Automatic Control of Madeira Wine Aging Process

Carlos M. Faria, Darío Baptista, José Carlos Marques and Fernando Morgado-Dias

Abstract Madeira wine aging processes are slow and costly. Besides, the value invested, the storage space, the energy costs and temperature control must also be considered. It has been shown that the aging process can include a heating step under controlled conditions, which can decrease the time necessary to achieve the characteristics of a typical fortified wine. The temperature control inside the tanks in the Madeira wine storage, mostly carried out manually, is the most important step to achieve a high quality wine. This work describes the procedure developed to implement, at low cost, an automatic temperature control inside the tanks in a pilot scale unit. Five different controllers were tested, where the Direct Inverse Control was the one that shows the best performance.

Keywords Madeira wine aging process · Automatic control · Artificial Neural Networks · Direct Inverse Control

1 Introduction

Madeira wine is an important product for the economy of Madeira. An important step to achieve Madeira wine quality is aging, which improves quality and general characteristics, but it requires monitoring during storage, mainly the temperature inside the tanks. This process is costly both due to the investment that can only be recovered after 3, 5, 10 or more years, the storage space and temperature requirements. It has been shown that similar characteristics can be achieved in shorter time if the wine is heated at specific and well controlled temperature. Based on exper-

C.M. Faria · J.C. Marques · F. Morgado-Dias (✉)
University of Madeira, Funchal, Portugal
e-mail: morgado@uma.pt

D. Baptista · F. Morgado-Dias
Madeira Interactive Technologies Institute, Funchal, Portugal

J.C. Marques
I3N - Institute of Nanostructures, Nanomodelling and Nanofabrication, Aveiro, Portugal

P. Garrido et al. (eds.), *CONTROLO 2016*, Lecture Notes in Electrical Engineering 402, DOI 10.1007/978-3-319-43671-5_66

Fig. 1 Image of the system with 10 tanks

iments 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 equipped with 10 stainless steel tanks (maximum capacity of 200 L each) and a tap water heater (maximum capacity of 150 L) was designed and installed at the University of Madeira. 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 (on-off valves). The maximum temperature of the heating system is 60 °C. The cooling is done through ambient heat dissipation keeping the valve closed. Figure 1 shows an image of this system.

The implementation of an automatic control system will allow a careful and efficient management and, consequently, a decrease of cost of this process.

2 Selection of the Sampling Period

To identify and control the system, an appropriate sampling period needs to be defined based on the rise time. By a rule of thumb, a relation between the rise time and the sampling period can be establish [1] and is given by:

$$N_r = \frac{T_r}{h} \approx 4 \text{ to } 10 \tag{1}$$

where T_r is the time between the 10 % up to 90 % of the rise time, h is the sampling period and N_r is the number of samples that the rise time should include.

Applying a step input in the system and taking into account the maximum number of samples that the time rise should include according rule of thumb (i.e. $N_r = 10$), the sampling period is approximately 30 min. However, it was verified

that within 30 min, the temperature variation can be too high. In consequence, in order to avoid the risk of losing important information, it was decided decrease the sampling period to 15 min (the half-value obtained by rule of thumb).

3 Development of ARX Models for System Identification

Two types of samples are used in this work namely the valve opening time that allows the passage of hot water and the temperature inside of tanks. The models will be tested against real data to verify the accuracy of prediction. For that, the dataset was divided into two sets: the train dataset and the validation dataset. 70 % of the data for training, i.e. it was used to identify the physical system, (750 samples) and the next 25 % of the data for test (250 samples). This method allows capturing enough information to develop models with a good performance to predict the behavior of the system.

Also, two different types of models were developed: linear and non-linear. On the one hand, the linear least squares was used to estimate the parameters of a linear model. Mathematically, the linear least squares is an approach fitting solution to model data. The best approximation is defined by the minimum value of sum of squared difference between the data values and their corresponding modeled values [2].

On the other hand, a non-linear model was developed, using an artificial neural network (ANN). ANNs are organized with layers and, inside it, there are elements called artificial neurons which process the information [3]. The Levenberg-Marquardt [4] and Backpropagation [5] algorithms were used to train the ANN. To evaluate these two training algorithms, ANNs with 4 and 7 hidden neurons were developed.

Several class models were tested. However, only the AutoRegressive with eXogenous class model (ARX) results are presented because it was the one, which presents best performance. The ARX models consists in inserting past values of input and output signals [6].

3.1 Order of the Models

To develop a model ARX, for a specific problem, it is necessary to know the number of inputs that the model should have. The main difficulty is to know how many past inputs are need without increasing the complexity unnecessarily. There is not an exact answer to solve this problem, but there are solutions that attempt to solve it [6].

A study about the order of system was performed to identify the best solution with less cost to implement but it allows the model to represent the system with high level of confidence. Figure 2 presents graphically the loss function (also known as cost

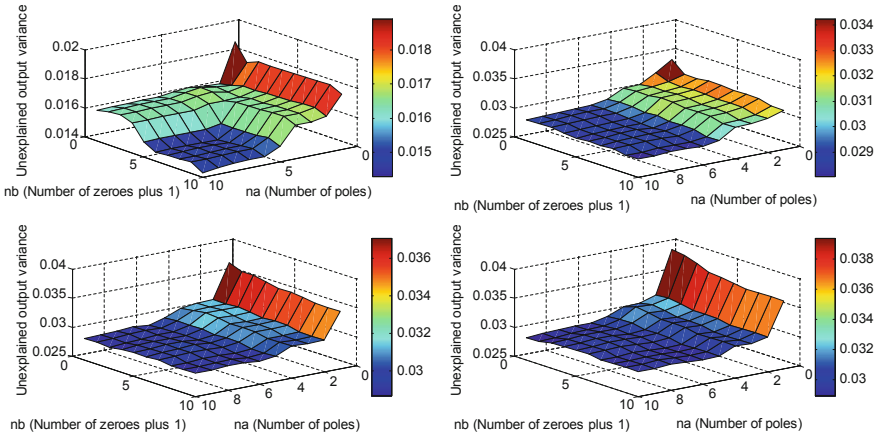

Fig. 2 Loss function to identify the order

function) of 1000 different combinations. This function represents the price to pay for inaccuracy of predictions. 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).

To select the best order, it should take into account the equilibrium between the complexity and the accuracy [6]. The way to select the order, without increase unnecessarily the complexity, is where the loss function begins to decrease slowly. So, looking the Fig. 2, the best choice is $na = 2$, $nb = 1$ e $nk = 1$.

3.2 Analysis of the Developed Models

Table 1 shows the percentage of matching between the output target and the output predicted by the models, the following equation was used:

$$Best\ Fit\ (\%) = 100 \times \frac{1 - \left\| Y_{target} - Y_{predict} \right\|}{\left\| Y_{target} - \overline{Y}_{target} \right\|} \tag{2}$$

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

			Best Fit (%)
Linear model ARX			95.87
Non-linear model ARX (ANN)	**4 neurons**	Levenberg-Marquardt	89.01
		Backpropagation	87.23
	7 neurons	Levenberg-Marquardt	80.71
		Backpropagation	81.20

where the Y_{target} is the output target and $Y_{predict}$ is the output predicted by the model and $\overline{Y}_{target}$ is the output target mean.

ANNs with different hidden neuron were developed. Here, it is presented only the two ANNs that had best performance. Comparing all models developed (linear and non-linear), it is concluded that the best options to represent the system is the linear model.

4 Development of the Control Structure

Different kinds of controllers were developed and tested, namely the Direct Inverse Control (DIC), the Internal Model Control (IMC), the Proportional-Integral-Derivative controller (PID), the Proportional-Integral-Derivative controller with Filter (PIDF) and the Model Predictive Control (MPC).

The DIC controller is an easy method to make control. It consists in connecting in series the inverse model and the process, as can be seen in Fig. 3. If the inverse model is of order superior to one, delayed samples of control and output are feedback to the inverse model input [7].

The IMC controller consists in a serial connection of the inverse model and the process. However, the inverse model received, as input, a signal. This signal reflects the perturbations that affect the system and the differences between the model and the process. Figure 4 shows a schematic of IMC controller [8].

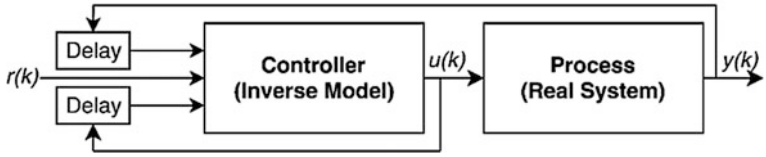

Fig. 3 Structure for direct inverse control. The signal $r(k)$ is the temperature set point inserted into the controller, $u(k)$ is the valve opening time that allows the passage of hot water, $y(k)$ is the temperature inside of tanks

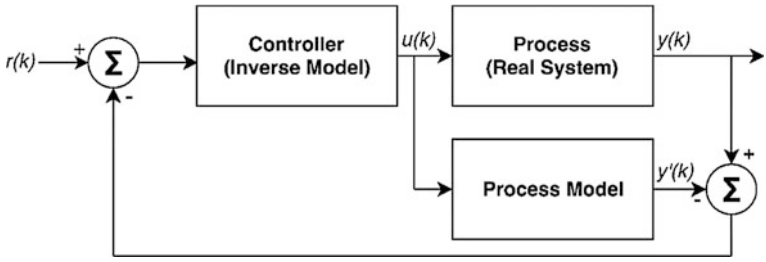

Fig. 4 Structure for internal model control. The signal $y^{\wedge}(k)$ is an estimative of the process

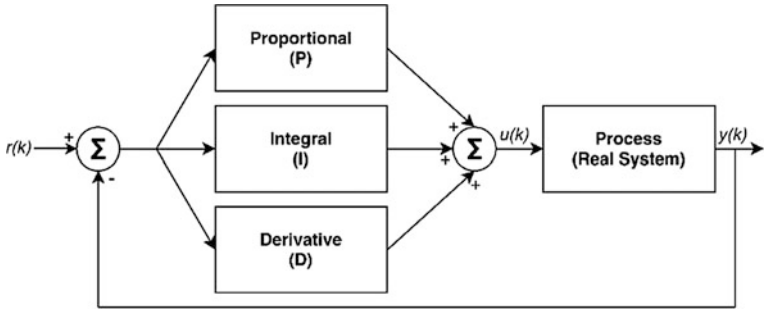

Fig. 5 Structure for PID control

The PID is named after its three correcting terms (proportional (K_p), integral (K_i), and derivative (K_d)) that are summed to calculate the output. Figure 5 shows a schematic of PID controller and Eq. 3 is its transfer function [9].

$$H_{PID}(s) = K_p + \frac{K_i}{s} + K_d\, s \tag{3}$$

The PIDF controller is the next generation to PID controller and the difference is a low pass filter to complement it [9]. Its transfer function, $H_{PIDF}(s)$ is given by:

$$H_{PIDF}(s) = \left(K_p + \frac{K_i}{s} + K_d\, s\right)\frac{1}{1 + T_f\, s} \tag{4}$$

where, T_f is a filter constant.

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 fed back into the main model, restarting the cycle [10]. Figure 6 shows the basic structure of a MPC.

Fig. 6 Structure for MPC control

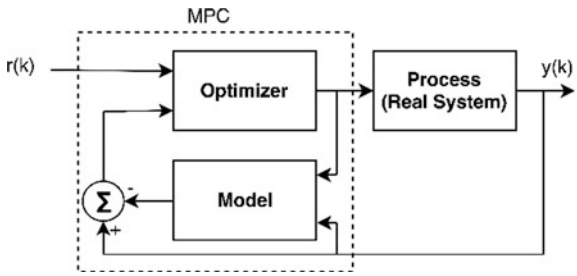

Table 2 Response step for the controllers PIDF, PID, IMC, MPC and DIC

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

4.1 Response Step Analysis

The parameters to consider in the response step analysis are the rise time, settling time, overshoot and time occurrence of the peak. Table 2 presents all controllers used.

The controller was developed using ARX models with 95.87 % of matching with the real system. When the model is replaced by the real system, the control will be present some mismatches. So, to minimize this effect, the controller with lowest overshoot was chosen. As a second requirement, the controller with a small settling time was chosen to achieve faster the set point temperature. The main requisite for the system is a small settling time to achieve faster the set point temperature and small overshoot. So, analyzing Table 2, it was verified that IMC controller presents a small settling time. However, looking at the value of overshooting it was verified that IMC controller presents an overshoot (0.00364) and DIC controller do not present it. So, the controller DIC was chosen because it does not present overshoot and its settling time presents only a difference of 4000 s comparing with the IMC.

4.2 Simulation Analysis

A simulation with all controllers was done to verify and consolidate the decision about the best controller to choose. Figure 7 shows the simulations done with each controller where it was held a set point variation from 5 °C to 25 °C and vice versa.

To determine the best relationship between the simulations of the system response using each controller and the set point signal, the Mean Square Error (*MSE*) and the correlation coefficient (*R*) values are used. The *MSE* is a measure to estimate the average of the squares of the difference between the desired response and the set point [11]. It is defined by:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(Y_{target} - Y_{predict}\right)^2 \tag{5}$$

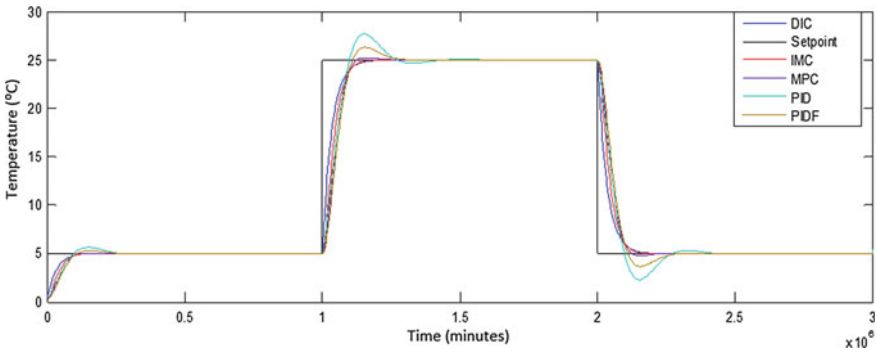

Fig. 7 Simulation for the controllers PIDF, PID, IMC, MPC and DIC

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

Controller	MSE of simulation	R of simulation
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

where N is the number of samples. However, it does not necessarily reflect whether a line fits the data tightly because the *MSE* depends on the magnitude of the data samples. The coefficient R solves that problem. R is a measure which indicates the linear relation between two variables [11]. By definition, the correlation coefficient is:

$$R = \frac{Cov(Y_{target},\ Y_{predict})}{\sigma_{Y_{target}}\sigma_{Y_{predict}}} \tag{6}$$

where σ is the standard deviation and Cov the covariance. Table 3 shows the values of MSE and R between the set point signal and the simulations of system response using each controller.

5 Controller Implementation in the Real System

The controllers were tested with the real system. Table 3 shows the values of MSE and R between the set point signal and the response of physical system using DIC (Table 4).

Figures 8 and 9 show two different responses of the real system using DIC control. As expected, there are disturbances and the results are slightly worse than the simulation tests.

Table 4 MSE and R values between the set point signal and the response of physical system using DIC for the tank 10 and 7

	MSE	R
Tank 7	3,0528	0,9028
Tank 10	3,8748	0,8987

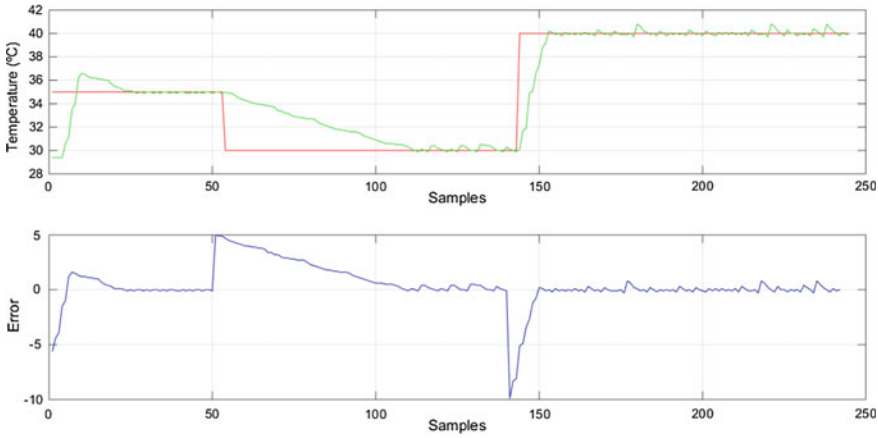

Fig. 8 Comparison between the set point temperature and the temperature inside the tank 10

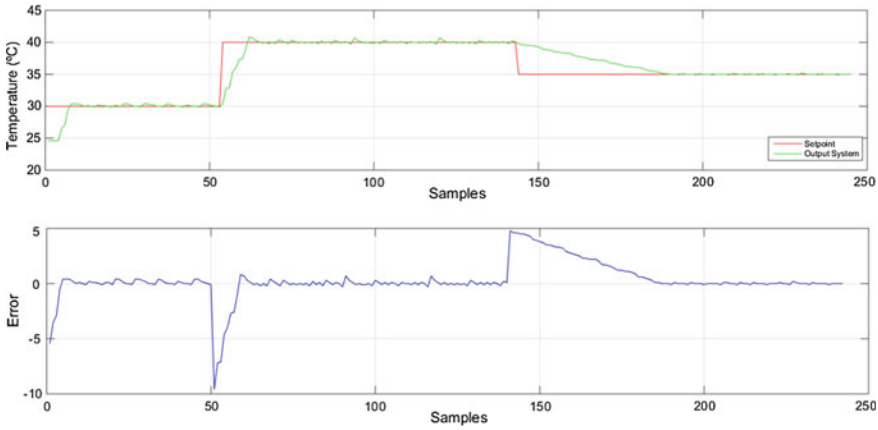

Fig. 9 Comparison between the set point temperature and the temperature inside the tank nº 7

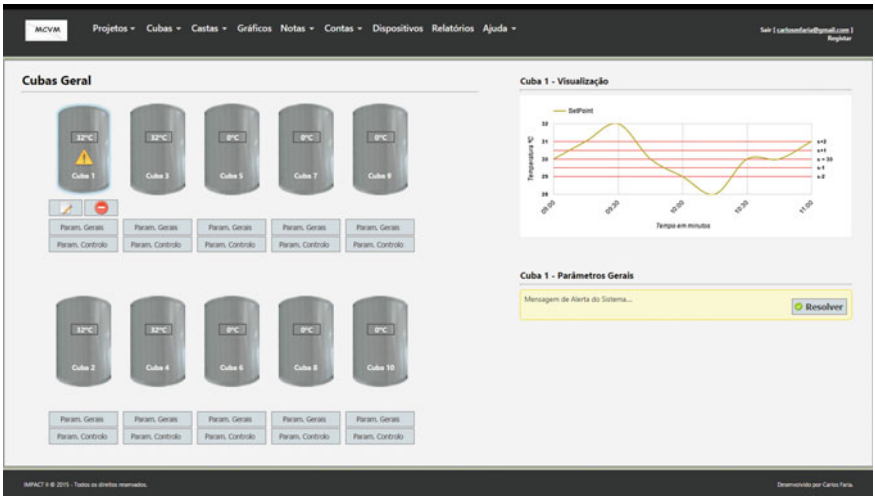

Fig. 10 General view of interface of tanks management interface

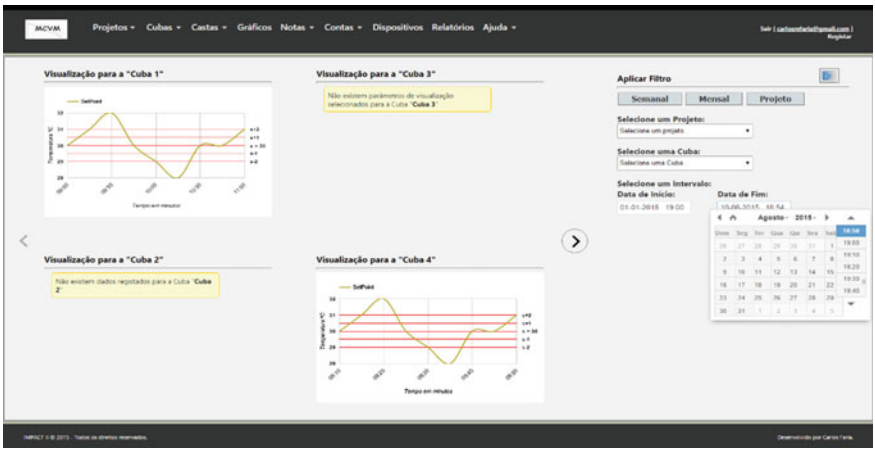

Fig. 11 General view of interface for a specific tanks management interface

A graphical interface was also developed for the users to easily monitor the temperature and set point. This interface can be seen in Figs. 10 and 11.

6 Conclusions

In this work a controller was developed to regulate the temperature inside of tanks to accelerate the aging process of Madeira wine. Two different kinds of models were developed with ARX class: linear and non-linear models. For this system, a

linear model presents a better result (95.87 %) and it was used to develop the controller. After that, several controllers were simulated. It was concluded that the controller DIC has a better performance because it presents a low settling time and a null overshoot. Besides, with this controller, there is a good matching between the response of physical system and the set point temperature (For tank 7: MSE = 3,0528 and R = 0,9028; For tank 10: MSE = 3,8748 and R = 0,8987).

Acknowledgments 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. Åström, K., Wittenmark, B.: Computer Controlled Systems. Prentice Hall (1997)
2. Valença, M.R.: Métodos Numéricos, 3th edn. Ferreira e Salgado, Press LLC (1993)
3. Dias, F.M., Antunes, A., Vieira, J., Mota, A.: A sliding window solution for the on-line implementation of the Levenberg-Marquardt algorithm. Eng. Appl. Artif. Intell. **19/1**, 1–17 (2006). IFAC
4. Baptista, D., Morgado-Dias, F.: Comparing different implementations for the Levenberg-Marquardt algorithm. In: 10th Portuguese Conference on Automatic Control (2012)
5. Siddique, M.N.H., Tokhi, M.O.: Training neural networks: backpropagation vs. genetic algorithms. In: Proceedings. IJCNN '01, vol. 4, pp. 2673–2678. IEEE (2001)
6. Morgado Dias, F.M.: Técnicas de controlo não-linear baseadas em Redes Neuronais: do algoritmo à implementação. Tese Universidade de Aveiro – Departamento de Electrónica e Telecomunicações (2005)
7. Morgado-Dias, F., Mota, A.M.: Direct inverse control of a Klin. In: 4th Portuguese Conference on Automatic Control. Guimaraes, Portugal (2000)
8. Gang, Q., Le, S., Ling, H.: A Practical application of IMC-PID controller in unmanned vehicle. TELKOMNIKA Indonesian J. Electr. Eng. **11**(6), 3228–3235 (2013)
9. Hassaan, G.A., Al-Gamil, M.A., Lashin, M.M.: Tuning of a PIDF controller used with a highly oscillating second order process. Int. J. Emerg. Technol. Adv. Eng. **3**(3) (2013)
10. Wang, Y., Boyd, S.: Fast model predictive control using online optimization. IEEE Trans. Control Syst. Technol. **18**(2), 267–278 (2010)
11. Principe, J.C., Euliano, N.R., Lefebvre, W.C.: Neural and Adaptative System: Fundamentals Through Simulations. Wiley, New York (1999)