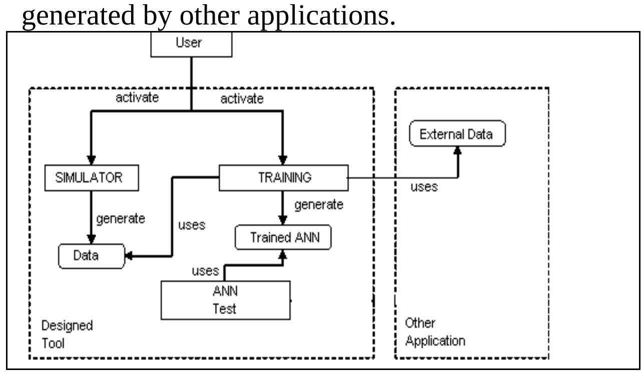

Figure 1 Computational tool functionality Scheme

The computer language used for developing the tool was LabVIEW™, which is a graphic programming language that allows creating easy and useful elegant interfaces. The programming in LabVIEW™ [8] is performed in a block diagram which contains the necessary code instructions for generating the front end that contains the graphics user interface.

The computational tool, developed in Spanish, possesses a main panel where the user can choose among two programs: the simulator and the program for training and testing ANN.

4.1. Processes Simulator

The designed program allows the simulation of any kind of linear or nonlinear processes (for example:

physical, chemical and mechanical processes) and its dynamics should be described by differential equations; in figure 2 the front panel of the simulation program can be observed.

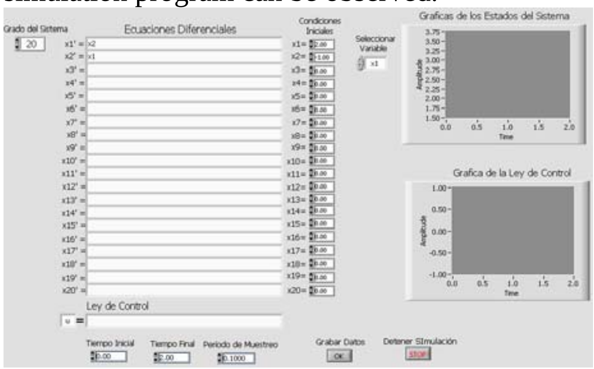

Figure 2. Simulation Front Panel.

LabVIEW™ possesses a group of methods for solving systems described by ordinary differential equations (ODE). It was used the Euler method, for developing the simulator, which solves "n" order systems; but in this case the order was limited to 20. This method requires the following parameters: initial and final simulation time, sampling period, system variables and its corresponding initial conditions and the group of differential equations. The method generates as output, in array format, the solutions of the state variables and the simulation times.
The design of the front panel for the state equations was given in such a way that depending on the order of the system it will be activated in screen the corresponding cells for introducing the equations and initial conditions of the system variables (See Figure 2). This is, if the order of the system is three, the user will only visualize three cells for introducing the equations, with the purpose of providing a comprehensible interface and easy for using.
Additionally, in the front panel there is an enabled cell for the control action of the system; this cell will always be available in cases where the system possesses an external entrance.
In the front panel two graphics are visualized, one for the system states variables and the other for the control signal, the graph of the system states allows the user to visualize a particular state: For doing this, it has a control that let the user to select the state that he wants to visualize (see figure 2).
For data storage, it was necessary to use the variety of functions for handling files that LabVIEW™ possesses. For this, there is a button in the front panel that allows keeping the simulation data of the system as can be seen in figure 3. The data is stored in a file in columns format, where the number of columns of the file depends on the order of the simulated system; for example, if the system is of 4th order, the data file will contain 4 columns. If the simulated system has an external entrance, then the data corresponding to this signal is stored like a column at the end of the file. The file contains a plain text format, example: filename.txt.

4.2 ANN Training

The designed tool for ANN training consists of an easy graphic interface, where the user can train the ANN using the data previously generated by the processes simulator or using external data, as can be seen in figure 1. In figure 3, the front panel is observed for training the ANN.
LabVIEW™ allows the connection with other programming languages including MatLab®. LabVIEW™ possesses a script (text) that can be used for executing MatLab® programs. The programming code for training ANN calls the appropriate functions of MatLab® Neural Networks Toolbox, and this MatLab® script is executed from LabVIEW™.

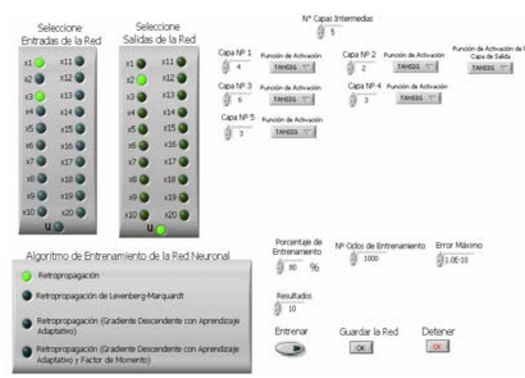

Figure 3. Front Panel for training ANN.

4.2.1 ANN Training Parameters

The MatLab® script receives several types of data [9], including: real and complex matrices, real and complex vectors and real and complex numbers. This allowed the pass of parameters for training ANN. This parameters are: number of layers of the ANN (layers), allowed maximum error (error), number of training cycles (nc), number of nodes in hidden layers (nd), activation function for each layer (t), input vector for the ANN (to), output vector for the net (s), data matrix, including a percentage for training of the ANN (TO) and the rest for testing the ANN (B). It can be seen in figure 5.
The interface allows the user to select the parameters in the following way:

- **ANN Inputs and Outputs:** The user has two cells that are available in the "ANN training" front panel, which are the indicators of the system state variables (see figure 4). The variable indicators are activated depending on the number of columns in the data file, where the user can select inputs and outputs; these cells are programmed in such a way that when a variable is activated as input variable, then it won't be able to be activated as output of the same ANN and vice versa. In case the system possesses an external entrance, it can be used as input or output for the ANN.
- **ANN Structure:** The structure of an ANN is conformed by its topological configuration. The structure is conformed by the number of layers, the number of neurons for each layer and the activation function per layer. In the front panel of ANN training module, there is a cell where the user introduces the number of layers of the ANN and automatically the cells corresponding to the number of hidden layers are activated. In these cells, it is defined the number of nodes for each layer, and is also possible to select the activation function of each one of them. The user can select the activation function inside a group of four functions which are: Hyperbolic Tangent (tansig), linear (purelin), sigmoid (logsig) and positive linear (poslin), (see figure 4). The output layer was designed in such a way that it automatically takes as many nodes as outputs have the ANN.
- **Other parameters**: Once configured the structure of the ANN, the user can choose the learning algorithm, among a group of four algorithms which are: backpropagation, Levenberg-Marquardt backpropagation, gradient descent with adaptive backpropagation and gradient descent with momentum and adaptive backpropagation [5, 10]. After having selected the learning algorithm, number of cycles, maximum error, percentage of data for training and testing, the user can also define the updating interval for presenting the results for the error.

When the ANN training is finished, the user will decide if he wants to keep or not the trained network. For it, there is a button in the front panel, which allows the user to keep the ANN with the name and location of his preference (see figure 4). The ANN is stored as a MatLab® file, with extension ".mat" which can be used in MatLab® or in the testing interface. The user can train and keep as many ANNs as he wants.

5 Conclusions

Inside the computer science techniques, the simulation is the one of the most widely used, but sometimes, the models that are used to simulate real systems are very complex and writing computer programs for describing them is a complicated task. Due to these complications and using the capabilities of LabVIEW™ for developing friendly interfaces, it was created a computational tool for simulating dynamical systems.

The communication between LabVIEW™ and MatLab® through the appropriate script, allowed creating a tool for training Artificial Neural Networks with a pleasant and very simple graphic user interface.

The designed tool can be of great utility in branches like the mechanical engineering, the physics and the chemistry, where the specialists will be able to interact in an intuitive way with the user interface, in order to simulate dynamical systems and/or train artificial neural networks. At the same time, it is considered that the developed tool can be potentially used in the engineering education, in order to demonstrate the behavior and functionality of artificial neural networks.

6 References

[1] Ernesto Kofman http://www.ni.com/latam

[2] Hilera Jose R., Martinez Victor J. *Redes Neuronales Artificiales Fundamentos, Modelos y Aplicaciones.* Editorial Alfaomega, México D.F 2000.

[3] Gordon Geoffrey. *Simulación de sistemas.* Editorial Diana, México D.F 1980.

[4] Igor Alvarado http://ww.simula.com/apuntessimulacion.doc

[5] Aguilar C. José, Rivas E. Francklin. *Introducción a las Técnicas de Computación Inteligente.* Universidad de Los Andes Mérida Venezuela. Junio, 2001.

[6] **(**Intelligent Agents: The Right Information at the Right Time), Don Gilbert http://w3.informatik.gu.se/~dixi/agent/read.htm

[7] Shinskey F.G. *Feedback Controllers Process Industry.* MacGrawHill. 1995.

[8] National Instruments Corporation, LabVIEW. User Manual. USA (2000).

[9] The MathWorks Inc., Matlab Language of Technical Computing. USA (1999).

[10] Hagan M., Demuth H. and Beale M. Neural Network Design. An International Thompson Publishing Company. 1996