A Novel All-in-One Real-Time Optimization and Optimal Control Method for Batch Systems: Algorithm Description, Implementation Issues, and Comparison with the Existing Methodologies

Francesco Rossi, Flavio Manenti,* and Guido Buzzi-Ferraris

Politecnico di Milano, Dipartimento di Chimica, Materiali e Ingegneria Chimica "Giulio Natta" Piazza Leonardo da Vinci 32, 20133 Milano, Italy

1. INTRODUCTION

Discontinuous and multistage processes were and are often still managed by means of traditional and heuristic recipes, conventional control loops, and/or manual operations. This is mainly due to their batch nature that requires frequent manual interventions, that is, to switch from operating to off conditions and to enable/disable cooling and heating operations or loading/unloading procedures. Moreover, the adopted conventional control methodology is often only partially effective to handle the set-point changes defined by batch production recipes and typical semibatch operation uncertainties.

For these reasons, many authors focused on batch processes to find efficient solutions to make them more automatic and better controlled. Most of the times model-based control techniques (proposed for the first time in ref 1) have been used with the aim of improving safety and optimizing intrinsically batch operations.[2−8] Several authors have also shown the potential for applying the dynamic optimization to batch systems,[9−11] studying in detail even the most appropriate control methodology to be selected and employed.[12,13] Moreover, both for optimal control and dynamic optimization, the best ways to perfom the process modeling have also been addressed.[14]

Nevertheless, the dynamic optimization is generally applied to real batch systems only offline so it has several similarities to the traditional recipe. Actually, although optimized, the set-point profiles are calculated a priori and are usually applied to the batch process offline. As a consequence, possible uncertainties or condition changes are not handled during the operations, reducing in practice the benefits of the procedure. This problem is dramatically emphasized with long batch operations such as fermentations or chemical vapor depositions, for example, the Siemens process for polysilicon production that requires several days of batch operation.[15] The main reasons why the dynamic optimization is rarely applied online may be identified as (i) the need for a sufficiently detailed but low-computational demanding process model; (ii) the complexity in the tuning of the control system; and (iii) the possible fluctuations, in the process variables, that may arise due to the interaction between the optimization and the process control layers.

To reduce the importance of some of the above-mentioned issues, the economic model predictive control has been proposed. Here the economic objective function of the dynamic optimization replaces the standard quadratic function employed in the model predictive control.[16,17] Some recent advances in this economic optimal control field include a Lyapunov-based economic model predictive control scheme for continuous processes where the objective function is made explicitly time-dependent in its economic weights.[18] Moreover, a linear optimal control methodology, where supply chain and scheduling-based objective functions are adopted, has also been studied in refs 19 and 20. However, these advanced model predictive control methods have been applied to batch systems in very few cases so far.

Therefore, it can be useful to investigate advanced model-based optimization and control methods, specifically designed for batch systems. The current paper is based on this topic. Indeed, a novel, model-based optimization and control method for batch and semibatch systems, the batch simultaneous model-based optimization and control (BSMBO&C), is proposed. This method allows both optimization and control of a discontinuous process in real-time by means of any user-defined performance indicator. A performance indicator is any mathematical function that is able to properly quantify the profitability of the process (not only economic or quadratic functions). In addition, the proposed strategy considers both the batch process manipulated variables and the batch cycle duration as degrees of freedom, which can be changed in order to take the controlled system to the highest profitability. This provides better efficiency and flexibility compared to the one guaranteed by the standard dynamic optimization and optimal control strategies, proposed in literature. Indeed, in the

Received: April 3, 2014
Revised: September 12, 2014
Accepted: September 13, 2014
Published: September 14, 2014

standard approaches the cycle duration is typically fixed and assigned a priori. Finally, the proposed methodology is designed to work online so it is perfectly able to handle possible perturbation changes that may occur during the batch cycle execution. The effect of each perturbation change can be either minimized, if negative, or maximized, if positive.

In terms of layout the paper is organized as follows: (i) the proposed algorithm description is provided; (ii) the implementation logics, essential to convert the method into a C++ tool, are conveyed; (iii) two test cases, used to prove the method effectiveness and flexibility and to compare its performance to that of the most common existing alternatives, are reported.

2. THE BSMBO&C ALGORITHM

The BSMBO&C algorithm derives from a coupling, a generalization, and an extension of nonlinear model predictive control (NMPC) strategies and dynamic real-time optimization (DRTO) techniques. It is specifically designed to be applied to discontinuous systems, and its aim is to provide both an online optimization and a process control at the same time. Inside it, the manipulated variable time-variant profiles and the batch operational time as well are simultaneously calculated by means of an optimization procedure, whose objective function is generic and completely customizable. Most of the times it is reasonable to define this objective function as an economic indicator for the controlled process but this is not mandatory at all. For instance, the objective function might also be a measure of the process environmental impact or an assessment of the produced amount of a certain product (i.e., yield, conversion, recovery) and so on. Moreover, the objective function can also be expressed in such a way that both online scheduling and pseudo-scheduling problems can be dealt with (further information on this topic is provided in section 2.2). Therefore, the BSMBO&C can be also considered as a significant improvement, limited to batch and semibatch systems, of a standard economic model predictive control (EMPC) strategy. Indeed, the batch cycle duration is added to the optimization and control problem degrees of freedom and the optimization problem objective function becomes completely customizable (even online scheduling and pseudo-scheduling problems can be solved).

To introduce the mathematical formulation of BSMBO&C scheme, let $\mathbf{d}$, $\mathbf{m}$, and $\mathbf{w}$ be the vectors of the perturbations, the manipulated variables, and the dependent variables of a batch process, respectively. Therefore, the process model can be written as

$$\begin{cases} \mathbf{I}_{\mathrm{M}} \dfrac{\mathrm{d}\mathbf{w}}{\mathrm{d}t} = \mathbf{f}(\mathbf{w},\ \mathbf{m},\ \mathbf{d}) \\ \mathbf{w}(t_0) = \mathbf{w}^0 \end{cases} \tag{1}$$

where $\mathbf{I}_{\mathbf{M}}$ is a diagonal matrix that can be either nonsingular, that is, the process model is an ODE system, or singular, that is, the process model is a DAE system. Once an initialization procedure, which is needed to set the user-defined settings and the controlled system initial condition, is completed, the proposed BSMBO&C methodology turns out to be based on the iterative application of a basic step. This step, constituted by several substeps, is continuously performed until a termination condition is fulfilled. The basic step is realized in this way:

I. An initial number of control intervals ($\mathbf{N}_{\mathrm{CI}}$) along with the standard width of each interval ($\Delta t_{\mathrm{CI}}^{0}$) is defined, or inherited from the previous basic step, for each manipulated variable.

II. Starting from a certain time instant (t^*), where the process operating condition (i.e., $\mathbf{d}$, $\mathbf{m}$, and $\mathbf{w}$ values) is known, the optimal manipulated variable profiles ($\mathbf{m}^{\mathrm{opt}}$) and the optimal residual batch operational time ($\Delta t_{\mathrm{BC}}^{\mathrm{opt}}$) are estimated through an optimization procedure (see section 2.2 for further details); in performing this optimization the manipulated variable trajectories are approximated via piece-wise constant functions.

III. The optimal trends of the manipulated variables ($\mathbf{m}^{\mathrm{opt}} t\ \epsilon\ [t^*, t^* + \Delta t_{\mathrm{CM}}]$) are implemented to the controlled system for a limited BSMBO&C algorithm-defined time interval (Δt_{CM}) and the response to the control action is measured and stored for future use.

IV. The initial time value for the next basic step (t_{new}^{*}) is evaluated by simply shifting t^* of Δt_{CM} units of time and the number of control intervals is updated (some details on the update logic used for $\mathbf{N}_{\mathrm{CI}}$ can be found in section 2.1).

The algorithm stop condition is satisfied when all the manipulated variables own a single control interval ($\mathbf{N}_{\mathrm{CI}} = 1$) and the optimal residual batch operational time ($\Delta t_{\mathrm{BC}}^{\mathrm{opt}}$) is lower than or equal to the minimum control interval standard width ($\Delta t_{\mathrm{CI,MIN}}^{0}$).

To further clarify how BSMBO&C algorithm works, a block diagram is provided in Figure 1. In that diagram the whole basic step and all the related substeps are explained by means of logic blocks. In detail (i) the oval-shaped blocks identify the substeps; (ii) the rectangular-shaped blocks represent either the user-defined data for the method initialization or the results coming from the substeps execution; (iii) the only rhomboidal block is used to symbolize the generic discontinuous system to be controlled.

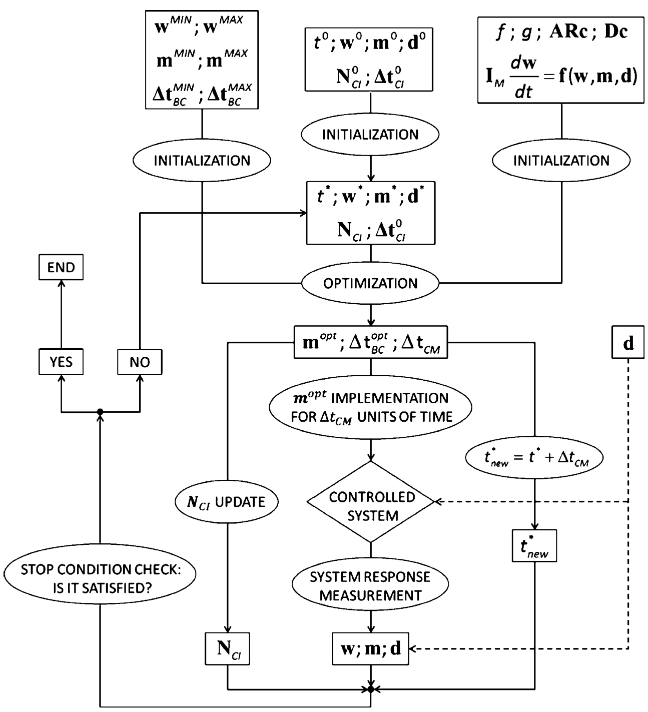

Figure 1. BSMBO&C algorithm block diagram.

The brief description of BSMBO&C methodology, reported in the current section, conveys the idea that this method is both an optimization tool and a control tool at the same time. It also suggests that the BSMBO&C is definitely able to drive a discontinuous process to profitable operating conditions and to handle random perturbations entering the controlled system, providing just-in-time optimal reactions and corrections. Moreover, the BSMBO&C novel capability of simultaneously managing both the time level and the manipulated variables level and the chance of using a completely generic and customizable objective function, make it very effective and flexible.

2.1. The Management of the Manipulated Variable Trends Discretization. BSMBO&C algorithm approximates the optimal manipulated variable trends by piece-wise constant functions. For each manipulated variable a specific discretization grid, defining the piece-wise approximant structure, is introduced. A standard grid for the *i*th manipulated variable is reported in Figure 2. Figure 2 suggests that each grid, thus each

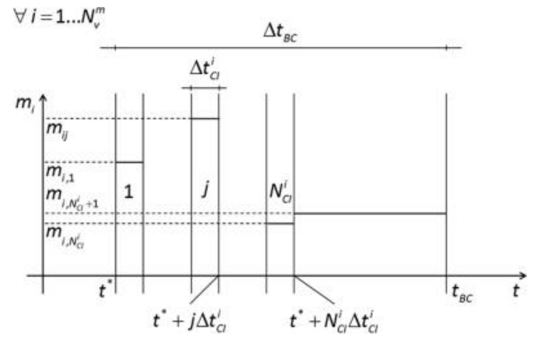

Figure 2. *i*th manipulated variable discretization grid structure.

manipulated variable, owns a certain number of control intervals (N_{CI}^i), all equivalent in terms of width (Δt_{CI}^i). Both the number and the width of the control intervals may vary during each BSMBO&C algorithm basic step: (i) N_{CI}^i, which starts from a user-defined initial value ($N_{CI}^{0,i}$) at the beginning of the simultaneous optimization and control procedure, can be updated at the end of the step; (ii) Δt_{CI}^i, which is set to $\Delta t_{CI}^{0,i}$ at the beginning of the step, can be adjusted several times during the optimization substep.

It is important to highlight that every manipulated variable relates to a specific $N_{CI}^i - \Delta t_{CI}^i$ couple that is independent to the other couples, i.e. each manipulated variable owns a discretization grid that is different to those of the other manipulated variables. This last feature is very important to save computational resources, thus improving the overall algorithm efficiency. Indeed, more control intervals can be allocated only to those manipulated variable profiles that really require them to be well-approximated.

The variation in each N_{CI}^i value is set in order to correctly handle to two opposite situations: (i) the case in which the estimated batch cycle duration (t_{BC}) overlaps the control horizon of the *i*th manipulated variable (i.e., $N_{CI}^i \Delta t_{CI}^{0,i}$); (ii) the case in which the ratio between the *i*th manipulated variable control horizon and the estimated residual batch cycle duration (Δt_{BC}) dramatically enlarges.

In the first case the N_{CI}^i update is performed with eq 2 while in the second case it is carried out with eq 3 (ε_i^0 is a dimensionless parameter indicating the ratio between the *i*th manipulated variable control horizon and an estimation of the residual optimal batch cycle duration in starting conditions). In all the situations in which neither the first nor the second case is the current one, N_{CI}^i is kept constant. It is relevant to add that the acronym *floor* in eq 2 and eq 3 stands for the approximation of a real number to the closest lower integer number.

$$N_{CI}^i = \max\left\{1;\ floor\left(\frac{\Delta t_{BC}}{\Delta t_{CI}^{0,i}}\right)\right\} \tag{2}$$

$$N_{CI}^i = \min\left\{N_{CI}^{0,i};\ floor\left(\varepsilon_i^0 \frac{\Delta t_{BC}}{\Delta t_{CI}^{0,i}}\right)\right\} \tag{3}$$

It is also important to highlight that the first of the above-mentioned cases must occur, more than once, before the BSMBO&C stop condition is satisfied, because of the batch nature of the system to which the algorithm is applied. The second above-mentioned case, instead, rarely occurs and it can only be caused by some perturbation changes effect.

The adjustment of a Δt_{CI}^i, instead, may be necessary in some iterations of the optimization substep, relating to each BSMBO&C algorithm basic step. In detail, when the estimated batch cycle duration overlaps the *i*th manipulated variable control horizon, the control interval width is rescaled with eq 4. It is relevant to highlight that each Δt_{CI}^i might be rescaled several times during an optimization substep. Moreover an already rescaled Δt_{CI}^i might also be restored to its original starting value ($\Delta t_{CI}^{0,i}$), that is, its standard value, if the estimated residual batch cycle duration goes back to be greater than the *i*th manipulated variable control horizon.

$$\Delta t_{CI}^i = \frac{\Delta t_{BC}}{N_{CI}^i + 1} \tag{4}$$

Handling the discretization of the manipulated variable optimal profiles with the reported methodologies allows BSMBO&C to effectively manage both the unavoidable trend of reduction of Δt_{BC} (typical of discontinuous systems) and the eventual nonmonotonic behavior of the same variable. The nonmonotonic behavior of Δt_{BC} may be caused by the presence of several perturbations affecting the controlled system. Moreover, the reported discretization management method also allows a minimization of the inaccuracy in the computation of the optimal manipulated variable trajectories, which may arise for the Δt_{BC} variations. Indeed, the BSMBO&C can independently adapt the width of each manipulated variable control horizon to the Δt_{BC} trajectory.

2.2. The Optimization Substep Features. Let $\mathbf{w}_{BC}$ be the dependent variable values in t_{BC} and let $\mathbf{w}^{CM}$ be the dependent variable values in ($t^* + \Delta t_{CM}$). Moreover, let $\mathbf{d}^*$, $\mathbf{m}^*$, and $\mathbf{w}^*$ be the process operating point in t^* and let **Dc** and **ARc** be tuning parameters of the BSMBO&C method. For a batch system with N_v^w dependent variables and N_v^m manipulated variables, the optimization problem that must be solved, at each basic step of BSMBO&C algorithm, is described in eq 5. This equation constitutes a constrained nonlinear (nonconvex) optimization problem where both nonlinear differential-algebraic and bound constraints are present. Its solution can be approached with different methodologies, but these issues are discussed in detail in section 3. For now, it is important to focus the attention on its mathematical structure and, in detail, on both the objective function and the nonlinear differential-algebraic constraints formulation.

The objective function ($f_{obj_{BSMBO\&C}}$) is constituted by (i) two user-defined performance functions (f and g); (ii) a derivative check term relating to the batch system dependent variables (eq 5, first line); (iii) an antiringing term (eq 5, second line).

All these three terms are combined as eq 5 shows. One such formulation has been selected because, by properly choosing f and g and accordingly setting **Dc** and **ARc**, almost any possible function can be achieved and so any possible optimization and control problem can be addressed. Indeed, an online optimization and control problem for a single batch cycle can be dealt with by setting f to a constant value of 1 and g to an indicator of a single batch cycle profitability. Instead, a pseudo-scheduling problem can be managed by defining f as the number of batch cycles that can be carried out in a predefined campaign time, at the current cycle duration and g as an indicator of a single batch cycle profitability. Finally, an online scheduling problem can be addressed by setting f to the number of batch cycles required to achieve a remaining scheduled production, at the current production level per cycle, and g to an indicator of a single batch cycle profitability.

It is clear that a wide range of different possibilities can be covered, also because, in the previous lines, g is described as a generic indicator of a single batch cycle profitability, that is, one out of a net income, a cost, a recovery, a yield, a conversion, and so on. Moreover, the listed cases are just examples, and other ad-hoc sets of f and g may be developed for specific batch systems to which the BSMBO&C method is applied. To end f and g features description, it is very important to point out that the function resulting from $f{\cdot}g$ must decrease if the performance of the controlled system increases and vice versa. This can be easily derived from eq 5 itself. It is now clear that the properties of $f_{\mathrm{obj}_{\mathrm{BSMBO\&C}}}$ are mainly determined by the choice of f and g. However, the antiringing and the dependent variables derivative check terms play a role in limiting the manipulated and the dependent variables oscillatory behavior, respectively. Therefore, these terms are essential to make BSMBO&C method return feasible outputs. Indeed, neither the manipulated nor the dependent variables can be subject to strong fluctuations on a relatively small time scale, mainly because of the consequent and unwanted stress on the equipment. Moreover, the dependent variables derivative check term is also very important when strongly sensitive systems, for example, a reactor that may be subject to runaway problems, have to be managed. In this case, this term helps to avoid a contingent control loss that may lead to relevant safety issues. The **Dc** and **ARc** values, which define how strong the effect of the antiringing and the dependent variables derivative check terms is, must be chosen through a tuning method. These value sets depend on both the features of the specific controlled system and f and g functions formulation (see section 2.3 for further details on the tuning logic).

$$
\begin{cases}
\min\limits_{m_{ij};\Delta t_{\mathrm{BC}};\mathbf{w}} \left\{ f(t^*, t_{\mathrm{BC}}, \mathbf{w}^*, \mathbf{w}_{\mathrm{BC}}) \left[g(t^*, t_{\mathrm{BC}}, \mathbf{w}^*, \mathbf{w}_{\mathrm{BC}}) + \dfrac{1}{\Delta t_{\mathrm{CM}}^2} \sum_{k=1}^{N_v^w} \mathrm{Dc}_k (w_k^{\mathrm{CM}} - w_k^*)^2 + \sum_{i=1}^{N_v^m} \dfrac{\mathrm{ARc}_i}{(N_{\mathrm{CI}}^i + 1)(\Delta t_{\mathrm{CI}}^i)^2} \sum_{j=1}^{N_{\mathrm{CI}}^i+1} (m_{ij} - m_{i,j-1})^2 \right] \right\} \\
\text{s. t.} \\
\begin{cases} \mathbf{I}_{\mathrm{M}} \dfrac{\mathrm{d}\mathbf{w}}{\mathrm{d}t} = \mathbf{f}(\mathbf{w}, m_{ij}, \mathbf{d}^*) \\ \mathbf{w}(t^*) = \mathbf{w}^* \end{cases} \\
\Delta t_{CM} = \min\limits_i \{\Delta t_{\mathrm{CI}}^i\} \\
t_{\mathrm{BC}} = t^* + \Delta t_{\mathrm{BC}} \\
m_{ij} \in [m_{ij}^{\min}; m_{ij}^{\max}];\ \Delta t_{\mathrm{BC}} \in [\Delta t_{\mathrm{BC}}^{\min}; \Delta t_{\mathrm{BC}}^{\max}]; \\
\quad w_k \in [w_k^{\min}; w_k^{\max}]
\end{cases}
\tag{5}
$$

The other relevant term, which has to be described in this section, is the set of differential-algebraic nonlinear constraints. This set of equations constitutes the dynamic model of the discontinuous system to be managed. Since nonlinear system models are supported, the BSMBO&C method can be used to perform an online scheduling based on nonlinear models. This operation may lead to significant improvements over the standard scheduling techniques based, instead, on linear models.

In conclusion, it is clear that the structure of the optimization substep allows the BSMBO&C algorithm to simultaneously control and optimize a discontinuous system on the basis of any performance criterion defined by the user. For example, single batch cycle, pseudo-scheduling, and online scheduling problems can be addressed. As a consequence, the BSMBO&C appears to be very general, flexible, and customizable, and seems to define a novel concept of universal simultaneous optimization and optimal control too. Moreover, even very sensitive systems can be efficiently handled thanks to the dependent variables derivative check term that guarantees the BSMBO&C scheme to avoid dangerous contingent control losses.

2.3. The BSMBO&C Tuning Parameters and Tuning Rules. Since BSMBO&C algorithm is also a control method, it requires a tuning procedure. The tuning parameters can be identified as follows: (i) the antiringing coefficients (**ARc**); (ii) the slope control coefficients (**Dc**); (iii) the initial number of control intervals (N_{CI}^0); (iv) the standard width of the control intervals (Δt_{CI}^0).

The tuning affects the BSMBO&C behavior as follows: (i) increasing/decreasing **Dc** and **ARc** causes the BSMBO&C scheme to be more/less robust but less/more performing; (ii) increasing/decreasing N_{CI}^0 guarantees better/worse performances but also higher/lower computational effort; (iii) increasing/decreasing Δt_{CI}^0 may take to worse or better performances depending on both the controlled system features and the chosen N_{CI}^0 (no generalization is possible in this case).

The optimal values of all the above-mentioned tuning variables strongly depend on both the controlled system nature and the f and g functions formulation and slightly depend on the perturbations value. Moreover, it is almost impossible to apply the standard error-based tuning criteria (ISE, IAE, ITAE etc.) so a heuristic method is the only possible choice. The proposed heuristic method is constituted by three steps: (i) N_{CI}^0 and Δt_{CI}^0 are fixed based on the open-loop dynamic behavior of the controlled system when all the manipulated variables are fixed to a nominal value, for example, the average between their maximum and minimum; (ii) by starting from the orders of magnitude of both the manipulated and dependent variables and the g function, a reasonable first attempt value for **Dc** and **ARc** is computed; typically **Dc** and **ARc** can be chosen in order to cause about 0.5−5% increase of g when the order of magnitude of a manipulated or controlled variable changes by about 0.01−10%; (iii) the first guess antiringing and slope control coefficients are refined using a sensitivity analysis.

It is important to point out that, unlike **ARc** coefficients, only a few of all the **Dc** elements have to be typically set to nonzero values, that is, those related to particularly sensitive dependent variables. Therefore, the total number of **Dc** and **ARc** coefficients to be determined is limited. The last important point to notice is that both **Dc** and **ARc** are dimensional values, whose units of measure make the antiringing and the dependent variables derivative check term compatible with the g function dimensions.

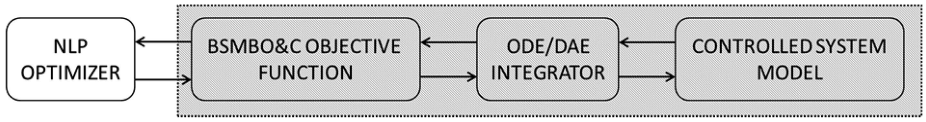

Figure 3. Black-box approach graphical view.

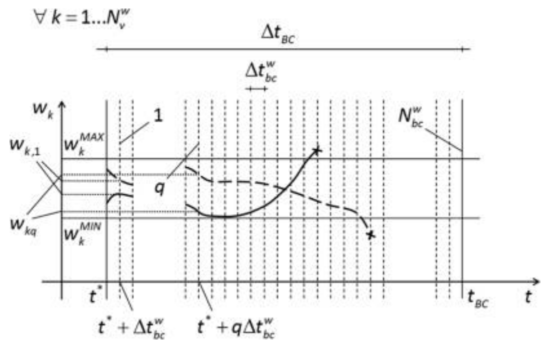

Figure 4. Graphical view of the boolean strategy for the case of a minimum (dashed line) and a maximum (solid line) bound violation.

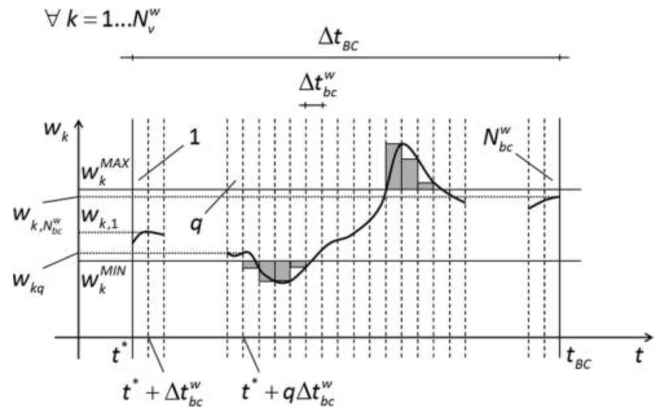

Figure 5. I_k^p approximation method in the weighted strategy.

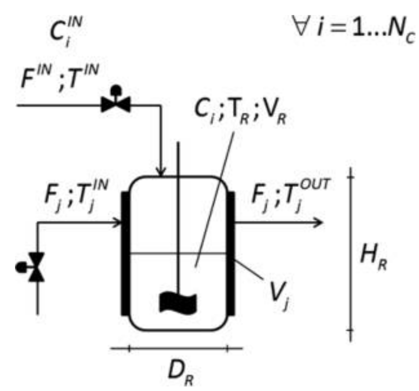

Figure 6. Fed-batch reactor drawing.

The usage of a dimensionless g function may simplify the tuning issues.

In conclusion, it is important to remember that the tuning rules reported here are only simple guidelines and rules of thumb. A more rigorous tuning method will be probably addressed in future works.

3. BSMBO&C METHOD IMPLEMENTATION

The formal structure of the BSMBO&C algorithm has already been described in section 2, but that abstract method has to be implemented into a numerical code. Therefore, some additional information on how to perform this implementation procedure must be provided. The coding is realized in C++, through a black-box approach (Figure 3). This choice allows the BSMBO&C tool to provide control moves that always relate to a feasible controlled system operating condition. Moreover, the use of a black-box approach also allows a limit of the number of degrees of freedom in the optimization substep, relating to each BSMBO&C algorithm basic step. (A trial version of the C++ tool is available for free at the SuPER group Web site (http://super.chem.polimi.it).)

Since a black-box approach has been selected, both a NLP optimizer and an ODE/DAE initial conditions integrator are required. The NLP optimizer is needed to solve the optimization substep. The ODE/DAE integrator, instead, is essential to integrate the controlled system model inside the BSMBO&C objective function ($f_{\text{obj}_{\text{BSMBO\&C}}}$). Indeed, for each $f_{\text{obj}_{\text{BSMBO\&C}}}$ evaluation a controlled system model integration must be performed.

The employed optimization and integration tools must be endowed of several features, described in detail in section 3.1, and have been taken from the BzzMath library.[21] Moreover, the optimization substep implementation is critical in order to get a final tool that is efficient and reliable: the methodologies employed in its realization are detailed in section 3.2. All the other implementation issues are more trivial and less interesting, thus they are not described in detail.

3.1. Optimization and Integration Tool Features. Both the NLP optimizer and the ODE/DAE initial conditions integrator must fulfill some minimum requirements in order to be suitable for the BSMBO&C method.

The optimizer must be sufficiently robust but, at the same time, efficient. Moreover, it must be able to handle discontinuous objective functions and, at least, bound constraints on the optimization variables. The chosen BzzMath library optimizer exploits an efficiency-improved Nelder−Mead simplex method, coupled with a penalty function approach for the management of the bounds on the optimization degrees of freedom. Therefore, it meets the above-mentioned requirements and is suitable for the BSMBO&C methodology. The reader should notice that the chosen optimization algorithm is only able to find local minima, even though the optimization substep may result in a nonconvex problem. Unfortunately, using a global optimizer is not feasible due to the subsequent excessive computational effort. However, such a global optimization strategy is likely to be employed in the near future thanks to both the increase in the CPUs clock and the parallel computing.

The integration routine, instead, must be very efficient and must be able to provide satisfying performances when the ODE/DAE system to be solved is strongly nonlinear. The selected solver of the BzzMath is based on efficiency-improved Gear multivalue methods, and it is specifically designed to handle also strongly nonlinear ODE/DAE systems. In addition, it is also configured to solve systems with an elevated ratio between the

Table 1. Process, Initial and Boundary Conditions, Reactor Structural Data, and Economic Value of the Components

Kinetic Scheme		
reactions	$A + B \rightarrow C$	(1)
	$C \rightarrow D$	(2)
	$2B \rightarrow E$	(3)
rate equations and kinetic parameters	$R_1 = k_1^0 \exp\left(-\frac{E_1}{RT}\right)C_a C_b$	$k_1^0 = 1.0880 \times 10^{10}$ m^3/(kmol*s)
		$k_2^0 = 9.2226 \times 10^{10}$ 1/s
	$R_2 = k_2^0 \exp\left(-\frac{E_2}{RT}\right)C_c$	$k_3^0 = 2.1780 \times 10^{10}$ m^3/(kmol*s)
		$E_1 = 8.40 \times 10^4$ kJ/kmol
	$R_3 = k_3^0 \exp\left(-\frac{E_3}{RT}\right)C_b^2$	$E_2 = 1.02 \times 10^5$ kJ/kmol
		$E_3 = 9.25 \times 10^4$ kJ/kmol
Boundary Variables Initial Value		
fed-batch feed inlet conditions	$F^{IN} = 0$ m^3/s	$T^{IN} = 298$ K
	$C_a^{IN} = 0$ kmol/m^3	$C_d^{IN} = 0$ kmol/m^3
	$C_b^{IN} = 1$ kmol/m^3	$C_e^{IN} = 0$ kmol/m^3
	$C_c^{IN} = 0$ kmol/m^3	
coolant inlet conditions	$F_j = 1 \times 10^{-3}$ m^3/s	$T_j^{IN} = 340$ K
Reactor Structural Parameters		
vessel size	$H_R = 2.10$ m	$D_R = 1.12$ m
jacket volume	$V_j = 0.3859$ m^3	
global heat transfer coefficient	$U = 1.085$ kW/(m^2·K)	
Thermodynamic Data		
reacting mixture components specific heat	$Cp_a = 75.31$ kJ/(kmol·K)	$Cp_d = 204.12$ kJ/(kmol·K)
	$Cp_b = 167.36$ kJ/(kmol·K)	$Cp_e = 334.73$ kJ/(kmol·K)
	$Cp_c = 217.57$ kJ/(kmol·K)	
coolant specific heat	$Cp_j = 4.186$ kJ/(kg·K)	
coolant density	$\rho_j = 1 \times 10^3$ kg/m^3	
heats of reaction	$\Delta H_{R,1} = -6.3200 \times 10^4$ kJ/kmol	$\Delta H_{R,3} = -1.0376 \times 10^5$ kJ/kmol
	$\Delta H_{R,2} = -1.5280 \times 10^5$ kJ/kmol	
Process Variable Constraints		
vessel filling	$V_R^{MAX} = 1.5516$ m^3	$V_R^{MIN} = 0$ m^3
reacting mixture temperature	$T_R^{MAX} = 373$ K	$T_R^{MIN} = 320$ K
jacket temperature	$T_j^{OUT,MAX} = 373$ K	$T_j^{OUT,MIN} = 273$ K
coolant inlet flow	$F_j^{MAX} = 0.1$ m^3/s	$F_j^{MIN} = 0$ m^3/s
fed-batch feed inlet flow	$F^{IN,MAX} = 5 \times 10^{-4}$ m^3/s	$F^{IN,MIN} = 0$ m^3/s
State Variables Initial Condition		
reacting mixture components concentration	$C_a^0 = 1$ kmol/m^3	$C_d^0 = 0$ kmol/m^3
	$C_b^0 = 0$ kmol/m^3	$C_e^0 = 0$ kmol/m^3
	$C_c^0 = 0$ kmol/m^3	
reacting mixture volume	$V_R^0 = 0.75$ m^3	
reacting mixture temperature	$T_R^0 = 340$ K	
jacket temperature	$T_j^{OUT,0} = 340$ K	
Miscellaneous Data		
reacting mixture components molar mass	$PM_a = 30$ kg/kmol	$PM_d = 130$ kg/kmol
	$PM_b = 100$ kg/kmol	$PM_e = 200$ kg/kmol
	$PM_c = 130$ kg/kmol	
universal gas constant	R = 8.314 kJ/(kmol*K)	
reacting mixture components economic value	$EV_a = 10$ €/kg	$EV_d = 0$ €/kg
	$EV_b = 30$ €/kg	$EV_e = 0$ €/kg
	$EV_c = 100$ €/kg	
coolant economic value	$EV_{coolant} = 1.5 \times 10^{-3}$ €/kg	

number of algebraic and differential equations. Therefore, it is perfectly suitable for the BSMBO&C method.

In addition both the adopted optimizer and the chosen ODE/DAE integrator exploit parallel computing: this results in a further efficiency improvement that is inherited by the BSMBO&C tool.

3.2. The Optimization Substep Implementation. As it is reported in section 2.2, the optimization substep requires the solution of a constrained NLP problem with both bound constraints and differential-algebraic nonlinear constraints. Nevertheless, thanks to the usage of the black-box approach (Figure 3), this very complex optimization problem is reduced to a simpler constrained NLP with bound constraints only. Indeed, the differential-algebraic nonlinear equations, that is, the controlled system model, are directly employed in the computation of the BSMBO&C objective function ($f_{obj_{BSMBO\&C}}$), thus they are automatically satisfied. Although the DAE equations have been formally removed from the constraints list, a new issue arises and has to be solved. Some of the bound constraints limit the feasible regions of the controlled system dependent variables ($\mathbf{w}$) but these variables do not formally belong to the optimization degrees of freedom any more. These bound constraints are quite cumbersome to handle so two different methodologies have been specifically developed for this purpose: the *boolean* strategy and the *weighted* strategy.

To introduce the *boolean* strategy description, it is necessary to specify a feature included in the BzzMath optimizer, adopted in the implementation of the BSMBO&C algorithm. This optimization tool let the user specify when the function to be minimized is infeasible (a function is infeasible when its evaluation causes a certain user-supplied infeasibility condition to be satisfied). If an infeasibility exception is thrown inside the minimization procedure, the optimization algorithm discards the current trial point, leaves the current search region and makes a new iteration. Coming now to the *boolean* strategy, it is based on these simple actions (see Figure 4):

(I) In each optimization substep of each BSMBO&C basic step, several evaluations of $f_{\mathrm{obj}_{\mathrm{BSMBO\&C}}}$ are required, and to perform each of these evaluations a controlled system model integration from t^* and $\mathbf{w}^*$ to t_{BC} and $\mathbf{w}_{\mathrm{BC}}$ is needed. For each $f_{\mathrm{obj}_{\mathrm{BSMBO\&C}}}$ calculation, the compliance of the dependent variables to their bound constraints is checked, using the model integration results, in $N_{\mathrm{bc}}^{\mathrm{w}}$ points belonging to the interval $(t^*;t_{\mathrm{BC}}]$ (for the sake of simplicity suppose those points to be uniformly distributed even though the real BSMBO&C tool does not exactly work in this way).

(II) If only one of the described bound checks fails, an infeasibility exception is thrown, thus causing the current trial set of manipulated variables and residual batch operational time to be discarded.

As a consequence, the optimal set of manipulated variables ($\mathbf{m}^{\mathrm{opt}}$) and the optimal residual batch operational time ($\Delta t_{\mathrm{BC}}^{\mathrm{opt}}$), achieved at the end of each optimization substep, should be such that the bounds on the controlled system dependent variables are globally satisfied inside the interval $(t^*;t_{\mathrm{BC}}]$.

The *weighted* strategy, instead, is based on an event-based penalty function method. It is similar, for some perspectives, to the *boolean* strategy but a relevant novelty is introduced. The strategy is based on these actions:

(I) In each optimization substep of each BSMBO&C basic step, several evaluations of $f_{\mathrm{obj}_{\mathrm{BSMBO\&C}}}$ are required, and to perform each of these evaluations, a controlled system model integration from t^* and $\mathbf{w}^*$ to t_{BC} and $\mathbf{w}_{\mathrm{BC}}$ is needed. For each $f_{\mathrm{obj}_{\mathrm{BSMBO\&C}}}$ calculation, the accordance of the dependent variable trends to their bound constraints is checked, using the model integration results, inside the interval $(t^*;\ t_{\mathrm{BC}}]$.

(II) If the kth dependent variable trend violates its maximum or minimum constraint in some subintervals included in $(t^*;\ t_{\mathrm{BC}}]$, a rough approximation of the integral of the absolute difference between the kth dependent variable profile and its maximum or minimum (I_k^p) is computed in these subintervals (see Figure 5 for a graphical insight).

(III) This procedure is repeated for each dependent variable, and a global penalty coefficient is calculated by means of a weighted sum of the previously evaluated integrals (I_k^p).

(IV) The global penalty coefficient is scaled to the standard BSMBO&C objective function ($f_{\mathrm{obj}_{\mathrm{BSMBO\&C}}}$) order of magnitude and to a penalty parameter (ξ) and then added to $f_{\mathrm{obj}_{\mathrm{BSMBO\&C}}}$ itself in order to build its penalized equivalent ($f_{\mathrm{obj}_{\mathrm{BSMBO\&C}}}^{\mathrm{p}}$).

(V) Only for the current iteration of the optimization substep, $f_{\mathrm{obj}_{\mathrm{BSMBO\&C}}}$ is replaced with $f_{\mathrm{obj}_{\mathrm{BSMBO\&C}}}^{\mathrm{p}}$, evaluated right now.

Thanks to this methodology the optimal set of manipulated variables ($\mathbf{m}^{\mathrm{opt}}$) and the optimal residual batch operational time ($\Delta t_{\mathrm{BC}}^{\mathrm{opt}}$), achieved at the end of each optimization substep, are likely to be such that the bounds on the controlled system dependent variables are globally satisfied or, at least, only slightly violated, inside the interval $(t^*;\ t_{\mathrm{BC}}]$. The presence and entity of the possible bound violations mainly depend on the value of the ξ coefficient but, typically, if a bound violation occurs, it equals a relative error around 0.1−1%.

Looking now at the above-written qualitative description of the *boolean* and *weighted* strategies, it comes out that both work in a discrete manner (Figures 4 and 5). The first performs $N_{\mathrm{bc}}^{\mathrm{w}}$ punctual bound checks, the second computes an integral (I_k^p) by means of a discretization based on $N_{\mathrm{bc}}^{\mathrm{w}}$ intervals of $\Delta t_{\mathrm{bc}}^{\mathrm{w}}$ time units. It is clear that the higher $N_{\mathrm{bc}}^{\mathrm{w}}$ is, the more accurate and reliable, but also the more computationally expensive, both the *boolean* and *weighted* strategies are. As a consequence, the $N_{\mathrm{bc}}^{\mathrm{w}}$ parameter must be carefully chosen based on the controlled system dynamic features by the BSMBO&C tool user. The faster its dynamics is, the higher the value of $N_{\mathrm{bc}}^{\mathrm{w}}$ must be.

Moving now from a qualitative toward a quantitative description of the *boolean* and *weighted* strategies, it results that the *boolean* strategy does not require a mathematical formalization because the qualitative and quantitative explanations are almost identical. Instead, the *weighted* strategy can be formalized in a set of equations (eq 6). Equation 6 quantitative formulation is chosen in order to ease and standardize the choice of the $f_{\mathrm{obj}_{\mathrm{BSMBO\&C}}}$ penalty parameter (ξ), whose influence on the numerical complexity of the optimization substep is huge. Indeed, by using the reported formulation, ξ is dimensionless and a value of 5−10 is satisfying for most of the practical cases (it means that no bound violations are typically observed with such a ξ value).

$$\begin{cases} f_{\mathrm{obj}_{\mathrm{BSMBO\&C}}}^{\mathrm{p}} = f_{\mathrm{obj}_{\mathrm{BSMBO\&C}}} + \varphi_{\mathrm{obj}}\xi \\ \displaystyle\sum_{k=1}^{N_{\mathrm{v}}^{\mathrm{w}}} \frac{I_k^p}{w_k^{\max} - w_k^{\min}} \\ \text{where:} \\ \varphi_{\mathrm{obj}} = \max\{1;\ |f_{\mathrm{obj}_{\mathrm{BSMBO\&C}}}|\} \\ I_k^p = \Delta t_{\mathrm{bc}}^{w} \displaystyle\sum_{q=1}^{N_{\mathrm{bc}}^{w}} (\alpha_{kq}|w_{kq} - w_k^{\max}| + \beta_{kq} \\ \quad |w_{kq} - w_k^{\min}|) \\ \alpha_{kq} = \begin{cases} 1 & \text{if } w_{kq} > w_k^{\max} \\ 0 & \text{otherwise} \end{cases} \\ \beta_{kq} = \begin{cases} 1 & \text{if } w_{kq} < w_k^{\min} \\ 0 & \text{otherwise} \end{cases} \end{cases} \tag{6}$$

Up to now, two different strategies for the dependent variable bounds handling have been described, but no information has been provided on which of the two options is the most performing or when one of the two alternatives is more efficient than the other. On the one hand, the *boolean* strategy guarantees no dependent variable bound violations at all when the

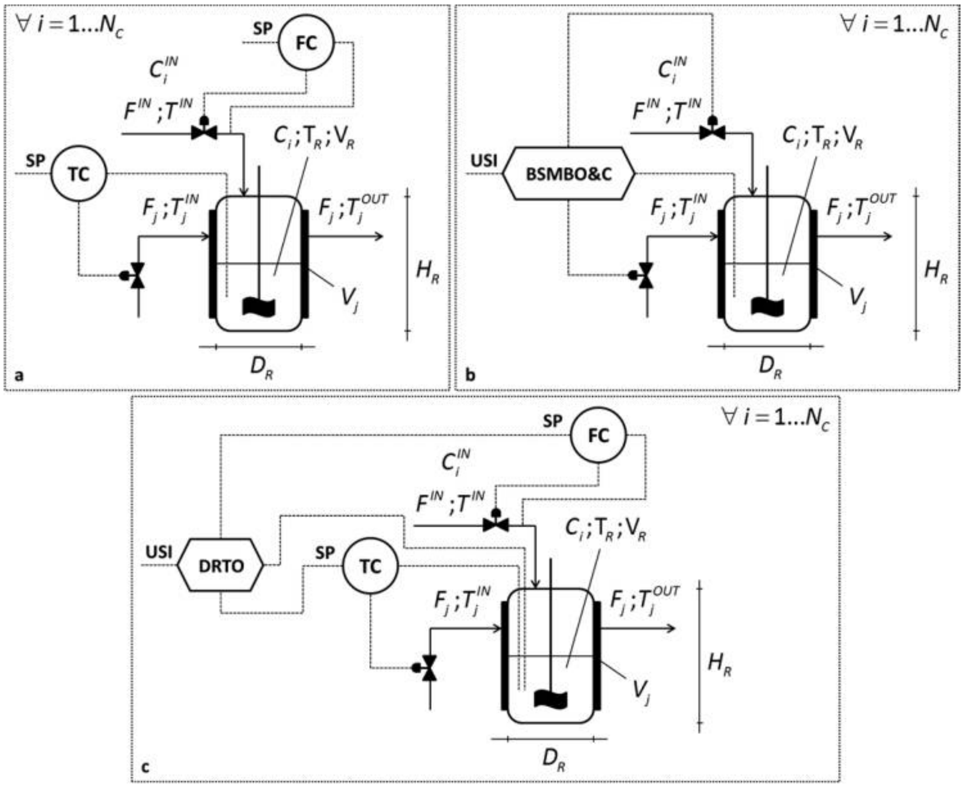

Figure 7. Fed-batch system in the OOS-PID (a), DRTO-PID (c), and BSMBO&C (b) layout.

optimization substep successfully converges, and slightly less computational effort because, once the first bound violation exception is thrown, the controlled system model integration can be aborted in the current $f_{\mathrm{obj}_{\mathrm{BSMBO\&C}}}$ calculation. On the other hand, the *weighted* strategy provides slightly superior results, in terms of $f_{\mathrm{obj}_{\mathrm{BSMBO\&C}}}$ values and significantly higher performances when the region of the $m_{ij} \cup \Delta t_{BC}$ space that guarantees the absence of dependent variable bound violations is very narrow (this condition is often observed when the controlled system is very sensitive).

As a result, none of the two proposed dependent variable bounds handling methodologies is always successful, so both of them are essential. Different controlled system dynamic behaviors and phenomenological complexity and a different controlled system sensitivity make either the *boolean* or the *weighted* strategy be the right choice.

Now that proper methods to manage the dependent variable bounds have been derived, the implementation of the optimization substep becomes trivial. Indeed, the NLP of eq 5 can be efficiently solved by using the black-box approach coupled with either the *boolean* or the *weighted* strategy and the chosen BzzMath optimizer (it is the BzzMath optimization algorithm that directly handles the bound constraints on both the manipulated variables and the residual batch cycle duration).

In conclusion, a proper way to implement the optimization substep has been presented in this section. Thanks to the BzzMath library optimizers and integrators, the described methodology is efficient enough to let the BSMBO&C algorithm work online but also stable enough to guarantee sufficient control robustness. The proof of this statement can be observed in the results of the case studies shown in section 4.

Table 2. Regulatory Control System Tuning Parameters, BSMBO&C Tuning Parameters and Allowed Batch Cycle Duration Range

BSMBO&C tuning parameters									
		manipulated variables							
		F_j					F^{IN}		
tuning parameters	**Arc**	1.2656×10^7 s^4/m^6					2.25×10^{11} s^4/m^6		
	N_{CI}^0	8					4		
	Δt_{CI}^0	45 s					60 s		
		dependent/state variables							
		C_a	C_b	C_c	C_d	C_e	V_R	T_j^{OUT}	T_R
tuning parameters	**Dc**	-	-	-	-	-	-	-	25.3125 s^2/K^2
BSMBO&C allowed batch cycle duration range									
maximum and minimum batch operational time						$t_{\mathrm{BC}}^{\mathrm{MAX}} = 2.16 \times 10^4$ s			$t_{\mathrm{BC}}^{\mathrm{MIN}} = 360$ s
regulatory control system tuning parameters									
proportional gain, integral time and derivative time		$K_C = 3.25 \times 10^{-4}$ m^3/(K·s)						$\tau_I = 120$ s	$\tau_D = 0.075$ s

4. BSMBO&C METHOD TEST CASES

The BSMBO&C has been described in detail in both its algorithm formulation and its practical implementation during the previous two sections. In the current section, instead, it is applied to a couple of test cases with the aim of proving its real effectiveness and comparing its performance to that of two well-established literature-based optimization and control methodologies. These other two literature-based methodologies are an offline optimization scheme, also called an offline optimal recipe-based scheme, coupled with a PID control loop (OOS-PID); and

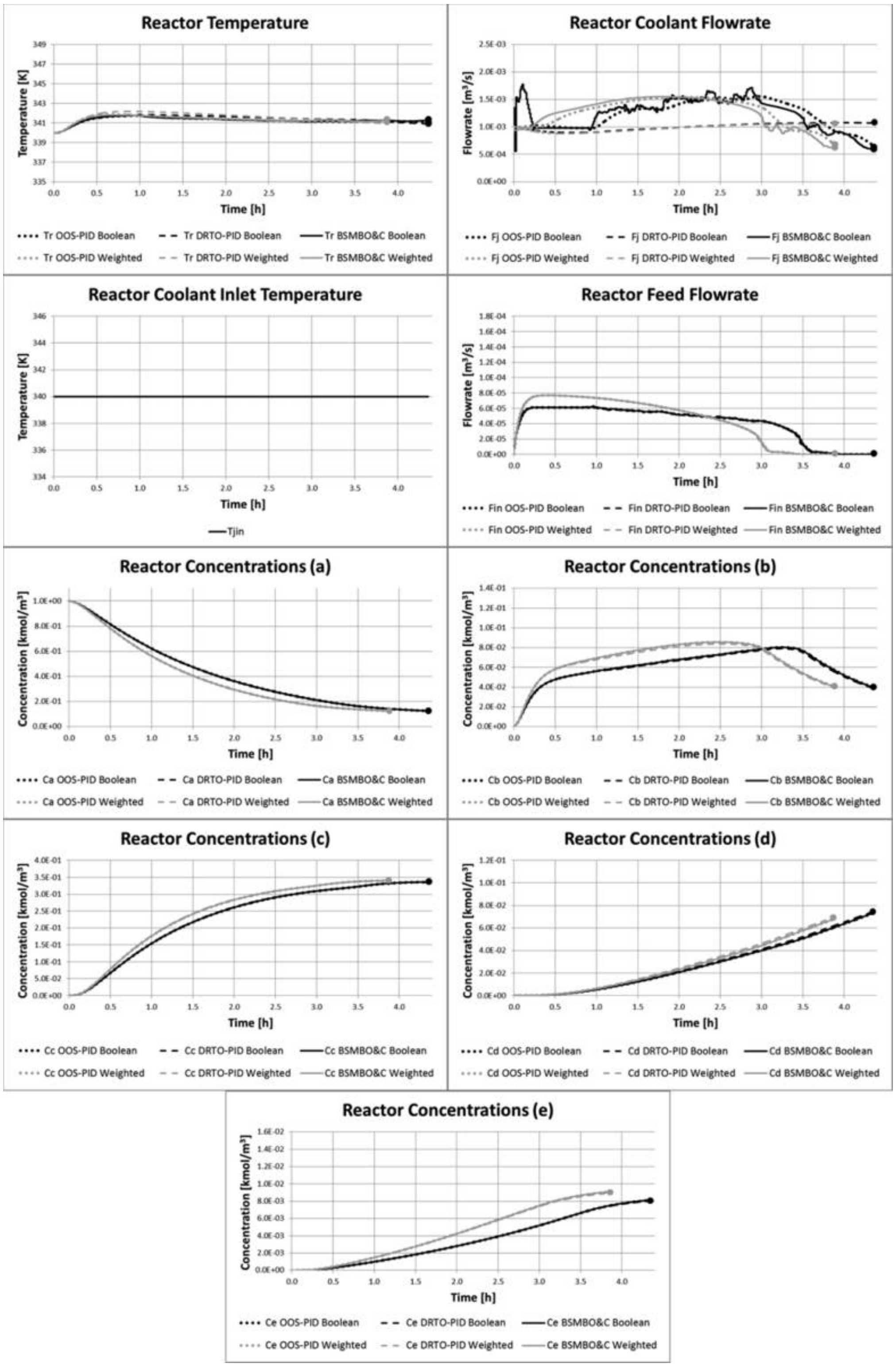

Figure 8. Single batch cycle optimization and control with no changes in T_J^{IN}.

a real-time dynamic optimization scheme coupled with a PID control loop (DRTO-PID).

The economic model predictive control technique, instead, has not been included in the previous list because its application to discontinuous processes is widespread neither in literature nor in the industrial practice so far. Moreover, a scheme where the DRTO is coupled to the NMPC has not been added to the previous list either. Indeed, it has been shown in ref 13 that, for batch systems, DRTO takes to better performances if coupled with conventional control networks.

Coming to a brief description of the two test cases, the first (section 4.2) is based on the comparison among the BSMBO&C algorithm performance and the OOS-PID and DRTO-PID methodologies effectiveness. The second case study (section 4.3) is entirely dedicated to the BSMBO&C and is aimed at proving its flexibility and the consequent advantages coming from this

Table 3. Economic Profitability of a Single Batch Cycle with No Changes in T_j^{IN}

	boolean strategy			weighted strategy		
	OOS-PID	DRTO-PID	BSMBO&C	OOS-PID	DRTO-PID	BSMBO&C
batch cycle duration [s]	15652.16	15652.16	15652.16	13948.60	13948.60	13948.60
g function [−]		0.5359	0.5373		0.5470	0.5490
net income [€]	3908.87	3898.84	3909.15	3993.81	3979.36	3994.23

Table 4. Economic Profitability of a Single Batch Cycle with Three Step-Changes in T_j^{IN}

	boolean strategy			weighted strategy		
	OOS-PID	DRTO-PID	BSMBO&C	OOS-PID	DRTO-PID	BSMBO&C
batch cycle duration [s]	15652.16	15652.16	9457.69	13948.60	13948.60	10797.63
g function [−]		0.4802	0.5664		0.4986	0.5702
net income [€]	2568.44	3493.58	4120.88	2973.98	3627.67	4148.38

feature. Moreover, while the first case study can be classified as a single batch cycle optimization problem, the second can be considered a pseudo-scheduling problem. No online scheduling problems are addressed in the current paper as one such problems will be probably studied in future works. Finally, notice that for both test cases, a discontinuous system, which the tests have to be based on, must be chosen. A fed-batch reactor, where three exothermic reactions take place, has been selected. The model of the just-mentioned system is reported in section 4.1.

4.1. Fed-Batch Reactor Modeling. As already suggested, the case studies, described in section 4.2 and 4.3, are based on a fed-batch reactor as a controlled system. The reactor drawing is shown in Figure 6.

Since the reaction mixture is supposed to be in the liquid phase and none of the mixture compounds is volatile in the allowed process thermal range, a single-phase piece of equipment is to be modeled. Moreover, to simplify the modeling issues, let the reactor and the cooling jacket be completely mixed and let all the thermodynamic properties be temperature-independent. Under these assumptions the fed-batch model is described by eq 7. In this equation, let N_R and N_C be the number of chemical reactions and components in the reacting system, R_l, ν_{il} and $\Delta H_{R,l}$ be the rate of the lth reaction, the stoichiometric coefficient of the ith component in the lth reaction and the heat of the lth reaction, U be the global heat transfer coefficient between the reacting mixture and the jacket cooling fluid, and Cp_i, Cp_j and ρ_j be the specific heat of the ith component in the reacting mixture, the coolant specific heat, and the coolant density.

The meaning of all the other employed symbols can be inferred from Figure 6. To avoid misunderstandings, remember also that, only for section 4, the subscript "i" refers to the general fed-batch system component and the subscript "j" identifies the parameters and variables related to the cooling jacket.

The ODE system reported in eq 7 requires several data to be practically solved: a set of initial conditions, a thermodynamic package, a kinetic scheme, and several reactor structural parameters. This necessary information, along with the economic value of the reacting system components, is included in Table 1. Notice that the fed-batch reactor configuration is partially derived from a well-known literature case in ref 22.

One last remark on the fed-batch reactor and, consequently, on its model concerns the identification of the perturbations and the dependent and manipulated variables. T^{IN}, T_j^{IN}, and C_i^{IN} are supposed to be perturbations, as it typically happens for these kinds of equipment. Therefore, the fed-batch reactor has eight dependent or state variables and two degrees of freedom, that is, two manipulated variables. In the current case, F^{IN} and F_j must be chosen as manipulated variables, whereas C_i, V_R, T_R, and T_j^{OUT} must be considered as dependent variables. Indeed, only F^{IN} and F_j can be physically adjusted during a batch cycle run. As a consequence, all the optimization and control schemes applied in the following test cases (section 4.2 and 4.3) will be compelled to manage two manipulated variables that will be F^{IN} and F_j.

$$\begin{cases} \dfrac{dC_i}{dt} = \dfrac{F^{IN}}{V_R}(C_i^{IN} - C_i) + \displaystyle\sum_{l=1}^{N_R} \nu_{il} R_l \\ \quad \forall\ i = 1 \cdots N_C \\ \dfrac{dV_R}{dt} = F^{IN} \\ \dfrac{dT_R}{dt} = \dfrac{4U}{D_R \sum_{i=1}^{N_C} C_i Cp_i}(T_j^{OUT} - T_R) \\ \quad + \dfrac{F^{IN} \sum_{i=1}^{N_C} C_i^{IN} Cp_i}{V_R \sum_{i=1}^{N_C} C_i Cp_i}(T^{IN} - T_R) \\ \quad - \dfrac{\sum_{l=1}^{N_R} \Delta H_{R,l} R_l}{\sum_{i=1}^{N_C} C_i Cp_i} \\ \dfrac{dT_j^{OUT}}{dt} = \dfrac{F_j}{V_j}(T_j^{IN} - T_j^{OUT}) \\ \quad + \dfrac{4UV_R}{D_R V_j \rho_j Cp_j}(T_R - T_j^{OUT}) \end{cases} \tag{7}$$

4.2. Test Case I. This first case study is aimed at making a comparison among the performances of the BSMBO&C, DRTO-PID, and OOS-PID methodologies, all applied to the semibatch reactor described in section 4.1. Scenarios, where both no and several perturbation changes occur, are analyzed. However, it is necessary to describe, at first, what DRTO-PID and OOS-PID schemes mean in this paper (the BSMBO&C logic has been widely addressed in the previous sections).

Few words are spent in the description of the DRTO-PID logic because it is widespread in literature. Here, an optimization loop, which works over the regulatory PID-based control system, computes in real-time, the optimal set-point for all the controllers which, in turn, provide the regulatory control actions.

The OOS-PID method, instead, is often still used in the industrial practice and must be explained in detail. Here, first of all, all the perturbations, which may influence the controlled system behavior, are set to a nominal value. Then, an offline optimization of the batch cycle, which leads to some optimal set-point trends, is performed. Finally, for each real batch cycle that is carried out these offline calculated set-points are used as online set-points of the PID-based regulatory control system.

Once the meanings of both DRTO-PID and OOS-PID methods have been clarified, it is possible to define how and with which settings these methods, along with BSMBO&C logic, are applied, for this case study, to the semibatch reactor that has been modeled in section 4.1. Figure 7 provides some of this information even though some additional contents, which are reported below, have to be added.

In terms of notation, the acronyms USI, SP, FC, and TC, included in Figure 7, stand for user supplied information, set-point, flow controller, and temperature controller.

By looking, once again, at Figure 7, it can be observed that the control loop of the reactor feed controls F^{IN} with F^{IN} itself. In other words, this loop identifies the controlled and manipulated variable. Moreover, since the mixture, fed to the reactor, is in the liquid phase, thus almost incompressible, the FCs can be considered perfect (the set-point is always equivalent to the controlled variable at any time instant). The reactor temperature control loop, instead, controls T_{R} with F_j and works with a PID algorithm. This loop has no peculiar features. As a consequence, the regulatory control system, which is employed in both the DRTO-PID and OOS-PID schemes, can be described by means of the expressions reported in eq 8. In this equation the following notation has been employed: K_{C} is the proportional gain, τ_{I} is the integral time, τ_D is the derivative time; F_j^{bias} is the reactor coolant flow bias; $F_{\mathrm{sp}}^{\mathrm{IN}}$ and $T_{\mathrm{R}}^{\mathrm{sp}}$ are the reactor feed and the reactor temperature set-points.

Finally, Figure 7 suggests that each controlled system dependent variable is supposed to be measurable and known in every time instant for both the DRTO-PID and the BSMBO&C scheme. If some of the dependent variables are not directly measurable, the easiest way to proceed is to suppose that their real values correspond to the values derived from the controlled system model.

It is clear that a tuning procedure for the PID controller, employed in both the DRTO-PID and the OOS-PID layout, must be chosen. This issue is not trivial since the optimal PID tuning parameters strongly depend on the batch cycle duration that, in the current case, can vary. The adopted tuning methodology is made of two steps: (I) several sets of optimal tuning parameters, related to different batch cycle durations, are evaluated by means of the minimization of the ISE indicator; (II) a pseudo-optimal set of PID parameters is derived from the data achieved in I.

By using this approach, the K_{C}, τ_{I} and τ_{D} values, employed in the current case study and summarized in Table 2, are achieved.

$$\begin{cases} F^{\mathrm{IN}} = F_{\mathrm{sp}}^{\mathrm{IN}} \\ F_j = F_j^{\mathrm{bias}} + K_{\mathrm{C}}\left[(T_{\mathrm{R}} - T_{\mathrm{R}}^{\mathrm{sp}}) \right. \\ \qquad \left. + \dfrac{1}{\tau_{\mathrm{I}}}\displaystyle\int_0^t (T_{\mathrm{R}} - T_{\mathrm{R}}^{\mathrm{sp}})\mathrm{d}t + \tau_{\mathrm{D}}\dfrac{\mathrm{d}}{\mathrm{d}t}(T_{\mathrm{R}} - T_{\mathrm{R}}^{\mathrm{sp}})\right] \end{cases} \tag{8}$$

Speaking now of the OOS-PID method, the only additional information that is required to apply it in the current case study is the set of optimal offline computed set-point profiles for F^{IN} and T_{R} (this set-points set is typically called the optimized recipe). These trajectories are approximated by means of an offline application of the BSMBO&C method itself. The method settings (f, g, **Arc**, **Dc**, etc.) used here are the same employed for the online simulations but all the perturbations, that is, T^{IN}, T_j^{IN} and C_i^{IN}, are set to a nominal value that is chosen to equal the initial value (see Table 1).

For the DRTO-PID logic more additional information is needed. Indeed, to make a meaningful comparison with the BSMBO&C methodology, a DRTO algorithm that is based on a black-box approach and shares the most relevant features with the BSMBO&C has been coded ad-hoc. The shared features include the objective function structure, including the tuning parameters selection type, the exploited numerical methods, the dependent variable bounds check strategies, and the manipulated variable trends discretization approach.

The settings of this ad-hoc DRTO-PID scheme, that is, f, g, **Arc**, **Dc**, etc., have been chosen to equal those used for the BSMBO&C method. Finally, since in the DRTO-PID approach the batch cycle duration (t_{BC}) is fixed, it is set to the values that derive from the optimal recipes evaluated for the OOS-PID method.

At last, coming to the BSMBO&C scheme, the adopted formulation of f and g, a set of tuning parameters, and the allowed batch cycle duration range must be provided. The chosen f and g functions are shown in eq 9, where the superscript "BC" stands for "evaluated in t_{BC}". Instead, the tuning parameters and the batch cycle maximum ($t_{\mathrm{BC}}^{\mathrm{MAX}}$) and minimum ($t_{\mathrm{BC}}^{\mathrm{MIN}}$) duration are summarized in Table 2. It is important to highlight that f and g are income-based functions for this test case. In detail, since a single batch cycle problem is studied, f is set to a constant value of 1 and g is defined as the dimensionless net income coming from a single batch cycle. This dimensionless net income is given by the dimensional net income, coming from a single batch cycle, divided by an order of magnitude of the maximum net income, achievable from a single batch cycle.

$$\begin{cases} f = 1 \\ g = \dfrac{[C_{\mathrm{a}}^0\mathrm{PM_aEV_a}V_{\mathrm{R}}^0 - (C_{\mathrm{c}}^{\mathrm{BC}}\mathrm{PM_cEV_c} + C_{\mathrm{d}}^{\mathrm{BC}}\mathrm{PM_dEV_d} + C_{\mathrm{e}}^{\mathrm{BC}}\mathrm{PM_eEV_e})V_{\mathrm{R}}^{\mathrm{BC}} + C_{\mathrm{b}}^{\mathrm{IN}}\mathrm{PM_bEV_b}\int_{t^*}^{t_{\mathrm{BC}}} F^{\mathrm{IN}}\mathrm{d}t + \rho_j\mathrm{EV_{coolant}}\int_{t^*}^{t_{\mathrm{BC}}} F_j\mathrm{d}t]}{C_{\mathrm{a}}^0 V_{\mathrm{R}}^0(\mathrm{PM_cEV_c} - \mathrm{PM_aEV_a} - \mathrm{PM_bEV_b})} \end{cases} \tag{9}$$

By using all the data reported in section 4 so far, four simulations have been carried out. Each of them includes a batch cycle execution performed with all the three optimization and control methodologies, that is, BSMBO&C, DRTO-PID, and OOS-PID. However, in the first two scenarios no perturbation changes occur, that is, T^{IN}, T_j^{IN} and C_i^{IN} are kept constant, but in the first the *boolean* strategy and in the second the *weighted* strategy are used. In the third and fourth scenario, instead, T_j^{IN} (the most critical perturbation) is given a piece-wise constant profile with three step-changes. Moreover, in the third simulation the *boolean* strategy is adopted and in the fourth simulation the *weighted* strategy is chosen. The results of the first two simulations are reported in Table 3 and Figure 8 while the data coming from the third and the fourth simulations are collected in Table 4 and Figure 9. It is relevant to highlight that for all the BSMBO&C-based scenarios addressed in the current case study the following optimization substep features can be observed: (i) The number of the optimization degrees of freedom varies from 3 to 13.

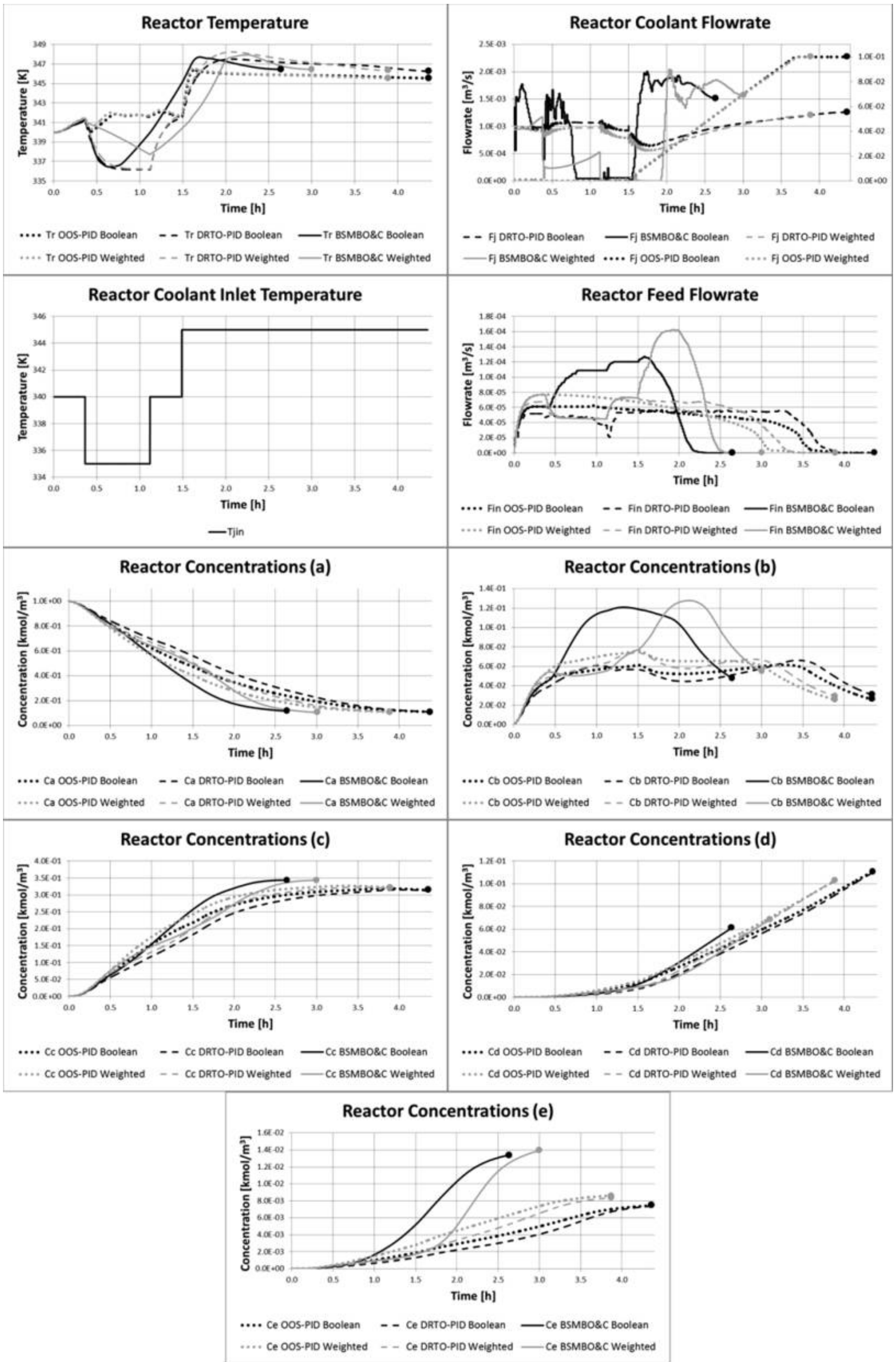

Figure 9. Single batch cycle optimization and control with three step-changes in T_J^{IN}.

(ii) The total number of variables, that is, the sum of the controlled system state variables and the optimization degrees of freedom, falls between 11 and 21. (iii) The number of differential-algebraic constraints equals 8. (iv) The number of bound constraints is twice the total number of variables, that is, it lies between 22 and 42. (v) The computational time required to complete a single optimization substep, which corresponds to the time required to compute a control move, is around 4 s.

For the features of the optimization substep relating to the DRTO-PID scheme simulations, no detailed discussion is reported. However, these features are similar to the ones already mentioned for the BSMBO&C-based simulations.

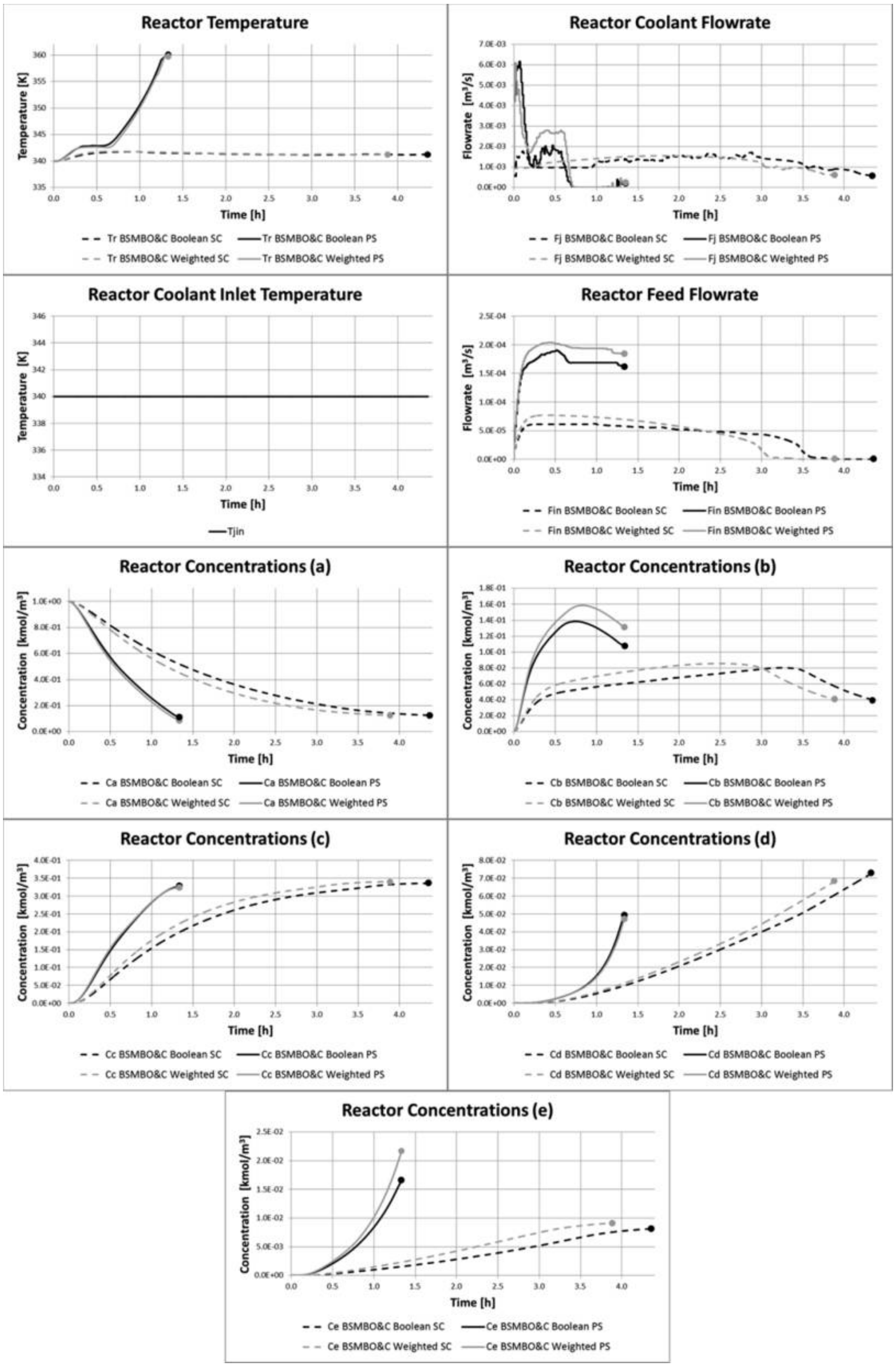

Figure 10. Pseudo-scheduling optimization and control with no changes in T_j^{IN}.

It is clear that the performance of the OOS-PID, DRTO-PID, and BSMBO&C schemes is almost the same when no variations in the reactor coolant inlet temperature occur. It can be derived both by looking at the single batch cycle net income in Table 3 and by observing that all the concentration and temperature trends in Figure 8 are almost identical. This sounds reasonable: when no perturbation changes occur, i.e. T_j^{IN} is constant, the DRTO-PID and the BSMBO&C methods take the reactor to an optimal operating condition that is almost the same defined by the offline computed optimal set-point trajectories used in the OOS-PID scheme. As a consequence, all the three methodologies take the semibatch reactor to

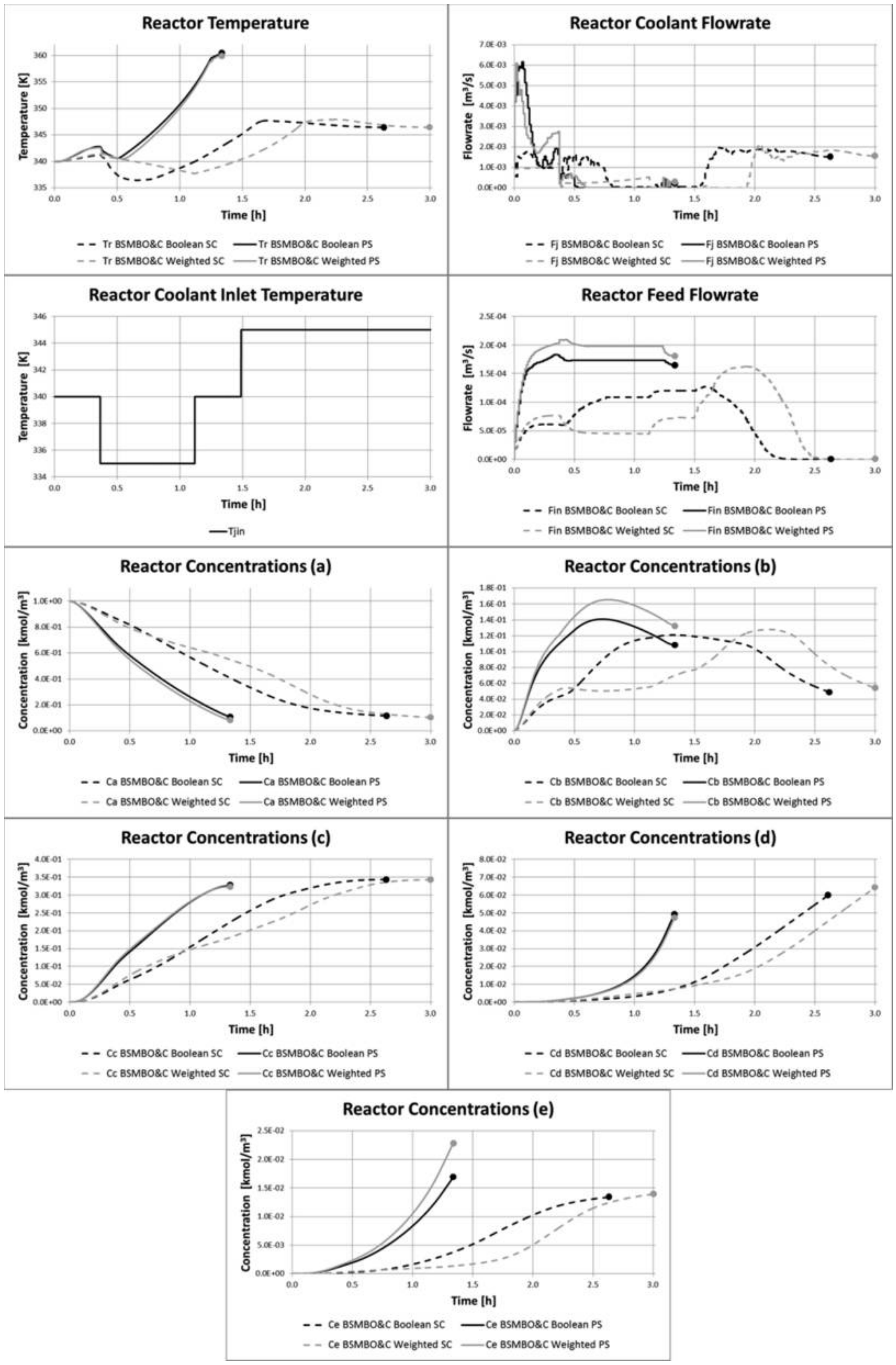

Figure 11. Pseudo-scheduling optimization and control with three step-changes in T_J^{IN}.

very similar operating points and achieve very similar performances.

The performance comparison among OOS-PID, DRTO-PID, and BSMBO&C methods gives completely different results when the reactor coolant inlet temperature is given a piece-wise constant profile with three step-changes (see Table 4). Indeed, the BSMBO&C methodology provides a much higher profitability than the DRTO-PID logic which, in turn, guarantees better incomes than the OOS-PID scheme. It appears reasonable that BSMBO&C and DRTO-PID methods perform better than the OOS-PID method in this circumstance. Indeed, the first two strategies can adjust the semibatch reactor operating condition in real-time taking into account the effect of the T_j^{IN} variation, while the third strategy is not able to do the same. The OOS-PID scheme recipe now is not optimal any more since it was optimized offline with the nominal value of T_j^{IN}. However, although both the BSMBO&C and DRTO-PID algorithms work in real-time, the BSMBO&C logic performs significantly better than the DRTO-PID logic. This happens because the BSMBO&C algorithm exploits both the manipulated variables (F^{IN} and F_j) and the batch operational time (t_{BC}) as degrees of freedom while the DRTO-PID methodology can only adjust F^{IN} and F_j. The last statement can be verified by means of a simple reasoning. At first, by looking at the trends in Figure 9, it can be seen that the BSMBO&C-based batch cycle ends approximately 2 h before the DRTO-PID-based batch cycle. Second, the "c" product concentration profile for the BSMBO&C-based batch cycle shows a maximum at the end of the cycle while the corresponding concentration profile for the DRTO-PID-based batch cycle highlights a maximum about 1 h before the end of the cycle. In addition, the semibatch system includes a side reaction that decomposes the valuable "c" product in useless and priceless "d" product (see kinetic scheme in Table 1). By connecting these three pieces of information, it results that the lower income achieved in the DRTO-PID-based batch cycle is a consequence of an excessive degradation of the valuable "c" product caused by an overlong batch cycle duration. Therefore, the BSMBO&C algorithm obtains higher profitability especially thanks to its capability of treating the batch cycle duration as an optimization variable.

At last, some remarks on the dependent variable bounds check strategies are provided. It appears that the *weighted* strategy is slightly superior than the *boolean* strategy from an economic point of view (Tables 3 and 4). Moreover, the *weighted* strategy provides smoother and more low-oscillating manipulated variable trajectories than those achieved with the *boolean* strategy (Figure 9). This is compliant to what has been written in section 3.2 and suggests that the *weighted* strategy is typically more suitable to be chosen if the extra computational effort, in respect to the *boolean* strategy, is not an excessive drawback.

4.3. Test Case II. This second case study is completely dedicated to the BSMBO&C algorithm and aims at proving the intrinsic flexibility that comes from its objective function structure (see eq 5). As for the first test case, the system on which the case study is based is the semibatch reactor modeled in section 4.1. Moreover, both scenarios with no perturbation changes and scenarios with several perturbation changes are explored. However, there is a relevant difference between the current case study and the previous one: here a pseudo-scheduling problem is addressed.

Before analyzing the test case in detail it is necessary to provide the exact meaning of pseudo-scheduling problem. In detail, a pseudo-scheduling problem is a problem for which, once a campaign duration ($t_{campaign}$) and a batch cycle dead time (t_{dead}) are fixed and assigned, the aim is to maximize the net income that can be achieved on the overall campaign. In one such problem, it is necessary to define how many batch cycles must be carried out and in which operating conditions each of them must be performed. All this must be determined online, that is, the number of batch cycles to be completed inside $t_{campaign}$ and each cycle operating conditions have to be dynamically evaluated and updated during each of the cycles execution. It is necessary to work online because the effect of any incoming perturbation change has to be taken into account.

The solution of such a problem is not trivial and may be approximated with an iterative procedure configured as follows: (I) A new batch cycle is executed and completed under the control of BSMBO&C method (in this circumstance, the BSMBO&C objective function must be set to account for the global campaign net income). (II) If the difference between $t_{campaign}$ and the sum of the durations of all the already performed cycles is less than the duration of the latest cycle the campaign end has been reached, otherwise a new batch cycle is started and the described steps are repeated again.

All the BSMBO&C method settings, which are needed for this specific case, have been chosen to equal the settings used for test case I. The only exception is the f and g functions formulation that needs to be redefined such that the net income of the overall batch campaign is given by $f{\cdot}g$. Hence, f is defined as the number of batch cycles that can be carried out in $t_{campaign}$ at the current operating conditions; g, instead, is set, as for case study I, to the net income of a single batch cycle divided by an order of magnitude of the maximum net income achievable in a single batch cycle. The new formulation of f and g is provided in eq 10 (the acronym *floor* stands for the approximation of a real number to the closest lower integer number) while the adopted values for $t_{campaign}$ and t_{dead} are summarized in Table 5.

$$\begin{cases} f = \mathit{floor}\left(\dfrac{t_{campaign}}{t_{BC} + t_{dead}}\right) \\ g = \dfrac{[C_a^0 PM_a EV_a V_R^0 - (C_c^{BC} PM_c EV_c + C_d^{BC} PM_d EV_d + C_e^{BC} PM_e EV_e) V_R^{BC} + C_b^{IN} PM_b EV_b \int_{t*}^{t_{BC}} F^{IN} dt + \rho_j EV_{coolant} \int_{t*}^{t_{BC}} F_j dt]}{C_a^0 V_R^0 (PM_c EV_c - PM_a EV_a - PM_b EV_b)} \end{cases} \tag{10}$$

By exploiting all the data provided in section 4 so far, the pseudo-scheduling problem could be addressed and solved but, for the sake of brevity, only a simplified version of the problem is reported here (an exact solution may be included in future works). Indeed, let us assume that for all the batch cycles of the campaign the perturbation trends, that is, the T^{IN}, T_j^{IN}, and C_i^{IN} profiles, are the same. With this assumption, the solution of the pseudo-scheduling problem can be achieved by solving only the first step of the iterative algorithm discussed above. Then, the first step solution can be simply used as the solution of all the other algorithm steps until the convergence condition is satisfied. This is exactly what has been done here. Therefore, only four simulations have been carried out. In the first two of them no perturbation changes occur, that is, T^{IN}, T_j^{IN}, and C_i^{IN} are kept constant. However, in the first scenario the *boolean* strategy is employed and in the second scenario the *weighted* strategy is used. In the third and fourth simulations, instead, T_j^{IN} (the most critical perturbation) is given the same piece-wise constant profile used for the test case I. However, in the third scenario the *boolean* strategy is exploited and in the fourth scenario the *weighted* strategy is chosen.

Observe that here the optimization substep features, relating to the BSMBO&C-based simulations, are essentially the same as that having been previously described for case study I. The only

Table 5. Production Campaign Duration and Batch Cycle Dead Time

campaign duration	batch cycle dead time
$t_{campaign} = 3.6 \times 10^4$ s	$t_{dead} = 360$ s

difference is that the computational time required to complete a single optimization substep, which corresponds to the time required to compute a control move, is lower and equals around 1.5 s. This occurs because the computational time depends almost only on the time required to perform a BSMBO&C objective function evaluation (BSMBO&C is implemented with a black-box approach). This computational time is lower here since the typical duration of a batch cycle in the current test case is lower than that of case study I, thus the controlled system model integration is much faster.

Coming to the results discussion, the trends for a campaign single batch cycle, coming from the first and second simulation, are reported in Figure 10, along with the corresponding trends relating to the BSMBO&C method and test case I. Instead, the profiles for a campaign single batch cycle, deriving from the third and fourth simulation, are shown in Figure 11, along with the corresponding profiles relating to the BSMBO&C method and test case I. It is important to highlight that, in Figures 10 and 11, the lines identified with the acronym "PS" are those evaluated in the current test case while the lines identified with the acronym "SC" are those inherited from test case I. Table 6, instead, summarizes the economic results of the campaign, deriving from the first two simulations. It also includes the economic results that would be achieved if, for each campaign single batch cycle, the operating condition was set to the corresponding one inherited from test case I. This is done to be able to easily make comparisons. Table 7 is the equivalent of Table 6 but it refers to the third and fourth simulations and their equivalents inherited from test case I.

Table 6. Economic Results of the Production Campaign with No Changes in T_J^{IN}

	boolean strategy		weighted strategy	
	test case I	test case II	test case I	test case II
number of batch cycles in the campaign [−]	2	7	2	7
single batch cycle duration [s]	15652.16	4782.86	13948.60	4782.86
g function [−]	0.5373	0.5407	0.5490	0.5490
single batch cycle net income [€]	3909.15	3933.63	3994.23	3993.96
campaign net income [€]	7818.29	27535.41	7988.45	27957.75

Table 7. Economic Results of the Production Campaign with Three Step-Changes in T_J^{IN}

	boolean strategy		weighted strategy	
	test case I	test case II	test case I	test case II
number of batch cycles in the campaign [−]	3	7	3	7
single batch cycle duration [s]	9457.69	4782.86	10797.63	4782.85
g function [−]	0.5664	0.5406	0.5702	0.5504
single batch cycle net income [€]	4120.88	3933.02	4148.38	4004.23
campaign net income [€]	12362.64	27531.13	12445.13	28029.59

Looking at Tables 6 and 7, it appears that the campaign management provided by the application of BSMBO&C algorithm with the f and g configuration related to the current case study is superior to the campaign handling that can be achieved by reusing the results (in terms of single-batch cycle operating conditions) computed in case study I. It means that, limited to the current problem, the f and g formulation used in this test case guarantees better results in respect to those ensured by the f and g configuration used in test case I. This is trivial to understand. Indeed, the batch cycle operating conditions achieved in case study I have been computed to be the optimal but only when a single batch cycle is performed. These operating conditions are optimal no more when it comes to deal with the management of a series of batch cycles with the aim of maximizing the income on a fixed time interval in which these several cycles must be completed. The reason why this last statement is correct is that, in this situation, it may be better to perform much more batch cycles with a slightly lower income per cycle than completing a much lower number of cycles but with a slightly higher net income per cycle. This is exactly what happens here. It can be easily understood by observing these peculiarities: (i) The single batch cycle durations for the simulations carried out in the current test case are much lower than the ones related to the simulations inherited by test case I (Figures 10 and 11). (ii) The number of batch cycles performed in the production campaign is much higher if the employed operating conditions for a single batch cycle are those coming from the current case study (Table 6 and 7). (iii) The achieved net income per cycle is typically slightly lower if the employed operating conditions for a single batch cycle are those coming from the current case study (Table 6 and 7).

Figures 10 and 11 also suggest that the optimal operating condition for a single batch cycle is significantly different between case study I and the current test case. In detail, the operating conditions of the current test case relate to higher average reactor temperatures and greater reactor feed rates. This is coherent with the need for speeding up the single batch cycle to be able to complete more cycles inside the campaign duration.

Observe also that the single batch cycles operating conditions, evaluated with the BSMBO&C settings relating to the current case study, are not significantly influenced by the T_J^{IN} trend. It means that the campaign profitability is slightly influenced by the perturbation changes. This is mainly a consequence of the short batch cycle duration that is achieved in this situation. Indeed, the shorter the batch operational time is, the less significant the perturbations effect typically is because the perturbations have a limited time period to act on the system.

Finally, it is interesting to observe that the *weighted* strategy for the dependent variable bounds management proves to be, once again, slightly better performing than its *boolean* equivalent.

To end this section, notice that this case study has not much importance by itself. Nevertheless, it can prove that the chosen structure of the BSMBO&C objective function (eq 5) is so general that very different batch production problems can be efficiently handled, if the correct f and g formulation is provided. Therefore, the BSMBO&C method can be declined in almost every user-defined configuration, each being suitable to handle a specific problem, and aims at becoming a universal model-based control and optimization algorithm for the batch world.

5. CONCLUSIONS

A novel real-time model-based optimization and control methodology for batch and semibatch systems, the BSMBO&C, has been proposed. It is able to simultaneously handle the dynamic optimization, nonlinear model predictive control, and optimization of the batch operational time, exploiting a user-defined objective function as performance indicator. Therefore, it is more efficient and flexible than similar alternatives described in literature. Indeed, the standard literature-based methods do not consider the batch cycle duration as a degree of freedom but fix it a priori and are typically designed to work only with either a quadratic or an economic objective function. In the paper, at first, the BSMBO&C mathematical formulation has been described in detail, then its implementation logics have been provided and discussed, and finally its application to a couple of test cases has been reported. The results coming from the case studies prove that the BSMBO&C is truly more effective than the standard methods proposed in literature. This is mainly due to its capability of treating the batch cycle duration as a degree of freedom. Moreover, the results coming from the test cases also show that the BSMBO&C is more general than its literature counterparts and suitable to be adapted to any possible need of the final user in terms of objective function definition. This proven flexibility and efficiency make BSMBO&C a universal model-based optimization and control tool for batch processes. A BSMBO&C free version can be downloaded at the Web site http://super.chem.polimi.it.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: flavio.manenti@polimi.it. Tel.: +390223993273. Fax: +390270638173.

Notes

The authors declare no competing financial interest.

■ REFERENCES

(1) Eaton, J. W.; Rawlings, J. B. Model predictive control of chemical processes. *Proc. Am. Control Conf.* **1991**, *2*, 1790.

(2) Balasubramhanya, L. S.; Doyle, F. J. Nonlinear control of a high-purity distillation column using a traveling-wave model. *AIChE J.* **1997**, *43*, 703.

(3) Mahadevan, R.; Agrawal, S. K.; Doyle, F. J. Differential flatness based nonlinear predictive control of fed-batch bioreactors. *Control Eng. Pract.* **2001**, *9*, 889.

(4) Abel, O.; Marquardt, W. Scenario-integrated on-line optimization of batch reactors. *J. Process Control* **2003**, *13*, 703.

(5) Nagy, Z. K.; Braatz, R. D. Robust nonlinear model predictive control of batch processes. *AIChE J.* **2003**, *49*, 1776.

(6) Joly, M.; Pinto, J. M. Optimal control of product quality for batch nylon-6,6 autoclaves. *Chem. Eng. J. (Loughborough, England)* **2004**, *97*, 87.

(7) Logist, F.; Vallerio, M.; Houska, B.; Diehl, M.; Van Impe, J. Multi-objective optimal control of chemical processes using ACADO toolkit. *Comput. Chem. Eng.* **2012**, *37*, 191.

(8) Vallerio, M.; Van Impe, J.; Logist, F. Tuning of NMPC controllers via multi-objective optimization. *Comput. Chem. Eng.* **2014**, *61*, 38.

(9) Greaves, M. A.; Mujtaba, I. M.; Barolo, M.; Trotta, A.; Hussain, M. A. Neural-network approach to dynamic optimization of batch distillation—Application to a middle-vessel column. *Chem. Eng. Res. Des.* **2003**, *81*, 393.

(10) Zavala, V. M.; Flores-Tlacuahuac, A.; Vivaldo-Lima, E. Dynamic optimization of a semi-batch reactor for polyurethane production. *Chem. Eng. Sci.* **2005**, *60*, 3061.

(11) Vite-Martínez, P.; Durán-Valencia, C.; Cruz-Maya, J. A.; Ramírez-López, A.; López-Ramírez, S. Optimization of reagents injection in a stirred batch reactor by numerical simulation. *Comput. Chem. Eng.* **2014**, *60*, 307.

(12) Pahija, E.; Manenti, F.; Mujtaba, I. M. Selecting the best control methodology to improve the efficiency of discontinuous reactors. *Comput.-Aided Chem. Eng.* **2013b**, *32*, 805.

(13) Pahija, E.; Manenti, F.; Mujtaba, I. M.; Rossi, F. Assessment of control techniques for the dynamic optimization of (semi-)batch reactors. *Comput. Chem. Eng.* **2014**, *66*, 269.

(14) Ponton, J. W.; Klemeš, J. Alternatives to neural networks for inferential measurement. *Comput. Chem. Eng.* **1993**, *17*, 991.

(15) Viganò, L.; Vallerio, M.; Manenti, F.; Lima, N. M. N.; Zuniga Linan, L.; Manenti, G. Model predictive control of a CVD reactor for production of polysilicon rods. *Chem. Eng. Trans.* **2010**, *21*, 523.

(16) Gopalakrishnan, A.; Biegler, L. T. Economic nonlinear model predictive control for periodic optimal operation of gas pipeline networks. *Comput. Chem. Eng.* **2013**, *52*, 90.

(17) Amrit, R.; Rawlings, J. B.; Biegler, L. T. Optimizing process economics online using model predictive control. *Comput. Chem. Eng.* **2013**, *58*, 334.

(18) Ellis, M.; Christofides, P. D. Economic model predictive control with time-varying objective function for nonlinear process systems. *AIChE J.* **2014**, *60*, 507.

(19) Subramanian, K.; Maravelias, C. T.; Rawlings, J. B. A state-space model for chemical production scheduling. *Comput. Chem. Eng.* **2012**, *47*, 97.

(20) Subramanian, K.; Rawlings, J. B.; Maravelias, C. T. Economic model predictive control for inventory management in supply chains. *Comput. Chem. Eng.* **2014**, *64*, 71.

(21) Buzzi-Ferraris, G.; Manenti, F. BzzMath: Library overview and recent advances in numerical methods. *Comput.-Aided Chem. Eng.* **2012**, *30*, 1312.

(22) Pahija, E.; Manenti, F.; Mujtaba, I. M. Optimization of batch and semi-batch reactors. *Comput.-Aided Chem. Eng.* **2013a**, *32*, 739.