Editorial Manager(tm) for International Journal of Advanced Manufacturing Technology
Manuscript Draft

Manuscript Number: IJAMT6621R1

Title: Object-oriented optimal controller for a batch dryer system

Article Type: Original Research

Keywords: Object-oriented programming; time delay systems; optimal control; batch dryer process.

Corresponding Author: Omar López-Ortega, PhD

Corresponding Author's Institution: Universidad Autónoma del Estado de Hidalgo

First Author: Liliam Rodríguez-Guerrero, M. S.

Order of Authors: Liliam Rodríguez-Guerrero, M. S.;Omar López-Ortega, PhD;Omar Santos-Sánchez, PhD

Abstract: We describe the synthesis of an optimal control law and its implementation using object-oriented programming in the Java language. Our aim is to optimize the performance of a batch dryer system, which presents several problems such as time delay in the input and overshoot. The experimental results indicate that the control law effectively reduces the energy applied to the actuators in order to reach and maintain the desired temperature of the process. As for the implementation in the Java language, it allows to use the control law regardless the operating system, while providing robustness and stability.

Response letter

David W Russell, PhD, FIEE

Regional Editor

International Journal of Advanced Manufacturing Technology

Dear Professor Russell,

This is to inform you about the corrections that have been made to candidate article "Object-oriented optimal controller for a batch dryer system", reference number IJAMT-S-10-01793.

Comment 1. References 4, 5, 7, 8, 10, 11, 29 (of the first submitted version) are hard to find.

As this regard, bibliography is now complete and traceable.

Comment 2. There is no error analysis reported between the theoretical and experimental studies.

The corrected version now contains subsection "Theoretical analysis of the expected error". We present a detailed description of the divergence between the implemented control law and its theoretical solution. Our analysis predicts that such deviation is bounded exponentially. This finding is corroborated in the experiments we performed.

Comment 3. There is no merit reporting the benefits of completed research for advanced manufacturing field.

In the corrected version, we include Section "Temperature control and manufacturing of dried products", where the importance of the control law is framed into the context of advanced manufacturing systems.

Comment 4. Research reports should be benchmarked with the findings of former researchers.

In the corrected version, we include more references and the findings are described within the importance of our experiment in the manufacturing field.

Comment 5. In 4.4 (of the first submitted version) some experimental validations are reported. However, any reader can not clearly draw these validations and results.

In the corrected version, the experimental results are explained with more detail. While the original version contains general validations for the three experiments, in this corrected version each experiment is explained. Also, we validate the expected theoretical deviation (comment 2) with experimental data.

Finally, we appreciate the reviewer's comments, which were helpful to improving both our research and the article we are submitting. With our best regards,

Omar López-Ortega

Corresponding author.

Object-oriented optimal controller for a batch dryer system

Liliam Rodríguez-Guerrero[1], Omar López-Ortega[2] and Omar Santos-Sánchez[3]
Área Académica de Sistemas Computacionales,
Universidad Autónoma del Estado de Hidalgo,
Carretera Pachuca-Tulancingo, km. 4.5, C. U.,
C. P. 42084 Pachuca, Hidalgo, México
e-mail : [1]profe_7492@uaeh.edu.mx, [2]lopezo@uaeh.edu.mx, [3]omarj@uaeh.edu.mx

February 14, 2011

Abstract

We describe the synthesis of an optimal control law and its implementation using object-oriented programming in the Java language. Our aim is to optimize the performance of a batch dryer system, which presents several problems such as time delay in the input and overshoot. The experimental results indicate that the control law effectively reduces the energy applied to the actuators in order to reach and maintain the desired temperature of the process. As for the implementation in the Java language, it allows to use the control law regardless the operating system, while providing robustness and stability.

Keywords: Object-oriented programming, time delay systems, optimal control, batch dryer process.

1 Introduction

A great number of industrial systems belong to the class of batch processes, where variables such as temperature, humidity, or pressure must be controlled to obtain a good performance of the plant. Batch dryers, particularly, are used to dehydrate vegetables and food, or to dry paper and plastic. Consequently, it is pivotal to control the temperature of the process, for which advanced techniques are being tested. Implementation of predictive control to regulate temperature is found in [1], [2]. Nevertheless, real-life processes still operate manually, and in some cases manual and automatic modes are used jointly [3].

Figure 1 displays a typical batch dryer. They normally possess an air-heater, which uses steam as heating media. The control system uses any given temperature sensor (TT). An air-to-open valve (TV) is the regulating element to avoid overheating of the product. A proportional-integral controller (TRC) compensates load changes due to variations in air flow rate. A hand switch (HS) activates a solenoid valve (HY) in the air line between the controller and valve. When HS is open, HY vents air from the valve motor closing TV. When HS is closed, HY is switched to close the vent and connect the controller output directly to the valve. Once the product is loaded in the dryer, automatic start-up is initiated when the operator closes the hand switch on the control panel.

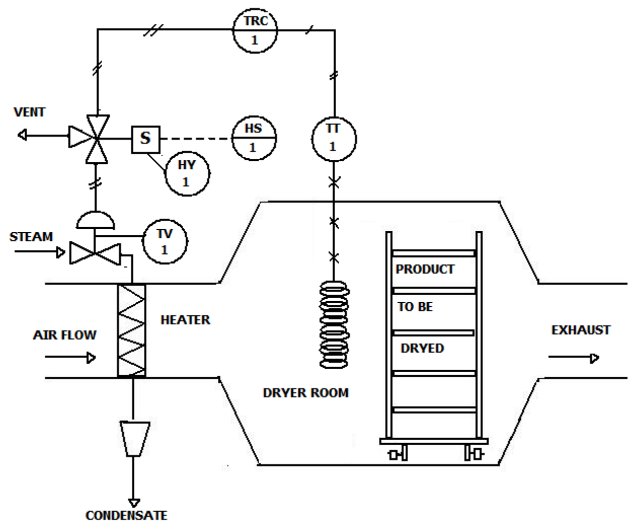

Figure 1: Batch dryer system

Firstly, the dryer is loaded with product. This is done at room temperature. Then, temperature is increased and it is held high during a pre-established time. By then it is assumed that the product is already dry and can be extracted. If it is assumed that a batch has been completed and the operator has opened HS, the steam flow shunts off to the heater, the product is removed and a new batch is loaded. As these operations take place temperature decreases and may even approach ambient conditions. When the operator closes HS for starting a new cycle, the steam flow is maximum and the temperature increases fast. The signal error should decrease and reduce the output pressure. But this does not occur even though the temperature reaches Set Point. This phenomenon represents an unnecessary waste of energy, yet it occurs due to a bad tuning of the controller when time delay is not considered. The time delay in the batch dryer process is provoked due to the distance of the source of heat to the product. Furthermore, the drying process must be started and controlled with little or no overshoot. However, the problem of start-up control is persistent. Consequently, the controller must:

- Reduce the effect due to time delay
- Reduce overshoot
- Minimize energy

To obtain a control law, it is necessary to analyze the transfer function of the plant. The batch dryer is described by the transfer function of equation (1), which represents the well known model given in [4].

$$\frac{Y(s)}{U(s)} = G(s) = \frac{Ke^{-hs}}{Ts+1}, \tag{1}$$

where h is the time delay, T is the time constant, and K is the gain of the system.

Despite the apparent simplicity of equation (1), it is not a trivial task to control time delay systems. Time delays cause undesired effects such as overshoot and the saturation of the actuators to reach and keep Set Points. Because it is challenging to reproduce this kind of systems in order to experiment with advanced control paradigms, we built a platform that emulates batch dryers. Since our platform has the same model of the industrial batch dryer system (equation 1), the results we present are valid also for larger plants. The implemented control strategy is an optimal control law for time-delay systems. Optimal control, presumably, improves the plants' performance if delays are inherent.

1.1 Antecedents

Several approaches based on optimal control have been proposed for time delay systems, such as the Maximum Principle [5], the suboptimal control (where time delays are considered as disturbances [6]) or the use of operators in infinite dimensional spaces [7], [8], [9]. Also, Dynamic Programming [10], [11], [12], and the use of complete type Lyapunov Krasovskii functionals [13] have been explored in order to obtain a suboptimal control [14]. The considered systems in those reports present delays in the state, in the input or both. If a system displays delays in the state and the optimal control uses state feedback, then the control presents distributed delays in the states [10]. As it is proved in [15], when the system depicts delays in the control, the controller has distributed delays in the control signal. Optimal controllers are rare in the industry due to their perceived complexity of implementation, but in this article we provide insights on how to exploit this type of control in industrial processes.

On the other hand, Java programming has been used marginally by the control field community. Specifically, in [16] a law to control helicopters is implemented and the system can be monitored and controlled on the Internet. A comparison with Java and Matlab is presented in [17]. It is reported that the control law implemented on the Java language runs faster than the same control law running on MatLab. Even though we are not concerned about running time, this finding is a positive aspect to consider for developing optimal control laws in Java. At this regard, commercial platforms such as MatLab or Labview are favored, but the object-oriented approach, particularly in Java, has not been explored sufficiently.

To summarize our contribution, we completed a mathematical analysis of the prototype batch dryer in order to derive the optimal control law. We employ the Simpson's rule to approximate the resultant control law. Then, we developed a fully functional object-oriented system that calculates the value of the control signal applied to the actuators. Our object-oriented implementation also considers the data acquisition process and the transmission of the control signal to the actuators.

This paper is organized as follows. First we describe the platform that was constructed to reproduce the batch dryer process. Then, we analyze mathematical concepts of optimal control and the theoretical error of the control law in regard to the batch dryer prototype. Afterwards, we describe the model and implementation of the object-oriented software. Shortly after experimental results are presented. Our findings are then contextualized within manufacturing systems. Finally, we draw some conclusions and perspectives for future work.

2 Batch dryer prototype

The batch dryer prototype is a closed box, which has three fans actuated by three DC motors (3V - 12V DC), a source of heat (an electrical grid, 17.5 volts AC as maximum signal), and a tunnel as output in the box. In our prototype, the actuators are the fans rather than the electrical grid, which stays on all the time. This fact increases the complexity of the control task. To reach or to maintain the desired Set Point, temperature must be regulated either by cooling the interior of the box with the three fans, or by letting temperature to rise by slowing the fans. Thus, the control signal must be smooth so the fans do not over-ventilate the box neither allow the inner temperature to surpass the Set Point.

Time delay is taken into consideration for the calculations. It is present due to the distance of the electrical grid to the output of the tunnel, where the product is dried. Figure 2 shows the instrumentation diagram of the batch dryer prototype.

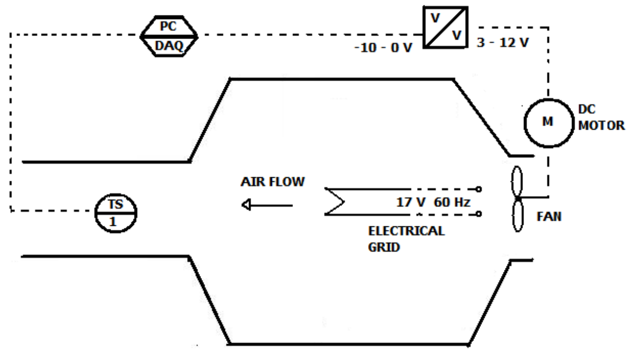

Figure 2: Instrumentation Diagram

The approximated model was obtained by the Ziegler Nichols method. Figure (3) shows the step response when the external temperature was 25^oC.

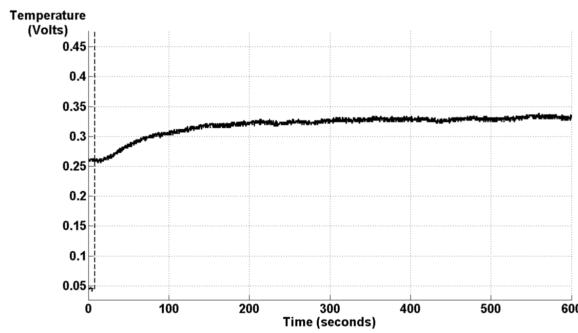

Figure 3: Step response of the temperature process

The transfer function is:

$$\frac{Y(s)}{U(s)} = G(s) = \frac{0.1e^{-14s}}{226s+1}. \tag{2}$$

2.1 Derivation of the optimal control law

In this part we present a summary of the analysis that we previously presented in [18] to derive the optimal control law. The difference differential equation of equation (1) is:

$$\dot{y}(t) = -ay(t) + bu(t-h), \tag{3}$$

where $a = 1/T$, $b = K/T$. The optimal control law u for time delay systems that we use is given in [15], where the performance index.

$$J = \int_h^\infty \left(x^2Q + v^2R\right) dt,$$

is minimum and it is modified for the infinite horizon problem. Here $Q \geq 0$, $R > 0$ and $v = u(t-h)$. The control law $v(t)$ can be found by classical methods of delayed-free optimal control:

$$v(t) = -Fy(t),$$

where $F = -R^{-1}bP$, and P satisfies a Riccati equation [19]. Thus, we have that $v(t) = -Fy(t)$. But only $y(t-h)$ is available as feedback to the controller, so the control law can be expressed as:

$$u(t-h) = -Fy(t) - 2aF \int_{-h}^{0} e^{-as} y(t+s) ds. \tag{4}$$

When nonlinear disturbances are considered in the model, and replacing the control law (4) in the nominal system (3) we derive the following closed loop system equation:

$$\dot{\bar{y}}(t) = a_0 \bar{y}(t) + \int_{-h}^{0} d(s) \bar{y}(t+s) ds + f(\bar{y}), \tag{5}$$

where

$$\begin{aligned} a_0 &= (-a - bF) \\ d(s) &= -2abFe^{-as}. \end{aligned}$$

disturbance $f(\bar{y})$ satisfies:

$$|f(\bar{y})| \leq \gamma |\bar{y}|, \quad \gamma > 0. \tag{6}$$

The stability analysis for finding delay-dependent sufficient conditions, which guarantee the robust stability in closed loop under nonlinear disturbances satisfying (6), is stated in the following lemma, originally proved in [18]:

Lemma 1 *System (5) is robustly stable under nonlinear disturbances if positive constants α_0 and β_0 exist such that the following inequalities are satisfied:*

$$\begin{aligned} 0 &< 2\alpha_0 \frac{(a + bF - \gamma)}{h} - \beta_0, \\ 0 &< 2Fb\alpha_0\beta_0 + 2a\alpha_0\beta_0 - h\beta_0^2 - 2\gamma\alpha_0\beta_0 - 4F^2a^2b^2h\alpha_0^2. \end{aligned} \tag{7}$$

In [18] we show that system of equation (2) in closed loop with control law (equation 4) satisfies stability conditions given by lemma 1.

2.2 Numerical approximation of the optimal control law

To implement the control law it is necessary to approximate the control law with distributed delays in the input that is equivalent to control law (4) by using a quadrature method. In this case we employed the composed Simpson's rule [20]. The numerical solution yields:

$$\begin{aligned} u(t-h) \simeq & -Fe^{ah}\bar{y}(t-h) - F[\frac{g}{3}(e^{ah}bu(t-2h) + bu(t-h)) \\ & + \frac{2g}{3}\sum_{k=1}^{q-1}(e^{-as_{2k}}bu(t+s_{2k}-h) + \frac{4g}{3}\sum_{k=1}^{q}(e^{-as_{2k-1}}bu(t+s_{2k-1}-h)] - adj, \end{aligned} \tag{8}$$

where q is the number of partitions, $g = \frac{h}{2q}$, $s_k = -h + \frac{h}{2q}k$, for $k = 0, 1, \cdots, 2q$.

We performed a change of variable in order to reach a Set Point (SP) different than zero, as we want to cool the box. This change of variable is $\bar{y}(t-h) = y(t-h) - SP$. We add a constant term $adj = \frac{SP}{(T)(b)}$ in the control law that defines the approximated value in the actuators to hold the temperature when the error is zero. Equation (8) is used to compute the control law. This computation is performed in two stages. We obtained the control law for the first h seconds, where the state of the system is considered as the initial condition value:

$$u(t-h) \;=\; -Fe^{ah}\bar{y}(t-h) - adj, \tag{9}$$

where

$$\bar{y}(t-h) = initial\_condition - SP.$$

All the h values are stored in an array, because the computation of the control law for all t requires previous data to predict the behavior of the system. In the second stage, equation (8) is employed to compute the control law after the initial h seconds of operation when the plant has reached a stable state. The previous analysis guarantees that the system is able to deal with non-linear, unstructured disturbances i. e. variation in the source of heat, change in the air flow, dead band of the actuators, to name but a few. However, there exists a deviation between the ideal model and the numerical approximation of the control.

2.3 Theoretical analysis of expected error

We present a theoretical analysis of the expected error inherent to the numerical solution of the control law. The actual implementation of the control law is affected by three kind of errors:

- External disturbances, such as fluctuation in ambient conditions
- Rounding errors due to floating-point calculation of the approximated control law
- Uncertain dynamics, such as hysteresis, actuators dead band, or saturation

Theoretically, the mentioned factors are not present when the ideal control law (equation 3) is solved analytically. Nevertheless, they must not affect greatly the performance of the object-oriented controller. This error or *deviation*, hereafter denoted as $\hat{e}$, between the solution of the ideal model and that of the actual implementation must fall within a tolerable operation region. This is what we are proving in the present error analysis, which will be later corroborated with the experimental results (Section 4). We remind that this deviation is not the error between the Set Point and the Process Variable. First, we analyze the ideal response of the control law. Equation (3) models the nominal system in closed loop with the approximated control law (equation 8). It possesses the following structure:

$$\begin{aligned} \dot{y}(t) &= \sum_{i=0}^{m} a_i y(t-h_i) \\ y(t) &= \varphi(t), \text{ for } t \in [-h, 0], \end{aligned} \tag{10}$$

Terms a_i in equation 10 are expanded:

$$\begin{aligned}
&a_0 = -a - b\left(F + \mu\right),\\
&a_1 = -4\mu e^{-a(-h+\frac{h(2q-1)}{2q})},\\
&a_2 = -2\mu e^{-a(-h+\frac{h(q-1)}{q})},\\
&a_3 = -4\mu e^{-a(-h+\frac{h(2q-3)}{2q})},\\
&a_4 = -2\mu e^{-a(-h+\frac{h(q-2)}{q})},\\
&\vdots\\
&a_{m-1} = -4\mu e^{-a(-h+\frac{h}{2q})},\\
&a_m = -\left(6\mu\right)e^{ah},\\
&\mu = \frac{2abFg}{3}\\
&h_0 = 0,\\
&h_1 = h - \frac{h(2q-1)}{2q},\\
&h_2 = h - \frac{h(q-1)}{q}\\
&\vdots\\
&h_m = h.
\end{aligned}$$

The analytical solution of equation (10) is:

$$y(t,\varphi) = Y(t)\varphi(0) + \sum_{i=0}^{m} a_i \int_{-h_i}^{0} Y(t - h_i - \theta)\varphi(\theta)d\theta, \tag{11}$$

where $Y(t)$ is the fundamental matrix of equation (10) which, according to [21], satisfies the following bounding conditions:

$$\begin{aligned}
&|Y(t)| \leq ke^{\alpha t},\ t \geq 0,\\
&\alpha > \alpha_0; \alpha_0 = \max\{Re(\lambda) : h(\lambda) = 0\}
\end{aligned} \tag{12}$$

In equation (12) $h(\lambda)$ is the characteristic equation of the nominal system (equation10), and $k(\alpha) > 0$. If the nominal system is stable, then $\alpha_0 < 0$, and α could be lesser or equal to zero. The nominal system (equation 10) provides the ideal response of the plant (equation 3) in closed loop when the control law (8) is used.

External disturbances are modeled as a bounded function $f(t)$:

$$\begin{aligned}
\dot{\bar{y}}(t) &= \sum_{i=0}^{m} a_i \bar{y}(t - h_i) + f(t)\\
\bar{y}(t) &= \bar{\varphi}(t),\ \text{for } t \in [-h, 0].
\end{aligned} \tag{13}$$

$f(t)$ in equation (13) is a bounded function ($|f(t)| \leq M$).

Floating point errors are described next. When the computer performs floating point operations (denoted as $fl(\cdot)$), rounding errors appear. These operations can be represented as described in [22].

$$fl(w\text{op}z) = w\text{op}z\left(1 + \delta\right), \quad |\delta| \leq u_{rouding}, \tag{14}$$

$u_{rouding}$ represents the computer precision.

Even though it is not straightforward to compute the final rounding error, we represent it as a nonlinearity in the nominal system (equation 10). As the control law depends of time delays h_i, the representation is:

$$\begin{aligned} \dot{\bar{y}}(t) &= \sum_{i=0}^{m} a_i \bar{y}(t-h_i) + f(t) + f_r(\bar{y}(t), \bar{y}(t-h_1), ..., \bar{y}(t-h_m)) \\ \bar{y}(t) &= \bar{\varphi}(t), \text{ for } t \in [-h, 0]. \end{aligned}$$

In equation (15) $f_r(\bar{y}(t), \bar{y}(t-h_1), \cdots, \bar{y}(t-h_m))$ models rounding errors of approximated control law (equation 8).

The condition (14) implies that

$$|f_r(\bar{y}(t), \bar{y}(t-h_1), \cdots, \bar{y}(t-h_m))| < L\,|\bar{y}(t), \bar{y}(t-h_1), \cdots, \bar{y}(t-h_m)|, \text{ for some } L > 0. \tag{15}$$

The effect of *Uncertain dynamics* is modeled as follows.

$$\begin{aligned} \dot{\bar{y}}(t) &= \sum_{i=0}^{m} a_i \bar{y}(t-h_i) + f(t) + f(\bar{y}(t), \bar{y}(t-h_1), ..., \bar{y}(t-h_m)) \\ \bar{y}(t) &= \bar{\varphi}(t), \text{ for } t \in [-h, 0] \end{aligned} \tag{16}$$

where

$$f(\cdot) = f_r(\cdot) + f_{ud}(\cdot). \tag{17}$$

In equation (17) f_{ud} represents the inclusion of uncertain dynamics. In the same equation $f(\cdot)$ satisfies a similar condition of that given by (15).

Therefore, equation (16) is the control law affected by the three kind of errors just analyzed. The analytical solution of the affected system:

$$\bar{y}(t, \bar{\varphi}) = y(t, \varphi) + \int_0^t Y(s)\left(f(s) + f(\bar{y}(s), \bar{y}(s-h_1), \cdots, \bar{y}(s-h_m))\right) ds,$$

$y(t, \varphi)$ is the solution for the nominal system (10).

Consequently, $\hat{e} = \bar{y}(t, \bar{\varphi}) - y(t, \varphi)$ is the error or deviation between the nominal and the implemented control law:

$$\begin{aligned} |\hat{e}| &= \left| \int_0^t Y(s)\left(f(s) + f(\bar{y}(s), \bar{y}(s-h_1), \cdots, \bar{y}(s-h_m))\right) ds \right| \\ &\leq \int_0^t \left| Y(s)\left(f(s) + f(\bar{y}(s), \bar{y}(s-h_1), \cdots, \bar{y}(s-h_m))\right)\right| ds \end{aligned}$$

Employing equations (12) and (15) on $f(\bar{y}(s), \bar{y}(s-h_1), \cdots, \bar{y}(s-h_m))$, and because $f(t)$ is a bounded function, it is derived:

$$|\hat{e}| \leq k \int_0^t e^{\alpha s}\left(M + L\,|\bar{y}(t), \bar{y}(t-h_1), ..., \bar{y}(t-h_m)|\right) ds. \tag{18}$$

Remark 2 *The asymptotic behavior of error or deviation $\hat{e}$ is determined by equation (18), proving that around the operation zone, equation (16) remains stable under nonlinearities. Hence, the implemented control law (equation 8) and the ideal system (equation 10) perform similarly. If external disturbances become significant (the value of M is high), then error $\hat{e}$ drifts from zero. In this situation, the controller can be made more robust. Parameters α and $k(\alpha)$ of equation (18) bound the error signal to operational limits. However more energy would be exerted if this were the case.*

Proposition 3 *The error function $\hat{e}(t)$ is non-increasing if the perturbed system of equation (16) is asymptotically stable.*

Corollary 4 *The nominal system (equation 10) must be stable to guarantee that error $\hat{e}$ is a non-increasing function in equation (13) when $t \to \infty$.*

Once the mathematical analysis of both, the control law and the expected error, has been completed, we now describe how to actually implement the numerical solution given in equation (8) via an object-oriented approach.

3 Design of the object-oriented optimal controller

To represent the object-oriented optimal controller we use the Unified Modeling Language (UML) [23]. The UML is a graphical language used to specify, build and visualize object-oriented computing systems. We provide three formal models that represent the structure and dynamics of the object-oriented optimal controller: Use case diagram, class diagram and collaboration diagram. Prior to the UML modeling, though, it was necessary to establish the functional requirements for controlling the batch dryer prototype. We also provide details about the computation of the control law with a series of flow diagrams. Even though we built the associated electronics, we are not providing insights about it.

The functional requirements are:

- Read data coming from the temperature sensor via serial port or USB port.
- Write the control signal to the serial port or USB port.
- Improve the quality of the input signal by using a digital filter.
- Store a series of past readings of the temperature in order to compute the control law.
- Calculate the control law according to the flow chart (see subsection 3.1)
- Display the control signal.

To implement the object-oriented optimal controller it was necessary to identify all the relevant variables and other parameters, which are described in the following table.

Parameter	Value	Unit	Type	Origin	Description
a	0.00442	$sec^{-}1$	constant	System	Constant of the system
b	0.00044	mV$sec^{-}1$	constant	System	Constant of the system
h	14	sec	constant	System	Time delay
T	226	sec	constant	System	Time constant
t	1	sec	variable	User	Sample time
q	100	-	variable	User	Partitions
F	70	-	variable	User	Controller gain
$y(t-h)$	-	mV	variable	Sensor	Temperature value
$\bar{y}(t-h)$	-	mV	input variable	Computer	Temperature, when adj is added
$u(t-h)$	-	V	output variable	Computer	Control law
s_{2k}	-	-	variable	Computer	Time shift at instant $2k$
s_{2k-1}	-	-	variable	Computer	Time shift at instant $2k-1$
SP	-	mV	variable	User	Set Point
adj	-	V	variable	Computer	Voltage value when the error is zero

Table 1. Relevant parameters to compute the optimal control law

3.1 Flow diagram for calculating the control law

We present the steps that are performed to compute the optimal control law.

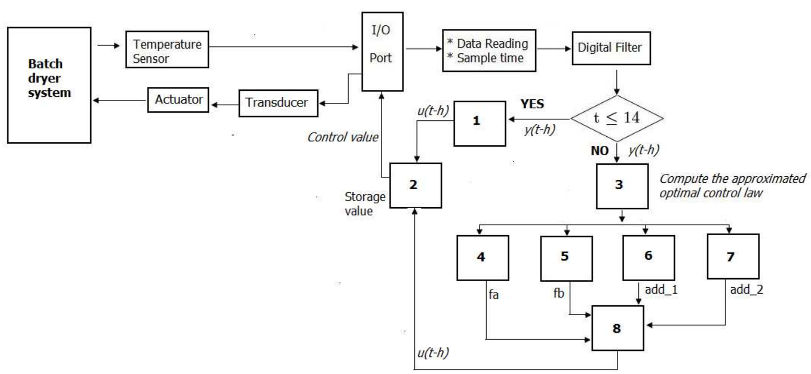

Figure 4: Flow diagram

Blocks of 1 to 8 are explained with more detail in the following paragraphs. Block 1 (Figure 5) computes equation (9) during the first 14 seconds, when the system has not surpassed the delay time.

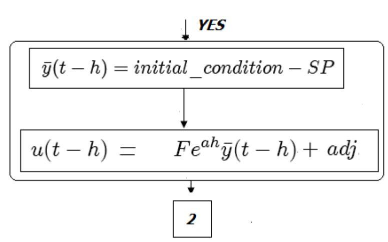

Figure 5: Block 1

Block 2 (Figure 6) indicates that the values obtained are stored in an array. Such values are used after the initial 14 seconds of operation.

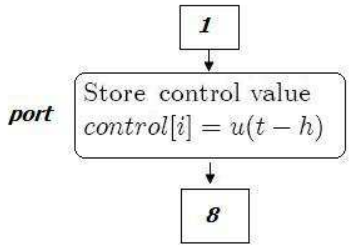

Figure 6: Block 2

When the initial 14 seconds are over, the control law must be computed with equation (8). To do so, we divide this procedure in several parallel blocks. Block 3 (Figure 7) is used to calculate the state of the system using the filtered value of the temperature sensor, for which the integration limits are fixed.

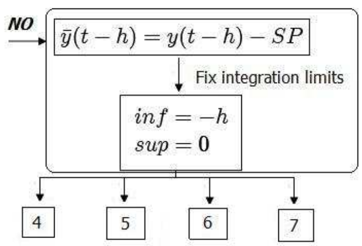

Figure 7: Block 3

Block 4 (Figure 8) computes the term $e^{ah}bu(t-2h)$. The elements of $u(t-2h)$ are obtained from the values stored previously in vector $control[]$.

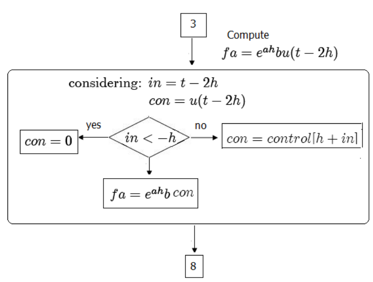

Figure 8: Block 4

Block 5 computes $bu(t-h)$ (Figure 9).

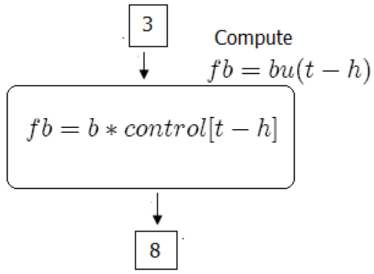

Figure 9: Block 5

Block 6 computes the first part of the summation (Figure 10). It is necessary to fix the number of partitions for the quadrature method. Vectors k and $s_{k..}$ must be calculated. In the next pseudo-code, loop cycles are used to calculate the value of $e^{-as_{2k}}bu(t+s_{2k}-h)$ and assemble all the partial computations.

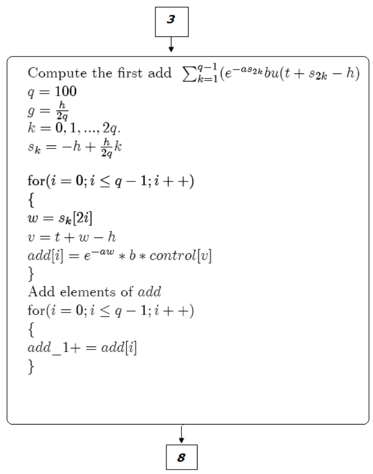

Figure 10: Block 6

Block 7 has the procedure to calculate the second summation, once the first h seconds pass (Figure 11).

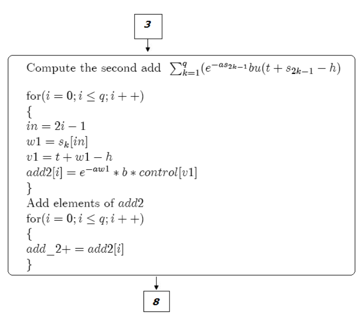

Figure 11: Block 7

After calculating the values of blocks 4 to 7, we compute the control value for the entire time span t (Figure 12).

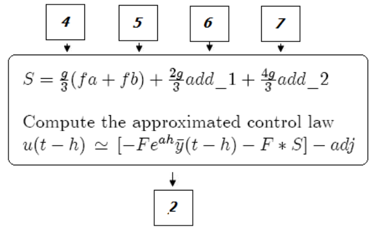

Figure 12: Block 8

Next, we present the formal models that describe the operation and structure of the object-oriented optimal controller.

3.2 Use case diagram

Use cases diagrams, as stated in [24], represent the things of value that the system performs for its *actors*, depicted as stick people. Use cases diagrams tell graphic stories about how the *actors use* the system to do something important, and what the system does to satisfy their needs. The purpose of the use case diagram is to summarize what the computing system does. *Actors* are the people or things that interact with the system. In our case, the object-oriented optimal controller must interact with sensors and the electronic interface associated to the batch dryer prototype. *Use cases*, drawn as ellipses, are the actions that the object-oriented optimal controller must perform to satisfy the needs of the batch dryer prototype. Actors and use cases are related by arrows. The use case diagram is presented in Figure 13.

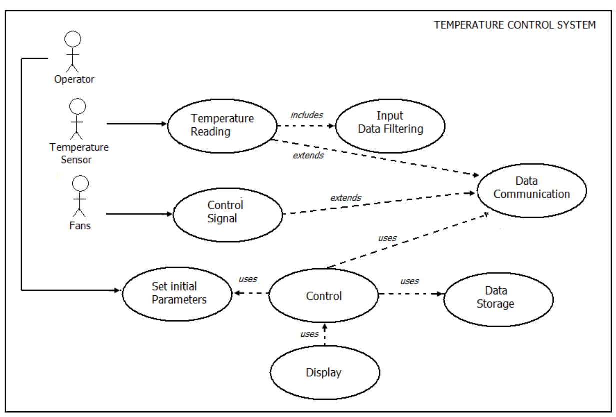

Figure 13: Use case diagram

3.3 Class diagram

Class diagrams are important models when building object-oriented systems. They are used to represent the static design view of the system. For each object, the diagram describes its identity, the relationships with other objects and the internal attributes and operations [25]. Figure 14 illustrates the class diagram.

The following classes comprise the object-oriented optimal controller:

- Class *Communication*. This class is in charge of reading temperature and sending data to the fans.

- Class *Digital filter*. This represents any digital filter to be implemented for improving quality of input data. Class *DigitalFilter* is extended by class *FirstOrderFilter* that computes the following:

$$w(k) = \frac{c}{d}m(k) - \frac{1}{d}w(k-1), \tag{19}$$

where c and d are scalar values, $m(k)$ is the current input and $w(k-1)$ is the previous output.

- Class *Control*. This class defines the method for implementing the control law. This is an abstract class being extended by class *OptimalControl*, where the actual optimal control law is codified. Both classes contain the codification to compute blocks 1 to 8 previously described. As a result, it is obtained the voltage value that is applied to the actuators.

- Class *Storage*. This class is used to store past values of the temperature.

- Class *Display*. This class shows the graphical results.

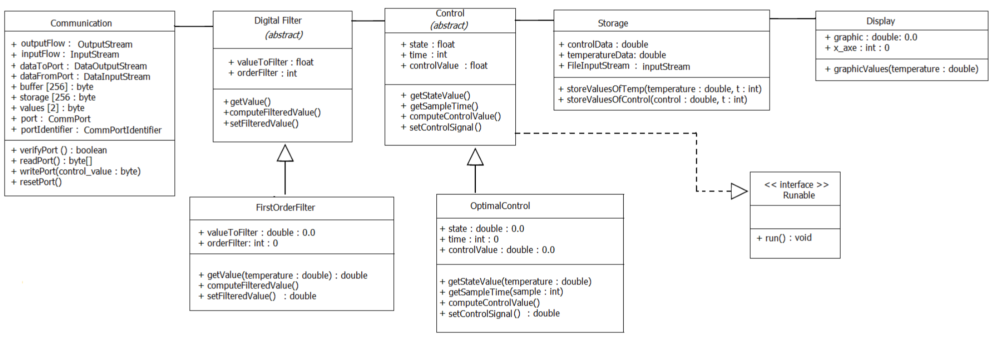

Figure 14: Class Diagram

3.4 Collaboration diagram

The collaboration among objects to control the batch dryer prototype is depicted in Figure 15. Class *Communication* reads the value of the process variable (temperature) from the RS-232 port of the computer. Class *Communication* selects the communication port, and sends and receives data streams. Two subclasses of the java.io package [26] are used. Those are *DataInputStream* and *DataOutputstream*.

When a data stream is received, it is necessary to convert such stream to a primitive data type. Input streams are converted to floating point values. This converted data indicates the temperature of the sensor. We keep track of the variable *time*, which is computed every second by Thread.sleep(1000). The time elapsed is stored, and then it is obtained by the *Control* class.

Parameters such as the initial conditions of the system, the controller gain, and the Set Point are stored in the class *Control*. For the implementation of the approximated control law (8) some considerations are made. For instance, the control law value is restricted to (0 - 12) volts because the actuators dictate so. The temperature and control values are sent to the class *Storage*, where the subclass of the java.io package *FileInputStream* is used to send data to a specific file. The class *Display* shows graphics, which presents the performance of the process variable and the control signal. Finally, class *Communication* sends the resultant control value to the serial port.

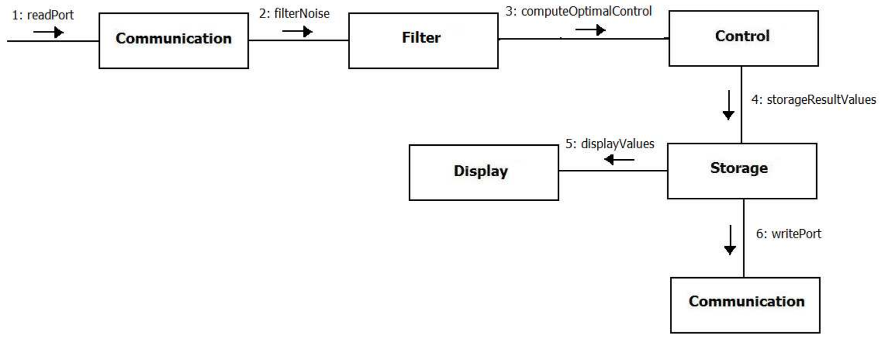

Figure 15: Collaboration diagram

We conducted a series of experiments to evaluate the performance of our object-oriented optimal controller.

4 Implementation and experimental results

The object-oriented optimal controller is implemented on a personal computer with an Intel Pentium 4 processor, 2 GHz, 1 GB RAM. The connection is made through the serial port, for which it was necessary to develop a communication protocol programmed into a microcontroller PIC16F877A of Microchip. On that hardware the analog to digital conversion of the temperature is made. The control signal is converted to a pulse weight modulation signal (PWM).

In the following, PV stands for process variable, in this case, the desired temperature inside the box. This variable is represented in the left side of the graphics. SP is the Set Point, and it is also depicted in the left side of the graphics. The control signal refers to the voltage applied to the fans, corresponding to the right side of the graphics. If the SP is higher than the temperature measured by the sensor, then the voltage applied to the fans decreases; conversely, if the SP is lower than the temperature measured by the sensor, then the voltage applied to the fans increases.

4.1 Experiment 1

The initial temperature in the box was 50°C, while the reference was set at 45°C. The external temperature at the initial moment was 25°C.

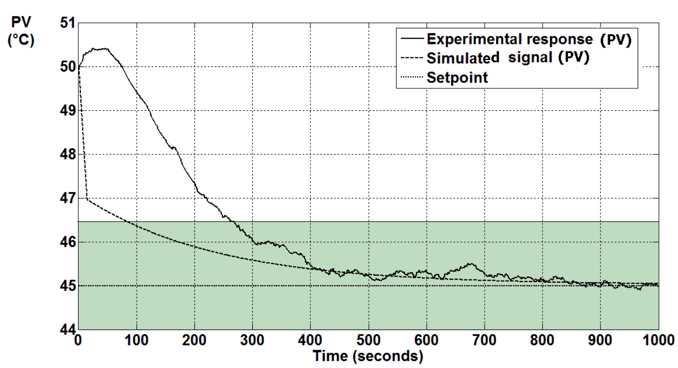

Figure 16: Process Variable reaching lower Set Point. Simulated and real responses

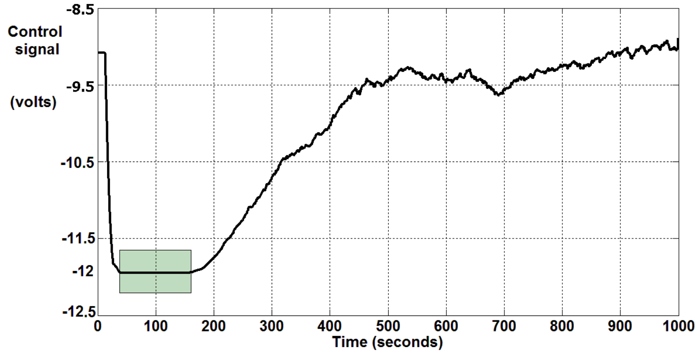

Figure 17: Control signal. Reaching lower Set Point

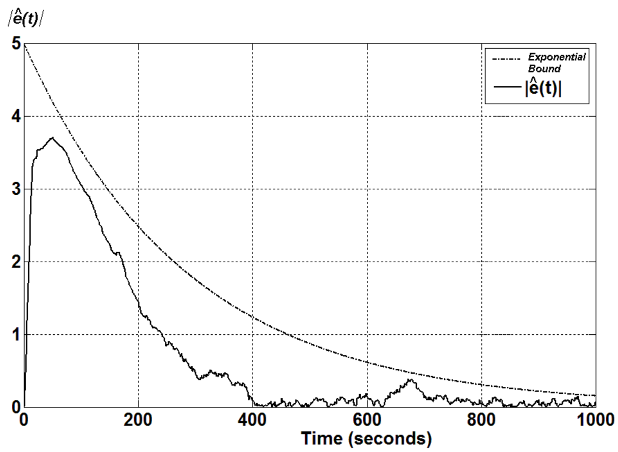

Figure 18: Deviation $\hat{e}$. Reaching lower Set Point

Data from experiment one yields:

- No overshoot. In figure (16) the shadowed area represents a $\pm$ 5% band. Temperature does not drift from this zone.
- Actuators saturate briefly. This is reflected in figure (17), where the controller only remains saturated for 100 seconds.
- Energy analysis from figure (17) yields the following data. Media is set at 10.79 volts. Standard deviation is 0.5306 volts. The area under the curve is 10786.74.
- Non-modeled dynamics (i.e. accumulated heat inside the box) provoke a large deviation $\hat{e}$ at the beginning of the experiment (Figure 18). However, it is confirmed that deviation $\hat{e}$ is bounded exponentially.

4.2 Experiment 2

In this experiment the initial temperature inside the box was 20°C, and the reference was set at 30°C. The external temperature at the initial moment was 20°C.

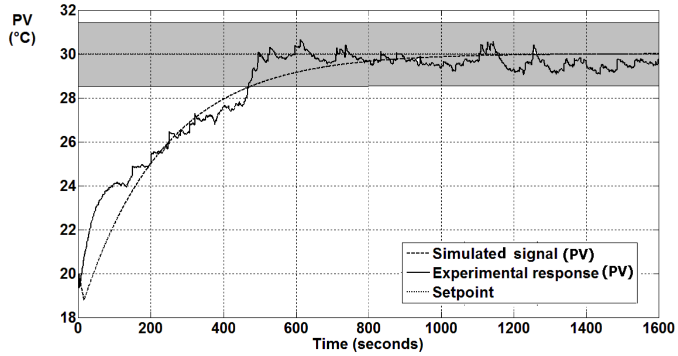

Figure 19: Process Variable reaching higher Set Point. Simulated and real responses

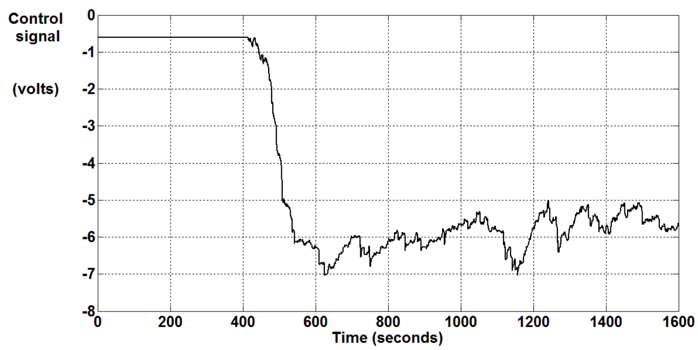

Figure 20: Control signal. Reaching higher Set Point

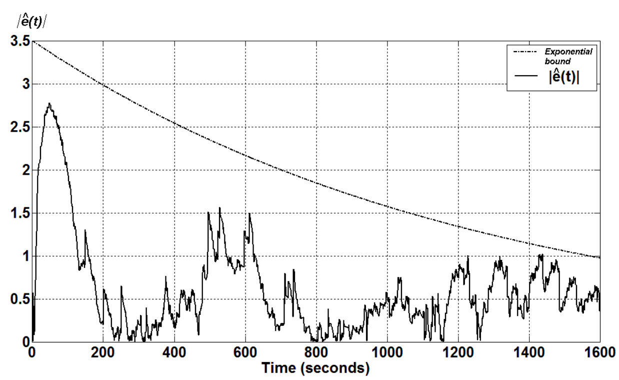

Figure 21: Deviation $\hat{e}$. Reaching higher Set Point

Data from experiment two yields:

- No overshoot. In figure (19) the shadowed area represents a $\pm$ 5% band. Temperature does not drift from this zone.
- Actuators do not saturate. This is reflected in figure (20), where maximum voltage applied is 7 volts.

- Energy analysis from figure (20) yields the following data. Media is set at -4.23 volts. Standard deviation is 1.25 volts. The area under the curve is 6849.84.
- In this experiment, SP > Initial conditions. Because of that, un-modeled dynamics do not influence the performance of the optimal controller. Figure (21) confirms that deviation $\hat{e}$ is bounded exponentially.

4.3 Experiment 3. Trajectory tracking

In this section we present an experiment where the optical controller has to adjust or adapt as different SP's where set on line. This is known as trajectory tracking. Even though this situation hardly takes place in drying processes, it is a demanding task for the optimal controller.

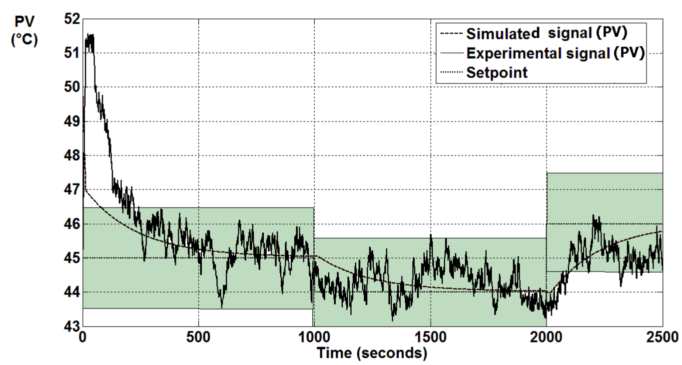

Figure 22: Process Variable following profile. Simulated and real responses.

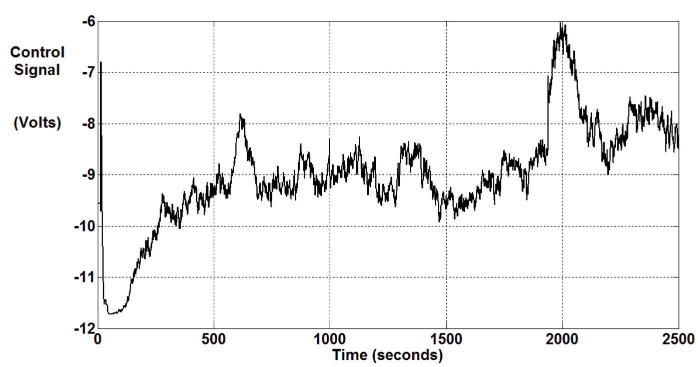

Figure 23: Control signal. Following profile

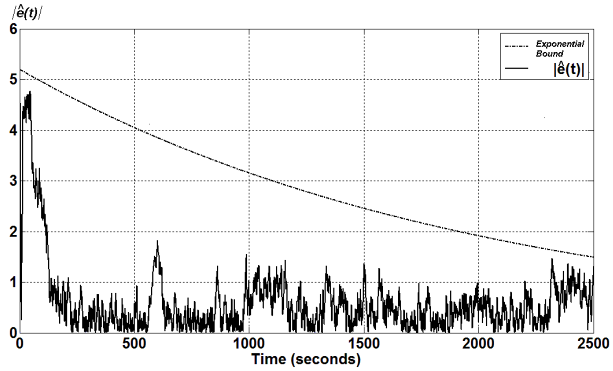

Figure 24: Deviation $\hat{e}$. Following profile

Data from experiment three yields:

- No overshoot. In figure (22) the shadowed area represents a $\pm$ 5% band. Temperature does not drift from this zone but in one reading.
- Actuators do not saturate. This is reflected in figure (23), where the controller sets a maximum voltage of 11.75 volts.
- Energy analysis from figure (23) yields the following data. Media is set at 8.99 volts. Standard deviation is 1.004 volts. The area under the curve is 23289.20.
- In this experiment, un-modeled dynamics do provoke instability. Nevertheless deviation $\hat{e}$ is bounded exponentially (Figure 24).

5 Temperature control and manufacturing of dried products

This section contains an elaboration on how optimal control theory can contribute to improve the performance of manufacturing systems that rely heavily on the regulation of temperature, particularly the process of drying products. According to [27] around 60,000 products need to be dried at different scales. In drying, unlike other industries, disturbances may be present at all times, and their impact over the quality of the final product is very strong. Whether it is fruits, wood, paper or scents, researchers concur studying the effects of temperature on the final dried product. For instance, the effect of the drying temperature on the volatile components and sensory acceptance of the Roselle (*Hibiscus sabdariffa*) extract in powder is investigated [28]. Another study presents the correlation of carotenoid content and temperature-time regimes during the drying of red pepper for paprika [29]. Also, caffeine content of guarana extract is investigated in [30], showing that the air and extract feed rates does not significantly affect the caffeine content, but the drying temperature is a determinant factor to consider. Specifically, caffeine losses were significant (1% level) for drying temperatures above 120C, while moisture content was lower than 3% for temperatures above 120C. The data showed that there is an optimum temperature (which must be kept without fluctuations) for the drying of guarana extracts.

In a similar fashion, a number of benefits are listed when drying temperature is controlled within range [31]:

- For a grain dryer, an optimization procedure allowed to decrease the drying cost by 33.6% from 2.29 to 1.52 . With optimal tuning of the heater power became 85% less than normal.

- For a grain dryer, the use of an on-line optimal controller led to an 18% decrease in drying time, a 6.4% decrease in fuel consumption, and a total cost decrease of 1.3%.
- In a beet sugar factory, the use of a model based predictive controller reduced the energy costs by 1.2% (18,900 per year) and decreased the downstream energy cost by 14,000 per year. The product yield increased by 0.86% worth 61,600 =year and off-specification production has decreased from 11 to 4%.

Industrial drying is one of the highest energy consuming processes [32]. Drying often finishes an industrial process and controls the product quality. The energy consumed in drying is usually described by some indices, such as the volume evaporation rate, steam consumption, energy efficiency (thermal efficiency), and overall energy consumption. Thus, enhancing the performance of the drying process to save energy, improve product quality as well as reduce environmental effect are important considerations to take into account from a manufacturing point of view.

The previous reports strengthen the importance of the optimal controller that we present in this contribution. Firstly, the technology we developed helps reducing the energy exerted to maintain the desired temperature. Secondly, Set Point is kept without undesired fluctuations, which is necessary to maintain product quality. Moreover, overshoot values are negligible, which prevents the batch from facing sudden heat peaks. Even though the optimal control approach has proved successful, emerging strategies based on soft-computing techniques to regulate the drying temperature have been reported. Such is the case of predictive control based on neural networks [33], [34].

On the other hand, the Java implementation of the optimal controller provides a number of advantages. Those related to the Java language in itself (portability, robustness, garbage collection), and those related to information availability. By having all the relevant data being processed on a computer, integration to higher levels (manufacturing execution systems, for instance) is feasible. In this sense, data coming from sensors and data related to the process's performance can be co-related to product parameters. These benefits are harder to obtain with traditional industrial controllers.

Our future research consists in studying the effects of different control laws on the final quality of *Annona diversifolia* [35]. We devise experimenting with optimal control, fuzzy control (Mamdani and Takagi-Sugeno-Khan) robust PID, among others. We are interested in determining nutrients content of the dried fruit for each control strategy employed to regulate temperature. With this, we will not only have data of the process's performance but also of the product quality, as determined by experts in the nutrition arena.

6 Conclusions

We present an object-oriented optimal controller that constitutes a solution for the previously described problems associated to batch dryers. In this contribution it was possible to make a mathematical analysis of a batch dryer system, which presents time delays, and to obtain an optimal control law. This law was the basis to develop an object-oriented optimal controller. Altogether, the proposed solution provides promising experimental results not only for batch processes, but also for any process that complies with equation (1).

Since the object-oriented design is modular, we can foresee numerous improvements. For instance, we will experiment with higher order filters, with different number of partitions to solve the numerical approximation of the control law, and even to employ another numerical method different than the Simpson's rule.

In short, the prototype we present in this paper has been useful to validate, in practice, theoretical claims proper of optimal control, which have not been documented so far within industrial settings.

References

[1] F. M'Sahli R. ben Abdennour M. Ksouri. Experimental nonlinear model-based predictive control for a class of semi-batch chemical reactors. *The International Journal of Advanced Manufacturing Technology*, 20(6):459 – 463, 2002.

[2] R. B. Abdennour F. M'Sahli and M. Ksouri. Nonlinear model-based predictive control using a generalized Hammerstein model and its application to a semi batch reactor. *The International Journal of Advanced Manufacturing Technology*, 20(11):844 – 852, 2002.

[3] H. B. Williams W. G. Andrew and L. M. Zoss. *Applied instrumentation in the process industry*, volume 4. Gulf, second edition edition, 1979.

[4] J. G. Ziegler and N. B. Nichols. Optimum settings for automatic controllers. *ASME Transactions*, 64:759 – 768, 1942.

[5] G. L. Kharatishvili. The maximum principle in the theory of optimal processes with delay (in Russian). *Doklady Akademii Nauk USSR*, 136:39 – 42, 1961.

[6] Malek-Zavarei M. and Jamshidi M. *Time delay systems, Analysis, Optimization and Applications*, volume 9. North Holland Systems and Control Series, The Netherlands, 1987.

[7] J. S. Gibson. Linear quadratic optimal control of hereditary differential systems: infinite dimensional Riccati equations and numerical approximations. *SIAM Journal on Control and Optimization*, 21(1):95 – 139, 1983.

[8] R. H. Kwong. A stability theory for the linear quadratic Gaussian problem for systems with delays in the state, control, and observations. *SIAM Journal on Control and Optimization*, 18:49 – 75, 1980.

[9] R. B. Vinter and R. H. Kwong. The infinite time quadratic control problem for linear systems with state and control delay. *SIAM Journal on Control and optimization*, 19(1):139 – 153, 1981.

[10] N. N. Krasovskii. On analytic design of optimal controllers for systems with time delay. *Prikladnaya Matematika i Mekhanika*, 26(1):39 – 51, 1962.

[11] D. W. Ross and I. Flügge-Lotz. An optimal control problem for systems with differential difference equation dynamics. *SIAM Journal on Control and Optimization*, 7:609 – 623, 1969.

[12] H. J. Kuhsner and D.I. Barnea. On the control of a linear functional differential equation with quadratic cost. *SIAM Journal on Control and Optimization*, 8(2):257 – 272, 1970.

[13] V. L. Kharitonov A. P. Zhabko. Lyapunov Krasovskii approach to the robust stability analysis of time delay systems. *Automatica*, 39(1):15 – 20, 2003.

[14] Omar Santos Sabine Mondié and Vladimir Kharitonov. Linear Quadratic Suboptimal Control for Time Delays Systems. *International Journal of Control*, 82(1):147 – 154, 2009.

[15] D. H. Chyung Y. Alekal, P. Brunovsky and E. B. Lee. The quadratic problem for systems with time delays. *IEEE Transactions on Automatic Control*, 16(6), 1971.

[16] J. Chen J. Zhang C. C. Ko, B. M. Chen and K. C. Tan. A Web-Based Laboratory on Control of a Two-Degrees-of-Freedom Helicopter. *Int. J. Engng Ed.*, 21(6):1017 – 1030, 2005.

[17] M. M T. Araujo A. P. Tereso, R. A. Novais and S. E. Elmaghraby. Optimal resource allocation in stochastic activity networks via the electromagnetism approach: a platform implementation in Java. *Control and Cybernetics*, 38(3), 2009.

[18] Omar Santos Liliam Rodríguez-Guerrero and Omar López-Ortega. Experimental results of a control time delay system using optimal control (In press). *Optimal Control, Applications and Methods*, 1:1 – 10, 2011.

[19] R. E. Kalman. Contributions to the Theory of Optimal Control. *Boletín de la Sociedad Matemática Mexicana*, 5:102 – 109, 1960.

[20] J. D. Hoffman. *Numerical Methods for Engineers and Scientists*. McGraw Hill, United States of America, 1992.

[21] J. K Hale. *Introduction to functional differential equations*. Springer Verlag, New York, USA, 1993.

[22] N. J. Higham. *Accuracy and stability of numerical algorithms*. SIAM, USA, 1996.

[23] M. Fowler and K. Scott. *UML distilled: A brief guide to the standard object modeling language*. Addison Wesley Longman Publishing, United States of America, second edition, 2000.

[24] Kurt Bittner and Ian Spencer. *Use case modeling*. Addison Wesley, USA, 2003.

[25] Grady Booch James Rumbaugh, Ivar Jacobson. *The Unified Modeling Language*. Addison Wesley Longman Inc, United States of America, 1999.

[26] Elliotte Rusty Harold. *Java I/O*. O'Reilly Media Inc., Sebastopol, CA. United States of America, second edition, 2006.

[27] A. S. Mujumdar. An overview of innovation in industrial drying: current status and RD needs. *Transport in Porous Media*, 66:3 – 18, 2007.

[28] Estarrón Espinoza M. Gómez-Leyva J. F. Gonzalez-Palomares, S. and I. Andrade-González. Effect of the temperature on the spray drying of roselle extracts (Hibiscus Sabdariffa L.). *Plant Foods for Human Nutrition*, 64:62 – 67, 2009.

[29] Hornero-Méndez D. Pérez-Gálvez, A. and M. I. Mínguez-Mosquera. Dependence of carotenoid content and temperature-time regimes during the traditional slow drying of red pepper for paprika production at La Vera county. *European Food Research and Technology*, 221:645 – 652, 2005.

[30] Bastos-Jairo K. Plagiarusi, R. S. and L. Freitas. Fluid Bed Drying of Guarana (Paullinia cupana HBK) Extract: Effect of Process Factors on Caffeine Content. *AAPS PharmSciTech*, 7(2):E1 – E7, 2006.

[31] P. Dufour. Control engineering in drying technology: Review and trends. *Drying technology*, 24:889 – 904, 2006.

[32] Gordienko-M. G. Voinovskii-A. A. Men'shutina, N. V. and T. Kudra. Dynamic criteria for evaluating the energy consumption efficiency of drying equipment. *Theoretical Foundations of Chemical Engineering*, 39(2):158 – 162, 2005.

[33] Chi Q.-Wang L. Zhao, C. and B. Wen. A model predictive control of a grain dryer with four stages based on recurrent fuzzy neural network. *Lecture Notes in Computer Science*, 4491:29 – 37, 2007.

[34] F. M'Sahli and R. Matlaya. A neural network model based predictive control approach: application to a semi batch reactor. *The International Journal of Advanced Manufacturing Technology*, 26(1 – 2):161 – 168, 2005.

[35] A.P. Julián-Loaeza N. F. Santos-Sánchez R. Valadez-Blanco B. S. Sánchez-Guzmán R. Salas-Coronado. Chemical composition, color, and antioxidant activity of three varieties of Annona diversifolia Safford fruits. *Industrial Crops and Products*, In press(1):1 – 7, 2011.