Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Intelligent Control Based on Reinforcement Learning for Batch Thermal Sterilization of Canned Foods ⋆

S. Syafiie * **Carlos Vilas** ** **Miriam R. Garcia** **
Fernando Tadeo * **Antonio A. Alonso** ** **Ernesto Martinez** ***

* *Department of Systems Engineering and Automatic Control, University of Valladolid, 47011 Valladolid, Spain, {syam|fernando}@autom.uva.es*
** *Process Engineering Group IIM-CSIC, Vigo, Spain, {miriamr|carlosvf|antonio}@iim.csic.es*
*** *National Research Council of Argentina, Avellaneda 3657, 3000-Santa Fe, Argentina, ecmarti@ceride.gov.ar*

Abstract: A control technique based on Reinforcement Learning is proposed for controlling thermal sterilization of canned food. Without using an *a-priori* mathematical model of the process, the proposed Model-Free Learning Controller (MFLC) aims to follow a temperature profile during two stages of the process: first heating by manipulating the saturated steam valve and then cooling by opening the water valve) by learning. From the defined state-action space, the MFLC agent learns the environment interacting with the process batch to batch and then using a tabular state-action mapping. The results show the advantages of the proposed technique for this kind of processes.

Keywords: Reinforcement Learning, Intelligent Process Control, Sterilization Process, Batch Process.

1. INTRODUCTION

In the food industry, it is very important to reduce the activity of harmful microorganisms for canned food, in order to reduce the health risk and increase the durability of the products. This is usually obtained through thermal processing (sterilization) in pressurized retorts using steam. Unfortunately, thermal processing also produces the deterioration of the organoleptic properties of the food. For this reason, appropriate control of the process is fundamental to guarantee the safety and quality of the product (Lewis [2006], Ramaswamy and Singh [1997]).

Thus, in the sterilization process the main control objective is to heat the canned food for a minimum time at a given temperature: long exposures or high temperatures deteriorate the product. This processing time and temperature are selected according to the mandatory degree of microorganism activity, measured off-line by estimating the microbiological lethality of the process.

Different strategies for control of the sterilization process have been proposed in the literature, such as adaptive control (Alonso et al. [1997]), online correction factor (Teixeira and Tucker [1997], Kuma et al. [2001]), optimal control (Kleis and Sachs [1999]) and receding horizon optimal control (Chalabi et al. [1999]). However, these controllers are difficult to design and need precise mathematical models of the process, so the most frequent control technique in industry is manual supervision of proportional Integral (PI) controllers.

To deal with problems of batch to batch variations and the complexity of the models for control, techniques based on learning would be adequate. From these, techniques that use Reinforcement Learning have been selected, as they provide a rigorous methodology for learning without detailed mathematical models of the controlled plant, using a simple algorithm suitable for real-time implementation (Sutton and Barto [1998]).

In particular, the MFLC approach, previously proposed by the authors (Syafiie et al. [2007a,b]), will be used to control the thermal processing, as it corresponds to a feasible implementation of Reinforcement Learning algorithms for Process Control. This technique is used because it is a simple technique that does not need a precise *a priori* model of the process, but incorporates basic knowledge of the process behavior. The MFLC controllers are based on Reinforcement Learning algorithms, so the control objective is the optimization of a desired performance index by learning to apply appropriate control actions through interaction with the plant. Learning is performed without requiring an explicit model of the plant: instead, the system's dynamics are learned and represented in action and reward functions.

The approach is based on Q-learning (Sutton and Barto [1998], Bertsekas and Tsitsiklis [1996]). However, the idea

⋆ Funded by mcyt-CICYT DPI2003-07444-C02 and DPI2007-66718-C04-02.

can be easily augmented to improve learning speed by applying other methodologies in the literature, such as lazy learning (Atkenson et al. [1997a,b]), near optimal closed-loop control (Ernst [2003]) and neural fitted q-iteration (Timmer and Riedmiller [2007]).

The problem at hand can be represented as a series of single-input single-output systems. However, the proposed approached can be extended to multiple-input multiple-output system using the ideas presented in Martin Riedmiller [1997].

It must be pointed out that no explicit mathematical model is used to design the control algorithm. However, basic knowledge of the process is used to fix control parameters (information from output range, control limitations, loop interactions, etc).

This article is organized as follows: First, a short presentation of the Thermal Sterilization Process is given in Section 2. The proposed technique to control the process by using MFLC is given in Section 3. The MFLC application in the sterilization process is given in Section 4. Finally, conclusions are given in Section 5.

2. BATCH THERMAL STERILIZATION PROCESS

The thermal sterilization processes for prepackaged food can be carried out in continuous or batch units (Lewis [2006], Ramaswamy and Singh [1997]). From a control point of view, the most challenging is the operation in batch units, which is the most frequent approach in industry, and is the one studied in this paper. It is now briefly described. For details of the process see (Alonso et al. [1997, 1998]).

In general, the sterilization process is carried out in batch steam retorts as depicted in Fig. 1.

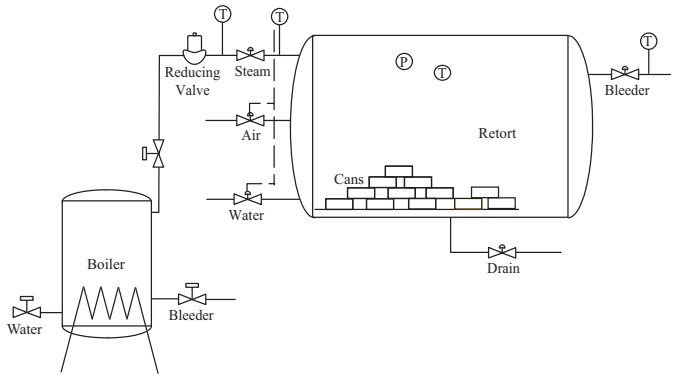

Fig. 1. Schematic of control equipment for batch sterilization.

A typical operation cycle involves several stages, namely venting, heating and cooling:

- *Venting*: In the first stage, steam is used to eliminate the air from the retort. At this stage, bleeder and drain valves are open. When the pressure in the retort, P_r, matches that corresponding to saturated steam at that temperature, P_s, it can be assumed that there is no air in the retort and heating can start.

- *Heating*: The objective of this stage is to follow a given temperature profile, prescribed by the desired microbiological lethality. At the critical point (the coldest point inside the product), lethality is defined as follows:

$$F_0 = \int_0^t 10^{\frac{T^k_{ref} - T(r_0)}{z}} dt \qquad (1)$$

where z represents a kinetic parameter, T^k_{ref} refers to temperature, $T(r_0)$ is the temperature at the critical point (see Ramaswamy and Singh [1997], Alonso et al. [1997]) and t is time. Since the lethality is affected by even small variations in the temperature, automatic control is required during this stage.

- *Cooling*: Once the heating stage concludes, the product is cooled with water down to room temperature. At the same time, air is injected into the vessel to avoid sudden pressure drops that could result in the bursting of product containers (cans). Pressure control during this stage is especially important for glass containers or conduction heated-type products where the existence of sharp temperature gradients between the inside and the outside of the product induces high differential pressure (Alonso et al. [1997, 1998]).

3. MFLC DESIGN TECHNIQUE

The MFLC technique (Syafiie et al. [2007a,b]), proposed to control the sterilization process, is a methodology based on learning using the Reinforcement Learning approach (Sutton and Barto [1998], Bertsekas and Tsitsiklis [1996]). In particular, it gives a feasible implementation of Reinforcement Learning for process control problems, by giving a precise but simple definition of symbolic states and actions, based on control objectives and the constraints on input and output variables. MFLC has been presented in detail by three of the authors in Syafiie et al. [2007a,b], so only a brief presentation is given here.

3.1 MFLC Architecture

The MFLC architecture is represented in Fig. 2: it is modular, based on a simple selection of states, actions and control signals, with the objective of being easily understood by the final user. At each sample time, the agent uses the "Policy" to select one action a_t from those available in the actual state s_t. Then, the selected action is converted to a control signal u_t in the "Calculation U" block. Based on the measured output, the "Situation" block estimates the next state and the corresponding reward. From this reward the so-called Q-value, which reflects the adequacy of the action, is updated in the "Critic" block.

As time goes by, actions are selected by the agent, and learning is carried out by criticizing them as "good" or "bad" depending on the resulting state. Every action that drives the system into the goal state is considered a good action and receives reward. However, actions that do not drive the system into the goal state are punished. A central part of the learning algorithm is the estimation of the Q-value, which gives numerically the benefit of applying action a_t when the system is in state s_t. This function is

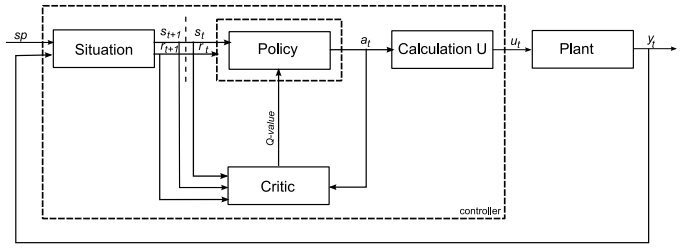

Fig. 2. MFLC architecture.

stored in a matrix $Q(s_t, a_t)$, called Q-matrix. To calculate the Q-values, it is necessary to take into account the current and future benefits: As it has been mentioned, when action a_t has been selected and applied to the plant, the system moves to a new state s_{t+1} and receives a reinforcement signal r_{t+1}. The value function for state-action pairs, $Q(s_t, a_t)$, is updated by the basic learning rule:

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_{b \in A_{s_{t+1}}} Q(s_{t+1}, b)] \quad (2)$$

where:

- $A_{s_{t+1}}$ is the set of possible actions in the new state.
- The *learning rate*, $\alpha \in (0, 1]$, is a tuning parameter used to optimize the speed of learning (Large learning rates make learning faster, but might induce oscillations).
- The *discount factor*, $\gamma \in (0, 1]$, is used to weight near-term reinforcements more than distant-future ones: If γ is small, the agent learns to respond only to short-term rewards; the closer γ is to 1 the greater the weight assigned to long-term reinforcements.

3.2 State Representation

A central issue in Reinforcement Learning algorithms is the definition of the states. In MFLC the states are defined based on the control objective and control constraints, as follows:

In a SISO implementation of the MFLC framework, the control objective is considered to maintain the desired output inside the band $r - d$ and $r + d$, as shown in Fig. 3. The width of this band is defined based on the tolerance of the system (which depends on measurement noise, disturbances and system specifications). This band is defined as the *goal band*, and corresponds to the *goal state*, where the agent should go and remain (it is now assumed, without loss of generality, that this is exactly in the middle of the working range).

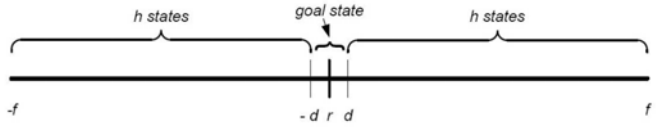

Fig. 3. Symbolic states definition in MFLC.

To describe the rest of the symbolic states, it is considered that the agent has h states from the goal state to the maximum positive or minimum negative error of the system, f (Selecting h is a trade-off: this number must be large enough to describe all the different behaviors of the process, but small enough to reduce computational time and the size of the Q-matrix).

If needed, the "length" of each state can be calculated as follows:

$$c = \frac{f - d}{h}. \quad (3)$$

Thus, the vector of symbolic states can be presented as follows:

$$g_j = \begin{cases} e - \omega_j \text{ IF } e \leq \omega_j \\ \omega_j - e \text{ OTHERWISE} \end{cases} \quad j \in [1, ..., 2h+1], \quad (4)$$

where e is the tracking error at instant t and the bound parameter ω_i can be presented as:

$$\omega_i = d + (i-1)c, i \in [1, ..., h] \quad (5)$$

(For negative errors, the bound parameter is trivial by changing signs).

The symbolic current state s_t, can then be evaluated by just:

$$s_t = \arg\max(g_j). \quad (6)$$

3.3 Action Representation

In the single-input single-output version of MFLC, which is the one used in this paper, the control signal $u_t \in \mathbb{R}$ is calculated by varying the previous control signal in a magnitude calculated from the difference of the numerical values of the selected optimal action, $a_t \in \mathbb{N}$, with respect to the *wait action*, a_w (action corresponding to maintaining the previous control signal). That is,

$$u_t = u_{t-1} + k(a_w - a_t). \quad (7)$$

This gives a PI-like structure, which simplifies initialization and tuning for the end user ($k \in \mathbb{R}$ is the tuning parameter). At each state there is only a finite set of possible actions (see Fig. 4), that are selected based on the system description. In particular, from the limitations on the minimum and maximum variations of the control signal, as follows:

If the incremental control is known to be bounded as follows:

$$\underline{\Delta u} \leq |\Delta u| \leq \overline{\Delta u}. \quad (8)$$

The number of total actions needed to satisfy the constraints can be calculated by:

$$N_a = 2h \left(round \left(\frac{\overline{\Delta u} - \underline{\Delta u}}{kh} \right) \right) + 1. \quad (9)$$

Rounding up is used, to satisfy the maximum bound (8). From (7), (8) and (9), the value corresponding to the wait action a_w, can be calculated as follows:

$$a_w = \frac{N_a + 1}{2}. \quad (10)$$

If there is no ovelapping, the number of actions in each state, n_a, are given by the following expression

$$n_a = \frac{N_a - 1}{2h}. \quad (11)$$

However, in practice, it is necessary to increase the number of available actions, by including some overlapping (see Fig. 4.), so that nonlinear action-to-space relations can be represented. As it is logical, not all the actions are available at each state: a state has only a subset of possible actions (only those that are physically reallistic). For example,

in our application, if the measured temperature is very low, the only actions available are those that increase the temperature. Thus, the number of actions in each state is

$$n_a^\beta = n_a(1+\beta), \tag{12}$$

where β is a parameter that gives the degree of overlapping with neighboring states (always selected such that n_a^β is integer).

Then, the available actions for every state go from a_p^j to a_b^j (except in the goal state, where there is only the wait action). The idea is presented in Fig. 4. Those available actions can be calculated as

$$\begin{aligned} a_p^j &= a_p^{j-1} + (j-1)v, \\ a_b^j &= a_p^j + n_a^\beta - 1, \end{aligned} \tag{13}$$

where $v = \beta\frac{n_a^\beta}{h}$ and a_p^{j-1} is the first action in the state j, calculated as

$$a_p^{j-1} = \begin{cases} 1, & \text{WHEN } j = 1 \\ 2a_w - a_b^{j-2}, & \text{WHEN } j = h+2 \end{cases}. \tag{14}$$

4. THERMAL CONTROL OF PREPACKAGED FOOD

This section explains the application of the MFLC strategy presented in section 3, to control the thermal sterilization process in a batch unit presented in section 2. In this study the application of MFLC is implemented on a detailed simulation of the sterilization process based on the mathematical model in partial differential equations developed and validated on some industrial plants by some of the authors (Alonso et al. [1997, 1998]). The first part of this section discusses the control strategy and continues with the system definition used in MFLC.

4.1 Control Strategy

As discussed in section 2, there are three crucial stages in controlling the sterilization process: Venting, Heating and Cooling.

The control strategy for these stages is shown in Fig. 5. As the venting stage can be controlled using a simple technique (keeping bleeder and drain valves fully open until the pressure inside the retort reaches the steam pressure), the proposed control application therefore concentrates on the heating and cooling stages. In fact the approaches are similar in these stages (although with different manipulated inputs and tuning parameters), so for lack of space only the design for the heating stage is discussed in detail.

During the heating stage, the control objective is to maintain the temperature inside the goal band by manipulating the steam valve. To evacuate the condensed water from the retort, the drain valve is open. Also, the bleeder valve is slightly open.

4.2 System Definition

During the heating stage, the objective is to maintain the retort temperature around $r = 121$C, with a tolerance of ± 1C. Thus, the goal band is $r-1 = 120$C to $r+1 = 122$C. The output range is considered to be ± 40C from the selected reference. Thus, following the ideas presented in

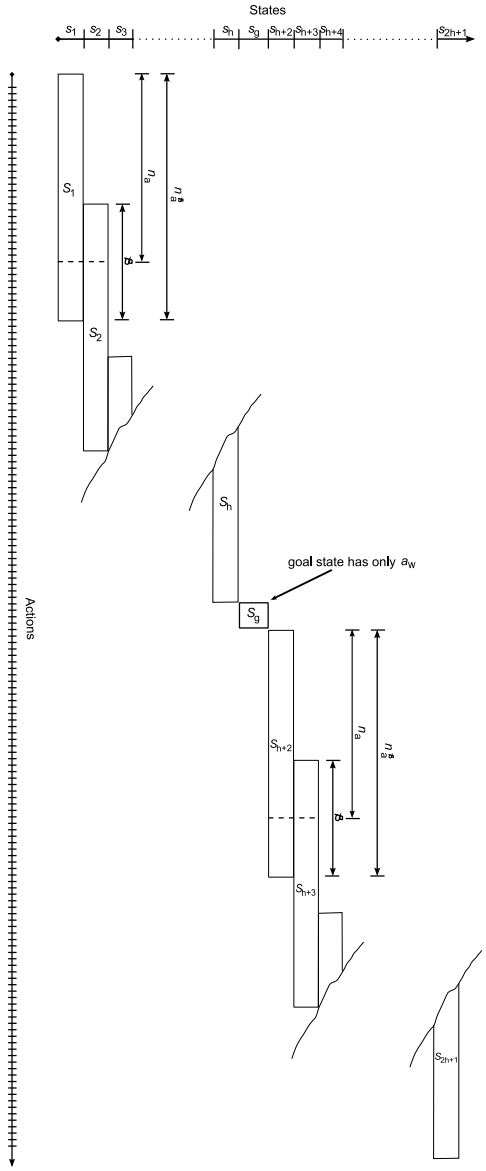

Fig. 4. State-Action space of Q-matrix in MFLC.

the previous section, there are 81 symbolic states, where state #41 corresponds to the goal state.

The actions are defined based on the possible control variations: it is known that at each sample time the signal must vary within the following bounds:

$$0.0001 \leq |\Delta u| \leq 0.001. \tag{15}$$

The tuning parameter is selected to be $k = 10^{-6}$, based on the control constraints and previous experience on the process. Thus, the total number of different available actions is 1831, where the wait action is action #1001. As there is some overlapping, the number of actions in every symbolic state is 878. Therefore, in state number 1 the actions are $\#1, \cdots, \#3878$; similarly, in state 2 the actions are $\#2, \cdots, \#879$, and so on, following (13).

From those available actions, the strategy for selecting one action is by *exploration* and *exploitation*. The agent explores those available actions to know the optimal value function by executing trial actions, following the ε-greedy

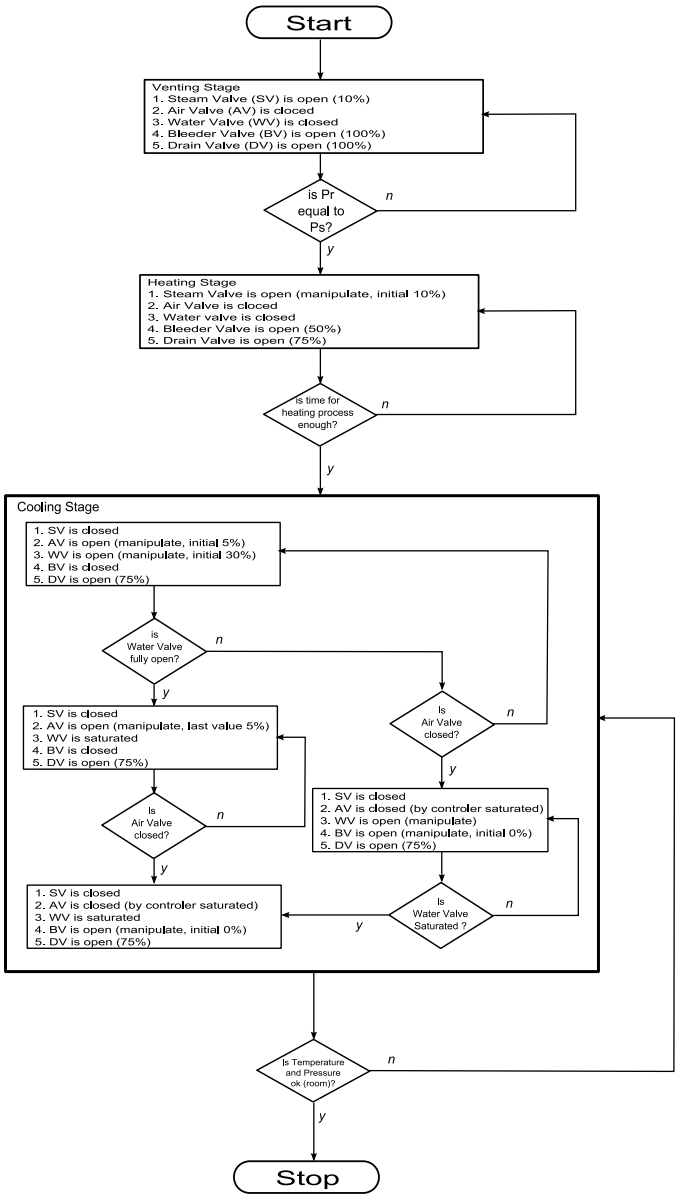

Fig. 5. Control logic

policy (Sutton and Barto [1998]). This means that the action which has maximum Q-value will be selected with $1-\varepsilon$ probability.

The goal of the control task is to maintain the process in the goal state (or drive it to the goal state if there has been any disturbance or change of reference). To achieve this, maximum reward is introduced for the action causing the process to be in the goal state. Actions that move the system away from the goal band are punished. Therefore, the reward is given as:

- 1.0 for actions causing the next state to be the goal state,
- 0.0 for actions moving the next state towards the goal state,
- −2.0 for actions not changing the state, or moving the next state away from the goal state.

Of course, more complex reward functions can be used, but this particularly simple reward function has been selected following the ideas by Smart [2002], which recommends not giving the agent a detailed path to achieve the goal, but only the goal, as the path assumed to be the most adequate might not be really the best (learning takes care of finding the most adequate approach).

4.3 Cooling Control Strategy

In this paper, the state-action space has been discussed in detail for the heating stage. For controlling temperature during the cooling stage the water valve is manipulated following the same strategy, although with a different Q-matrix, and changing the sign of the gain k in (7), because in this case the input/output gain is negative. The state-action space for maintaining pressure inside the retort is defined as in the heating stage.

5. RESULTS AND DISCUSSION

The temperature responses for controlling the sterilization process with 100 cans inside the retort using the proposed MFLC are shown in Fig. 6, showing the learning process after several batches. After venting, the first 5000 seconds correspond to the heating process, where the steam valve is manipulated. When the process is switched from heating to cooling, the pressure inside the retort is maintained. In this transition phase, the steam valve is closed and the air valve is open to maintain the pressure. Then, for cooling over the next 80 minutes, the water valve is open.

It is possible to see that the proposed controller correctly regulates the temperature inside the retort during the heating process, without affecting the cooling process. Moreover the main variable (temperature within the cans, estimated using simulation), remains within the desired bound over the required minimum time. The pressure inside the retort is shown in Fig. 8. Clearly, the RL agent is able to handle the pressure drop inside the retort. Moreover, the control signals, shown in Fig. 7, are smooth and fulfil the control constraints.

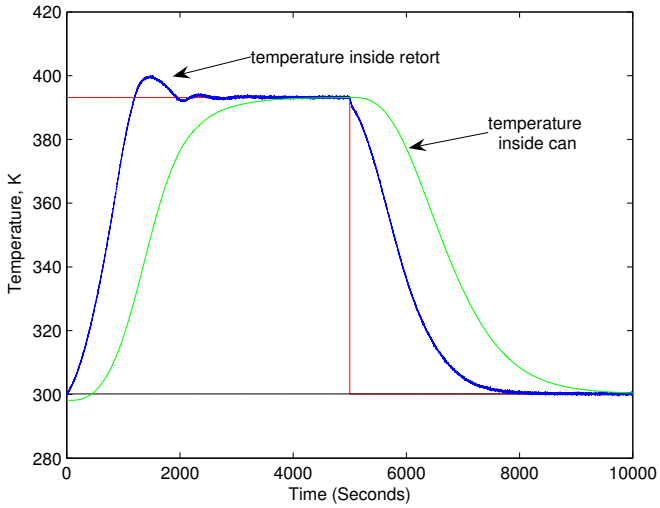

Fig. 6. Temperature responses for heating and cooling for one cycle, after learning.

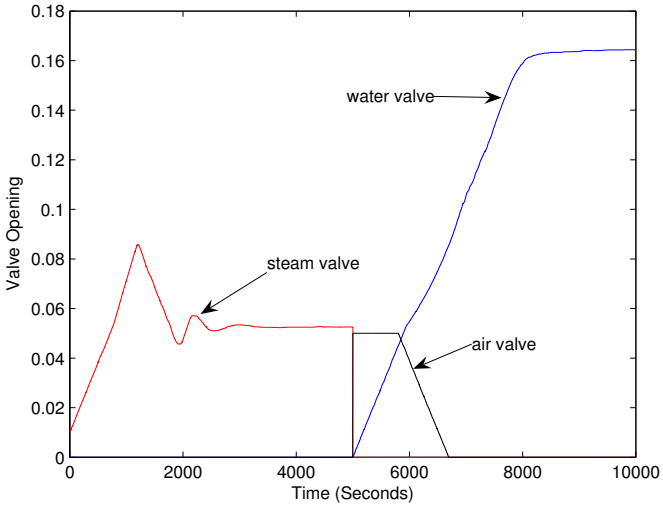

Fig. 7. Control signals: opening of saturated steam valve and water valve.

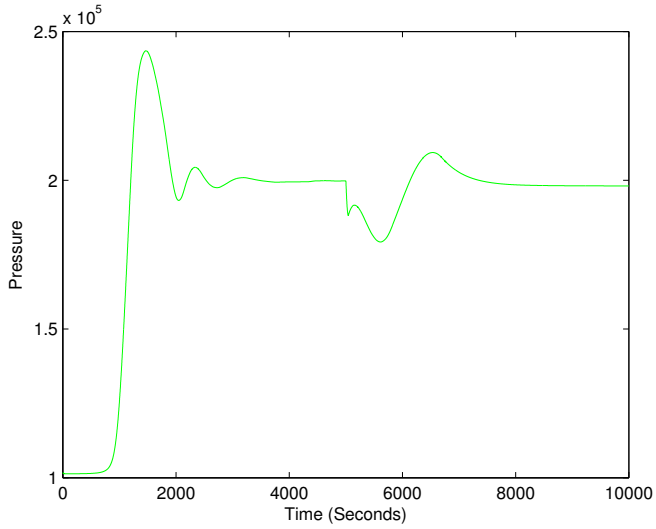

Fig. 8. Pressure inside the retort.

To summarize, we can conclude that the MFLC manages the process satisfactorily, without needing a precise mathematical model of the process.

6. CONCLUSIONS

An alternative procedure to the control of the sterilization process of cans in the food industry has been presented, based on regulation using the proposed MFLC algorithm, which is based on Reinforcement Learning ideas. The proposed MFLC automation strategy is appealing for this kind of process: since they are very uncertain, it is not practical to use a precise model of the process for control, and the batch operations are specially adequate for learning controllers. The preliminary results presented show that the proposed controller makes possible to maintain the temperature inside the cans within specifications, allowing safe consumption of the contents.

REFERENCES

M. J. Lewis, Thermal Processing, in J.G. Brennan, *Food Processing Handbook*, Wiley, 2006, pp. 33–70.

H. S. Ramaswamy and R. P. Singh, Sterilization Process Engineering, in K. J. Valentas, E. Rotstein, R. P. Singh, *Handbook of Food Engineering Practice*, CRC Press, New York, 1997.

A. A. Alonso, J. R. Banga and R. P. Martin, A Complete Dynamic Model for the Thermal Processing of Bioproducts in Batch Units and its Application to Controller Design, *Chemical Engineering Science*, vol. 52, no. 8, 1997, pp. 1307–1322.

A. A. Teixeira and G. S. Tucker, On-line Retorts Control in Thermal Sterilization of Canned Foods, *Food Control*, vol. 8, no. 1, 1997, pp. 13 – 20.

M. A. Kumar, M. N. Ramesh and S. Nagaraja Rao, Retrofitting of a Vertical Retort for On-line Control of the Sterilization Process, *Journal of Food Engineering*, vol. 47, 2001, pp. 89 – 96.

D. Kleis and E. W. Sachs, Optimal Control of the Sterilization of Prepackaged Food, *SIAM Journal on Optimization*, vol. 10, no. 4, 1999, pp. 1180 – 1195.

Z. S. Chalabi, L. G. van Willigenburg and G. van Straten, Robust Optimal Receding Horizon Control of the Thermal Sterilization of Canned Foods, *Journal of Food Engineering*, vol. 40, 1999, pp. 207–218.

R. S. Sutton and A. G. Barto, *Reinforcement Learning: an Introduction*, The MIT Press, Cambridge, MA, 1998.

S. Syafiie, F. Tadeo and E. Martinez, Learning to Control pH Processes at Multiple Time Scales: Performance Assessment in a Laboratory Plant, *Chemical Product and Process Modeling*, vol. 2, no. 1, 2007, article no 7.

S. Syafiie, F. Tadeo and E. Martinez, Model-Free Learning Control of Neutralization Process Using Reinforcement Learning, *Engineering Application Of Artificial Intelligence*, vol. 20, pp. 767 – 782, 2007.

D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, Belmont, Massachusetts, 1996.

C. G. Atkenson, A. W. Moore and S. Schaal, Locally Weighted Learning, *Artificial Intelligence Review*, Vol. 11, pp. 11–73, 1997.

C. G. Atkenson, A. W. Moore and S. Schaal, Locally Weighted Learning for Control, *Artificial Intelligence Review*, Vol. 11, pp. 75–113, 1997.

Damien Ernst, *Near optimal closed-loop control. Application to electric power systems*, PhD thesis at University of Lige, Belgium, 2003.

S. Timmer and M. Riedmiller, Fitted Q Iteration with CMACs, In Proceedings of *the International Symposium on Approximate Dynamic Programming and Reinforcement Learning* (ADPRL), Honolulu, USA, April 2007.

Martin Riedmiller, Learning Control for Continuous MIMO Dynamical Systems Using Neuro Dynamic Programming, in proceeding of *Third European Workshop on Reinforcement Learning*, Rennes, France, October 13-14, 1997.

A. A. Alonso, J. R. Banga and R. P. Martin, Modeling and Adaptive Control for Batch Sterilization, *Computers and Chemical Engineering*, vol. 22, no. 3, 1998, pp. 445–458.

W. D. Smart, Making Reinforcement Learning Work on Real Robots, PhD thesis at Brown University, 2002.