

# Modelling and control of Madeira Wine aging process using directive inverse control

Carlos Faria<sup>1</sup>, Darío Baptista<sup>2</sup>, José Carlos Marques<sup>1,3</sup> and Fernando Morgado Dias<sup>1,2,a</sup>

<sup>1</sup>*University of Madeira, Funchal, Portugal*

<sup>2</sup>*Madeira Interactive Technologies Institute, Funchal, Portugal*

<sup>3</sup>*IN - Institute of Nanostructures, Nanomodelling and Nanofabrication, Aveiro, Portugal*

**Abstract.** Madeira Wine aging processes are slow and costly due to the value invested, the storage space and the temperature control. The aging process can be speed up if the wine is kept under certain conditions with a higher temperature. To get a high quality wine, an important step is the temperature control inside the tanks to get a high quality wine. Nowadays, it is mostly done in a manual way. To improve the process, this work describes an automatic temperature control method developed and implement, with low cost, inside the tanks of a pilot scale tank. It was verified that a Direct Inverse Control is the solution that shows the best performance.

**Keywords:** wine aging process, controller, automatic control, regression models, direct inverse control.

## 1 Introduction

Madeira wine is an important product for the economy of Autonomous Region of Madeira (ARM). An important phase to achieve Madeira wine quality is aging, which improves its quality. However, it requires monitoring the temperature inside the tanks during storage. This process is costly both due to the storage space, temperature requirements and the investment (normally the investment takes 3, 5, 10 or more years of aging before it is possible to recover it). It has been shown that similar features can be achieved in shorter time if the wine is heated at specific and well controlled temperature. Based on experiments introduced in the process since the XVIII century, the heating step must be carried out at about 45°C for at least 3 months.

For this purpose, a pilot scale unit was designed and installed at the University of Madeira (UMa), with: 10 stainless steel tanks with a maximum capacity of 200 liters each and a tap water heater with a maximum capacity of 150 liters.

Figure 1 presents an image of the system installed in UMa. In the system, heating is achieved through opening a set of valves allowing the passage of hot water in a loop circuit, which meanders inside the tanks. These valves can only be commanded through open and close commands and take about 2 minutes to switch from one point to the other. The cooling is obtained by closing the valve and through ambient heat dissipation. So, the variables used in this work are the valve opening time that allows the passage of hot water (as input) and the temperature inside of tanks (as output). The use

---

<sup>a</sup> Corresponding author: morgado@uma.pt

of valve opening time was the solution found to control the system since it does only allow open and close positions.



**Figure 1.** Picture of the pilot scale system with 10 tanks.

## 2 Models for system identification

The most important part for identification is to represent the dynamics of a system through a mathematical model. In this work, two kinds of models were developed: linear and non-linear models.

### 2.1 Linear Models

Mathematically, the Linear Least Squares (LLS) is an approach fitting solution to model data for fitting. The LLS minimizes the Sum of Squared Error (SSE) estimate [1].

$$SSE = \sum_{i=1}^n \varepsilon_i^2 \quad (1)$$

Where  $n$  is the number of samples and  $\varepsilon_i$  is error of sample  $i^{th}$  given by:

$$\varepsilon_i = y_i - \hat{y}_i \quad (2)$$

Where  $y_i$  is the  $i^{th}$  value sample of system and  $\hat{y}_i$  is the  $i^{th}$  fitted value provided by the model.

### 2.2 Non Linear Models

The kind of non-linear model developed was an Artificial Neural Network (ANN). ANNs are organized in layers and, inside it, there are elements called artificial neurons which process the information. Mathematically, the artificial neuron is the sum of the product between the inputs and its weights and the result is applied to an activation function where a response is generated [2].

$$y = f\left(W_0 + \sum_{k=1}^n x_k W_k\right) \quad (3)$$

Where  $y$  is the neuron output,  $x_k$  is the  $k^{th}$  input,  $W_k$  is the weight of  $k^{th}$  input and  $W_0$  is the bias weight. To compute the weight's value a training algorithm is used (in this work the Levenberg-Marquardt [3] and Backpropagation [4] algorithms were used). Typically, the training algorithm seeks to minimize the Mean Squared Error (MSE) defined by

$$MSE = \frac{1}{n} \sum_{i=1}^n \varepsilon_i^2 \quad (4)$$

Where  $n$  is the number of samples and  $\varepsilon_i$  is given by equation 2.

### 3 The order of the ARX class models

The AutoRegressive with eXogenous class of linear model (ARX) consists in inserting past values of input and output signals into the linear model [5]. Mathematically, the ARX structure of linear models is given by:

$$y(k) + a_1 y(k-1) + \dots + a_{na} y(k-na) = b_1 y(k-nk) + \dots + b_{nb} y(k-nb-nk) \quad (5)$$

The parameters  $na$  and  $nb$  are the orders of the ARX model ( $na$  is number of past output terms and  $nb$  is the number of past input terms),  $nk$  is the delay (also called the dead time in the system).

The classes of non-linear models (such as ANN) are direct extensions of the corresponding linear classes. Thus, the same inputs and internal structure is used for an ANN. This kind of solution was proposed by several authors [6].

### 4 Control structure

In this work the following controllers were developed and simulated: Direct Inverse Control (DIC), Internal Model Control (IMC), Proportional-Integral-Derivative controller (PID), Proportional-Integral-Derivative controller with Filter (PIDF), Model Predictive Control (MPC).

#### 4.1 Direct Inverse Control

The DIC controller consists in connecting in series the inverse model and the process [7]. Figure 2 presents a DIC block diagram. If the inverse model has an order superior to one, delayed samples of control and output are feedback to the inverse model input [7].

#### 4.2 Internal Model Control

The IMC controller is similar to DIC (i.e. a serial connection of the inverse model and the process). However, the inverse model receives a signal that reflects the perturbations that affect the system (i.e. the difference between the model and the process). Figure 3 shows a block diagram of IMC controller [8].

#### 4.3 Proportional-Integral-Derivative controller with and without filter

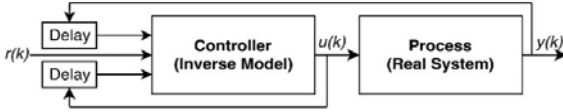
The PID is named after its three correcting terms, proportional ( $K_p$ ), integral ( $K_i$ ) and derivative ( $K_d$ ), that are added to compute the output. Figure 4 presents a block diagram of a PID controller and equation 6 is its transfer function,  $H_{PID}(s)$  [9]. The PIDF controller is the next generation to PID controller. The PIDF is similar to PID and it is complemented by a low pass filter [9]. Its transfer function,  $H_{PID}$ , is given by equation 7 where  $T_f$  is a filter constant.

$$H_{PID}(s) = K_p + \frac{K_i}{s} + K_d s \quad (6)$$

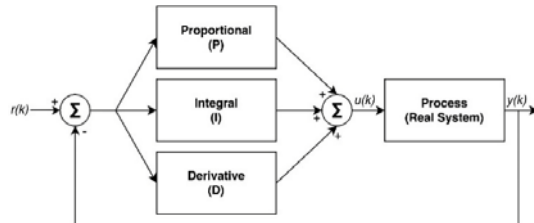
$$H_{PID}(s) = \left( K_p + \frac{K_i}{s} + K_d s \right) \frac{1}{1 + T_f s} \quad (7)$$

## 4.4 Model Predictive Control

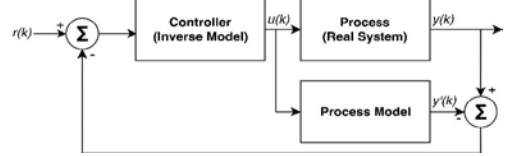
The MPC controller is used to predict the future evolution of the process to optimize the control signal. The output of the model combines with the output of the system to give a prediction of the future error of the system. This error is fed into an optimizer and it gives the predicted future inputs, which are feedback into the main model, restarting the cycle [10]. Figure 5 shows the basic structure of a MPC.



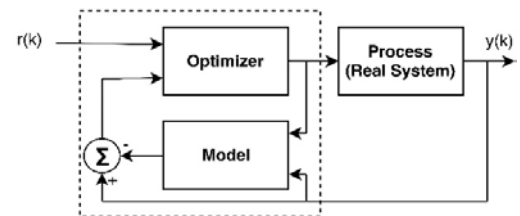
**Figure 2.** Structure for the Direct Inverse Control. The signal  $r(k)$  is the temperature introduced into the controller (i.e. set point),  $u(k)$  is the valve opening time that allows the passage of hot water,  $y(k)$  is the temperature inside the tanks



**Figure 4.** Structure for PID control.



**Figure 3.** Structure for the Internal Model Control. The signal  $r(k)$  is the set point,  $u(k)$  is the valve opening time,  $y(k)$  is the temperature inside of tanks and  $y^{\wedge}(k)$  is an estimative of the process



**Figure 5.** Structure for the MPC control.

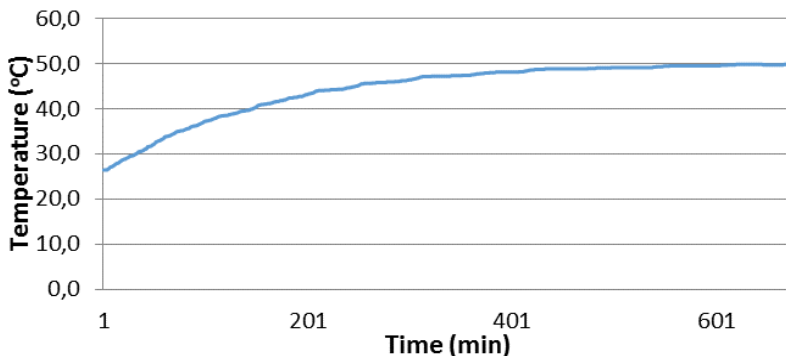
## 5 Procedure to develop the controller

### 5.1 Sampling period

According to [11] there is a rule of thumb to establish a relation between the rise time and the sampling period, given by:

$$N_r = \frac{T_r}{h} \approx 4 \text{ to } 10 \quad (8)$$

Where  $T_r$  is the time between from 10% until 90% of the rise time,  $h$  is the sampling period and  $N_r$  is the number of samples that the time rise should include.



**Figure 6.** The rise time of the system for tank n° 1.

Figure 6 shows the rise time of the system. It is starting around of 25°C because is the temperature inside the laboratory where the tanks are located. Analysing figure 10 the following can be verified: 10% of the rise time (i.e. 22th minute) the temperature is 28.8 °C and 90% of the rise time (i.e. 365th minute) the temperature is 47.6 °C.

Assuming that the maximum number of samples that  $T_r$  should include according rule of thumb (i.e.  $N_r = 10$ ), the sampling period is around 30 minutes. But, it was verified that after 30 minutes, the temperature variation between samples was quite high. To avoid the risk of losing important information, it was decided to develop models using a sampling period of 15 and 30 minutes.

## 5.2 Development of the models

The main difficulty to develop an ARX model is to know the  $na$  and  $nb$  values without increasing the complexity unnecessarily. A study about the order of system was performed to identify the best solution with less cost to implement. This function represents the price to pay for inaccuracy of predictions. Figures 7 and 8 present graphically the loss function of 1000 different combinations using the data whose sampling period was of 15 minutes and 30 minutes, respectively. It is verified that data with 15 minutes of sampling period presents a lowest cost function value than the data with 30 minutes of sampling period. So, to develop a model, the best choice is to use the data with 15 minutes of sampling period.

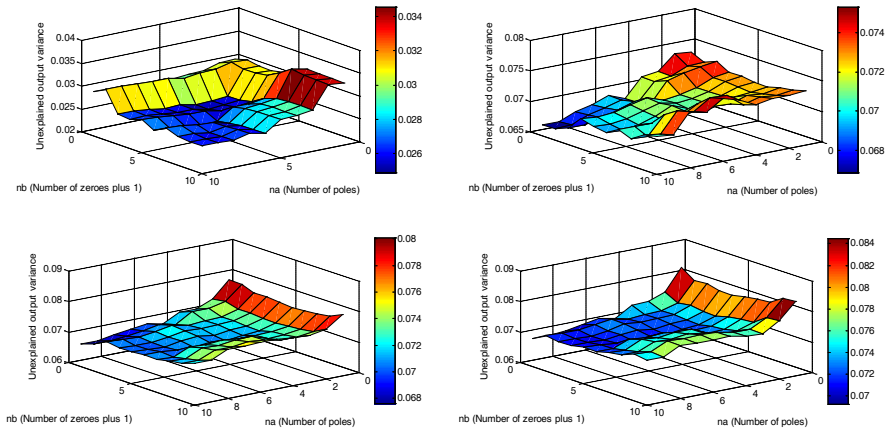


Figure 7. Loss function to identify the order (sampling period of 30 minutes).

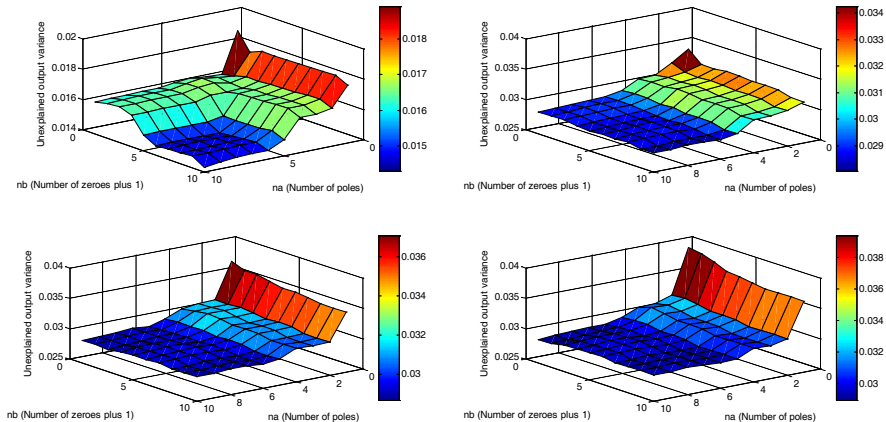


Figure 8. Loss function to identify the order (sampling period of 15 minutes).

To select the best order, within the data with 15 minutes of sampling period, it has to be taken into account the equilibrium between complexity and accuracy [12]. The way to select the order, without increasing unnecessarily the complexity, is to find the point where the loss function begins to decrease slowly. So, looking at the figures, the best choice is  $na=2$ ,  $nb=1$  e  $nk=1$ .

Table 1 presents the best fit (in %) between the output target and the output predicted. To calculate the best fit, the following equation was used the equation 9. Here, the  $Y_{target}$  is the output target and  $Y_{predict}$  is the output predicted by the model. Comparing all models developed, it is concluded that the best options to represent the system is the linear model.

$$Best\ Fit\ (\%) = 100 \times \frac{1 - \|Y_{target} - Y_{predict}\|}{\|Y_{target} - \bar{Y}_{target}\|} \quad (9)$$

**Table 1.** Best Fit (%) of all models developed for the different algorithms

			Best Fit (%)
<b>Linear Model ARX</b>			95.87%
<b>Non-Linear Model ARX (ANN)</b>	<b>4 neurons</b>	Levenberg-Marquardt	89.01%
		Backpropagation	87.23%
	<b>7 neurons</b>	Levenberg-Marquardt	80.71%
		Backpropagation	81.20%

### 5.3 Step response of the control system

The parameters to consider in the response step analysis are the rise time, settling time, overshoot, peak value, time occurrence of the peak and closed-loop stability. The main requisite for the system is to achieve faster the set point temperature. For that, it is necessary to choose the controller with a small settling time and small overshoot.

Table 2 presents the features of the step response of all controllers developed. By analysis of table 2, it was shown that IMC presents the smaller settling time but, in contrast, it presents an overshoot of 0.00364. However, the DIC presents the second smaller settling time and it has not overshoot. So, the DIC was chosen because it does not present overshoot and its settling time presents a small difference when compared with the IMC.

**Table 2.** Response step for the controllers PIDF, PID, IMC, MPC and DIC

Controller	PIDF	PID	IMC	MPC	DIC
<b>Rise time (seconds)</b>	7,20 e+04	7,20 e+04	7,20 e+04	7,20 e+04	7,20 e+04
<b>Settling time (seconds)</b>	2,38 e+05	2,49 e+05	1,24 e+05	3,60 e+05	1,28 e+05
<b>Overshoot (%)</b>	6,6	13,6	0,00364	1	0
<b>Peak</b>	1,07	1,14	1	1,01	1
<b>Time occurrence of the peak (seconds)</b>	1,55 e+05	1,51 e+05	2,00 e+05	1,48 e+05	1,80 e+05
<b>Closed-loop stability</b>	Stable	Stable	Stable	Stable	Stable

### 5.4 Simulation analysis

A simulation with all controllers was done to verify and consolidate the decision about the best controller to choose. Figure 9 presents a simulation done with each controller where the set point has a variation from 5 °C to 25°C and vice-versa.

To determine the best matching between the simulations of the system response using each controller and the set point signal, the MSE and the correlation coefficient, R, values are used. The MSE was explained in section 2.2 and it is defined using equation 4. However, the MSE does not necessarily reflect whether a line fits the data tightly because the MSE depends on the magnitude of the data samples. The correlation coefficient solves that problem because it is a measure which indicates the relation between two variables [12].

Table 3 presents the MSE and R between the set point signal and the simulations of system response for each controller. These values show that DIC presents the best solution.

6 DIC controller implementation in the real system

The controllers were tested with the real system. Figures 10 to 13 present tests with the response of real system using DIC control in tanks 7, 8, 9 and 10 respectively. A graphical interface was also developed for the users to easily monitor the temperature and set point. This interface can be seen in figures 14 to 17. Figures 14 and 15 show the management side of the graphical interface, where it is possible to define set points and monitor the temperature. Figure 16 shows the project management part, where the users and tanks can be removed, added and configured. Figure 17 shows an example of the reports that can be generated in the interface.

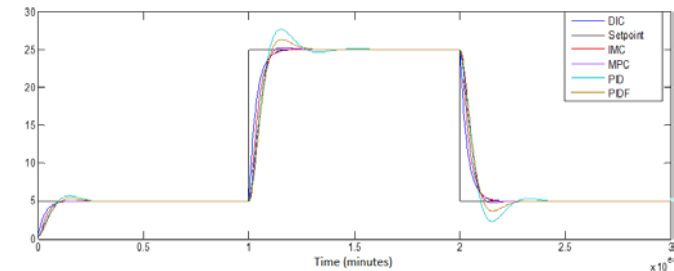


Figure 9. Simulation for the controllers PIDF, PID, IMC, MPC and DIC.9

Table 3. MSE and R between the set point signal and the simulations of system response for each controller

Controll er	MSE	R
DIC	5,4135	0,9727
IMC	8,4797	0,9572
MPC	10,4711	0,9472
PID	10,0318	0,9500
PIDF	11,7054	0,9413

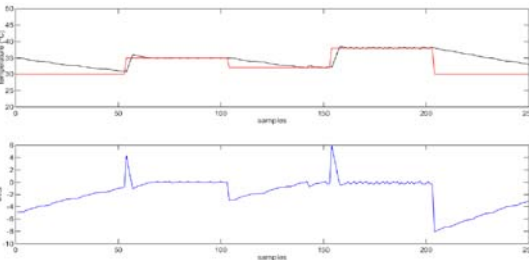


Figure 10. Comparison between the set point and the temperature inside tank 7 (red line: set point; black line: temperature inside the tank 7; blue line: error).

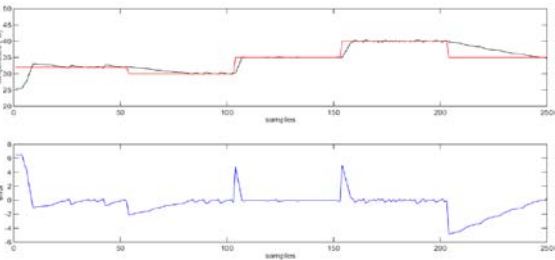


Figure 11. Comparison between the set point and the temperature inside tank 8 (red line: set point; black line: temperature inside the tank 8; blue line: error).

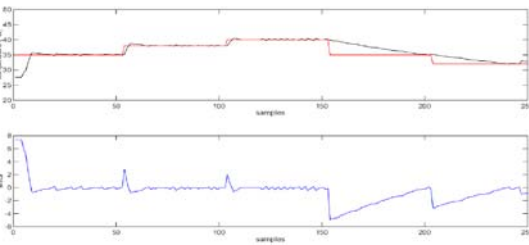


Figure 12. Comparison between the set point and the temperature inside tank 9 (red line: set point; black line: temperature inside the tank 9; blue line: error).

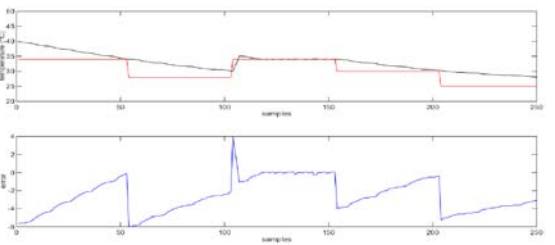


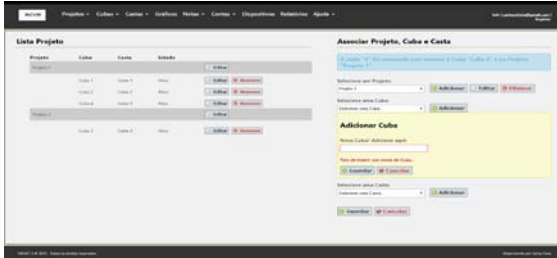
Figure 13. Comparison between the set point and the temperature inside tank 10 (red line: set point; black line: temperature inside the tank 10; blue line: error).



**Figure 14.** General view of the tank's management interface.



**Figure 15.** General view of a specific tank's management interface.



**Figure 16.** Project management interface



**Figure 17.** Reports interface

## 7 Conclusions

In this work a controller was developed to regulate the temperature inside the tanks to accelerate the aging process of Madeira wine. Before developing a model to represent the system's behavior with good performance, data collection was performed. To do that, a sampling period was chosen by analyzing the system's rise time. It was concluded, according rule of thumb, that the sampling period should be around 30 minutes. But, models were also developed using 15 minutes as a sampling period to make sure that important information was not lost. These showed to have a better performance.

Then, using the data collected, it was concluded that to get a model, to represent the system behavior with good performance, should be used data whose sampling period is 15 minutes and the model order was be  $na=2$ ,  $nb=1$  and  $nk=1$ . After that, a linear and non-linear models was developed using the data with 15 minutes sampling period collected previously. For this system, a linear model presents a better result (95.87% of matching) and it was used to develop the controller.

Several controllers were simulated. The DIC was chosen because it does not present overshoot and its settling time ( $1.28e+05$ ) presents a small difference when compared with the IMC. Besides, with the DIC, there is a good matching between the response of real system and the set point temperature.

## Acknowledgements

Acknowledgments to Portuguese Foundation for Science and Technology for their support through project PEst-OE/EEI/LA0009/2011.

Acknowledgments to ARDITI - Agência Regional para o Desenvolvimento da Investigação Tecnologia e Inovação through the support provided by the FSE - Madeira 14-20.

## References

1. B. Hassibi, A. H. Sayed, T. Kailath, *Indefinite-Quadratic Estimation and Control: A Unified Approach to H2 and H-infinity Theories* (Philadelphia, SIAM, 1999).



2. S. Wright, M. Tshilidzi, Artificial Intelligence Techniques for Steam Generator Modelling, *School of Electrical and Information Engineering*, (2007).
3. D. Baptista, F. Morgado-Dias, Comparing different implementations for the Levenberg-Marquardt algorithm, *10th Portuguese Conference on Automatic Control*, 15-20 (2012).
4. M.N.H. Siddique, M.O. Tokhi, Training Neural Networks: Backpropagation vs. genetic algorithms. Proceedings. International Joint Conference on Neural Network 2001-IEEE, **4**, 2673-2678 (2001).
5. T. Ozaki, *Time Series Modelling of Neuroscience Data*. (New York, CRC Press, 2012).
6. K. S. Narendra and K. Parthasarathy, Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, **1**(1), 4-27 (1990).
7. F. Morgado-Dias, A. M. Mota, Direct Inverse Control of a Klin. *4<sup>th</sup> Portuguese Conference on Automatic Control*, Guimarães, Portugal (2000).
8. Q. Gang, S. Le, H. Ling, A Practical Application of IMC-PID Controller in Unmanned Vehicle. TELKOMNIKA, *Indonesian Journal of Electrical Engineering*. **11**(6), 3228-3235, (2013).
9. G. A. Hassan, M. A. Al-Gamil, M. M. Lashin, Tuning of a PIDF Controller Used With a Highly Oscillating Second Order Process. *International Journal of Emerging Technology and Advanced Engineering*, **3**(3), 943- 945, (2013).
10. Y. Wang, S. Boyd, Fast Model Predictive Control Using Online Optimization. *IEEE Transactions on Control Systems Technology*. **18**(2), 267-278 (2010).
11. K. Åström; B. Wittenmark. *Computer Controlled Systems* (New York, Prentice Hall, 1997).
12. F. M. Morgado Dias, Técnicas de controlo não-linear baseadas em Redes Neurais: do algoritmo à implementação. *PhD Dissertation of Aveiro University* (2005).
13. J. C. Príncipe, N. R. Euliano, W. C. Lefebvre. *Neural and Adaptive System: Fundamentals through Simulations* (New York, Wiley, 1999).