Nonlinear Dynamics and Systems Theory, **10** (1) (2010) 65–76

# Training a Neural Network Using Hierarchical Genetic Algorithm for Modeling and Controlling a Nonlinear System of Water Level Regulation

I. Ben Omrane* and A. Chatti

*Institut National des Sciences Appliquées et de Technologie INSAT, Centre Urbain Nord BP 676 - 1080 Tunis Cedex, Tunisie*

Received: November 28, 2008; Revised: January 11, 2010

**Abstract:** In this paper, we present a new approach of Hierarchical Genetics Algorithms (HGA), and the improvement brought compared to the backpropagation algorithm for the simultaneous determination of the structure and the learning of a Multilayer Perceptron (MLP). The neural model found by the two methods are employed separately in a non-linear system for water level regulation. A comparison study will therefore be presented.

**Keywords:** *hierarchical genetic algorithms; neural networks; backpropagation algorithm; training; multilayer perceptron; optimization; modeling and controlling; nonlinear systems.*

**Mathematics Subject Classification (2000):** 49J35, 34A34 92C20.

## 1 Introduction

The use of artificial neural network is an approach that has its origins in the study of nervous tissue. In fact, the operation of an artificial neuron is by analogy with that of the nerve cell.

Neural network consists of a set of artificial neurons interconnected by weights whose values affect the behaviours of the whole structure. The rules under which the adjustment operation is carried out connections characterize the learning algorithm of network. Due to the massively parallel structure and ability to reproduce arbitrary behaviours from examples, neural networks are an interesting tool for solving various problems [1–4].

* Corresponding author: imenbo7@yahoo.fr

 

Since this learning phase is the basis of a good run, we will focus on it. Once the architecture of a neural network has been chosen, it is necessary to make learning to determine the values of weight allowing the output of the neural network to be as close as possible to the target. This learning takes place through the minimization of a function, called cost function, based on examples of the learning basis and the neural network output. This function determines the goal. This minimization can be done through several algorithms called learning algorithms.

In this work, we will mainly focus on the backpropagation algorithm and try to improve their learning process by using a new approach called Hierarchical Genetic Algorithms. This approach will operate to leave the local minima which is the disadvantage of the backpropagation algorithm. Then we move to the implementation of such NN for modeling and controlling an unit of water level regulation. The results of their implementation are compared and the advantage of HGA over backpropagation is released.

## 2 Description of the Water Level Regulation Unit

The process that will be used throughout the experiments, the block diagram of which is given in Figure 2.1, is made mainly of two tanks ($T1$ and $T2$), a drain valve which is manually controlled, a sensor level placed inside tank $T2$ and a pump controlled directly through computer.

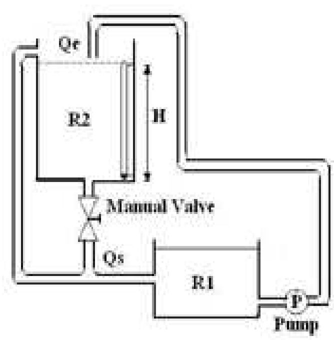


**Figure 2.1:** Description of the studied process.

The pump draws the liquid in the Tank to be conveyed to Tank $T2$ with a flow rate of $[0, 2.31]l/mn$. Depending on the liquid level $H$ in Tank $T2$, the DC motor, which controls the pump, receives an order to advance the flow of entry $Q_e$. The command signal of the pump ranges from 0 to $12v$; the conversion from analog to digital signal produces a value between 0 and 255 [8].

## 3 Presentation of Training Algorithms Used

### 3.1 Backpropagation algorithm

The learning process of the backpropagation algorithm is an iterative procedure that aims to find the weight of connections minimizing the mean square error cost function $J$

committed by the network throughout the learning date; the cost function $J$ is given by:

$$J = \frac{1}{2}\sum_{k=1}^{N}[y - y_m]^2 \tag{3.1}$$

with:
$N$: Number of examples,
$k$: samples $k, k = 1, 2, \ldots, N$,
$y$: output of the process,
$y_m$: NN output.

$W1$ define the weight matrix that characterizes the connection between input and hidden layer and $W2$ the matrix of weight between the output layer and hidden layer. The neuronal model is governed by the following equation:

$$S(k) = h(W_2 g(W_1.E(k)) \tag{3.2}$$

with :
$y_m(k)$ : output vector number $k$,
$h$ : activation function of exit neurons layer,
$g$: activation function of hidden layer,
$E(k)$ : entry vector or stimulus number $k$.
We have:
$S(k) = g(W1.E(k))$
Neural network weights are up to date as following :

$$W_{new} = W_{old} - \mu\frac{\partial J}{\partial W}, \tag{3.3}$$

where $\mu$ is the step of learning.
Variation of weight in matrix $W1$ et $W2$ is defined by :

$$\frac{\partial J}{\partial W_2} = \frac{\partial J}{\partial y_m(k)}.\frac{\partial y_m(k)}{\partial E_i(k)}.\frac{\partial E_i(k)}{\partial W_2}, \tag{3.4}$$

$$\frac{\partial J}{\partial W_2} = [y_m(k) - y(k)].\frac{\partial h(E_i(k))}{\partial E_i(k)}.\frac{\partial E_i(k)}{\partial W_2}, \tag{3.5}$$

$$\frac{\partial J}{\partial W_1} = [y_m(k) - y(k)].\frac{\partial h(E_i(k))}{\partial E_i(k)}.\frac{\partial E_i(k)}{\partial S_i}.\frac{\partial g(E_j(k))}{\partial E_j(k)}.\frac{\partial E_j(k)}{\partial W_1}. \tag{3.6}$$

This algorithm remains questionable since its convergence is not proven. Its use can lead to deadlock in a local minimum of the error surface. Its effectiveness depends mostly on a large number of parameters to be fixed by the user: the step gradient, the parameters of sigmoid functions, network architecture (number of layers, number of neurons per layer ... ), initialization of weights ...

This learning method has limitations, including:
– The topology of NN must be defined firstly;
– Very sensitive to local minima.

### 3.2 Hierarchical genetic algorithm

The HGA is used to optimize parameters and topology of NN. The advantage of this approach is that genes of chromosome are classified into two categories (hierarchy). This approach is ideal to represent the relations between:

– NN layers number;
– neurons in hidden layers;
– synaptic weights associated with genes on a chromosome.

We start by building a HGA that selects a structure of a MLP (number of neurons in the hidden layer) and then make learning. Therefore, in contrast with RPG, the HGA will go through several different structures and will make learning. To do so, it will be an evaluation function called fitness (or cost). This function is to minimize the same criterion of the backpropagation algorithm:

$$J = \frac{1}{2}\sum_{k=1}^{N}[y - y_m]^2. \tag{3.7}$$

Figure 3.1 illustrates the principle of operation of the new strategy.

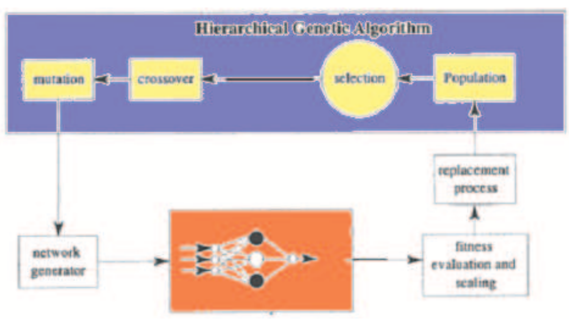


**Figure 3.1:** Principle of the new strategy.

Concerning the chromosome coding, we used a matrix instead of a vector. The first matrix's line is encoded with a sequence of "0" and "1" which indicate the existence or not of the neuron in the hidden layer. The remaining lines contain the real numbers that represent all the input connections and output neurons in the hidden layer (weights).

**Example of a chromosome coding and its equivalent in NN:**

Consider the following chromosome (Figure 3.2).

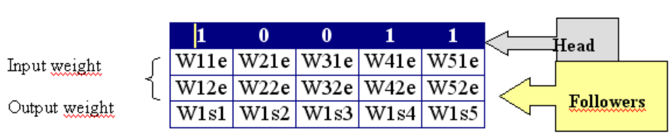


**Figure 3.2:** The chromosome Code.

Correspondence between Neurons network and chromosome is given by Figure 3.3.

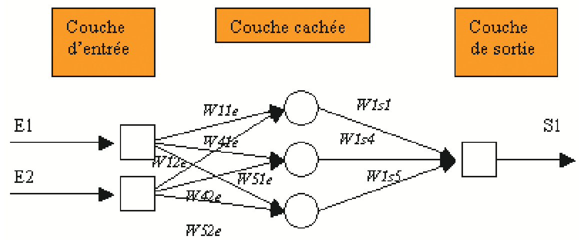


**Figure 3.3:** Translation of the chromosome toward NN.

Different genetic operators used later will allow the determination of a new structure and a new redistribution of weight.

## 4 Modeling Process

The use of learning techniques in process control can overcome the difficulties caused by the strong non-linearity. The main interest in neural networks for control is their ability to easily model non-linear systems by learning.

According to the control structure, direct model and inverse model of the process are necessary. We present in this section direct and inverse model of the unit water level regulation. These models will be used in determining the control law based on the internal model [2–4].

To obtain both the direct and inverse model of the regulation water level unit, we have excited the system by a rich signal frequency and with an amplitude that varies between 0 and 2 l/min in order to obtain an output that varies between 0 and 0.4 m. This is the sequence of learning.

We divide the database into two parts,one of which serves to learning and the other in the neuronal model validation, Figure 4.1 shows the sequence that has been used.[8]

### 4.1 Direct Neural Model ($DNM$)

The DNM builds a non-linear function that estimates the outputs of the process through old data of its inputs and outputs. In the following (Figure 4.2), a DNM is presented to be used in the sequel. The learning process of a DNM is presented, first through the backpropagation algorithm and secondly by adding our HGA.

For the backpropagation algorithm we are forced to give in advance the architecture of neural network. After several tests, we considered a non-looped network of 2 layers with 2 inputs, 10 hidden neurones in sigmoid activation function and a linear output neuron.

For HGA, we generated randomly some individuals for the first generation in which we injected the backpropagation solution. This injection has the primary effect of prohibit divergence and expulsion of the solution space.

Figure 4.3 shows the generalization error between the actual output of the process and the output of the model developed in the two cases considered. This error has the

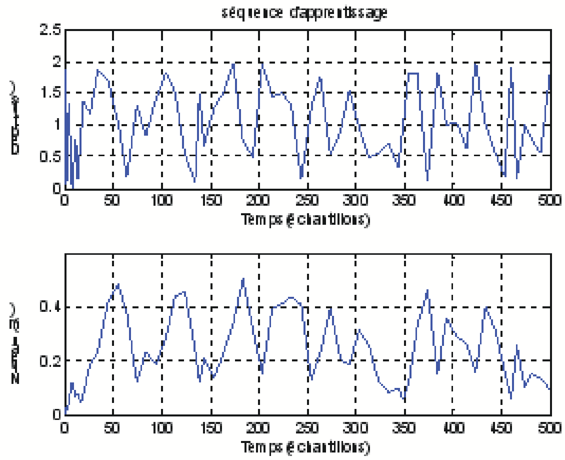


**Figure 4.1:** Sequence of training and test.

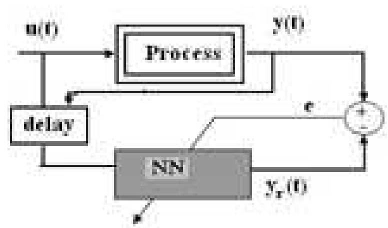


**Figure 4.2:** Direct model training.

maximum value 0.2 for backpropagation with fluctuations greater than that of HGA. Indeed, we can conclude that the result given by our HGA has considerably improved the backpropagation. So we will keep the direct model developed by our HGA.

### 4.2 Inverse neural model ($INM$)

The goal is to identify inverse model parameters through learning process; that is to find weights that render the behavior of the NN as close as possible to the desired control signal. The inverse NN model of the process is built using a NN made of 3 inputs, 10 hidden neurones with a sigmoid activation function and one neurone linear output. The learning algorithm used is the backpropagation algorithm. Most of the algorithms in NN used for learning the inverse neural model, determine the control error ($u_{ref}$-$u_r$), that is the difference between the desired reference $u_{ref}$ and the obtained control for the inverse model $u_r$. For learning the INM of level regulation unit, we applied a technique of direct

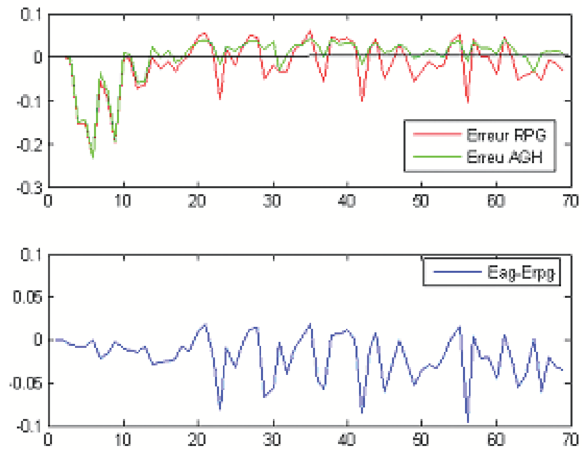


**Figure 4.3:** Variation of the error for the two methods of direct training.

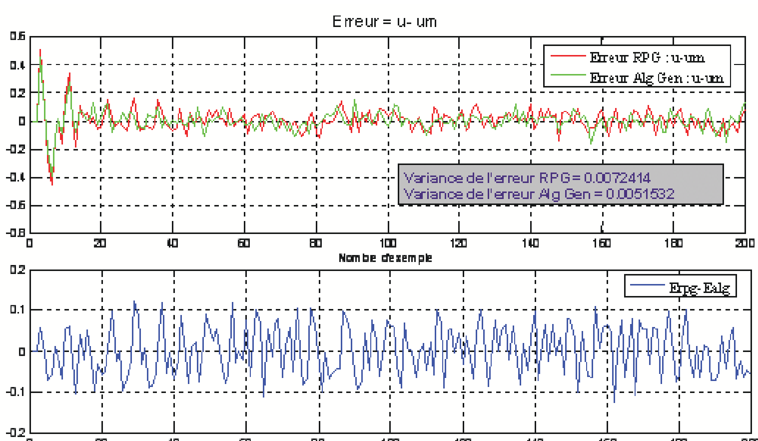


**Figure 4.4:** Prediction error after the use of backpropagation and HGA.

supervised training which minimizes the following cost criterion:

$$J = \frac{1}{2} \sum_{k=1}^{N} [u_{ref} - u_r]^2 \tag{4.1}$$

with
$N$: number of samples,
$u_{ref}$ : control signal desired,
$u_r$ : output of neural model.

The learning process of the NN is performed using the backpropagation algorithm. This algorithm assures a convergence to a minimum, it is however worth to notice that this minimum can not be a global one. To overcome this obstruction, we will introduce the HGA.

**Validation of the INM.**

In order to test the validity of the model immediately after its learning, we apply a sequence of tests to the NN then we compare the resulted tests to the desired output. Figure 4.4 shows the obtained results and the prediction error to evaluate the performance of the NN.

As for the direct model, the HGA has improved the results provided by the backpropagation.

## 5 System Neural Control

In this section, we aim to control the water-level system regulation by using the generated neural models. We will make use of the INM for the direct control of the inverse model and then we apply simultaneously the INM and DNM for the control of internal model.

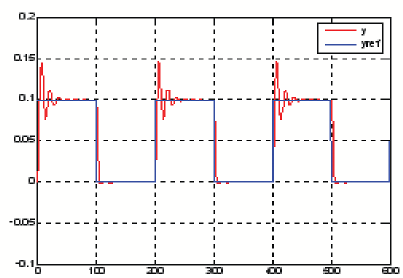


**Figure 5.1:** Water level control by INM (backpropagation).

### 5.1 Direct control of the inverse model

The principle of the control law designed for the regulation process of the water-level is to calculate every sampling step of the pump flow to reach the desired level. The inverse model, previously presented, receives reference inputs for the water level, it should then generate the appropriate control law for the pump. In the first experiment, we used the INM that we found after learning process through the method of backpropagation.

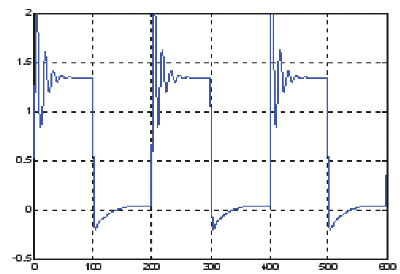

**Figure 5.2:** Control signal by INM (backpropagation).

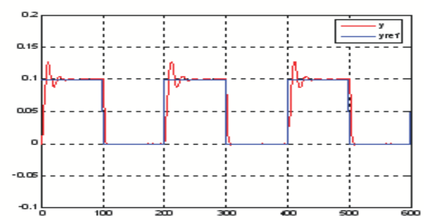


**Figure 5.3:** Water level control by INM (HGA).

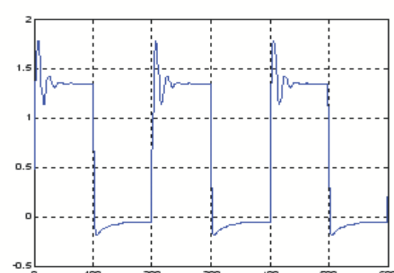

**Figure 5.4:** Control signal by INM (HGA).

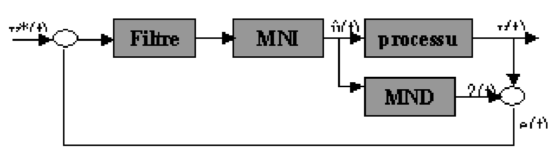


**Figure 5.5:** IMC structure.

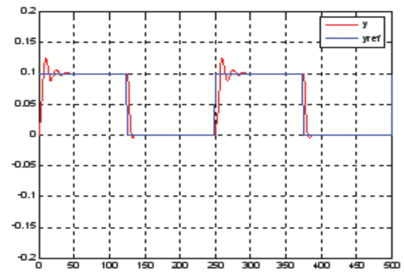


**Figure 5.6:** Water level control by IMC (backpropagation).

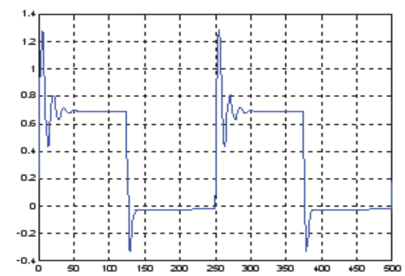

**Figure 5.7:** Control signal by IMC (backpropagation).


**Figure 5.8:** Water level control by IMC (HGA).

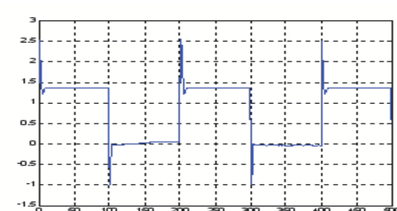

**Figure 5.9:** Control signal by IMC (HGA).

Figure 5.1 illustrates the evolution of the reference input and the output and Figure 5.2 gives the control signal generated for water level regulation.

In the second experiment, we used the INM that we established after learning process through the HGA. Figure 5.3 illustrates the evolution of the references input and the output and Figure 5.4 shows the control signal generated for the system to regulate water level.

This experience confirms even more improvements brought by the HGA with respect to the results given by the backpropagation algorithm. The first improvement is made in the modeling (a more appropriate model) and the second is confirmed at the control level, overshoot and fluctuations are less important (reduced).

### 5.2 Internal model control (IMC)

The internal model control structure, applied to our control system requires the use of inverse model as a controller and the direct model as internal model. These models are established in the previous paragraphs and are generated by two different learning algorithms, the backpropagation and HGA. When using the control law based internal model, the controller is placed in cascade with the control system, whereas the direct model is placed in parallel. The block diagram of the control law is shown in Figure 5.5.

We show the results of the simulations for the choice of the filter transfer function bellow: $F(z) = \frac{0{,}2}{z-0{,}8}$. This filter is used to eliminate fluctuations. As the first result, we present the IMC of which the INM and the DNM are generated through learning process using the backpropagation algorithm. Figure 5.6 illustrates the evolution of the reference and the output. Figure 5.7 plots the control signal generated by the IMC. Finally, we apply the inverse model control law which uses the INM and the DNM that are generated by learning through HGA. Figure 5.8 illustrates the evolution of the reference input and the output. Figure 5.9 shows the control signal generated by the IMC for our control system.

We note that the output of the control process in the case of learning through the HGA (Figure 5.8) is better than that of the backpropagation (Figure 5.6). This is evident since the HGA further minimizes the error output $(y*(t) - y(t))$ compared to backpropagation. We can say that the HGA has allowed us to leave the local minimum found by backpropagation. Indeed, we note that the response of the internal model does not oscillate, but small peak on each variation of the reference input.

## 6 Conclusion

This validation on a real system has allowed us to show that we have achieved our main objective, which is overcoming the defects of the backpropagation algorithm using a new algorithm and approach that is the HGA. Indeed this algorithm can take into account a large number of MLP and make the learning process by implementing its various operators. So we conclude that HGA meets our needs.

## References

[1] Rivals, I. *Modélisation et commande de processus par réseaux de neurones; application au pilotage d'un véhicule autonome.* Thése de Doctorat, Université de Paris 6, 1995.

[2] Rivals, I., Personnaz, L., Dreyfus, G. and Canas, D. Real-time control of an autonomous vehicle: a neural network approach to the path-following problem. In: *5th International Conference on Neural Networks and their Applications*, 1993, pp. 219–229, Nîmes.

[3] S. Bel Hadj Ali, Sur la commande par modéle interne de processus dynamiques. Thése de Doctorat, Ecole Nationale d'Ingénieurs de Tunis, 2003, Tunis.

[4] Yaich, A. *Sur la modélisation et la commande des systémes en incluant les concepts des réseaux de neurones et de la logique floue*. Thése de Doctorat, Ecole Nationale d'Ingénieurs de Tunis, 2002, Tunis.

[5] Chen, S., Billings, S.A. and Grant, P.M. Nonlinear system identification using neural networks. *Int. J. Control* **51**(6) (1990) 1191–1214.

[6] Anders,U. and Korn, O. Model selection in neural networks. *Neural Networks* **12** (1999) 309–323.

[7] Chatti, A., and Ayari, I. *Identification and control of a non linear process using a neural network approach*. e-STA, 2005.

[8] Chatti, A, Gallah, T., Ayari, I., Borne, P. and Benrejeb, M. *Modélisation et commande neuronales d'une unité de régulation de niveau d'eau*. e-STA, 2005.

[9] Borne, P., Benrejeb, M. and Haggege, J. *Les rseaux de neurone- Prsentation et application*. TECHNIP, Paris, 2007.

[10] Borne, P. Artificial neural Networks Learning. *REE* (9) (2006) p. 37.

[11] Borne, P. and Benrejeb, M. Artificial neural networks. Application to modelling and process control. *REE* (9) (2006) p. 55.