

# Optimum Operation of a Beer Filtration Process

Cesar de Prada, Smaranda Cristea, Rogelio Mazaeda, and Luis G. Palacín

**Abstract** This paper deals with the optimum operation of a beer filtration process that uses membranes for this task. Due to fouling, the operation requires cleaning, which damages the membranes, and creates a discontinuous operation. The optimal economic operation can be defined in terms of minimizing the number of chemical cleanings, as well as the use of energy, when processing a certain amount of beer in a given time. The problem is hybrid in nature, due to the discontinuities created by the cleanings. The corresponding optimization problem is formulated in the framework of predictive control but integrating the economic operation as target of the controller and different time scales. Also, instead of using binary variables for representing the discontinuities, the problem employs a sequential approach, embedding them in the dynamic simulation of the process model combined with a control parameterization that allows computing the solution in terms of the continuous variables that represent its degrees of freedom. Results of the optimal operation are presented.

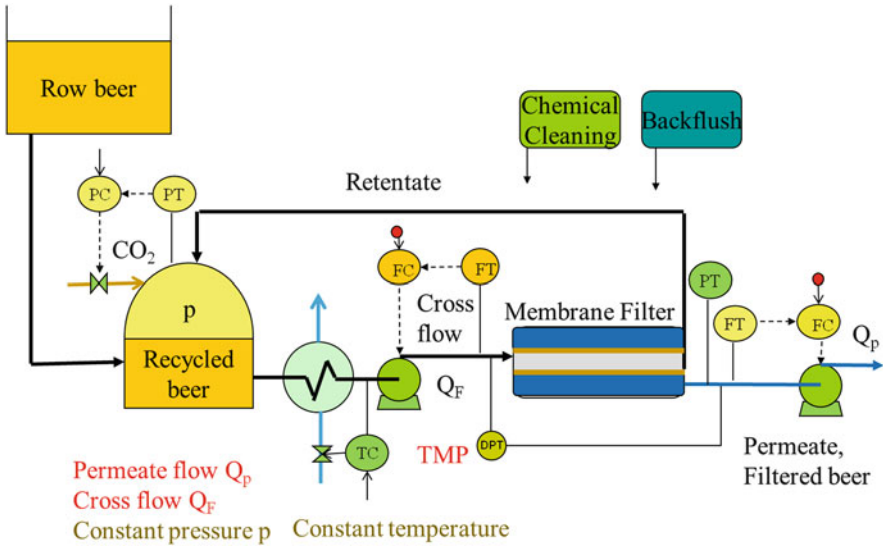
## 1 Introduction

In breweries, beer is filtered in order to remove different types of impurities helping obtaining in this way the desired color and taste. Today, this operation is performed in many factories using special membranes that allow pass the beer and retain the impurities. The process follows the schematic displayed in Fig. 1: from a pressurized tank of filtered beer, this is pumped to a set of membranes where the separation takes place. One fraction goes through the membrane as permeate while the rest, the retentate, returns to the recycled beer tank, circulating again in closed loop until all beer has been processed.

The membranes considered in this paper are of hollow fiber type, like the ones in Fig. 2: a bunch of fibers each one being a long pipe with microscopic holes in its

---

C. de Prada (✉) • S. Cristea • R. Mazaeda • L.G. Palacín  
Department of Systems Engineering and Automatic Control, University of Valladolid,  
Valladolid, Spain  
e-mail: [prada@autom.uva.es](mailto:prada@autom.uva.es); [smaranda@autom.uva.es](mailto:smaranda@autom.uva.es); [rogelio@cta.uva.es](mailto:rogelio@cta.uva.es);  
[palacin@cta.uva.es](mailto:palacin@cta.uva.es)



**Fig. 1** Schematic of the basic control system of the beer filtration process

surface that allow permeating the beer but not the impurities. The beer flows along the fibers flowing out partially as permeate, while the non-filtered portion leaves the fiber by the other end. From the point of view of the beer filtration, these are of two kinds: those that do not cross the membrane and form a cake layer in the inside, acting as an additional filter, and those that get trapped in the membrane pores obstructing them partially.

The operation of the plant is usually performed using the control system represented in Fig. 1. A pressure control loop maintains the pressure in the recycled beer tank injecting  $CO_2$  gas to avoid foam formation, while two flow control loops fix the values of permeate and cross flow and another loop helps maintaining the recycled beer temperature using a heat exchanger. In addition, there are measurements of the permeate side pressure and the trans-membrane pressure. The cake layer grows as time passes due to new deposits of impurities and, as a result, the trans-membrane pressure (TMP) that is required to maintain the flow of filtered beer increases. When this pressure has risen up to a certain value ( $TMP_{max}$ ), the filter needs to be cleaned to restore normal operation, which is done with the so called backflush cleanings (BF). These remove the cake layer, but not all the impurities attached to the membrane pores, so that, when the process is re-started again, the initial trans-membrane pressure required to maintain the permeate flow is higher than in the previous cycle, and the time required to reach the  $TMP_{max}$  decreases, as can be seen in Fig. 3. After several cycles, this time  $t$  is too short and a deeper cleaning is needed, the so called chemical cleaning (CIP) that restores the membrane to its original state, but producing damage so shortening its life

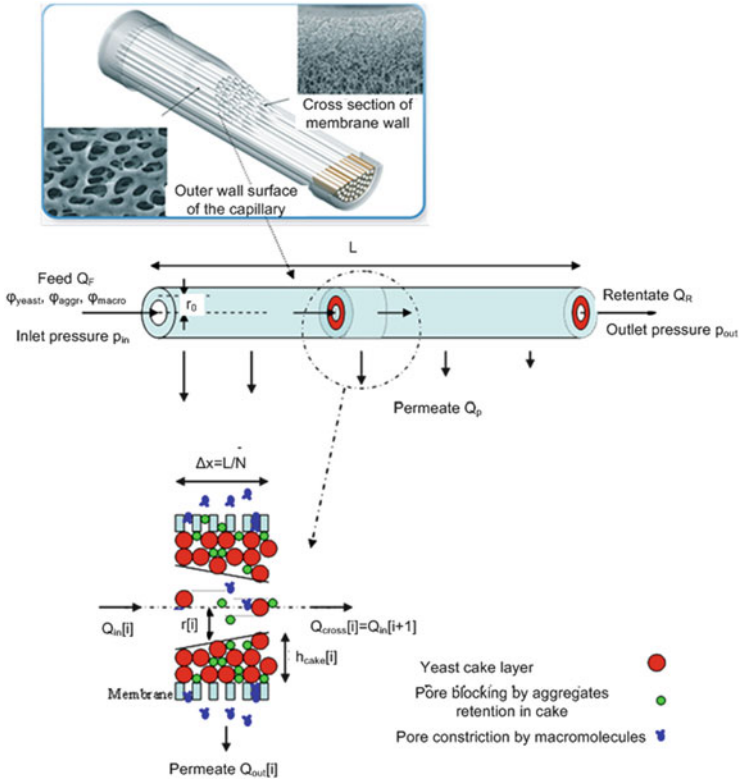


Fig. 2 Schematic of a membrane, fiber and a section with the internal cake layer and different types of impurities

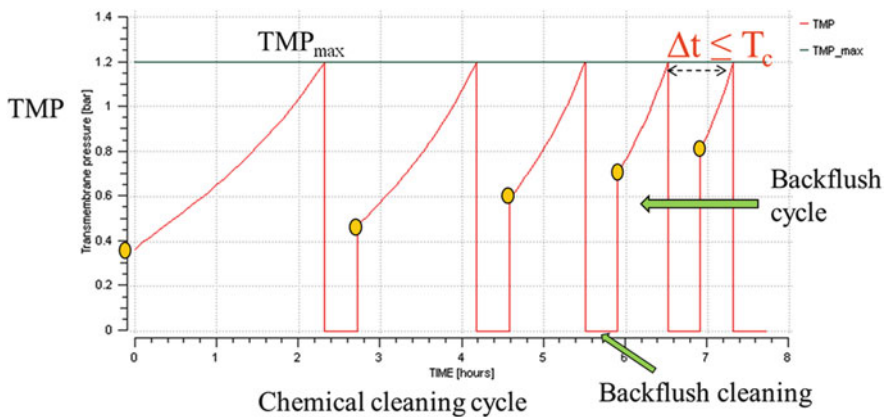


Fig. 3 Time evolution of the trans-membrane pressure between two chemical cleanings, including several backflush cleanings

The usual target is to filter a certain amount of beer to be delivered to the clients by a specified time. The beer to be filtered is stored in a raw beer tank and is supplied to the recycled beer tank to be processed. Besides fulfilling this target, there is a clear interest in performing the task in the best possible way, both from economic and process points of view. The economics involve the energy pumping costs and the cleaning costs, while the process operation requires fulfilling a set of constraints and enlarge the membrane life, minimizing the number of chemical cleanings needed to process a certain amount of beer, because membranes are quite expensive.

The optimal plant operation involves decisions at different levels, in particular, the following values have to be chosen: the number of chemical cleanings (*CIPs*), the number of backflashes per *CIP*, the value of the cross and permeate flow set points ( $Q_F$ ,  $Q_P$ ) and the maximum trans-membrane pressure ( $TMP_{max}$ ) between backflashes that allow processing the specified amount of beer by the required final time. Notice that besides different levels, there are also discrete and continuous variables implicated. The natural approach is, then, to formulate a set of hierarchical optimization problems like in [3], where a three layer optimization is proposed, with the two upper ones solving MIP problems to schedule production, providing the number of cleaning cycles, and the one below them and above the control layer using NLP optimization to determine the best  $TMP_{max}$  and flows. Nevertheless, this approach involves a lot of computation and it is not suitable for real-time application.

This paper presents an alternative that merges economic optimization and control and uses especial parameterizations to solve the problem using a small number of NLP problems in a single layer. This provides an efficient way of solving the problem and shows a way of dealing with mix-integer dynamic optimization problems that can be applied in other contexts as well. The paper is organized as follows: after this introduction, the optimization problem is formulated in Sect. 2, then, Sect. 3 shows results obtained with a particular beer assignment and gives implementation directions. The paper ends with some conclusions and references.

## 2 Optimizing Control

The main elements of the proposed approach are synthesized next. The first one is integration of economic optimization and dynamic control in the framework of Model Predictive Control (MPC), what is called optimizing control, [4, 6, 8]. The approach takes decisions about the manipulated process variables at a certain time instant according to the solution of a dynamic optimization problem, where the cost function is directly the economic aim one tries to optimize. The solution is applied to the process and the next sampling time the problem is repeated, updated with the new information collected from the process. The second element recognizes the fact that the behavior of the process between two chemical cleanings (*CIP*) is uniform and that not all the decision variables mentioned in the different time scales involved are fully independent. In fact, given the cross and permeate flow and a  $TMP_{max}$ ,

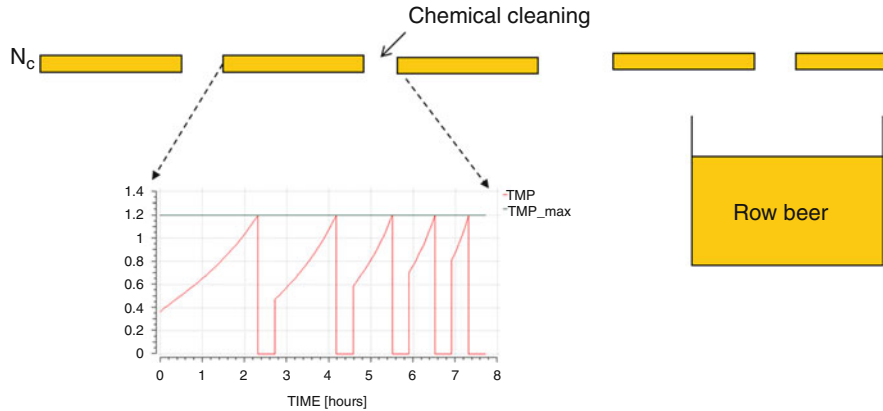


Fig. 4 Sequence of  $N_C$  chemical cleaning cycles and detail of one of them

and assuming that a minimum time between backflushes,  $T_c$ , has been fixed, the number of backflushes between two chemical cleanings and the total duration of the CIP,  $t_c$ , can be determined, and, hence, the number of CIPs, as some global constraints concerning the total amount of beer to be processed must be satisfied. In particular, with reference to Fig. 4, these constraints are:

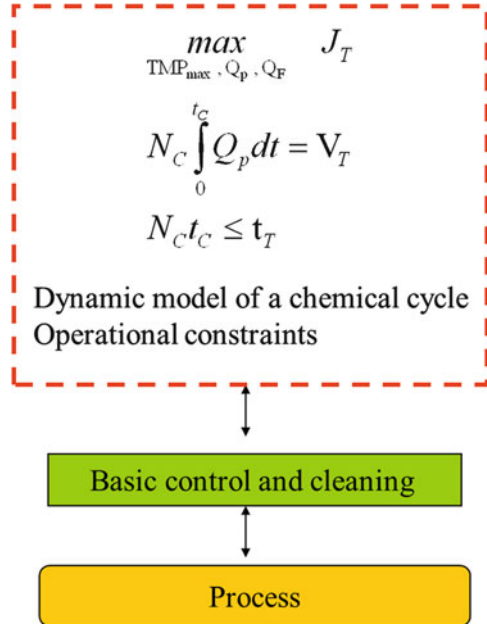
$$\begin{aligned}
 N_C \int_0^{t_c} Q_p dt &= V_T \\
 N_C t_c &\leq t_T
 \end{aligned}
 \tag{1}$$

Where  $N_C$  is the number of CIPs,  $t_c$  is the elapsed time of a chemical cycle,  $V_T$  is total amount of filtered beer to be processed, which is known,  $Q_p$  the permeate flow and  $t_T$  the assigned time for filtering. Notice that  $N_C$  from (1) can be a real number, as the last CIP cycle can finish incomplete. So, the global problem can be reduced to the optimization of a chemical cleaning cycle, assuming of all them uniform, provided that the constraints (1) are explicitly considered in the optimization. In this way, the large time scale covered by the processing of the beer assignment can be shortened and the corresponding time scale integrated in a shorter time scale problem.

This problem is displayed in Fig. 5. The cost function to maximize is the global one, covering the total number of CIP cycles, but the decision variables and the dynamic model are reduced to the ones of a chemical cycle. In particular, the cost function  $J_T$  is formulated as the economic target:

$$J_T = N_C \left[ \begin{array}{l} \alpha_0 \int_0^{t_c} Q_p dt - \alpha_1 \int_0^{t_c} p Q_p dt \\ -\alpha_2 \int_0^{t_c} p_F Q_F dt - \alpha_3 \int_0^{t_c} Q_F dt - \alpha_4 N_B - \alpha_5 \end{array} \right]
 \tag{2}$$

**Fig. 5** The optimizing control problem and its implementation



Which includes several terms with  $\alpha_i$  representing prices:  $\alpha_0$  price of beer,  $\alpha_1$  price of energy in permeate pump,  $\alpha_2$  price of energy in cross flow pump,  $\alpha_3$  price of cross flow cooling,  $\alpha_4$  cost of a backflush,  $\alpha_5$  cost of a chemical cleaning and  $N_C$  the number of *CIPs*. Notice that the first term tries to maximize the filtered beer per *CIP*, contributing indirectly to maximizing the membrane life which expands for a certain number of *CIPs*.

The constraints are the ones that apply to a *CIP*, with the addition of the global ones (1), and include ranges for the process variables and other operational limitations. Concerning the decision variables, the two flows  $Q_F$  and  $Q_P$ , cross flow and permeate, as well as the maximum trans-membrane pressure to fire a backflush cleaning,  $TMP_{max}$ , were chosen. They can be given different parameterizations, compromising between increasing the degrees of freedom of the problem and increasing the computation time.

The third element deals with the way the optimization is performed considering the discontinuities over a *CIP*. The embedded logic approach [8] has been used, which follows a sequential method for solving the dynamic optimization problem, including the discontinuous operation as part of the model simulation according to Fig. 6.

The dynamic model follows [9] and comprises equations for hydraulic flows, growing cake, membrane fouling, etc. It has been implemented in the simulation environment EcosimPro [1], an object oriented language implementing a good treatment of discontinuities and facilities for optimization according to a sequential approach. The integration of the model starts from the current point and implements

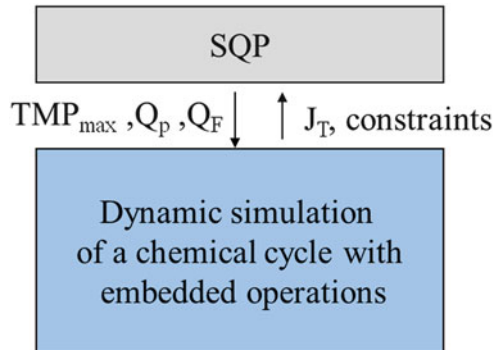


Fig. 6 Logic embedded approach to discontinuous dynamic optimization

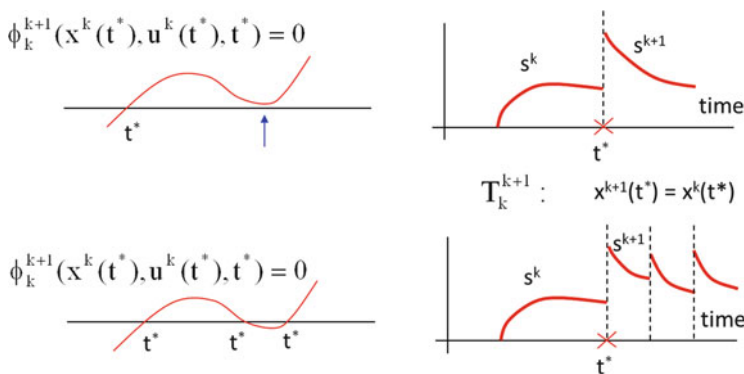


Fig. 7 Transition function and sensibilities with state discontinuities

a backflush cleaning each time the  $TMP$  reaches the value  $TMP_{max}$ , starting again the filtration from the new conditions of the membrane pores until the elapsed time between two backflushes is less or equal to  $T_C$ , in which case a chemical cleaning takes place and the simulation ends. So,  $N_C$  and  $J_T$  are evaluated, as well as the constraints and their values are sent back to the optimization algorithm. As no explicit integer variables are used, in principle, an NLP method such as SQP (Sequential Quadratic Programming), could be used.

Nevertheless, notice that the discontinuities involved are not input discontinuities but state ones and this may create problems in the computation of the cost function gradient in the optimization. This type of discontinuities are fired when a certain transition function  $\phi(x, u)$  in the model changes sign, according to the time evolution of the model states  $x$ , as in the upper part of Fig. 7, where a change from model  $k$  to model  $k + 1$  at time instant  $t^*$  is represented besides the corresponding sensibilities  $s$ . Of course, the sensibilities, defined as the derivatives of the states with respect to the decision variables, may jump at  $t^*$ , but, in principle, this jump does not mean that the gradient of the cost function is discontinuous.

The problem may appear if the situation represented in the bottom of Fig. 7, takes place, because here, a small change in the decision variable, changes the number of discontinuities, impeding the right computation of the gradients. This is exactly the situation with our problem, in which a small change in the value of  $TMP_{max}$  may change by one the number of backflushes, that is of discontinuities of a *CIP*.

To avoid this problem, an alternative was used in the paper, based in recognizing that the integer number of possible backflushes typically doesn't change much in practice. It consist in solving in parallel a number of similar problems, but, each one with a fixed and different number of backflushes  $N_B$ , and then select the one with the best cost function  $J_T$ . As  $N_B$  is fixed, the above mentioned problem disappears and an NLP method can be used, which speeds up the computations. Of course, the price to pay is repeating the NLP problem a small number of times (3–6 being adequate) but it compensates as there is no need to formulate the problem as a mixed-integer optimization one. In order to impose a fix number of backflushes in the operation, notice that two new constraints must be added in order to obtain realistic solutions:

$$\begin{aligned} \Delta t_{N_B} &\leq T_c \\ \Delta t_{N_B-1} &> T_c \end{aligned} \quad (3)$$

That corresponds with forcing the last backflush to fulfill the condition of a chemical cleaning and the previous one not to fulfilling it.

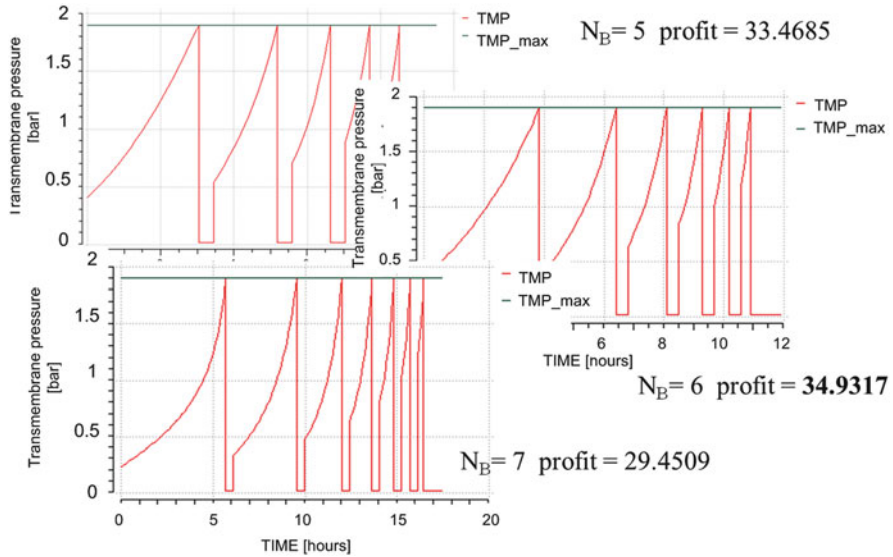
### 3 Results and Implementation

The proposed approach has been tested with a realistic simulation of the beer filtration process, coupled to an optimizing controller as the one described in section two. The numbers correspond to a small pilot plant and are not representative of an industrial installation, but describe quite well the possibilities of the method. The assignment was to process a volume of  $V_T = 0.051 \text{ m}^3$  in less than 5 days. The process constraints were given by:

$$\begin{aligned} 0.008 &\leq Q_F \leq 0.021 \quad \text{m}^3/\text{h} \\ 0.6 &\leq TMP_{max} \leq 1.9 \quad \text{bars} \\ 0.0002 &\leq Q_P \leq 0.0007 \quad \text{m}^3/\text{h} \end{aligned} \quad (4)$$

In Fig. 8, results of three parallel dynamic optimizations, each one identical, but with different values of  $N_B$ , taken as 5, 6 and 7, are given. In this case, the associated cost function  $J_T$  to the case  $N_B$  equal to 6 is the best one, so that this value was selected. The optimal solution corresponds to a benefit of 34.9317 Euros, and the assignment was performed in less than 5 days, with 9.66 *CIPS*, which means 9 full cycles and a partial one at the end. The optimal decision variables were:





**Fig. 8** Optimal solution of three parallel cases with  $N_B$  selected as 5, 6 and 7 backflushes

Regarding implementation, it is worth mentioning that the differential algebraic system of equations (DAE), resulting after discretization along the fiber axial direction of the original partial differential equations (PDE) first-principle model, reaches a substantial size: with 16 discretizations, it presents a number of 647 equations in total, with 68 dynamic states.

The required numerical optimization to obtain the best possible working regime for the installation is of a dynamical nature. In order to be able to apply the standard NLP techniques, the so called direct sequential approach [2] is adopted in this case. The input to the plant, or control vector, is parameterized, and then, the resulting initial value problem (IVP) is solved, meaning that the dynamic equations of the model are numerically integrated for a duration corresponding to a complete *CIP* cycle. The modelling and numerical integration has been implemented by means of the EcosimPro tool. The specific integration software used was DASSL [7], while the numerical programming has been performed by an SQP-type algorithm as implemented in SNOPT [5].

The obtained computation time of around 10 min, using a 1.83 GHz standard PC with 1 GB of RAM, it is not negligible. It can be attributed to the combination of already mentioned factors, in particular the size of the model and the control vector parameterization strategy adopted. In relation with the latter, it should be emphasized that the model has to be integrated for a whole *CIP* cycle horizon at every step of the optimization algorithm. Furthermore, this integration is performed, not once, but several times for each SQP iteration, in order to automatically obtain

**Table 1** Decision variables values

Variable	Value	Units
$N_B$	6	
$Q_F$	0.008	m <sup>3</sup>
$TMP_{max}$	1.9	bars
$Q_p$	0.00048444	m <sup>3</sup> /h
Chemical cleaning cycle	11.8916	h
Computation time	623	s

information concerning the gradients using discrete perturbations for each input. In any case, although there is room for improvement, the control processing time already obtained, opens the door to real-time implementation given the time scales of the process.

Further work is needed to include several elements that are required in an industrial plant. Among them, it is necessary to update the model with on-line measurements from the plant in order to tune it according to the different beer types and membrane quality that one can find over the process operation. At the same time, closed loop operation requires the incorporation in the problem formulation of the shrinking time horizon that appears in real-time operation as time passes. In relation with this, it should be said that the values for the three input degrees of freedom appearing in Table 1, are the optimal constant values which were obtained for the entire exercise in an open loop fashion. The implementation of a closed loop receding horizon scheme, would, on the other hand, immediately provide, at each sampling instant, new values for the same control inputs, that would be optimal at the new conditions, taking into consideration the presence of disturbances.

## 4 Conclusions

A novel approach has been presented to reformulate an optimization problem that combines different time scales and decision levels as well as continuous and discrete decisions. Furthermore, the model presents a variable structure in the sense that it has several discontinuities whose existence cannot be known beforehand since they depend on the dynamical evolution of the state. The approach is based on the idea of optimizing control, mixing economic optimization and dynamic control, as well as in the use of embedded logic optimization with a sequential approach to dynamic optimization. Finally, the discontinuity analysis has led to incorporate parallel computation to solve the hybrid problem in an efficient way. The method has been tested in a simulated process giving promising results.

## References

1. Agrupados, E.: EcosimPro, User Manual. [www.ecosimpro.com](http://www.ecosimpro.com) (2010)
2. Biegler, L.T., Grossmann, I.E.: Retrospective on optimization. *Comput. Chem. Eng.* **28**, 1169–1192 (2004)
3. Blankert, B.: Short to medium term optimization of dead-end ultra-filtration. PhD thesis, University of Groningen, The Netherlands (2007)
4. Engell, S.: International scientific & technical encyclopedia (iste). *J. Process Control* **17**, 203–219 (2007)
5. Gill, P.E., Murray, W., Saunders, M.A.: Software for large-scale nonlinear programming. Department of Management Science and Engineering, Stanford University, Stanford (2008)
6. González, A.I., Zamarreño, J.M., Prada, C.: Controlador predictivo con significación económica: aplicaciones simuladas a procesos batch. *Ingeniería Electrónica, Automática y Comunicaciones* **XXIII**, 26–32 (2002)
7. Petzold, L.: A Description of DASSL: A Differential/Algebraic System Solver. Technical report 82-8637. Sandia National Laboratories, Livermore, CA, USA (1982)
8. Rodríguez, M., Sarabia, D., Prada, C.: Hybrid predictive control of a simulated chemical plant. In: *Taming Heterogeneity and Complexity of Embedded Control*, pp. 617–634. ISTE, London/Newport Beach (2007)
9. Van der Sman, R.G.M.: Simulation of confined suspension flow at multiple scale lengths. *Soft Matter* **22**, 4376–4387 (2009)