Evenet2000 as Software Application for System Identification and Control

E.J. González, R.L. Marichal, A.F. Hamilton, L. Moreno

Abstract-- In this paper, advantages of using Evenet-2000, a Java-Based neural network toolkit, for system identification and control are presented. This powerful tool, although it was initially designed for training if neural networks, has been showed highly useful in system identification and control problems. These problems are treated as optimization ones, considering the global system as a neural network to train. Applying learning methods (such as Descent Gradient or Genetic Algorithms) to these problems the authors have obtained the results that are presented in this paper.

Index Terms-- Neural networks, software tools

I. Introduction

Many system identification and control problems are based in the optimization of some parameters. In the same way, the training of a neural network consists of finding the best values for the weights of the network. Because of this similarity of methods, the authors consider the application of neural networks training methods to control problems [1][5]. As in the case of the neural networks, one of the main difficulties in this application lies in establishing the gradient algorithm formulas [2][6][7].This process usually requires long, complicated and careful calculations. There are many tools for the weight optimization of neural networks, but the most of them offers only standard architectures such as the multilayer perceptron (MLP). On the other hand, there are other tools such as the new MATLAB neural network toolbox that allows its users design and train its own neural network architecture.

However for this paper, the authors have preferred using Evenet2000, a Java-based neural network toolbox developed at the University of La Laguna, because this tool offers some features such as the possibility of designing the criterion function and its open source. This tool develops an approach to derive gradient algorithms for time-dependent neural networks, using the *Signal Flow Graph theory* [2][3]. This approach is based on a set of simple block diagram transformation and manipulation rules. Moreover, the designed structure makes it not to be limited to algorithms based on the gradient.

The aim of this paper is presenting a suite of problems where the application of these optimization methods could be useful. The authors suggest the neural network-based controllers, system identification and controller optimizations.

II. System Simulation

Evenet2000 allows simulating the evolution of a discrete system. As example, the authors have chosen an interconnected tanks system (Figure 1). This plant is a SISO system, whose input is the water flow that goes into Tank 1 (q), while height in Tank 2 (h_2) is the system output. Transfer function for this system is:

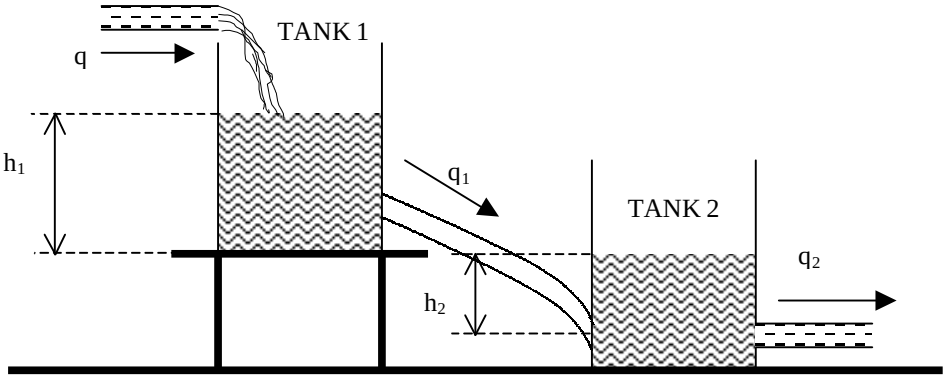

Fig 1. Interconnected Tanks System Scheme

Department of Física Fundamental y Experimental, Electrónica y Sistemas. University of La Laguna. Spain.

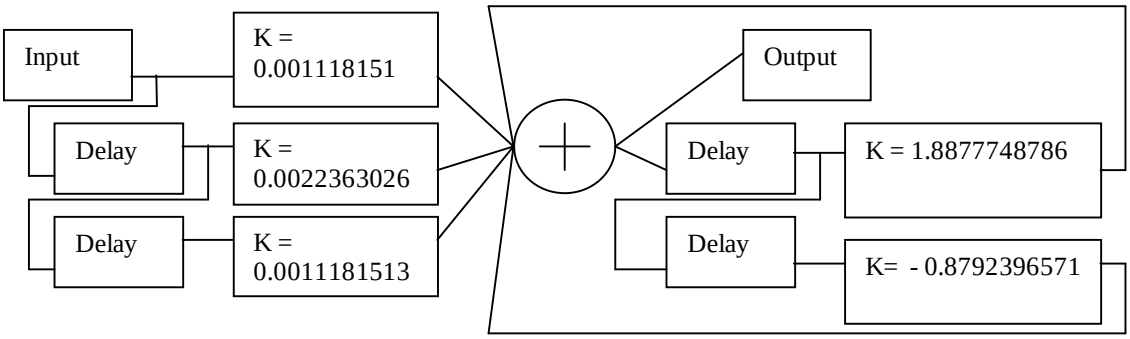

Fig. 2: Plant Implementation

$$\frac{H_2(s)}{Q(s)} = \frac{R_2}{(1+s\cdot R_1\cdot C_1)(1+s\cdot R_2\cdot C_2)+s\cdot R_2\cdot C_1} \qquad (1)$$

where R_i is the "resistance" to the pass of water in the tank i and C_i is the section of the tank i. If $R_1 = R_2 = 3$ sec·cm^{-2}, $C_1 = 10$ cm^2 and $C_2 = 7$ cm^2 and applying bilinear transformation with period T=1 sec, the following discrete transfer function is obtained.

$$\frac{H_2(z)}{Q(z)} = \frac{0.001118 + 0.002236z^{-1} + 0.001118z^{-2}}{1 - 1.8777487z^{-1} + 0.8792396z^{-2}} \qquad (2)$$

This equation can be implemented the way shown in Figure 2, using Evenet2000 modules.

With this implementation, it is possible simulating the behaviour of this system. For example, a step function as input. For this, a pattern file of 600 training pairs is taken, with input = 1, while the output do not care. Training parameters are limit of steps = 1 and learning rate = 0. Results obtained are showed in Figure 3. These results have been compared with the obtained with other programs such as MATLAB or Octave.

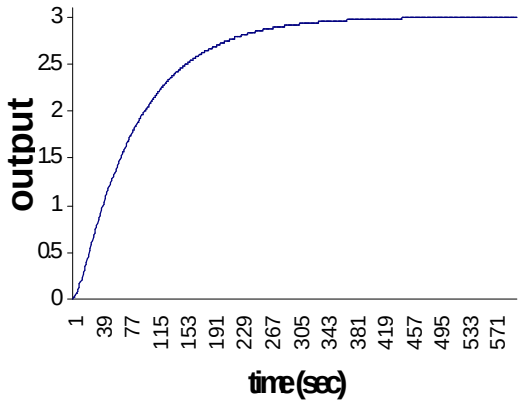

Fig. 3. Simulation of the tank system with step function as input

III. Neural Network-Based Controllers

In this section, Evenet-2000 is used for designing a neural-network based controller. As example, the tank system analysed above is used. Once the plant has been implemented through Evenet2000 modules, it is saved in a text file that can be used in other designs. The chosen control scheme is a closed-loop system, whose implementation is the shown in Figure 4, where the control system is a neural network whose scheme is shown in Figure 5. Non-standard neural network architecture has been chosen to prove Evenet2000 capacity for train any arbitrary neural network. Moreover, this design tries to give some memory to the neural network, choosing tanh(x) as neurone activation function.

Neural network input (that is the error function of the closed-loop e(k)), is delayed nine times creating nine new inputs: e(k-1),…,e(k-9). The ten inputs are connected to a five neurones layer that is fully connected to a totally interconnected three neurones layer. The output of the third neurone is included as input in the first layer. This output is weighted and translated to generate the neural network output, which is the command of the control system. As can

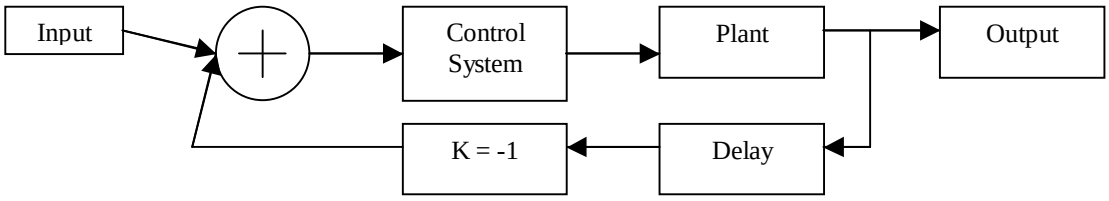

Fig. 4. Control Scheme Implementation

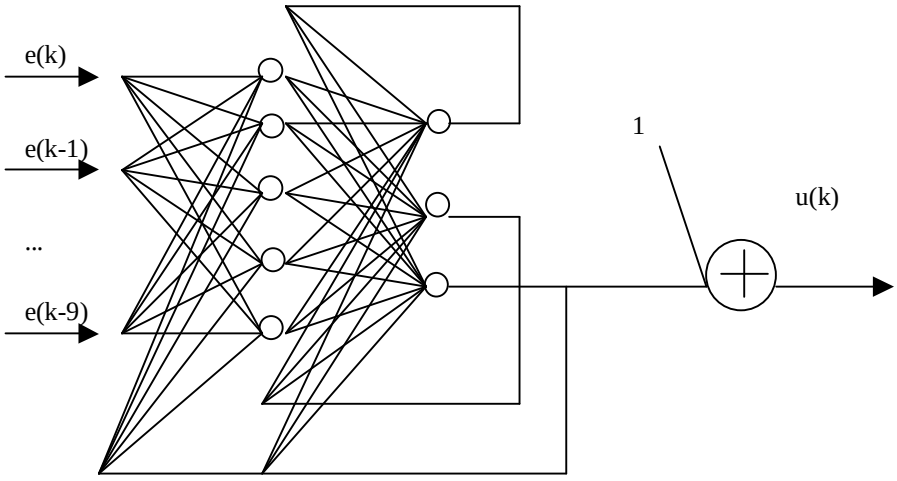

Fig. 5. Neural Network Controller Scheme

be seen, establishing the gradient algorithm formulas of this architecture is not immediate. So, it is important to remark that Evenet2000 allows its users to avoid this work.

Trying to obtain better results for the training, specially for the offset, weights have been optimised to learn the pattern obtained from the global system behaviour when the control system is a proportional-integral controller with k = 0.93, T_i = 0.00134 and input r(t)=10. Training algorithms has been again Conjugated Gradient and Golden Search as learning rate optimisation algorithm. In this case training finishes after 12000 iterations, with random initial weight set and obtained error 0.1. Goodness of this learning is tested presenting a variable reference. These references are chosen trying to offer the global system values both lower and higher than training reference. Response of the system is shown in Figure 6. As can be seen, stationary output is very closed (less than 5%) to reference value in all cases, although output is higher than reference when it is lower than training one while when reference is higher than

Table 1. Parameters of P, PI and PID controllers of a tank system obtained for several criterion functions

Controller	$J = \Sigma (e(K))^2$	$J = \Sigma \lvert e(k) \rvert$	$J = \Sigma k * \lvert e(k) \rvert$
P	K = 9.358991 First peak value: 14.855 Final value: 9.687294	K = 14.571063 First peak value: 16.585 Final value: 9.796877	K = 19.650787 First peak value: 17.965 Final value: 9.848589
PI	K = 7.031003 T_i = 3.313618e-3 First peak value: 14.074 Final value: 10.000000	K = 4.264689 T_i= 1.092376e-2 First peak value: 13.120 Final value: 10.000000	K = 2.398842 T_i= 1.228744e-2 First peak value: 11.700 Final value:10.000000
PID	K = 10.034202 T_i = 1.79263e-2 T_d =12.817345 First peak value: 12.455 Final value:10.000000	K = 11.014376 Ti = 1.114656e-2 Td = 7.48808 First peak value: 11.205 Final value:10.000000	K = 9.395098 T_i = 1.117276e-2 T_d = 7.293749 First peak value: 10.565 Final value:10.000000

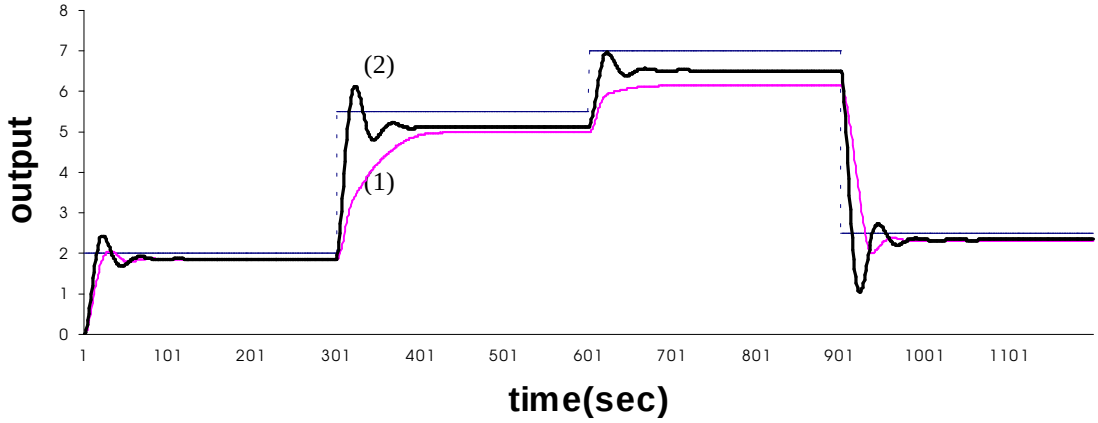

Fig. 6. Behavior of the tank system with neural network controller (1) and original PI controller (2)

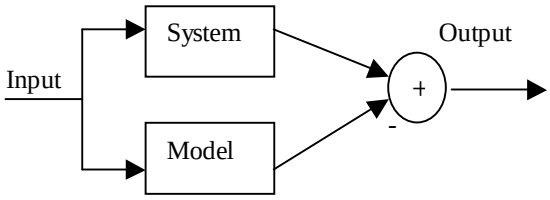

Fig. 7. Scheme in System Identification Problem

training one, opposite case occurs (stationary output is lower than reference value).

IV. System Identification

Evenet-2000 allows identifying a system with a neural network (actually, with any system with several weights to optimize). This problem is treated as an optimization problem, with 0 as desired output in the scheme shown in Figure 7. This scheme is implemented through Evenet2000 modules.

Trying to obtain a better model for the system, a sum of sinusoidal functions with different values of frequency is recommended as input. For testing the goodness of the identification, the authors recommend implementing a system simulation with an input with random values.

Evenet-2000 allows identifying a system with a neural network (actually, with any system with several weights to optimize). This problem is treatedas an optimization problem, with 0 as desired output in the scheme shown in Figure 7. This scheme is implemented through Evenet2000 modules.

Trying to obtain a better model for the system, a sum of sinusoidal functions with different values of frequency is recommended as input. For testing the goodness of the identification, the authors recommend implementing a system simulation with an input with random values.

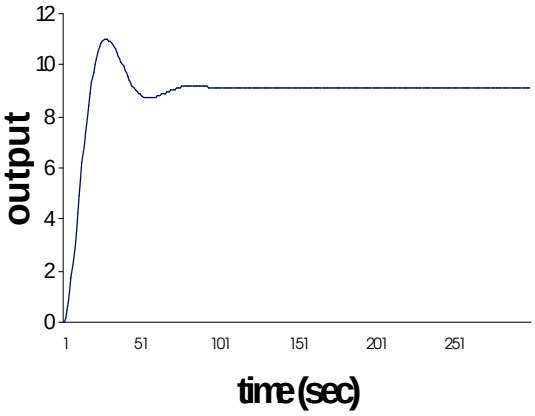

Fig. 8. Behavior of the tank system when the parameters are trained with limit =11 for the height in the second tank

V. Criterion Function

In real systems, there are usually physical limitations. For example, the command has a limit value. In the case of the tank system, for the second tank has a limited height. These situations are implemented in Evenet2000, defining a criterion function, for the variables that have these limitations, giving a high value of the criterion function when the variable is out of the defined limits. Figure 8 shows the behavior of the tank system defining a limit =11 for the height of the second tank, and consign = 10.

VI. Optimization in Control Problems

As Evenet-2000 modules are designed for optimization problems, they can be applied to control problems. For

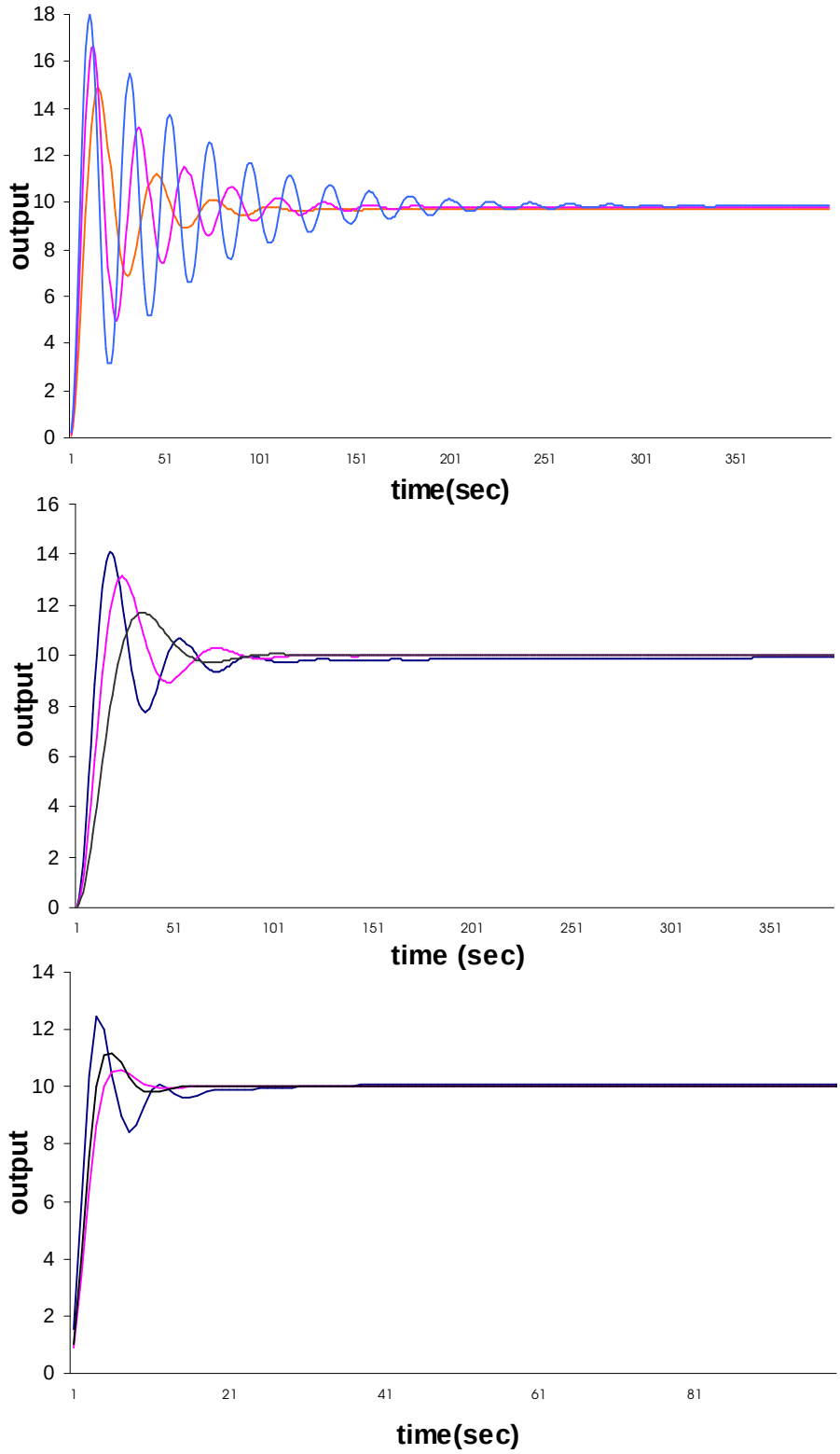

Figure 9: Behavior of the tank system controlled by a P, PI and PID controller whose parameters are obtained minimizing several criterion functions

example, finding the parameters of a PID controller that minimize a criterion function such as a quadratic in error, absolute of error and time * absolute of error.

This problem is treated as following.The training pattern consists of 1000 patterns, with the desired output equal to the consign.

As example, Figure 9 shows the behavior of the tank system detailed above, controlled by a P, PI and PID controller whose parameters are trained for minimizing several criterion functions J and consign = 10. The obtained parameters are showed in Table 1.

VII. CONCLUSIONS

The aim of this paper has been presenting a suite of problems where the application of optimization methods in Evenet2000 could be useful. The authors have detailed system simulation, neural network-based controllers, system identification, optimization in control problems and limitations in criterion functions.

REFERENCES

[1] Acosta L., Marichal G.N., Moreno L., Rodrigo J.J., Hamilton A., Méndez J.A. (1999). Robotic system based on neural network controllers. *Artificial intelligence in Engineering*, 13, number 4, pp 393-398.

[2] Campolucci P., Marchegiani A., Uncini A., Piazza F. (1997). Signal-Flow-Graph Derivation of On-line Gradient Learning Algorithms. *IEEE International Conference on Neural Networks*, Houston (USA).

[3] González E.J., Moreno L., Hamilton A., Piñeiro J.D., Marichal R., Marichal, G.N. (2000). Evenet2000: A New Java-Based Neural Network Toolkit. *Proceedings of the Second ICSC Symposium on Engineering of Intelligent Systems*, Paisley, June 2000.

[4] Gonzalez E.J., Hamilton A., Moreno L., Sigut J., Marichal R. (2001) Evenet-2000: Designing and Training Arbitrary Neural Networks in Java. *Bio-Inspired Applications of Conectionism*, Springer Verlag, Lectures Notes in Computer Science, 2085.

[5] Haykin, S. S. (1998) Neural Networks: A Comprehensive Foundation. Prentice Hall.

[6] Osowski S., Herault J. (1995) Signal Flow Graphs as an Efficient Tool for Gradient and Exact Hessian Determination. *Complex Systems*, 9, 1995.

[7] Wan E. A. and Beaufays F. (1994) Relating real-time backpropagation and backpropagation through time. An application of flow graph interreciprocity. *Neural computation*, 6, number 2, pp. 296-306.